STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, March 2025, pp 900–921.
Published online in International Academic Press (www.IAPress.org)

Interior-point methods for monotone linear complementarity problems based on the new kernel function with applications to control tabular adjustment problem

Goran Lesaja [1,4,*], Anna Oganian [2], Tifani Williams [1], Ionut Iacob [1], Mehtab Iqbal [3]

[1] *Mathematical Sciences, Georgia Southern University, Statesboro, Georgia, USA*
[2] *National Center for Health Statistics, Centers for Disease Control and Prevention, Hyattsville, MD, USA*
[3] *Clemson University, Clemson, South Carolina, USA*
[4] *Corvinus Institute for Advanced Studies, Corvinus University, Budapest, Hungary*

Abstract
We present a feasible kernel-based interior point method (IPM) to solve the monotone linear complementarity problem (LCP) which is based on an eligible kernel function with a new logarithmic barrier term. This kernel function defines the new search direction and the neighborhood of the central path. We show the global convergence of the algorithm and derive the iteration bounds for short- and long-step versions of the algorithm.
We applied the method to solve a continuous Control Tabular Adjustment (CTA) problem which is an important Statistical Disclosure Limitation (SDL) model for protection of tabular data. Numerical results on a test example show that this algorithm is a viable option to the existing methods for solving continuous CTA problems. We also apply the algorithm to the set of randomly generated monotone LCPs showing that the initial implementation performs well on these instances of LCPs. However, this limited numerical testing is done for illustration purposes; an extensive numerical study is necessary to draw more definite conclusions on the behavior of the algorithm.

Keywords Linear Complementarity Problem, Short-step and Long-step Interior-Point Methods, Iteration Bounds, Polynomial Complexity, Control Tabular Adjustment Problem

AMS 2010 subject classifications 90C33, 90C51

DOI: 10.19139/soic-2310-5070-2322

1. Introduction

In this paper, we consider the monotone Linear Complementarity Problem (LCP) in the standard form: Given a positive semi-definite matrix $M \in \mathbf{R}^{n\times n}$ and a vector $q \in \mathbf{R}^n$, find a pair of vectors $(x, s) \in \mathbf{R}^{2n}$ such that

$$-Mx + s = q \qquad x, s \geq 0 \qquad xs = 0, \tag{1}$$

where xs denotes the component-wise product of vectors x and s. Monotone LCPs are commonly used for both theoretical and practical purposes. The importance of LCPs stems from the fact that Karush-Kuhn-Tucker (KKT) optimality conditions of important optimization problems such as Linear Optimization (LO) and Quadratic Optimization (QO) can be formulated as LCPs (see [21]). In addition, many important practical problems in game theory, engineering, economics, and optimal control can be formulated as LCPs, see [16, 20].

*Correspondence to: Goran Lesaja (Email: goran@georgiasouthern.edu). Mathematical Sciences, Georgia Southern University, Statesboro, Georgia 30458, USA.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Due to the theoretical and practical importance of LCPs, efficient methods for solving LCPs are of significant interest. The Interior-Point Methods (IPMs) have revolutionized optimization theory and practice in the last three decades and have been proven to be quite efficient in solving different classes of LCPs that were inaccessible or difficult to solve using simplex-type pivoting methods. The path-following IPMs are based on following a *central path* trajectory of the LCP approximately by staying within a predetermined neighborhood of the central path towards a solution of LCP. The main tool in this process is Newton's method which is used to determine the search direction at each iteration.

There are two types of path-following IPMs, IPMs with short-step updates at each iteration, and long-step updates at each iteration. Long-step IPMs take more aggressive steps that can deviate farther away from the central path, while short-step IPMs take more conservative steps staying closer to the central path. Most of the IPMs are based on the classical logarithmic barrier function [22] that is stated in the sequel in (7). The long-step versions of these IPMs have $O\left(n\log\frac{n}{\epsilon}\right)$ iteration bound while short-step versions have $O\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ iteration bound which is for order of magnitude better. However, in practice, the situation is the opposite, long-step IPMs perform much better than short-step IPMs. This is considered the *irony* of IPMs, the term coined by Renegar [32]. Throughout the years a considerable effort has been devoted to designing long-step IPMs with improved theoretical iteration bounds to reduce, or even close, the gap between the theoretical complexities of short- and long-step IPMs. Several approaches can be observed.

The first approach was to consider higher-order methods based on the classical logarithmic barrier function [31]. The second approach was the IPM developed by Ai and Zhang [1] which was based on two key ideas: a new wide neighborhood and the decomposition of Newton's directions into two components. In both of these approaches, the gap between short- and long-step versions of the IPMs was closed on the expense of more computational work.

The third approach was a fruitful idea of kernel-based IPMs introduced by Peng, Roos, and Terlaky [29] that led to long-step IPMs with an improved iteration bound. They considered the class of self-regular kernel functions (SRKFs). Subsequently, Bai, Roos, and El Ghami [7] introduced a different class of kernel functions, a class of eligible kernel functions (EKFs), and obtained a similar improvement for long-step IPMs for LO. These results were extended to $P_*(\kappa)$-LCPs in [30] for SRKFs and in [22] for EKFs. For some instances of SRKFs and EKFs they obtained $O\left(\sqrt{n}(\log n)\log\frac{n}{\epsilon}\right)$ iteration bound for long-step IPMs which is a significant improvement when compared with $O\left(n\log\frac{n}{\epsilon}\right)$ for long-step IPM based on the classical logarithmic kernel function.

In this paper, we introduce a new kernel function which is a modification of the classical logarithmic kernel function and show that it is EKF and SRKF. The general convergence analysis of the IPMs based on the general EKFs presented in [22] is used to show the convergence of the IPM based on the new EKF. The iteration bounds for short- and long-step versions of the IPM are derived, which turns out to be as good as for the classical logarithmic kernel function.

It is also shown that the IPM developed in this paper provides an alternative efficient method to solve the Controlled Tabular Adjustment (CTA) problem [24] which is usually solved using the simplex method.

CTA is a method of Statistical Disclosure Limitation (SDL) that was first introduced in [10, 17]. The SDL is an increasingly important area of research and practice for the statistical agencies that collect data from individuals or enterprises and then release it to the public, researchers, and policymakers for statistical analysis and research. Prior to such a release, the collected data have to undergo some SDL procedure in order to guarantee the privacy and confidentiality of data providers. The goals of such procedures are two-fold: minimize the risk of disclosure of confidential information about data providers and, at the same time, maximize the amount of released information, that is, maximize the utility of the data for the legitimate data users. These are conflicting goals and therefore SDL practice as a whole can be thought of as a search for the solution of complex and multifaceted optimization problem: maximize the utility of the released data, subject to some upper bound on disclosure risk. The way utility and risk are formulated depends on the scenario of data release and on the data format.

Data can be released in two basic formats: microdata - a collection of individual records, and tabular data - a table of cumulative data that is obtained from cross-tabulations of attributes from microdata. CTA is a perturbative method of protecting tabular data when a specified subset of its cells, called sensitive cells, must be modified to avoid the re-identification of an individual respondent. The goal of CTA is to guarantee that the modified value of a sensitive cell is outside of the *disclosure interval*- an interval that is determined by the data protector (usually a

statistical agency). The remaining cells are minimally adjusted to satisfy table equations which usually represent the requirement that the sum of elements in each row and column should be constant and remain unchanged. Hence, the goal of CTA is to find the closest *safe* table to the original table with respect to the constraints outlined above. The closeness of the original and modified table is measured by the weighted distance between the tables with respect to a certain norm. Most commonly used norms are ℓ_1 and ℓ_2 norms. Thus, the problem can be formulated as a minimization problem with the objective function being a particular weighted distance function and constraints being table equations and lower and upper bounds on the cell values. The ℓ_2-CTA reduces to a Quadratic Optimization problem while the ℓ_1-CTA is a convex but nonsmooth problem that can be reformulated as a Linear Optimization (LO) problem. However, the number of variables and inequality constraints doubles. Alternatively, in [24] a novel second-order cone (SOC) reformulation of ℓ_1-CTA is proposed that does not increase the dimension of the problem as much. As it is shown in [24], conic reformulation of ℓ_1-CTA is a viable alternative to LO reformulation of the problem.

For the sake of completeness, the general formulations of CTA and its various reformulations are given in the Numerical Results section of the paper. The IPM developed in this paper is applied to an LO formulation of a CTA problem that appears as a test problem in [24] and several other papers. It is shown to work well confirming the fact that it is a viable alternative for solving continuous CTA. We also apply the algorithm to the small set of randomly generated monotone LCPs showing that the initial implementation performs well on these instances of LCPs. However, this limited numerical testing is done for illustration purposes. More sophisticated implementation, extensive numerical testing, and comparison to other methods are needed to draw more definite conclusions about the practical performance of the method and are subject of future research.

The outline of the paper is as follows. The generic barrier-based IPM is presented in Section 2. The definitions of the kernel function, EKFs, and SRKFs are presented in Section 3.1. The new kernel function and corresponding barrier function are discussed in Section 3.2. The analysis of the algorithm based on the new kernel function and derivation of the iteration bounds for short- and long-step variants of the method are presented in Section 4. The application of the method to a CTA problem and preliminary numerical results are presented in Section 5 and the concluding remarks are outlined in Section 6.

2. Barrier-based IPMs

As already indicated in the previous section, in this paper we consider the monotone LCP in the standard form (1). We assume that the LCP satisfies the interior-point condition (IPC), which means that there exists a point $x^0 > 0$ such that $s^0 = Mx^0 + q > 0$, indicating that the interior of the feasible region is not empty. The IPC can be assumed without loss of generality, since the original LCP can be embedded in the larger LCP with the known starting point as described in Section 5.1 in [21].

Most IPMs are based on the use of Newton's method. However, it is well-known using the direct application of Newton's method to the system (1) will most likely fail at the second equation called the *complementarity equation*. Hence, the idea is to replace the complementarity equation in (1) with a parametrized equation $xs = \mu e$ with parameter $\mu > 0$ and e being a vector of all ones. Therefore, we consider the parametrized system

$$\begin{aligned} -Mx + s &= q, \\ xs &= \mu e, \\ x, s &> 0\,. \end{aligned} \tag{2}$$

Since we assume that IPC holds and M is a positive-semidefinite matrix, it is well known that the parametrized system (2) has a unique solution, for each $\mu > 0$; see Lemma 4.1 in [21]. This solution is denoted as $(x(\mu), s(\mu))$ and it is called the μ-center of the LCP. The set of μ-centers, with μ running through all positive real numbers, forms a homotopy path, which is called *the central path* of the LCP. Under the above assumptions, if $\mu \to 0$, the limit of the central path exists and it is a solution of LCP (1).

The limiting property of the central path mentioned above leads naturally to the main idea of the IPMs for solving LCP: Trace the central path while reducing μ at each iteration. However, tracing the central path exactly

would be computationally too costly and inefficient. It has been shown that it is sufficient to trace the central path approximately within a certain neighborhood of the central path. The neighborhood of the central path is determined using a certain proximity measure to the central path which in this paper is determined by the kernel function and the corresponding barrier function.

Suppose an iterate (x, s) of the IPM in the neighborhood of the central path is known. We then decrease μ to $\mu := (1-\theta)\mu$, for some value of the barrier parameter $\theta \in (0,1)$. Applying Newton's method to (2) leads to the following Newton's system for the search directions $(\Delta x, \Delta s)$:

$$\begin{aligned} -M\Delta x + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs. \end{aligned} \tag{3}$$

Since M is a positive semidefinite matrix, the above system has a unique solution for any $(x, s) > 0$.

For the analysis of IPMs it is important to associate any pair $(x, s) \geq 0$ and $\mu > 0$ with the *variance vector*

$$v := \sqrt{\frac{xs}{\mu}}. \tag{4}$$

Note that the pair (x, s) coincides with the μ-center $(x(\mu), s(\mu))$ if and only if $v = e$. Furthermore, it is also important to introduce the following *scaled search directions* d_x and d_s

$$d_x := \frac{v\Delta x}{x}, \quad d_s := \frac{v\Delta s}{s}, \tag{5}$$

where the operations are component-wise product and division. Using (4) and (5) the system (3) can be rewritten as

$$\begin{aligned} -\bar{M}d_x + d_s &= 0, \\ d_x + d_s &= v^{-1} - v, \end{aligned} \tag{6}$$

where $\bar{M} := DMD$, with $D := X^{1/2}S^{-1/2}$, $X := \mathrm{diag}\,(x)$, $S := \mathrm{diag}\,(s)$.

A crucial observation is that the right-hand side of the second equation of (6) equals minus the gradient of the function

$$\Psi_c(v) := \sum_{i=1}^{n} \left(\frac{v_i^2 - 1}{2} - \log v_i \right), \tag{7}$$

where v_i represents the i-th component of the variance vector v. In other words,

$$d_x + d_s = -\nabla\Psi_c(v). \tag{8}$$

This equation is called the *scaled centering equation*. Its importance arises from two facts, first, it essentially defines the search direction, and second, it defines a proximity measure to the central path.

One may easily verify that $\nabla^2\Psi_c(v) = \mathrm{diag}\,(e + v^{-2})$. Since this matrix is positive definite, $\Psi_c(v)$ is strictly convex. Moreover, since $\nabla\Psi_c(e) = 0$, it follows that $\Psi_c(v)$ attains its minimal value at $v = e$, with $\Psi_c(e) = 0$. Thus, $\Psi_c(v)$ is nonnegative everywhere and vanishes if and only if $v = e$, that is, if and only if (x, s) is a μ-center, i. e., $x = x(\mu)$ and $s = s(\mu)$. Hence, we see that the μ-center $(x(\mu), s(\mu))$ can be characterized as the minimizer of the function $\Psi_c(v)$. Thus, the second important feature of the function $\Psi_c(v)$ is that it essentially serves as a 'proximity' measure of closeness for (x, s) with respect to the μ-center.

The above observations regarding the function $\Psi_c(v)$ lead to an obvious generalization: We can replace $\Psi_c(v)$ by any strictly convex function $\Psi(v)$, $v \in \mathbf{R}^n_{++}$, such that $\Psi(v)$ is minimal at $v = e$ and $\Psi(e) = 0$. Thus, the new scaled centering equation becomes

$$d_x + d_s = -\nabla\Psi(v). \tag{9}$$

The function $\Psi(v)$ is called a *(scaled) barrier function*. Hence, different barrier functions lead to different Newton's directions, as they are calculated from Newton's system

$$\begin{aligned} -\bar{M}d_x + d_s &= 0, \\ d_x + d_s &= -\nabla\Psi(v). \end{aligned} \tag{10}$$

Since

$$\Psi(v) = 0 \quad \Leftrightarrow \quad \nabla\Psi(v) = 0 \quad \Leftrightarrow \quad v = e,$$

the function $\Psi(v)$ still serves as a proximity measure of closeness to the μ-center $(x(\mu), s(\mu))$ on the central path.

Introducing a parameter $\tau > 0$ as a *threshold value*, the inequality $\Psi(v) \leq \tau$ defines a τ-neighborhood of the central path.

$$N_\Psi(\tau) = \left\{ v \in \Re^n_{++} \ : \ \Psi(v) \leq \tau \right\} \tag{11}$$

In the sequel we also use norm-based proximity measure, which is defined by

$$\delta(v) := \frac{1}{2}\|\nabla\Psi(v)\|. \tag{12}$$

The function $\delta(v)$ is often called *proximity function* and it is easy to see that

$$\delta(v) = 0 \quad \Leftrightarrow \quad v = e. \tag{13}$$

From the above discussion, the main idea of the IPM based on the appropriate barrier function follows. Given the current iterate (x, s), we start the new *outer iteration* by targeting the new μ^+-center obtained by reducing μ by a factor $1 - \theta$ with $0 < \theta < 1$, that is, $\mu^+ = (1 - \theta)\mu$. This implies the change in the variance vector v, $v^+ = \frac{v}{\sqrt{1-\theta}}$. Most likely, after this step, the value of the barrier function will exceed the threshold value, that is, $\Psi(v^+) > \tau$, and hence, the algorithm enters the *inner iteration loop*. The inner iteration loop consists of computing the scaled search directions d_x and d_s at the current iterate and the current value of μ^+ from Newton's system (10), which we rename to be a new μ. Next, we compute the corresponding search directions Δx and Δs from d_x and d_s by using (5). Then, the new inner iterate is calculated as

$$x^+ = x + \alpha\Delta x, \quad s^+ = s + \alpha\Delta s \tag{14}$$

with the appropriately calculated step size α. If necessary, the procedure is repeated, however, with keeping μ fixed, until the iterate belongs to the τ-neighborhood (11) of the current μ-center $(x(\mu), s(\mu))$, that is, until $\Psi(v) \leq \tau$. This process is repeated until μ is small enough, which is usually measured by $n\mu < \epsilon$. At this stage, we have found an ϵ-approximate solution of LCP (1).

Note that the algorithm can be started, since, as we discussed at the beginning of this section, we may assume that a strictly feasible point (x^0, s^0) is given, and this point can be chosen such that $\Psi(v^0) \leq \tau$ which means it is in the τ-neighborhood of the μ-center.

The generic form of the barrier-based IPM for monotone LCP is given in Figure 1.

The parameters τ, θ and the step size α in the algorithm should be tuned in such a way that the number of iterations required by the algorithm is as small as possible.

Two main types of the algorithm can be distinguished: short-step method and long-step method, according to the value of the barrier-update parameter θ. The long-step method is characterized by the fact that θ is a fixed constant ($\theta \in (0, 1)$), independent of the dimension n of the problem, whereas the short-step method uses a value of θ that depends of the dimension of the problem, with $\theta = O\left(\frac{1}{\sqrt{n}}\right)$.

The resulting iteration bound depends on a careful selection of these parameter values. It also heavily depends on the choice of the barrier function.

Generic barrier-based IPM for LCP

Input:
a threshold parameter $\tau \geq 1$;
an accuracy parameter $\varepsilon > 0$;
a fixed barrier update parameter θ, $0 < \theta < 1$;
a starting point (x^0, s^0), such that $\Psi(v^0) \leq \tau$, where
$\mu^0 = \frac{(x^0)^T s^0}{n}$, and $v^0 = \sqrt{\frac{x^0 s^0}{\mu^0}}$;

begin
$x := x^0$; $s := s^0$; $\mu := \mu^0, v := v^0$;
while $n\mu \geq \varepsilon$ **do**
begin
$\mu := (1-\theta)\mu$;
$v = \frac{v}{\sqrt{1-\theta}}$;
while $\Psi(v) > \tau$ **do**
begin
calculate search direction $(\Delta x, \Delta s)$ using (5) and (10);
determine a step size α;
update $x := x + \alpha\Delta x$; $s := s + \alpha\Delta s$;
$v := \sqrt{\frac{xs}{\mu}}$;
end
end
end

Figure 1. Generic barrier-based IPM for LCP

3. Kernel functions and their properties

3.1. Eligible kernel functions

Following the discussion in the previous section and considering the original logarithmic barrier function (7), we restrict ourselves to the case where a barrier function $\Psi(v)$ is separable with identical coordinate functions $\psi(v_i)$. All barrier functions considered in the literature on this topic are of this type. Thus,

$$\Psi(v) = \sum_{i=1}^{n} \psi(v_i), \tag{15}$$

where $\psi(t) : (0, +\infty) \to [0, +\infty)$ is twice differentiable, convex, and attains its minimum at $t = 1$, with $\psi(1) = \psi'(1) = 0$. Following the terminology introduced in [6, 7, 28], we call the univariate function $\psi(t)$ the *kernel function* of the barrier function $\Psi(v)$ (15). In the following, we state the formal definition of the kernel function.

Definition 3.1
The univariate function $\psi : (0, \infty) \to [0, \infty)$ is called a *Kernel Function* (KF) if it satisfies the following conditions:

$$\begin{aligned} \psi'(1) = \psi(1) &= 0, && \text{(KF-a)}\\ \psi''(t) &> 0, && \text{(KF-b)}\\ \lim_{t\downarrow 0}\psi(t) = \lim_{t\to\infty}\psi(t) &= \infty. && \text{(KF-c)} \end{aligned}$$

The first class of KF, the class of Self-Regular KFs (SRKF), that leads to the improved complexity of long-step IPMs was proposed by Peng, Roos, and Terlaky in [29]. Subsequently, the class of Eligible Kernel Functions (EKFs) was introduced in [7] that leds to the same improved complexity of long-step IPMs. It is worth noting that these two classes are not the same, however, they have a nonempty intersection. Below we state the formal definitions of SRKF and EKF.

Definition 3.2
The KF defined in Definition 3.1 that satisfies the following additional properties:

C.1 There exist positive constants $\nu_2 \geq \nu_1 > 0$ and parameters $p \geq 1,\ q \geq 1$ such that

$$\nu_1\left(t^{p-1}+t^{-1-q}\right) \leq \psi''(t) \leq \nu_2\left(t^{p-1}+t^{-1-q}\right),\ \ \forall t \in (0,\infty);$$

C.2 For any $t_1, t_2 > 0$

$$\psi\left(t_1^r t_2^{1-r} \leq r\psi(t_1)+(1-r)\psi(t_2)\right),\ \ \forall r \in [0,1];$$

is called a Self-Regular Kernel Function (SRKF). We call parameter q the barrier degree and parameter p the growth degree of the function $\psi(t)$.

Definition 3.3
The KF defined in Definition 3.1 that satisfies the following additional properties

$$\begin{aligned} t\psi''(t)+\psi'(t) &> 0, \quad t<1, && \text{(EKF-a)}\\ \psi'''(t) &< 0, \quad t>0, && \text{(EKF-b)}\\ 2\psi''(t)^2-\psi'(t)\psi'''(t) &> 0, \quad t<1, && \text{(EKF-c)}\\ \psi''(t)\psi'(\beta t)-\beta\psi'(t)\psi''(\beta t) &> 0, \quad t>1,\ \beta>1, && \text{(EKF-d)} \end{aligned}$$

is called an Eligible Kernel Function (EKF).

Remark 3.4
In [7] another condition is also discussed, namely,

$$t\psi''(t)-\psi'(t)>0, \quad t>1. \tag{16}$$

This condition is listed because conditions (16) and (EKF-b) imply condition (EKF-d) (Lemma 4.4 in [7]). The reason for the introduction of condition (16) is that it is easier to check condition (16) than (EKF-d) which is more technically involved.

3.2. The new EKF with logarithmic barrier term

It is easy to observe that the barrier function (15) is a separable function with the same univariate function for each component v_i of the variance vector v. If we define this univariate function as

$$\psi_c(t) = \frac{t^2-1}{2} - \ln t, \tag{17}$$

then the barrier function (15) becomes logarithmic barrier function as stated in (7), i.e. $\Psi(v) = \Psi_c(v)$. The related kernel function $\psi_c(t)$ is called the *logarithmic kernel function*. These are the classical logarithmic barrier and kernel

functions that have been used the most in the theory and practice of IPMs. Most optimization software codes are also based on these functions. It is not hard to show that the logarithmic kernel function is EKF and SRKF. Term $\frac{t^2-1}{2}$ is called the growth term and $-\ln t$ is called the barrier term.

However, many other SRKFs and EKFs have been proposed and IPMs based on them analyzed and iteration bounds calculated. The literature on this topic is rich. Without an attempt to be complete, we list several relevant references [22, 6, 7, 9, 23, 29, 30, 27, 19, 18, 14].

The EKFs that appeared in the literature can be classified regarding the barrier term into four main groups, EKFs with logarithmic, rational, exponential, trigonometric barrier terms, or some combination of them. There are quite a number of instances of EKFs from last three groups, however, there are very few with logarithmic barrier terms that are different than the classical logarithmic EKF (17).

In this section we introduce a new KF with a logarithmic barrier term that is different than the barrier term of the classical logarithmic EKF, $-\ln t$ and show that it is EKF and SRKF. The definition of the new kernel function is as follows:

$$\psi(t) = \frac{t^2-1}{2} + 2\ln\left(1+\frac{1}{t}\right) - 2\ln(2), \quad t > 0, \tag{18}$$

and the corresponding scaled barrier function is

$$\Psi(v) = \sum_{i=1}^{n} \frac{v_i^2-1}{2} + 2\ln\left(1+\frac{1}{v_i}\right) - 2\ln 2 \qquad v_i > 0. \tag{19}$$

The new kernel function has a standard growth term $\frac{t^2-1}{2}$ and a logarithmic barrier term $2\ln\left(1+\frac{1}{t}\right) - 2\ln 2$. It is worth noting that this is the only other kernel function in the literature with a pure logarithmic barrier term, however, different than the barrier term $-\ln t$ that appears in the classical logarithmic kernel function (17).

The first three derivatives of the new kernel function (18) are:

$$\psi^{'}(t) = t - \frac{2}{t^2+t}, \tag{20}$$

$$\psi^{''}(t) = 1 + 2\frac{1+2t}{(t^2+t)^2}, \tag{21}$$

$$\psi^{'''}(t) = -4\frac{3t^2+3t+1}{(t^2+t)^3}. \tag{22}$$

It is easy to see that the new kernel function satisfies all properties of a kernel function stated in the Definition 3.1 Furthermore, $\psi^{''}(t)$ is monotone decreasing for $t > 0$. Since $\psi(t)$ is at least twice differentiable, it is completely determined by the following integral

$$\psi(t) = \int_1^t \int_1^{\xi} \psi^{''}(\zeta) d\zeta d\xi. \tag{23}$$

In the next lemma, it is shown that (18) is EKF. We need to show that the four conditions for eligibility stated in the Definition 3.3 are satisfied.

Lemma 3.5
The kernel function in (18) is EKF since it satisfies all conditions in Definition 3.3.

Proof
Using the first three derivatives of the new kernel function 20 - 22, we easily check eligibility conditions in Definition 3.3.

$$\begin{aligned}
t\psi^{''}(t)+\psi^{'}(t) &= 2t+2t\tfrac{1+2t}{(t^2+t)^2}-\tfrac{2}{t^2+t}=2t+\tfrac{2}{(1+t)^2} && >0, \quad t<1,\\
t\psi^{''}(t)-\psi^{'}(t) &= 2t\tfrac{1+2t}{(t^2+t)^2}+\tfrac{2}{t^2+t}=\tfrac{2(2+3t)}{t(1+t)^2} && >0, \quad t>1,\\
\psi^{'''}(t) &= -4\tfrac{3t^2+3t+1}{(t^2+t)^3} && <0, \quad t>0,\\
2\psi^{''}(t)^2-\psi^{'}(t)\psi^{'''}(t) &= \tfrac{2(t^7+4t^6+6t^5+18t^4+33t^3+24t^2+10t+4}{t^3(1+t)^4} && >0, \quad t<1.
\end{aligned}$$

□ □

It is shown in [22], that these conditions are essential in proving the convergence of the generic algorithm and deriving good iteration bounds. The following lemma that provides the quadratic bounds for EKFs is also important in the subsequent analysis of the algorithm.

Lemma 3.6 (Lemma 4.2 in [22])
Given any EKF $\psi(t)$, the following inequalities hold:

$$\frac{1}{2}\psi^{''}(1)(t-1)^2<\psi(t)<\frac{1}{2}\psi^{''}(t)(t-1)^2, \quad t<1, \tag{24}$$

$$\frac{1}{2}\psi^{''}(t)(t-1)^2<\psi(t)<\frac{1}{2}\psi^{''}(1)(t-1)^2, \quad t>1. \tag{25}$$

Observe that for $t=1$, the above inequalities hold as equalities. For the new EKF (18), the quadratic bounds above reduce to the following bounds.

Corollary 3.7
Given the EKF (18), the following inequalities hold:

$$\frac{1}{2}(t-1)^2<\psi(t)\leq\frac{1}{2}\psi^{'}(t)^2, \qquad t<1; \tag{26}$$

$$\frac{1}{2}(t-1)^2<\psi(t)<\frac{5}{4}(t-1)^2, \qquad t>1. \tag{27}$$

Proof
From (21) it immediately follows $\psi^{''}(t)>1$ for $t>0$ which leads to left inequalities in (26) and (27). The right side of the right inequality in (27) follows from the fact that $\psi^{''}(1)=\frac{5}{2}$. The right inequality in (26) follows from (23)

$$\psi(t)=\int_1^t\int_1^\xi\psi^{''}(\zeta)d\zeta d\xi\leq\int_1^t\int_1^\xi\psi^{''}(\xi)\psi^{''}(\zeta)d\zeta d\xi=\frac{1}{2}\psi^{'}(t)^2,$$

where the inequality in the derivation above is due to $\psi^{''}(t)>1$ for $t>0$. Hence, (26) holds not only for $0<t<1$ but for entire $t>0$. □ □

Below, we show that the new EKF (18) is also SRKF.

Lemma 3.8
The kernel function in (18) is SRKF since it satisfies all conditions in Definition 3.2.

Proof
Proof of $C1$:
We first show the upper bound in $C1$.

$$\begin{aligned}
\psi''(t) &= 1+2\tfrac{1+2t}{(t^2+t)^2}\\
&\leq 1+2\tfrac{2+2t}{(t^2+t)^2}\\
&= 1+4\tfrac{1}{t^2(1+t)}\\
&\leq 1+4\tfrac{1}{t^2}\\
&\leq 4\left(1+\tfrac{1}{t^2}\right).
\end{aligned}$$

Hence, $\nu_2 = 4$ and $p = q = 1$. Next, we show the lower bound in $C1$.

$$\begin{aligned}\psi''(t) &= 1 + 2\frac{1+2t}{(t^2+t)^2} \\ &\geq 1 + \frac{1+t}{(t^2+t)^2} \\ &= 1 + \frac{1}{t^2(1+t)}\end{aligned}$$

We have two cases:

Case I: If $0 < t < 1$, then $1 + t < 2$ implying $\frac{1}{t^2(1+t)} > \frac{0.5}{t^2}$. Therefore, it follows $\psi''(t) \geq .5\left(1 + \frac{1}{t^2}\right)$.

Case II: If $t \geq 1$, then $.5 \geq \frac{.5}{t^2}$ implying $1 + \frac{1}{t^2(1+t)} \geq .5 + \frac{.5}{t^2} + \frac{1}{t^2(1+t)}$. Since $\frac{1}{t^2(1+t)} \geq 0$, it follows that $\psi''(t) \geq .5\left(1 + \frac{1}{t^2}\right)$.

Hence, $\psi''(t) \geq .5\left(1 + \frac{1}{t^2}\right)$ for $t > 0$. We see that $\nu_1 = 0.5$ and again $p = q = 1$.

Proof of $C2$:

The proof of $C2$. follows from the (EKF-a). In [29] it is shown that these two conditions are equivalent. □

4. Analysis of the algorithm for new EKF

In this section, we analyze the algorithm presented in Figure 1 for the specific barrier function (19) derived from the new EKF (18). In the sequel, we refer to the algorithm in Figure 1 with this specific barrier function as simply the Algorithm.

We first give an overview of the analysis of the algorithm with general EKF. The analysis is performed in the following four major steps.

- During the outer iteration the value of a barrier function, which serves as a proximity measure, increases. It is important to find an upper bound on that increase.
- Next, we determine the default step size that leads to a sufficient decrease of the barrier function during each inner iteration.
- The lower bound on the decrease of the barrier function during each inner iteration is then found.
- Combining previous steps, we can determine the upper bound on the number of iterations necessary to find ε-approximate solution of the LCP for long-step and short-step methods.

We will not repeat this general analysis here, instead, we refer the reader to [22] where the details of the analysis and the proofs can be found. We will especially use the Scheme from [22] which streamlines the analysis of the generic algorithm and calculations of iteration bounds for short- and long-step versions of the method. However, to better follow the process, we first state the definition of the following two functions that play a crucial role in the analysis of the generic algorithm and derivation of the iteration bounds and are frequently used in the Scheme.

Definition 4.1

We define two inverse functions, ϱ and ρ, as follows:

(i) The function $\varrho : [0, \infty) \longrightarrow [1, \infty)$ is the inverse function of $\psi(t)$, $t \geq 1$.

(ii) The function $\rho : [0, \infty) \longrightarrow (0, 1]$ is the inverse function of $-\frac{1}{2}\psi'(t)$, $t \in (0, 1]$.

We will adapt the Scheme for the new EKF (18) and derive the iteration bounds of the short-step and long step versions of the Algorithm. We first state the steps, as they are stated in Scheme, and then make the derivations for the new EKF (18).

Step 1: *Solve the equation* $-\frac{1}{2}\psi'(t) = s$ *to get* $\rho(s)$*, the inverse function of* $-\frac{1}{2}\psi'(t), t \in (0, 1]$*. If the equation is hard to solve, derive a lower bound for* $\rho(s)$.

It is important to mention that variable s used in the derivations in this step and subsequent steps is different than the variable s used in the definition of the LCP (1). It is quite clear which one is used from the content where it appears.

We have the following derivation;

$$\begin{array}{rcl} s & = & -\frac{1}{2}\psi'(t) \\ s & = & -\frac{1}{2}\left(t - \frac{2}{t^2+t}\right) \\ \frac{2}{t^2+t} & = & t + 2s \\ \frac{1}{t} & \leq & 1 + 2s \end{array}$$

The inequality above follows from the fact that $t \leq 1$ which implies $t^2 \leq t$ and $t^2 + t \leq 2t$ that leads to $\frac{2}{t^2+t} \geq \frac{1}{t}$. Hence, we have

$$\rho(s) = t \geq \frac{1}{1+2s}. \tag{28}$$

Step 2: *Calculate the decrease of the barrier function $\Psi(v)$ during the inner iteration in terms of the proximity measure $\delta(v)$ for the default step size $\tilde{\alpha}$.*

In what follows $\delta(v)$ is denoted simply as δ. From Lemma 5.10 in [22], the lover bound on the decrease of the barrier function, $f(\alpha) = \Psi(v_+) - \Psi(v)$ is given as

$$f(\tilde{\alpha}) \leq -\tilde{\alpha}\delta^2, \tag{29}$$

where $\tilde{\alpha}$ is a default step size

$$\tilde{\alpha} = \frac{1}{\psi''\left(\rho\left(2\delta\right)\right)}. \tag{30}$$

For $\psi''(t)$ given by (21), we have the following derivation for $t \in (0, 1]$,

$$\psi''(t) = 1 + 2\frac{1+2t}{(t+t^2)^2} < 1 + 2\frac{2+2t}{(t+t^2)^2} = 1 + 2\frac{2(1+t)}{t^2(1+t)^2} = 1 + \frac{4}{t^2(1+t)} < 1 + \frac{4}{t^2}, \tag{31}$$

where the last inequality is due to the fact that $1/1 + t < 1$ for $t > 0$. Hence, from (28), (30) and (31) we derive the lower bound for the default step size

$$\tilde{\alpha} = \frac{1}{\psi''\left(\rho\left(2\delta\right)\right)} > \frac{1}{1 + \frac{4}{(\rho(2\delta))^2}} \geq \frac{1}{1 + 4(1+4\delta)^2} \tag{32}$$

Step 3: *Solve the equation $\psi(t) = s$ to get $\varrho(s)$, the inverse function of $\psi(t), t \geq 1$. If the equation is hard to solve, derive lower and upper bounds for $\varrho(s)$.*

From the definition (18) of $\psi(t)$, we have

$$\frac{t^2-1}{2} = s - 2\ln\left(1 + \frac{1}{t}\right) + 2\ln(2). \tag{33}$$

Since $-2\ln\left(1+\frac{1}{t}\right) + 2\ln(2)$ is monotone increasing function with respect to $t \geq 1$, (33) reduces to the following inequality

$$\frac{t^2-1}{2} \geq s,$$

which implies that

$$\varrho(s) = t \geq \sqrt{2s+1}. \tag{34}$$

Using (27) in Corollary 3.7, we have

$$s = \psi(t) \geq \frac{1}{2}(t-1)^2,$$

where equality is obtained for $t = 1$. This immediately leads to the following inequality

$$\varrho(s) = t \leq 1 + \sqrt{2s}. \tag{35}$$

Hence, in inequalities (34) and (35) we obtained lower and upper bounds on $\varrho(s)$

$$\sqrt{2s+1} \leq \varrho(s) \leq 1+\sqrt{2s}.$$

Step 4: *Derive a lower bound for δ in terms of $\Psi(v)$.*
We have

$$\Psi(v) = \sum_{i=1}^{n} \psi(v_i) \leq \sum_{i=1}^{n} \frac{1}{2}\psi'(v_i)^2 = \frac{1}{2}\left\|\nabla\Psi(v)\right\|^2 = 2\delta(v)^2,$$

where the inequality in the derivation above is due to the right inequality in (26) which is valid not only for $t<1$ but for entire $t>0$, as shown in the proof of the Corollary 3.7. Hence, we immediately obtain the lower bound for δ in terms of Ψ

$$\delta(v) \geq \sqrt{\frac{1}{2}\Psi(v)}. \tag{36}$$

This lower bound leads to yet another one. Since in the Algorithm we assume the threshold parameter $\tau \geq 1$ and in inner iterations $\Psi(v) \geq \tau$, the inequality (36) reduces to

$$\delta(v) \geq \sqrt{\frac{1}{2}} > \frac{1}{2}. \tag{37}$$

Step 5: *Substitute the results of Step 3 and Step 4 into Step 2 and find a valid inequality of the form*

$$f(\tilde{\alpha}) \leq -\beta\Psi(v)^{1-\gamma}$$

for some positive constants β and γ, with $\gamma \in (0,1]$ as small as possible.

The inequality in (37) can be written as $1 \leq 2\delta$ and used to further transform the lower bound for default step size $\tilde{\alpha}$ in (32)

$$\tilde{\alpha} > \frac{1}{1+4(4\delta+1)^2} \geq \frac{1}{4\delta^2+4(2\delta+4\delta)^2} = \frac{1}{148\delta^2}. \tag{38}$$

Substituting (36) and (38) into the (29) we obtain

$$f(\tilde{\alpha}) \leq -\tilde{\alpha}\delta^2 \leq -\frac{1}{148\delta^2}\delta^2 = -\frac{1}{148}. \tag{39}$$

Hence, we have

$$f(\tilde{\alpha}) \leq -\beta\Psi(v)^{1-\gamma} = -\frac{1}{148},$$

implying that

$$\beta = \frac{1}{148}, \quad \gamma = 1. \tag{40}$$

Step 6: *Calculate an upper bound for Ψ_0 which is the value of the barrier function after outer iteration in which μ is reduced to $\mu^+ = (1-\theta)\mu$, that is, $\Psi_0 = \Psi(v^+) = \Psi\left(\frac{v}{\sqrt{1-\theta}}\right)$.*

The upper bound is given in Corollary 5.2 in [22]

$$\Psi_0 \leq n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right) \leq \frac{n}{2}\psi''(1)\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}-1\right)^2. \tag{41}$$

We have the following derivation

$$\begin{aligned}
\Psi_0 &\leq \tfrac{n}{2}\tfrac{5}{2}\left(\tfrac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}-1\right)^2 \\
&= \tfrac{5n}{4(1-\theta)}\left(\varrho\left(\tfrac{\tau}{n}\right)-\sqrt{1-\theta}\right)^2 \\
&\leq \tfrac{5n}{4(1-\theta)}\left(1+\sqrt{\tfrac{2\tau}{n}}-\sqrt{1-\theta}\right)^2 \\
&\leq \tfrac{5n}{4(1-\theta)}\left(\theta+\tfrac{\sqrt{2\tau}}{\sqrt{n}}\right)^2 \\
&= \tfrac{5}{4(1-\theta)}\left(\theta\sqrt{n}+\sqrt{2\tau}\right)^2.
\end{aligned} \tag{42}$$

The first inequality follows from (41) by substituting $\psi''(1) = \frac{5}{2}$ from (21). The second inequality follows from (35) while the third inequality is the result of the well-known inequality $1 - \sqrt{1-\theta} = \frac{\theta}{1+\sqrt{1-\theta}} \leq \theta$.

Step 7: *Calculate an upper bound for the total number of iterations N by substituting the results of Step 5 and Step 6 into the expression*

$$N \leq \frac{\Psi_0^{\gamma}}{\theta\beta\gamma} \log \frac{n}{\epsilon}.$$

Hence, we have the following derivation of the upper bound of iterations of the Algorithm

$$\begin{aligned} N &\leq \tfrac{\Psi_0}{\theta \frac{1}{148}} \log \tfrac{n}{\epsilon} \\ &\leq \tfrac{148}{\theta} \tfrac{5}{4(1-\theta)} \left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \tfrac{185}{\theta(1-\theta)} \left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2 \log \tfrac{n}{\epsilon}, \end{aligned} \tag{43}$$

where the first inequality is due to (40) and the second inequality is due to (42).

Step 8: *Set $\tau = \Theta(n)$ and $\theta = \Theta(1)$ to calculate an iteration bound for long-step method, and set $\tau = \Theta(1)$ and $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$ to calculate an iteration bound for short-step method.*

Short-step method: Substituting $\tau = \Theta(1)$ and $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$ into (43) we obtain

$$\begin{aligned} N &\leq \tfrac{185}{\theta(1-\theta)} \left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \frac{185}{\Theta\left(\frac{1}{\sqrt{n}}\right)\left(1-\Theta\left(\frac{1}{\sqrt{n}}\right)\right)} \left(\Theta\left(\tfrac{1}{\sqrt{n}}\right)\sqrt{n} + \sqrt{2\Theta(1)}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \frac{185}{\Theta(1)\left(\frac{1}{\sqrt{n}}\right)\left(1-\Theta(1)\left(\frac{1}{\sqrt{n}}\right)\right)} \left(\Theta(1)\left(\tfrac{1}{\sqrt{n}}\right)\sqrt{n} + \sqrt{2\Theta(1)}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \tfrac{185}{\Theta(1)\Theta(1)} \left(\Theta(1) + \Theta(1)\right)\sqrt{n} \log \tfrac{n}{\epsilon} \\ &= \tfrac{185}{\Theta(1)}\Theta(1)\sqrt{n} \log \tfrac{n}{\epsilon}. \end{aligned} \tag{44}$$

Hence, for short-step method we match the best possible iteration bound

$$N = O\left(\sqrt{n} \log \frac{n}{\epsilon}\right). \tag{45}$$

Long-step method: Substituting $\tau = \Theta(n)$ and $\theta = \Theta(1)$ into (43) we obtain

$$\begin{aligned} N &\leq \tfrac{185}{\theta(1-\theta)} \left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \tfrac{185}{\Theta(1)(1-\Theta(1))} \left(\Theta(1)\sqrt{n} + \sqrt{2\Theta(n)}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \tfrac{185}{\Theta(1)\Theta(1)} \left(\Theta(1)\sqrt{n} + \Theta(1)\sqrt{n}\right)^2 \log \tfrac{n}{\epsilon} \\ &= \tfrac{185}{\Theta(1)}\Theta(1) n \log \tfrac{n}{\epsilon}. \end{aligned} \tag{46}$$

Hence, for long-step method we get the following iteration bound

$$N = O\left(n \log \frac{n}{\epsilon}\right). \tag{47}$$

Hence, we conclude that the iteration bounds of the long- and short-step methods are of the same order of magnitude as for the classical logarithmic barrier function.

Remark 4.2
In [29] it is shown that a long-step IPM based on the SRKF with a barrier term $q \geq 1$ has an $O\left(n^{\frac{q+1}{2q}} \log \frac{n}{\varepsilon}\right)$ iteration bound while the iteration bound of short-step IPMs is the same as for the short-step IPMs based on EKFs, $O\left(\sqrt{n} \log \frac{n}{\varepsilon}\right)$.

Since the new kernel function (18) is SRKF with $q = 1$, the iteration bound of the large-step version of the Algorithm is $O\left(n \log \frac{n}{\varepsilon}\right)$ confirming the result (47) obtained when considering new kernel function (18) as an EKF.

Remark 4.3
In this remark we discuss an interesting observation regarding the complexity of the long-step version of the Algorithm. In Step 1 of the analysis of the Algorithm in Section 4, we have to solve the equation

$$\frac{2}{t^2 + t} = 2s + t, \tag{48}$$

for t or, if it is hard to solve the equation, find the lover bound on t. We can do that in a different way than in Step 1 in Section 4. The left-hand side of the equation (48) can be transformed to

$$\frac{2}{t^2 + t} = \frac{2}{t^2(1 + 1/t)}.$$

Since $t \in (0, 1]$, we can assume that there exists $t_0 > 0$, however small, such that $t \in [t_0, 1]$. If we denote $\lambda = 1 + 1/t_0$, then $1 + 1/t \leq \lambda$, hence,

$$\frac{2}{t^2(1 + 1/t)} \geq \frac{2}{\lambda t^2}.$$

Thus, from (48) we can derive the following inequality

$$\frac{2}{\lambda t^2} \leq 2s + 1,$$

where the right side of the inequality follows from the fact that in (48) $2s + t \leq 2s + 1$ for $t \leq 1$. Hence, we have the following lower bound on $t = \rho(s)$ that is different than the one in (28) and depends on the parameter λ

$$t = \rho(s) \geq \sqrt{\frac{2}{\lambda(2s + 1)}}.$$

If we follow the steps of the analysis of the Algorithm as in Section 4, after some calculations, we would derive the iteration bound for long-step version of the Algorithm to be $O\left(\lambda\sqrt{n} \log \frac{n}{\epsilon}\right)$.

Hence, with this derivation, the iteration bound of the long-step version of the Algorithm is of the same order of magnitude as the iteration bound for the short-step version. However, this derivation has a serious drawback, it depends on the constant λ which can be fixed arbitrarily and, hence, can be exponentially large if the t_0 is exponentially small. Introducing such a constant for even well structured problems such as LO makes this derivation obsolete.

However, in our opinion, this observation still has value because it gives insight into finding and/or creating a kernel function for which short- and long-step versions of the related IPM would have iteration bounds of the same order of magnitude. This is an interesting topic for further research.

5. Numerical results

Numerical experiments were implemented in Python and carried out on a regular PC operating on a 64-bit Operating System Windows 10 machine, lntel(R) Core(TM) i7-10700K CPU @ 3.80GHz, 16.0 GB RAM.

5.1. Numerical results for CTA Model

5.1.1. The CTA Model

In this subsection we provide the formulation of CTA problem as an important example of the conic problem to which the IPM developed in this paper can be efficiently applied.

The following CTA formulation is given in [24] and several other papers: Given the following set of parameters:

(i) A set of cells $a_i, i \in \mathcal{N} = \{1, \ldots, n\}$. The vector $a = (a_1, \ldots, a_n)^T$ satisfies a certain linear system $Aa = b$ where $A \in \mathbf{R}^{m \times n}$ is an $m \times n$ matrix and $b \in \mathbf{R}^m$ is m-vector. The system usually describes the fact that the sum of elements in each row and column should remain unchanged, i.e. constant.
(ii) A lower, and upper bound for each cell, $l_{a_i} \leq a_i \leq u_{a_i}$ for $i \in \mathcal{N}$, which are considered known by any attacker.
(iii) A set of indices of sensitive cells, $\mathcal{S} = \{i_1, i_2, \ldots, i_s\} \subseteq \mathcal{N}$.
(iv) A lower and upper protection level for each sensitive cell $i \in \mathcal{S}$ respectively, lpl_i and upl_i, such that the released values must be outside of the interval $(a_i - lpl_i,\ a_i + upl_i)$.
(v) A set of weights, $w_i, i \in \mathcal{N}$ used in measuring the deviation of the released data values from the original data values.

A CTA problem is a problem of finding values $z_i,\ i \in \mathcal{N}$, such that $z_i,\ i \in \mathcal{S}$ are safe values and the weighted distance between released values z_i and original values a_i, denoted as $\|z - a\|_{l(w)}$, is minimized, which leads to solving the following optimization problem

$$\begin{array}{ll} \min\limits_{z} & \|z - a\|_{l(w)} \\ s.t. & Az = b, \\ & l_{a_i} \leq z_i \leq u_{a_i},\ i \in \mathcal{N}, \\ & z_i,\ i \in \mathcal{S} \text{ are safe values.} \end{array} \tag{49}$$

As indicated in the assumption (iv) above, safe values are the values that satisfy

$$z_i \leq a_i - lpl_i \text{ or } z_i \geq a_i + upl_i,\ i \in \mathcal{S}. \tag{50}$$

By introducing a vector of binary variables $y \in \{0, 1\}^s$, the constraint (50) can be written as

$$\begin{array}{lll} z_i \geq & -L\,(1 - y_i) + (a_i + upl_i)\,y_i, & i \in \mathcal{S}, \\ z_i \leq & Ly_i + (a_i - lpl_i)\,(1 - y_i)\,, & i \in \mathcal{S}, \end{array} \tag{51}$$

where $L \gg 0$ is a large positive number. Constraints (51) enforce the upper safe value if $y_i = 1$ or the lower safe value if $y_i = 0$.

Replacing the last constraint (49) in the CTA model with constraints in (51) leads to a mixed integer convex optimization problem (MIOP) which is, in general, a difficult problem to solve; however, it provides solutions with high data utility [11]. The alternative approach is to fix binary variables up front, which leads to a CTA that is a continuous convex optimization problem because all binary variables are replaced with values 0 or 1. The continuous CTA is easier to solve; however, the obtained solution may have a lower data utility because the optimal solution of the continuous CTA is either a feasible or infeasible solution of the corresponding MIOP depending on the values that were assigned to the binary variables. The strategies on how to avoid a wrong assignment of binary variables that may result in the MIOP being infeasible are discussed in [12, 13].

In what follows, we consider a continuous CTA where binary variables in MIOP are fixed with certain values of 0 or 1, and vector z is replaced by the vector of *cell deviations* $x = z - a$. Then, the CTA (49) reduces to the following convex optimization problem:

$$\begin{array}{ll} \min\limits_{x} & \|x\|_{l(w)} \\ s.t. & Ax = 0, \\ & l \leq x \leq u\,, \end{array} \tag{52}$$

where upper and lower bounds for $x_i,\ i \in \mathcal{N}$ are defined as follows:

$$l_i = \begin{cases} upl_i & \text{if } i \in \mathcal{S} \text{ and } y_i = 1 \\ l_{a_i} - a_i & \text{if } (i \in \mathcal{N} \setminus \mathcal{S}) \text{ or } (i \in \mathcal{S} \text{ and } y_i = 0) \end{cases} \tag{53}$$

$$u_i = \begin{cases} -lpl_i & \text{if } i \in \mathcal{S} \text{ and } y_i = 0 \\ u_{a_i} - a_i & \text{if } (i \in \mathcal{N} \setminus \mathcal{S}) \text{ or } (i \in \mathcal{S} \text{ and } y_i = 1)\,. \end{cases} \tag{54}$$

The two most commonly used norms in (52) are the ℓ_1 and ℓ_2 norms. For the ℓ_2-norm, (52) reduces to the following ℓ_2-CTA model:

$$\begin{array}{ll} \min\limits_{x} & \sum\limits_{i=1}^{n} w_i x_i^2 \\ s.t. & Ax = 0, \\ & l \le x \le u\,. \end{array} \tag{55}$$

The above problem is a standard QO problem that can be efficiently solved using IPM or other methods.

For the ℓ_1-norm, (52) reduces to the following ℓ_1-CTA model:

$$\begin{array}{ll} \min\limits_{x} & \sum\limits_{i=1}^{n} w_i \left|x_i\right| \\ s.t. & Ax = 0, \\ & l \le x \le u\,. \end{array} \tag{56}$$

The ℓ_1-CTA model (56) above is a convex optimization problem; however, the objective function is not differentiable at $x = 0$. Since most of the algorithms, including IPMs, require differentiability of the objective function, (56) needs to be reformulated.

The standard reformulation is the transformation of (56) into the following LO model:

$$\begin{array}{ll} \min\limits_{x^-,x^+} & \sum\limits_{i=1}^{n} w_i \left(x_i^+ + x_i^-\right) \\ s.t. & A\left(x_i^+ - x_i^-\right) = 0, \\ & l^+ \le x^+ \le u^+, \\ & l^- \le x^- \le u^-, \end{array} \tag{57}$$

where

$$x^+ = \begin{cases} x & \text{if } x \ge 0 \\ 0 & \text{if } x < 0, \end{cases} \qquad x^- = \begin{cases} 0 & \text{if } x > 0 \\ -x & \text{if } x \le 0. \end{cases} \tag{58}$$

The drawback of the LO reformulation above is that the number of variables and inequality constraints doubles. In [24] an alternative SOC reformulation of ℓ_1-CTA is proposed where the dimension of the problem does not increase as much. It is based on the fact that the absolute value has an obvious SOC representation since the epigraph of the absolute value function is exactly SOC, that is,

$$t_i = |x_i| \quad \longrightarrow \quad \mathcal{K}_i = \left\{(x_i, t_i) \in \mathbf{R}^2 \ : \ t_i \ge \sqrt{x_i^2}\right\}.$$

A SOC formulation of the l_1-CTA (56) is given below

$$\begin{array}{ll} \min\limits_{x} & \sum_{i=1}^{n} w_i t_i \\ s.t. & Ax = 0, \\ & (x_i, t_i) \in \mathcal{K}_i; \quad i = 1, \ldots, n\,, \\ & l \le x \le u\,. \end{array} \tag{59}$$

The IPM developed in this paper can be directly applied to the LO formulation in (57) of ℓ_1-CTA because LO can straightforwardly be formulated as a monotone LCP or, alternatively the Algorithm can easily be modified to solve a LO formulation directly. In the next subsection, we implement the method on a CTA test example used in [24] and several other papers. These numerical results show that the Algorithm is a viable option for solving CTA problems, however, more sophisticated implementation, extensive numerical testing, and comparison with other methods, are needed to draw more definite conclusion.

5.1.2. Numerical results for the test example

We first consider the test example of a table considered in Figure 1 (table (a)) in [24] which is also listed as table (a) in Figure 2 below.

Original table

$\mathbf{10}_{(\mathbf{3})}$	15	11	9	45
8	10	12	15	45
10	12	11	$\mathbf{13}_{(\mathbf{5})}$	46
28	37	34	37	136

(a)

Safe table: LO ℓ_1-IPM

13.11	15.30	11.25	5.34	45
8.19	10.60	12.55	13.66	45
6.70	11.10	10.20	**18**	46
28	37	34	37	136

(b)

Safe table: LO ℓ_1-Simplex

13	15	11	6	45
10	10	12	13	45
5	12	11	**18**	46
28	37	34	37	136

(c)

Figure 2. Results of the test example (rounded to two decimal places).

The continuous CTA model based on the table (a) is formulated in the following way:

- The linear constraints are obtained from the requirement that the sum of the elements in each row (or column) remains constant and is equal to the corresponding component in the last column (or row) of table (a).
- The sensitive cells are cells a_1 and a_{12}. For both of them the upper safe values are enforced, which are listed in the parentheses in the lower right corners of the cells, $upl_1 = 3$ and $upl_{12} = 5$ respectively. Hence, in the transformed tables the upper safe value of the cell a_1 should be 13 or above and for a_{12} the upper safe value should be 18 or above.
- For the nonsensitive cells the lower and upper bounds are set to be zero and positive infinity respectively, that is, $l_{a_i} = 0$ and $u_{a_i} = \inf$ for $i = 2, \ldots, 11$.
- The weights in the objective function are set to have the value one, that is, $w_i = 1$ for $i = 1, \ldots, 12$.

Table (b) is a safe table obtained by solving LO formulation of ℓ_1-CTA using the Algorithm in Figure 1. For comparison we also list the safe table (c) obtained by solving it using the Simplex method in MOSEK solver [2].

5.2. Numerical results for a set of randomly generated monotone LCPs

5.2.1. Randomly generated monotone LCPs

A monotone LCP is randomly generated in the following way:

1. Randomly generate a $m \times n$ matrix A using random generator subroutine in Python, matrix A = numpy.random.randint(from, to, size=(m,n)).
2. Calculate a matrix $M = A^T A$. By construction matrix M is positive semidefinite.
3. Calculate a vector q such that $q = e - Me = (I - M)e$, where vector e is a n-vector of all ones, and I is a $n \times n$ identity matrix. This selection of vector q ensures that $x = e,\ s = e$ is an obvious solution of the LCP, hence, the feasible region of the LCP is not empty.

We consider problems of dimension $n = 10, 20, 50, 100$.

5.2.2. Numerical results for monotone LCPs

A set of randomly generated monotone LCPs described in the previous subsection is solved using the Algorithm with two EKFs, the new EKF (18), and the classical logarithmic kernel function (17). The classical logarithmic

kernel function is used in most implementations of IPMs in optimization software packages and serves as a benchmark for comparison with IPMs based on different kernel functions.

The point $x = e,\ s = e$ is an obvious starting point for the Algorithm because $\mu^0 = 1$ and $v^0 = e$ implying that $x = e,\ s = e$ is a μ-center, i.e., lies on the central path of the LCP.

We have used the following values of the parameters:

- *Accuracy parameter:* We used two values, $\varepsilon = 0.001$ (lower accuracy) and $\varepsilon = 0.00001$ (higher accuracy).
- *Threshold parameter:* We used two values $\tau = 3$ (narrower neighborhood) and $\tau = 10$ (wider neighborhood).
- *Barrier parameter:* For the long-step method we used two values $\theta = 0.9$ (more aggressive steps) and $\theta = 0.5$ (less aggressive steps). For the short-step method we only used $\theta = \frac{1}{\sqrt{n}}$.
- *Step size:*
  *D*efault step size: From (32), the lower bound for a step size is $\alpha = \frac{1}{1+4(1+4\delta)^2}$ which is then taken as a default step size. The default step size for classical logarithmic kernel function is exact, $\alpha = \frac{1}{1+\left(2\delta+\sqrt{1+4\delta^2}\right)^2}$, see [7].
  Practical step size: We have also tried an aggressive step size that is practical but does not guarantee the convergence of the Algorithm. In essence, it is a scaled minimal ratio type step which is obtained as follows. First, calculate $\alpha_x = \left\{-\frac{x_i}{\Delta x_i} \ :\ \Delta x_i < 0\right\}$ and $\alpha_s = \left\{-\frac{s_i}{\Delta s_i} \ :\ \Delta s_i < 0\right\}$. Next, calculate $\alpha_{\max} = \min\{\alpha_x,\ \alpha_s\}$, and then finalize the calculation of the step size $\alpha = \gamma\alpha_{\max}$, where $\gamma = 0.9$, or $\gamma = 0.95$, or even $\gamma = 0.99$, assuring that we slightly step back from the boundary of the feasible region. We call this step size *maximal step size* and denote it as $\max\alpha$.

In Table 1, the number of iterations and the CPU times for both EKFs and both choices of step sizes are given. A few comments on the results shown in the table are in order.

- The number of iterations for the default step size for the new kernel function is significantly higher than for the classical logarithmic kernel function. The reason is that the default step size is calculated exactly for the classical kernel function while for the new kernel function we use a lower bound that may not be the best estimate of the step size, however, it does serve the purpose of deriving good theoretical iteration bounds for the new kernel function.
- In the first four rows of the Table 1, LCPs were solved by the long-step method using an aggressive reduction of μ with $\theta = 0.9$, and moderately aggressive reduction with $\theta = 0.5$. In the last row of the table a short-step method was used with θ depending on the dimension of the problem, hence, the reduction of μ at each iteration is smaller.
- If we look at the first two rows of the table, we can see that the number of iterations is smaller for $\theta = 0.5$ because we expect the value of the barrier function after the μ-update to increase less, hence, also deviates less from the central path than for $\theta = 0.9$, requiring less inner iterations necessary to return to the τ-neighborhood of the central path.
- Comparing the first and third rows of the table, we can observe that a wider τ-neighborhood for $\tau = 10$ requires more iterations than the narrower τ-neighborhood for $\tau = 3$.
- Finally, the third and fourth rows of the table show that in the case of higher accuracy $\varepsilon = 0.00001$, the Algorithm requires more iterations than in the case of the lower accuracy $\varepsilon = 0.001$, which is to be expected.
- The columns with α_{max} in the table represent the number of iterations when the practical (aggressive) step size is used that converges to the ε-approximate solution of LCP significantly faster than the default step size. However, it does not guarantee convergence. We can observe that in this case long-step methods are much faster than the short-step methods, which is expected and observed in many, if not all, practical implementations of these types of IPMs [2]. It is also worth noting that for α_{max} step size the number of iterations for the classical kernel function and new kernel function are identical in almost all cases except one where they differ for only one iteration. The related CPU times show more differences, albeit small ones, than the iteration counts. Actually, in several instances the CPU times of IPM based on a new kernel function are slightly better than for the IPMs based on the classical logarithmic kernel function. Hence, the last two

columns of the table show that with aggressive step sizes, IPMs based on the new kernel function behave equally as good as IPMs with classical logarithmic kernel function.

θ	τ	ϵ	n	# iterations CPU time (sec)			
				New Ψ Default step size	Classical Ψ Default step size	New Ψ α max	Classical Ψ α max
0.9	3	0.001	10	11926 1.03125	2693 0.234375	7 0.0	7 0.0
			20	24185 3.984375	5568 0.984375	9 0.0	9 0.0
			50	47518 31.96875	11146 8.375	9 0.0	9 0.015625
			100	83333 249.28125	19875 55.578125	10 0.015625	10 0.015625
0.5	3	0.001	10	4683 0.40625	980 0.09375	14 0.0	15 0.0
			20	7303 1.21875	1555 0.25	15 0.0	15 0.0
			50	13438 8.609375	2933 2.3125	16 0.015625	16 0.0
			100	22676 70.40625	5051 12.921875	17 0.015625	17 0.125
0.9	10	0.001	10	19900 1.71875	4619 0.40625	7 0.0	7 0.0
			20	36685 6.171875	8585 1.515625	9 0.0	9 0.0
			50	63781 43.125	15138 10.703125	8 0.0	8 0.0
			100	104006 297.796875	24961 71.28125	9 0.0	9 0.078125
0.9	10	0.00001	10	32524 2.859375	7548 0.6875	10 0.0	10 0.015625
			20	53907 9.234375	12623 2.140625	11 0.0	11 0.0
			50	92695 64.15625	22026 14.453125	10 0.109375	10 0.015625
			100	149994 452.671875	35988 104.734375	11 0.0	11 0.0
$1/\sqrt{n}$	3	0.001	10	3747 0.3125	767 0.078125	25 0.0	25 0.0
			20	5099 0.859375	1045 0.171875	40 0.015625	40 0.0
			50	7904 6.15625	1627 0.921875	71 0.0	71 0.0
			100	11469 33.625	2362 7.40625	110 0.125	110 0.03125

Table 1. Results of solving monotone LCPs using Algorithm.

The numerical results presented in Table 1 are for illustration purposes, however, they still show that with aggressive step size, IPMs with new kernel function and classical logarithmic kernel function are comparable. A more sophisticated implementation that includes a procedure to select a starting point, detecting and handling the ill-conditioning, taking advantage of sparsity and block structure of problems, extensive numerical testing on a larger set of test problems, and comparisons with IPMs based on other kernel functions are needed to draw more definite conclusions about the computational behavior of the Algorithm. This is an interesting and large enough topic for future research whose results we intend to present in a separate paper.

6. Concluding remarks

In this paper, we consider a monotone LCP which is the most commonly used class of LCPs. The method of solving a monotone LCP is a feasible barrier-based IPM, the outline of which is given in Figure 1.

The class of EKFs was introduced in [7] to improve the theoretical performance of long-step versions of kernel-based IPMs. It is well known that most long-step methods perform better in practice but have worse theoretical iteration bounds than short-step methods. The best iteration bound for most short-step methods is $O(\sqrt{n}\log\frac{n}{\epsilon})$ while the best iteration bound for long-step methods is $O(\sqrt{n}\log n\log\frac{n}{\epsilon})$ which is achieved for several specific EKFs [7, 22]. Similar results were obtained for the class of SRKFs introduced in [30, 29].

We introduced a new kernel function (18) with logarithmic barrier term that is different than the barrier term $-\ln t$ of the classical logarithmic kernel function (17). It is proved that the new kernel function is EKF adding to the small pool of the EKFs with pure logarithmic barrier term. It is shown in Remark 4.2 that the new kernel function is also SRKF.

The generic barrier-based IPM presented in Figure 1 is adapted for the barrier function based on the new kernel function and is called throughout the text simply the Algorithm. The Algorithm was analyzed and iteration bounds for short- and long-step versions of the Algorithm were derived. It is shown that the iteration bounds of large- and short-step versions have the same order of magnitude as large- and short-update versions of the IPM based on the classical logarithmic kernel function, namely $O(\sqrt{n}\log\frac{n}{\epsilon})$ for short-step version and $O(n\log\frac{n}{\epsilon})$ for long-step version of the Algorithm. These results were confirmed by the fact that the new kernel function is also SRKF.

It is important to mention that the goal of the paper is mostly theoretical, to introduce a new kernel function, examine its properties as EKF and SRKF, analyze the IPM based on this new kernel function, and derive the iteration bounds for short-step and long-step versions of the Algorithm.

The numerical experiments in Section 5 are limited to show the validity of the Algorithm. The Algorithm is applied to solve a continuous CTA problem which is an important Statistical Disclosure Limitation (SDL) model for the protection of tabular data. Numerical results on a standard test example in Figure 2 show that this algorithm is a viable option to the existing methods for solving continuous CTA. We also applied the Algorithm to a small set of randomly generated monotone LCPs. Numerical results are presented in Table 1 and show that the Algorithm performs well on this small set of test problems. Furthermore, in the same table, we compared the Algorithm with the IPM based on the classical logarithmic barrier function, which is a benchmark kernel function used in most implementations of IPM, and showed that for the practical, aggressive, step-sizes they perform equally as good. However, more extensive numerical testing on a larger set of test problems, and comparisons with IPMs based on other kernel functions, are needed to draw more definite conclusions about the behavior of the Algorithm. This is an interesting topic for future computational research.

Other possible research directions, that are more theoretical, include modifications of the Algorithm to solve different formulations of the CTA model as well as more general classes of LCPs, such as sufficient LCPs and LCPs over symmetric cones. An interesting but difficult topic for future research is connected to the observation in Remark 4.3 that may be helpful in searching for the the kernel function for which short- and long-step IPMs have the same complexity, namely $O(\sqrt{n}\log\frac{n}{\epsilon})$.

Acknowledgment

This paper is dedicated to Professor Sanjo Zlobec on the occasion of his 80th Birthday.

Statements and Declarations

Disclaimer:
The findings and conclusions in this paper are those of the authors and do not necessarily represent the official position of the Centers for Disease Control and Prevention.

Data availability statement:
The authors confirms that all data generated or analyzed during this study are included in this paper.

Funding and/or Competing interests:
No funding was received to assist with the preparation of this manuscript.

REFERENCES

1. W. Ai and S. Zhang. A $O(L\sqrt{n})$ iteration primal-dual path-following method, based on wide neighborhoods and large updates, for monotone LCP. *SIAM J. Optim.*, 16: 400–417, 2005.
2. E. D. Andersen, *MOSEK guide*. *h*ttps://www.mosek.com/documentation/2022
3. E. D. Andersen and Y. Ye. On a homogeneous algorithm for the monotone complementarity problem. *Math. Program.*, 84: 375-399, 1999.
4. M. Anitescu, G. Lesaja, and F. A. Potra. An infesible interior-point predictor-corrector algorithm for the P_*-geometric LCP. *Appl. Math. Optim.*, 36:203–228, 1997.
5. M. Anitescu, G. Lesaja, and F. A. Potra. Equivalence between different formulations of the linear complementarity problem. *Optim. Methods Softw.*, 7:265-290, 1997.
6. Y. Q. Bai and C. Roos. A polynomial-time algorithm for linear optimization based on a new simple kernel function. *Optim. Methods Softw.*, 18(6):631-646, 2003.
7. Y. Q. Bai, M. El Ghami, and C. Roos. A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization. *SIAM J. Optim.*, 15(1): 101-128, 2004.
8. Y. Q. Bai, G. Q. Wang, and C. Roos. A new Primal-Dual Interior-Point Algorithm for Second-Order Cone Optimization. *Optim. Methods Softw.*, 11-12:141–182, 2005.
9. Y. Q. Bai, G. Lesaja, C. Roos and G. Q. Wang. A Class of Large- and short-step Primal-dual Interior-point Algorithms for Linear Optimization. *J. Optim. Theory Appl.*, 138(3):341-359, 2005.
10. J. Castro. Minimum-distance controlled perturbation methods for large-scale tabular data protection. European J. Oper. Res., 171, 39-52, 2006.
11. J. Castro and J. A. Gonzalez. Assessing the information loss of controlled adjustment methods in two-way tables. *Privacy in Statistical Databases 2014*, LNCS, 8744, 79-88, 2014.
12. J. Castro and J. A. Gonzalez. A fast CTA method without complicating binary decisions. Documents of the Joint UNECE / Eurostat Work Session on Statistical Data Confidentiality, Statistics Canada, Ottawa, 1-7, 2013
13. J. Castro and J. A. Gonzalez. A multiobjective LP approach for controlled tabular adjustment in statistical disclosure control. Working paper, Department of Statistics and Operations Research, Universitat Politecnica de Catalunya, 2014.
14. G. -M. Cho and M. -K. Kim. A new large-update interior point algorithm for $P_*(\kappa)$-LCPs based on kernel functions. *Appl. Math. Comput.*, 182: 1169–1183, 2006.
15. S. J. Chung. NP-completeness of the linear complementarity problem. *J. Optim. Theory Appl.*, 160(3): 393–399, 1989.
16. R. W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, MA, 1992.
17. R. A. Dandekar and L. H. Cox. Synthetic tabular Data: an alternative to complementary cell suppression. Manuscript, U.S. Energy Information Administration, 2002.
18. L. Derbal and Z. Kebbiche. Theoretical and numerical result for linear optimization problem based on a new kernel function. *J. Sib. Fed. Univ. Math. Phys.*, 12: 160–172, 2019.
19. M. El Ghami, Z. Guennoun, S. Bouali, and T. Steihaug. Interior-point methods for linear optimization based on a kernel function with a trigonometric barrier term. *J. Comput. Appl. Math.*, 236(15): 3613-3623, 2012.
20. F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problem*. Springer, New York, 2003.
21. M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A Unified Approach to Interior Point Algorithms for Linear Complementarity problems.* Lecture Notes in Computer Science 538, Springer-Verlag, New York, 1991.
22. G. Lesaja and C. Roos. Unified analysis of kernel based interior-point methods for $P(\kappa)$-linear complementarity problems. *SIAM J. Optim.*, 20(6): 1014-1039, 2010.
23. G. Lesaja, G. Q. Wang, and D. T. Zhu. Interior-point methods for Cartesian $P_*(\kappa)$-linear complementarity problems over symmetric cones based on the eligible kernel functions. *Optim. Methods Softw.*, 27: 827–843,. 2012
24. G. Lesaja, J. Castro, and A. Oganian. Second Order Cone formulation of Continuous CTA Model. Lecture Notes in Computer Science 9867, Springer., 41-53, 2016.
25. N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pages 131–158. Springer Verlag, New York, 1989. Identical version in : *Proceedings of the 6th Mathematical Programming Symposium of Japan, Nagoya, Japan*, pages 1–35, 1986.

26. R. D. C. Monteiro and S. J. Wright. A superlinear infeasible-interior-point affine scaling algorithm for LCP. *SIAM J. Optim.*, 6: 1-20, 1996.
27. J. Peng, C. Roos, and T. Terlaky. New complexity analysis of primal-dual Newton methods for $P_*(\kappa)$ - nonlinear complementarity problems. In H. Frenk, C. Roos, T. Terlaky and S. Zhang, eds., *High Performance Optimization,* pages 245-265, Kluwer Academic Publication, Boston, 1999.
28. J. Peng, C. Roos, and T. Terlaky. Self-regular functions and new search directions for linear and semidefinite optimization. *Math. Program., Ser. A,* 93:129-171, 2002.
29. J. Peng, C. Roos, and T. Terlaky. *Self-Regularity: A New Paradigm for Primal-Dual Interior-Point Algorithms*. Princeton University Press, 2002.
30. J. Peng, C. Roos and T. Terlaky. New complexity analysis of primal-dual Newton methods for $P_*(\kappa)$ linear complementarity problems. In: H. Frenk, C. Roos, T. Terlaky and S. Zhang (editors), High Performance Optimization, pp. 245–265, Kluwer Academic Publishers, Boston, MA, 1999.
31. X. Liu and F. A. Potra. Corrector-predictor methods for sufficient linear complementarity problems in a wide neighborhood of the central path. *SIAM J. Optim.*, 17: 871–890, 2006.
32. J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization.* MPS/SIAM Ser. Optim., SIAM, Philadelphia, 2001.
33. C. Roos, T. Terlaky, and J. Ph. Vial. *Theory and Algorithms for Linear Optimization. An Interior-Point Approach.* Springer Science, 2005.
34. H. Väliaho. P_*-matrices are just sufficient. *Linear Algebra Appl.*, 239: 103–108, 1996.
35. G. Q. Wang, Y. Q. Bai, and C. Roos. Primal-Dual Interior-Point Algorithms for Semidefinite Optimization Based on a Simple Kernel Function. *J Math Model Algor.*, 4:409-433, 2005.
36. S.J. Wright and Y. Zhang. A superquadratic infeasible-interior-point method for linear complementarity problems. *Math. Program.*, 73: 269-289, 1996.
37. Y. Ye. On homogeneous and self-dual algorithms for LCP. *Math. Program.*, 76: 211-221, 1997.
38. P. A. Zhang and X. S. Li. An infeasible-start path-following method for monotone LCPs. *Math. Comput. Model.*, 38: 23-31, 2003.