Appl. Numer. Math. 220 (2026) 144–166

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Research Paper

A feasible interior-point method with full-Newton step for $P_*(\kappa)$-weighted linear complementarity problem via the algebraically equivalent transformation

Xiaoni Chi [a,b,c], Lin Gan [a,b], Zhuoran Gao [a,c], Jein-Shan Chen [d,*]

[a] School of Mathematics and Computing Science, Guilin University of Electronic Technology, 541004, Guilin, China
[b] Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, 541004, Guilin, China
[c] Center for Applied Mathematics of Guangxi (GUET), 541004, Guilin, China
[d] Department of Mathematics, National Taiwan Normal University, 116059, Taipei, Taiwan

ARTICLE INFO

Keywords:
$P_*(\kappa)$-weighted linear complementarity problem
Feasible interior-point method
Full-Newton step
Algebraically equivalent transformation
Polynomial complexity

ABSTRACT

This paper investigates feasible interior-point method (IPM) with full-Newton step for $P_*(\kappa)$-weighted linear complementarity problem (WLCP). In particular, by applying the algebraically equivalent transformation (AET) for linear optimization, we obtain the new search directions by solving the perturbed Newton system. The AET of the Newton system is based on the kernel function $\varphi(t) = t - \sqrt{t}$, which is used for solving WLCP for the first time. At each iteration, our algorithm takes only full-Newton steps. Therefore, no line-searches are needed to update the iterates. We show the strict feasibility of the full-Newton step and the polynomial iteration complexity of our algorithm under suitable assumptions. Some numerical experiments demonstrate the effectiveness of the proposed algorithm.

1. Introduction

For a given vector $q \in \mathbb{R}^n$ and a given matrix $M \in \mathbb{R}^{n\times n}$, the weighted linear complementarity problem (WLCP) [1] is to find vectors $(x, s) \in \mathbb{R}^n \times \mathbb{R}^n$ satisfying

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

where $\omega \geq 0$ is a weight vector, and xs represents the componentwise (Hadamard) product of vectors x and s. It is clear that the WLCP (1) reduces to the linear complementarity problem (LCP) [2] with $\omega = 0$. The importance of the WLCP lies on the fact that it can be applied to a large class of problems from economics, engineering and science, such as the Fisher market equilibrium problem [1], Arrow-Debreu competitive market equilibrium [3], the linear programming and weighted centering problem (LPWC) [4], perfect price discrimination market model [5], Eisenberg-Gale market [6] and so on.

For convenience, we denote the feasible region and the solution set of the WLCP (1) by

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

and

$$\mathcal{F}^* := \{(x,s) \in \mathcal{F} : xs = \omega\},$$

* Corresponding author.
E-mail addresses: chixiaoni@126.com (X. Chi), ganlin163@126.com (L. Gan), jschen@math.ntnu.edu.tw (J.-S. Chen).

https://doi.org/10.1016/j.apnum.2025.10.003
Received 11 May 2024; Received in revised form 17 September 2025; Accepted 4 October 2025
Available online 10 October 2025
0168-9274/

respectively. We write the strictly feasible set of the WLCP (1) as

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

Throughout this paper, we always suppose that $\mathcal{F}^0 \neq \emptyset$ and M in WLCP (1) is a $P_*(\kappa)$-matrix. For κ being a nonnegative number, a matrix M is called a $P_*(\kappa)$-matrix, if the following is satisfied:

$$(1+4\kappa)\sum_{i\in I_+} x_i(Mx)_i + \sum_{i\in I_-} x_i(Mx)_i \geq 0, \quad \forall\, x \in \mathbb{R}^n.$$

Here $J = \{1, 2, \cdots, n\}$, $I_+ = \{i \in J : x_i(Mx)_i \geq 0\}$, and $I_- = \{i \in J : x_i(Mx)_i < 0\}$. The set of all P_*-matrices is defined as

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

The P_*-matrix is column sufficient, which was first proved by Kojima et al. [7]. Guu and Cottle [8] presented row sufficiency of a P_*-matrix. Väliaho [9] further showed that a matrix is a P_*-matrix if and only if it is sufficient. We call problem (1) a $P_*(\kappa)$-WLCP if M is a $P_*(\kappa)$-matrix, which is the main target of this paper.

In optimization theory, $P_*(\kappa)$-matrices play a significant role for LCP. LCP with general matrices is NP-complete [10]. Consequently, it makes sense to seek matrix classes M where the related LCPs are solvable in polynomial time [11]. Kojima et al. [7] proved that $P_*(\kappa)$-matrices can ensure the global convergence of IPM for LCP. Monotone LCP is the KKT optimality conditions for convex quadratic programming and linear programming [2]. In addition, many real-world problems can be directly formulated as monotone LCP. $P_*(\kappa)$-LCP includes monotone LCP as a special case, allowing it to accommodate more complex systems and constraints.

IPM is a popular approach for linear optimization due to its well-behaved polynomial complexity [12–14], which has been used to solve optimization problems including semidefinite LCP [15], symmetric optimization [16,45], second-order cone optimization [17,18], etc. Besides, various IPMs have been proposed for the $P_*(\kappa)$-LCP. For instance, Kojima et al. [7] demonstrated the existence of central paths for $P_*(\kappa)$-LCP and extended a primal-dual IPM for linear optimization to $P_*(\kappa)$-LCP for the first time. Miao [19] designed a Mizuno-Todd-Ye (MTY) predictor-corrector method for $P_*(\kappa)$-LCPs, which has the polynomial iteration complexity $O((1+\kappa)\sqrt{n}L)$, where L is the encoding size of the problem. Illés et al. [20] proved that the IPM converges to a maximally complementary solution of $P_*(\kappa)$-LCP in polynomial time. Amini and Peyghami [21] gave a large update IPM for $P_*(\kappa)$-LCP through a kernel function, which has $O\left(q(1+2\kappa)\sqrt{n}(\log n)^{\frac{q+1}{q}}\log(n/\varepsilon)\right)$ iteration bound with parameter $q \geq 1$, $q \in \mathbb{N}$. Gurtuna et al. [22] introduced a corrector-predictor IPM with superlinear convergence for $P_*(\kappa)$-LCP. Lee et al. [23] gave a IPM for $P_*(\kappa)$-LCP using a new class of kernel functions. Wang et al. [24] generalized the primal-dual logarithmic barrier method [25] for linear optimization to $P_*(\kappa)$-LCP, and showed that their algorithm has local quadratic convergence and polynomial iteration complexity $O\left((1+4\kappa)\sqrt{n}\log(n/\varepsilon)\right)$. Lesaja and Potra [26] presented an adaptive full-Newton step infeasible IPM for sufficient horizontal LCP.

The algebraic equivalent transformation (AET) technology is an important way to obtain the search direction of IPM. By transforming the nonlinear equations of the central path using continuously differentiable monotonically increasing functions, new search directions are defined that may yield more efficient algorithms. In the AET technique based on $\varphi(t) = t$, the central path is not transformed. Darvay [27] first proposed the square root function to give the search direction. Subsequently, Darvay et al. [28] proposed an IPM for LP, which has a new direction based on the new function $\varphi(t) = t - \sqrt{t}$. Later, the IPM using AET technology based on $\varphi(t) = t - \sqrt{t}$ has extended to $P_*(\kappa)$-LCP [29,30]. Kheirfam and Haghighi [31] proposed an IPM for $P_*(\kappa)$-LCP, whose search direction is obtained based on AET function $\varphi(t) = \frac{\sqrt{t}}{2(1+\sqrt{t})}$.

In addition, IPMs and smoothing algorithms have been adopted for solving the WLCPs [1,32–36]. For example, Potra [1] gave the general optimization of the Fisher market equilibrium problem, for which a smooth central path and two IPMs for the monotone WLCP were presented. Subsequently, by introducing the notion of the sufficient WLCP, Potra [37] extended the characterization of the sufficient LCP to the sufficient WLCP, and designed a corrector-predictor (CP) IPM for its numerical solution. In 2020, Asadi et al. [32] introduced a full-Newton step IPM for solving the monotone WLCP, and proved the method has the best-known iteration bound for IPMs. Zhang [36] considered a smoothing Newton algorithm for the monotone WLCP and proved that it possesses global and local quadratic convergence. Tang [35] developed a modified damped Gauss-Newton algorithm for solving the non-monotone WLCP with global convergence and local quadratic convergence, based on a derivative-free non-montone line search. Chi et al. [38] studied the existence and uniqueness of weighted horizontal WLCP in the Euclidean Jordan algebras.

Motivated by the aforementioned approaches, this paper aims to extend the full-Newton step IPM for $P_*(\kappa)$-LCP [30] to $P_*(\kappa)$-WLCP. The search directions are obtained by applying the technology of AET to the Newton system. In the AET of the Newton system, we adopt the function $\varphi(t) = t - \sqrt{t}$. We only use full-Newton steps along the search directions, which avoids calculating the step size. Under suitable assumptions, we prove the feasibility of the full-Newton steps, and derive the polynomial complexity of our algorithm for $P_*(\kappa)$-WLCP.

The remainder of this paper is summarized as below. In Section 2, we propose a feasible IPM with full-Newton step for solving $P_*(\kappa)$-WLCP. The feasibility of the full-Newton step and convergence of the algorithm are derived in Section 3. The iteration bound is given in Section 4, and some numerical experiments are implemented in Section 5; some numerical findings are presented therein. The conclusion is drawn in Section 6.

The notations utilized throughout this paper are clarified as follows. Let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{R}_+$ denote the set of nonnegative real numbers. $\mathbb{R}^n$ denotes the n-dimensional vector space, while $\mathbb{R}^{n\times n}$ denotes the space of $n \times n$-dimensional matrices. The nonnegative orthant of $\mathbb{R}^n$ is represented by $\mathbb{R}^n_+$. For $x, s \in \mathbb{R}^n$, xs is the Hadamard product of vectors x and s, i.e.,

$xs = (x_1 s_1, \dots, x_n s_n)^T$; x/s denote the vector with components x_i/s_i, i.e., $\frac{x}{s} = \left(\frac{x_1}{s_1}, \dots, \frac{x_n}{s_n}\right)^T$. The minimal and maximal elements of a vector x are denoted by $\min(x)$ and $\max(x)$, respectively. The Euclidean norm of a vector x is denoted as $\|x\|$, while the infinity norm of x is represented by $\|x\|_\infty$. For an arbitrary function $f : \mathbb{R} \to \mathbb{R}$ and a vector x, $f(x)$ denotes the vector $f(x) = (f(x_1), \dots, f(x_n))^T$. I is the unit matrix and O is the zero matrix. e denotes the all-ones vector whose dimension n is determined by the context.

2. The full-Newton step IPM for $P_*(\kappa)$-WLCP

For subsequent analysis, we recall the concept of the central path and review how to obtain the search directions based on a continuously differentiable function. Given a strictly feasible starting point $(x^0, s^0) \in \mathcal{F}^0$, we denote

$$\omega(t) := (1-t)\omega + t x^0 s^0, \tag{2}$$

where $t \in (0, t_0]$ with $t_0 = 1$. Then, the central path of $P_*(\kappa)$-WLCP (1) is the set of the points $\{(x, s; t) \mid t \in (0, 1]\}$ satisfying

$$s = Mx + q, \quad xs = \omega(t), \quad x \geq 0, \quad s \geq 0. \tag{3}$$

Without loss of generality, we assume that the interior-point condition (IPC) [37] holds, i.e., there exists a pair of vectors $(x^0, s^0) > 0$ such that $s^0 = Mx^0 + q$. Since M is a $P_*(\kappa)$-matrix and the IPC holds, the perturbed system (3) has a unique solution $(x(t), s(t))$ for any $t \in (0, t_0]$ (cf. Potra [37, Proposition 3]), which is called t-center. The central path of $P_*(\kappa)$-WLCP is the set of t-centers, which is defined as

$$C := \{(x(t), s(t)) : t \in (0, 1]\}.$$

Let $t_0 = 1$. The value of t is gradually reduced by multiplying t by a factor $(1-\theta)$ in each iteration. Let $t_+ := (1-\theta)t$ be the value of t updated after one iteration, where $\theta \in (0, 1)$. By decreasing t, we ensure the appropriate adjustment of $\omega(t)$, which allows the IPM to iteratively converge towards a solution. Apparently, when t tends to zero, the central path converges to a solution of $P_*(\kappa)$-WLCP (1) [37].

2.1. Determination of the search direction and the definition of κ'

Next, we elaborate how to obtain the search directions by applying AET, which is an idea from Darvay [27]. Consider a continuously differentiable function $\varphi : (\eta, +\infty) \to \mathbb{R}_+$, where $0 \leq \eta < 1$ is a fixed value. For any vector $v \in \mathbb{R}^n$, we define $\varphi(v)$ as follows:

$$\varphi(v) \in \mathbb{R}^n, \quad [\varphi(v)]_i = \varphi(v_i).$$

Then, the system (3) can be rewritten as

$$\begin{aligned} s &= Mx + q, & x &\geq 0, \\ \varphi\left(\frac{xs}{\omega(t)}\right) &= \varphi(e), & s &\geq 0. \end{aligned} \tag{4}$$

By applying Newton's method to system (4), it yields

$$\begin{aligned} -M\Delta x + \Delta s &= 0, \\ s\Delta x + x\Delta s &= a, \end{aligned} \tag{5}$$

where

$$a := a(t) = \frac{\omega(t)\left(\varphi(e) - \varphi\left(\frac{xs}{\omega(t)}\right)\right)}{\varphi'\left(\frac{xs}{\omega(t)}\right)}.$$

Due to M being a $P_*(\kappa)$-matrix, and x and s are positive vectors, the system (5) uniquely defines $(\Delta x, \Delta s)$.

For notational simplicity and ease of analysis, we introduce the scaled vector and the scaled search directions as

$$v := \sqrt{\frac{xs}{\omega(t)}}, \quad d_x := \frac{v\Delta x}{x}, \quad d_s := \frac{v\Delta s}{s}. \tag{6}$$

Then, combining (5) and (6) leads to

$$\begin{aligned} s\Delta x + x\Delta s &= \omega(t) v (d_x + d_s), \\ d_x d_s &= \frac{\Delta x \Delta s}{\omega(t)}. \end{aligned} \tag{7}$$

In addition, by (6) and (7), the system (5) is recast as

$$\begin{aligned} -\overline{M} d_x + d_s &= 0, \\ d_x + d_s &= p_v, \end{aligned} \tag{8}$$

where

$$d := \sqrt{\frac{x}{s}}, \quad D := \mathrm{diag}(d), \quad W(t) = \mathrm{diag}(\omega(t)), \quad \overline{M} = \sqrt{W^{-1}(t)}DMD\sqrt{W(t)},$$

and

$$p_v := \frac{\varphi(e) - \varphi(v^2)}{v\varphi'(v^2)}.$$

The handicap of a matrix M is defined by $\bar{\kappa} := \inf\{\kappa : M \in P_*(\kappa)\}$. Notice that the scaling of M is not symmetric, i.e, matrix M is multiplied by different matrices from left and right to get matrix $\overline{M}$. Therefore, the handicap of matrix $\overline{M}$ is different from the handicap of matrix M.

Now we are in a position to show that $\overline{M} \in P_*(\kappa')$ and derive the formula for κ'. Let $\omega \in \mathbb{R}^n_+$ be a nonnegative weight vector, where ω_i is the i-th component of ω, $i = 1, 2, \dots, n$. Define

$$J := \{1, 2, \dots, n\}, \; J_0 := \{i \in J : \omega_i = 0\}, \; J_+ := \{i \in J : \omega_i > 0\},$$

and $\chi \in \mathbb{R}^n$, where

$$\chi_i := \begin{cases} 0, & \text{if } i \in J_0; \\ 1, & \text{if } i \in J_+. \end{cases} \tag{9}$$

Then $\omega \in \mathbb{R}^n_+$ can be classified into the following three classes:

i) All components of ω are zero, namely all $i \in J_0$, i.e., $J = J_0$;
ii) All components of ω are positive, namely all $i \in J_+$, i.e., $J = J_+$;
iii) Some components of ω are zero and the others are positive, namely $J = J_0 \cup J_+$.

We will prove the relationship between κ and κ' by considering the above three cases. To simplify calculations, we let $x^0 s^0 > \omega$. i) and ii) It follows from (2) and Kojima et al. [7, Theorem 3.5] that $\overline{M}$ is a $P_*(\kappa')$-matrix, where κ' satisfies

$$\begin{aligned} \frac{1+4\kappa'}{1+4\kappa} &= \max_{t\in(0,1]} \frac{\max \omega(t)}{\min \omega(t)} = \max_{t\in(0,1]} \frac{\max\{t x^0 s^0 + (1-t)\omega\}}{\min\{t x^0 s^0 + (1-t)\omega\}} \\ &= \begin{cases} \dfrac{\max(x^0 s^0)}{\min(x^0 s^0)}, & \text{if } J = J_0; \\ \dfrac{\max(x^0 s^0)}{\min \omega}, & \text{if } J = J_+. \end{cases} \end{aligned} \tag{10}$$

iii) The case when ω has both zero and positive coordinates at the same time is discussed below. Let $\overline{\varsigma} \in \mathbb{R}^n$, $\varsigma(t) = D\sqrt{W(t)}\overline{\varsigma}$ and

$$\zeta_i = \chi_i \varsigma_i(t) + (1-\chi_i)\frac{\varsigma_i(t)}{\sqrt{t}} = \begin{cases} \dfrac{\varsigma_i(t)}{\sqrt{t}}, & \text{if } i \in J_0; \\ \varsigma_i(t), & \text{if } i \in J_+, \end{cases}$$

where χ is given by (9) and $t \in (0, 1]$. Then for each $i \in J$,

$$\begin{aligned} \overline{\varsigma}_i(\overline{M}\overline{\varsigma})_i &= \overline{\varsigma}_i\Big(\sqrt{W^{-1}(t)}DMD\sqrt{W(t)}\overline{\varsigma}\Big)_i \\ &= \omega_i^{-1}(t)\Big(D\sqrt{W(t)}\overline{\varsigma}\Big)_i\Big(MD\sqrt{W(t)}\overline{\varsigma}\Big)_i \\ &= \frac{\varsigma_i(t)(M\varsigma(t))_i}{\omega_i(t)} = \begin{cases} \dfrac{\varsigma_i(t)(M\varsigma(t))_i}{t(x^0 s^0)_i}, & \text{if } i \in J_0; \\ \dfrac{\varsigma_i(t)(M\varsigma(t))_i}{\omega_i(t)}, & \text{if } i \in J_+, \end{cases} \\ &= \begin{cases} \dfrac{\zeta_i(M\zeta)_i}{(x^0 s^0)_i}, & \text{if } i \in J_0; \\ \dfrac{\zeta_i(M\zeta)_i}{\omega_i(t)}, & \text{if } i \in J_+. \end{cases} \end{aligned} \tag{11}$$

By (11), we define

$$\begin{aligned} I_+ &:= \{i \in J : \overline{\varsigma}_i(\overline{M}\overline{\varsigma})_i \ge 0\} = \{i \in J : \varsigma_i(t)(M\varsigma(t))_i \ge 0\} = \{i \in J : \zeta_i(M\zeta)_i \ge 0\}, \\ I_- &:= \{i \in J : \overline{\varsigma}_i(\overline{M}\overline{\varsigma})_i < 0\} = \{i \in J : \varsigma_i(t)(M\varsigma(t))_i < 0\} = \{i \in J : \zeta_i(M\zeta)_i < 0\}. \end{aligned}$$

Since $M \in P_*(\kappa)$, then for $\zeta \in \mathbb{R}^n$,

$$(1+4\kappa)\sum_{i\in I_+} \zeta_i(M\zeta)_i + \sum_{i\in I_-} \zeta_i(M\zeta)_i \ge 0. \tag{12}$$

Taking into account the fact that $x^0s^0 > \omega$ and using (2) and (11), we get for every $\overline{\varsigma} \in \mathbb{R}^n$

$$
\begin{aligned}
&(1+4\kappa')\sum_{i\in I_+}\overline{\varsigma}_i(\overline{M}\overline{\varsigma})_i + \sum_{i\in I_-}\overline{\varsigma}_i(\overline{M}\overline{\varsigma})_i\\
&= (1+4\kappa')\sum_{i\in I_+}\frac{\varsigma_i(t)(M\varsigma(t))_i}{\omega_i(t)} + \sum_{i\in I_-}\frac{\varsigma_i(t)(M\varsigma(t))_i}{\omega_i(t)}\\
&= (1+4\kappa')\sum_{i\in I_+\cap J_+}\frac{\varsigma_i(t)(M\varsigma(t))_i}{\omega_i(t)} + (1+4\kappa')\sum_{i\in I_+\cap J_0}\frac{\varsigma_i(t)(M\varsigma(t))_i}{t(x^0s^0)_i} + \sum_{i\in I_-\cap J_+}\frac{\varsigma_i(t)(M\varsigma(t))_i}{\omega_i(t)} + \sum_{i\in I_-\cap J_0}\frac{\varsigma_i(t)(M\varsigma(t))_i}{t(x^0s^0)_i}\\
&= (1+4\kappa')\sum_{i\in I_+\cap J_+}\frac{\zeta_i(M\zeta)_i}{\omega_i(t)} + (1+4\kappa')\sum_{i\in I_+\cap J_0}\frac{\zeta_i(M\zeta)_i}{(x^0s^0)_i} + \sum_{i\in I_-\cap J_+}\frac{\zeta_i(M\zeta)_i}{\omega_i(t)} + \sum_{i\in I_-\cap J_0}\frac{\zeta_i(M\zeta)_i}{(x^0s^0)_i}\\
&\geq (1+4\kappa')\frac{\sum_{i\in I_+\cap J_+}\zeta_i(M\zeta)_i}{\max_{i\in J_+}(x^0s^0)} + (1+4\kappa')\frac{\sum_{i\in I_+\cap J_0}\zeta_i(M\zeta)_i}{\max_{i\in J_0}(x^0s^0)} + \frac{\sum_{i\in I_-\cap J_+}\zeta_i(M\zeta)_i}{\min_{i\in J_+}\omega} + \frac{\sum_{i\in I_-\cap J_0}\zeta_i(M\zeta)_i}{\min_{i\in J_0}(x^0s^0)}\\
&\geq \frac{(1+4\kappa')}{\max(x^0s^0)}\sum_{i\in I_+}\zeta_i(M\zeta)_i + \max\left\{\frac{1}{\min_{i\in J_+}\omega}, \frac{1}{\min_{i\in J_0}(x^0s^0)}\right\}\sum_{i\in I_-}\zeta_i(M\zeta)_i\\
&= (1+4\kappa)\max\left\{\frac{1}{\min_{i\in J_+}\omega}, \frac{1}{\min_{i\in J_0}(x^0s^0)}\right\}\sum_{i\in I_+}\zeta_i(M\zeta)_i + \max\left\{\frac{1}{\min_{i\in J_+}\omega}, \frac{1}{\min_{i\in J_0}(x^0s^0)}\right\}\sum_{i\in I_-}\zeta_i(M\zeta)_i\\
&= \max\left\{\frac{1}{\min_{i\in J_+}\omega}, \frac{1}{\min_{i\in J_0}(x^0s^0)}\right\}\left[(1+4\kappa)\sum_{i\in I_+}\zeta_i(M\zeta)_i + \sum_{i\in I_-}\zeta_i(M\zeta)_i\right]\\
&\geq 0,
\end{aligned}
\tag{13}
$$

where the last inequality is due to (12). Moreover, the fourth equality in (13) holds by following the idea of Kojima et al. [7, Theorem 3.5] in the context of LCP, where $(1+4\kappa)\max\left\{\frac{1}{\min_{i\in J_+}\omega}, \frac{1}{\min_{i\in J_0}(x^0s^0)}\right\}$ is the coefficient of $\sum_{i\in I_+}\zeta_i(M\zeta)_i$ chosen such that there exists the nonnegative term $(1+4\kappa)\sum_{i\in I_+}\zeta_i(M\zeta)_i + \sum_{i\in I_-}\zeta_i(M\zeta)_i$ in the fifth equality. By (13), we have shown that $\overline{M} \in P_*(\kappa')$ in the case of ω with $J = J_0 \cup J_+$, where κ' satisfies

$$
\frac{(1+4\kappa')}{\max(x^0s^0)} = (1+4\kappa)\max\left\{\frac{1}{\min_{i\in J_+}\omega}, \frac{1}{\min_{i\in J_0}(x^0s^0)}\right\},
$$

i.e.,

$$
\frac{1+4\kappa'}{1+4\kappa} = \max\left\{\frac{\max(x^0s^0)}{\min_{i\in J_+}\omega}, \frac{\max(x^0s^0)}{\min_{i\in J_0}(x^0s^0)}\right\} = \frac{\max(x^0s^0)}{\min\left\{\min_{i\in J_+}\omega,\ \min_{i\in J_0}(x^0s^0)\right\}}. \tag{14}
$$

Combining (9), (10) with (14) yields the following relationship between κ' and κ in all three cases

$$
\frac{1+4\kappa'}{1+4\kappa} = \frac{\max(x^0s^0)}{\min\{\chi\omega + (e-\chi)x^0s^0\}} = \begin{cases} \dfrac{\max(x^0s^0)}{\min(x^0s^0)}, & \text{if } J = J_0;\\[2ex] \dfrac{\max(x^0s^0)}{\min\omega}, & \text{if } J = J_+;\\[2ex] \dfrac{\max(x^0\ s^0)}{\min\left\{\min_{i\in J_+}\omega,\ \min_{i\in J_0}(x^0s^0)\right\}}, & \text{if } J = J_0\cup J_+. \end{cases}
$$

Table 1
Analysis of IPMs for $P_*(\kappa)$-LCP or $P_*(\kappa)$-WLCP.

Proximity measure	Algorithm	Complexity
$\frac{\|v^{-1}-v\|}{2}$	CP IPM for WLCP [40]	$O\left((1+4\hat{\kappa})\sqrt{n}\log\frac{\frac{5}{4(1+4\kappa)}\max(x^0s^0)+\|x^0s^0-\omega\|}{\varepsilon}\right)$
$\frac{\|\tilde{v}^{-1}-\tilde{v}\|}{2}$	IPM for LCP [39]	$O\left((1+2\kappa)^2 n\log\frac{\max\{(x^0)^Ts^0,\|s^0-q-Mx^0\|\}}{\varepsilon}\right)$
$\left\|\frac{\tilde{v}-\tilde{v}^2}{2\tilde{v}-e}\right\|$	CP IPM for LCP [29]	$O\left((1+2\kappa)\sqrt{n}\log\frac{9(x^0)^Ts^0}{8\varepsilon}\right)$
$\left\|\frac{\tilde{v}-\tilde{v}^2}{2\tilde{v}-e}\right\|$	IPM for LCP [30]	$O\left((1+\kappa)\sqrt{n}\log\frac{(x^0)^Ts^0}{\varepsilon}\right)$
$\left\|\frac{v-v^2}{2v-e}\right\|$	IPM for WLCP	$O\left((8+13\kappa')\left\|\frac{x^0s^0-\omega}{\chi\omega+(e-\chi)\min(x^0s^0)}\right\|\log\left(\frac{\frac{\max(x^0s^0)}{2(1+4\kappa')}+\|x^0s^0-\omega\|}{\varepsilon}\right)\right)$

Therefore, the definition of κ' is given by

$$
\begin{aligned}
\kappa' :=& \frac{(1+4\kappa)\max(x^0s^0)-\min\{\chi\omega+(e-\chi)x^0s^0\}}{4\min\{\chi\omega+(e-\chi)x^0s^0\}}\\
=& \begin{cases} \dfrac{(1+4\kappa)\max(x^0s^0)-\min(x^0s^0)}{4\min(x^0s^0)}, & \text{if } J=J_0;\\[2mm] \dfrac{(1+4\kappa)\max(x^0s^0)-\min\omega}{4\min\omega}, & \text{if } J=J_+;\\[2mm] \dfrac{(1+4\kappa)\max(x^0s^0)-\min\left\{\min\limits_{i\in J_+}\omega,\ \min\limits_{i\in J_0}(x^0s^0)\right\}}{4\min\left\{\min\limits_{i\in J_+}\omega,\ \min\limits_{i\in J_0}(x^0s^0)\right\}}, & \text{if } J=J_0\cup J_+. \end{cases}
\end{aligned}
\tag{15}
$$

In Table 1, the polynomial complexity of some existing IPMs for $P_*(\kappa)$-LCP in Darvay et al. [29,30], Lee et al. [39] or $P_*(\kappa)$-WLCP in Chi et al. [40] and our algorithm for $P_*(\kappa)$-WLCP are listed, where $\tilde{v}=\sqrt{\frac{xs}{\mu}}$ with $\mu>0$ and $\hat{\kappa}=\frac{1}{4}\left(\frac{(1+4\kappa)\max(x^0s^0)}{\min\omega}-1\right)$, κ' is given in (15) and χ is defined by (9). In the case of using AET function $\varphi(t)=t$, the norm-based proximity measure obtained is $\frac{1}{2}\left\|v^{-1}-v\right\|$. Whereas, when using the AET function $\varphi(t)=t-\sqrt{t}$, the proximity measure is $\left\|\frac{v-v^2}{2v-e}\right\|$. Lee et al. [39] gave a full-Newton step infeasible IPM for $P_*(\kappa)$-LCP using a class of $(1/t)$-bound kernel functions. Using AET with $\varphi(t)=t-\sqrt{t}$, Darvay et al. [29] proposed a corrector-predictor IPM for $P_*(\kappa)$-LCP, and presented a short-step IPM for $P_*(\kappa)$-LCP [30]. The polynomial complexity of the above-mentioned IPMs can be analyzed through three aspects. Firstly, the polynomial complexity of IPMs is related to $\sqrt{n}$, while the polynomial complexity of infeasible IPMs is related to n. Secondly, the polynomial complexity of IPMs for $P_*(\kappa)$-LCP is influenced by several factors, including the accuracy parameter ε, the size of the problem, the handicap of M, the starting points x^0 and s^0. Furthermore, for $P_*(\kappa)$-WLCP, an additional influencing factor is the weight vector ω. Thirdly, our algorithm has a different polynomial iteration complexity than CP IPM for $P_*(\kappa)$-WLCP [40]. The analysis of full-Newton step IPM in this article differs from that studied in Chi et al. [40], which focuses on CP IPM for $P_*(\kappa)$-WLCP and is based on different AET function $\varphi(t)=t$. The algorithm in Chi et al. [40] does not use any transformation of the central path. In Section 3, we will compute the polynomial complexity of the IPM for $P_*(\kappa)$-WLCP with the weight vector $\omega\geq 0$.

2.2. An IPM with full-Newton step for $P_*(\kappa)$-WLCP

This paper considers the function $\varphi(t)=t-\sqrt{t}$ used by Darvay for linear optimization [28]. Indeed, by direct calculations, we obtain

$$p_v=\frac{2(v-v^2)}{2v-e}, \tag{16}$$

where $v_i\in\left(\frac{1}{2},+\infty\right)$. Notice that p_v is invertible in the case of $v>\frac{e}{2}$.

We further define a norm-based proximity measure $\delta(x,s;t)$ as

$$\delta(x,s;t):=\frac{\|p_v\|}{2}=\left\|\frac{v-v^2}{2v-e}\right\|. \tag{17}$$

Taking into account the fact that $v > 0$, one can easily verify that

$$\delta(x, s; t) = 0 \iff v = v^2 \iff v = e \iff xs = \omega(t).$$

In other words, the proximity function $\delta(x, s; t)$ measures the distance between the current iterate (x, s) and the corresponding center path $(x(t), s(t))$. Define the τt-neighborhood of the central path as

$$\mathcal{N}(\tau, t) := \left\{(x, s) \in \mathcal{F}^0 : \delta(x, s; t) \le \tau t\right\},$$

where $\tau > 0$ is a threshold and $t \in (0, 1]$ is a parameter.

A feasible IPM with full-Newton step is proposed for solving $P_*(\kappa)$-WLCP (1) and shown as in Algorithm 1, which employs the function $\varphi(t) = t - \sqrt{t}$. More specifically, the algorithm traces the central path approximately and guarantees that the iterates (x, s) stay in the τt-neighborhood of the central path for a threshold $\tau > 0$ and $t \in (0, t_0]$ with $t_0 = 1$. Notice that Algorithm 1 starts with a strictly feasible initial point (x^0, s^0), which satisfies $\delta(x^0, s^0; t_0) \le \tau t_0$. Since the weight vector $\omega \ge 0$, the condition $x^0 s^0 > \omega$ ensures that the initial point is also in the strictly feasible region $\mathcal{F}^0$. Moreover, we assume $x^0 s^0 > \omega$ in order to simplify the expression for κ', which is the handicap of the matrix $\overline{M}$ (see pages 5–7). In fact, $x^0 s^0 > \omega$ is a sufficient and not necessary condition for the strict feasibility of the initial point. If we do not use it, then it would be enough to assume that $x^0 s^0 > 0$. Besides, t is reduced to t_+ by multiplying by $(1 - \theta)$, that is, $t_+ = (1 - \theta)t$, where the update parameter $\theta \in (0, 1)$. From solving system (8), we obtain the scaled search direction d_x and d_s, and derive the original search direction $(\Delta x, \Delta s)$ by (6). Then, taking a full-Newton step, we achieve the new iterate (x^+, s^+). With suitable choices of θ and τ, the new iterate also lies in the τt_+-neighborhood of the central path. We repeat the above procedure until we find an optimal solution $(x, s) \in \mathcal{F}^0$ satisfying the stopping criterion $\|xs - \omega\| \le \varepsilon$.

Algorithm 1 The feasible IPM with full-Newton step for $P_*(\kappa)$-WLCP.

Input

An accuracy parameter $\varepsilon > 0$;
A barrier update parameter $\theta \in (0, 1)$;
A weight vector $\omega \ge 0$, a vector $q \in \mathbb{R}^n$ and a matrix $M \in P_*(\kappa)$.
Let $(x^0, s^0) \in \mathcal{F}^0$ such that $x^0 s^0 > \omega$ and $\delta(x^0, s^0; t_0) \le \tau t_0$, where $t_0 = 1$
and $\tau = \dfrac{1}{2\sqrt{1 + (1 + 4\kappa')^2}}$;

begin
  $x := x^0, \quad s := s^0, \quad t := t_0;$
  while $\|xs - \omega\| > \varepsilon$ **do**
  begin
    Based on (16), calculate (d_x, d_s) from (8) and obtain $(\Delta x, \Delta s)$ via (6);
    Update the iteration point:
    $x := x + \Delta x, \quad s := s + \Delta s;$
    Update:
    $t := (1 - \theta)t.$
  end
end

3. Analysis of the algorithm

As mentioned, after a full-Newton step, we obtain the new iteration point $x^+ = x + \Delta x$ and $s^+ = s + \Delta s$. In this section, we first prove the strict feasibility of the new iteration point, i.e., $x^+ > 0$, $s^+ > 0$. Then, we discuss the effect of the full-Newton step on the proximity function but keeping t fixed. Finally, we also study the effect of t-update on the proximity function.

To proceed, as a consequence of Wang et al. [24, Lemma 3.2 and Lemma 3.3] and (15), we present two lemmas.

Lemma 1. *Let $\delta := \delta(x, s; t)$ and $x^0 s^0 > \omega$. One has*

$$\|d_x d_s\|_\infty \le (1 + 4\kappa')\delta^2,$$

where κ' is given in (15).

Proof. From the second equality of system (8) and (17), we have

$$\delta = \frac{1}{2}\|d_x + d_s\| = \frac{1}{2}\|p_v\|.$$

In the proof of Wang et al. [24, Lemma 3.2], $\delta = \frac{1}{2}\|d_x + d_s\| = \frac{1}{2}\|p_v\|$ which is the same as in our case for $P_*(\kappa)$-WLCP. Lemma 3.2 in Wang et al. [24] gives an upper bound on the norm of the sum of the solutions of a Newton system for $P_*(\kappa)$-LCP. Here the difference is that in the coefficient matrix of the Newton system $\overline{M}$ is a $P_*(\kappa')$-matrix. Thus, by the result of Wang et al. [24, Lemma 3.2], we have

$$\|d_x d_s\|_\infty \le (1+4\kappa')\delta^2,\ \|d_x d_s\| \le \sqrt{1+(1+4\kappa')^2}\delta^2.$$

□

Lemma 2. *Let $\delta := \delta(x,s;t)$ and $x^0 s^0 > \omega$. Then, we have*

$$\|d_x d_s\| \le \sqrt{1+(1+4\kappa')^2}\delta^2.$$

Proof. Similar to Lemma 1, based on the result of Wang et al. [24, Lemma 3.3], one has

$$\|d_x d_s\| \le \sqrt{1+(1+4\kappa')^2}\delta^2.$$

□

Lemma 3. *Let $(x^0, s^0) \in \mathcal{F}^0$. Suppose that $\delta < \frac{1}{\sqrt{1+4\kappa'}}$ and $v > \frac{e}{2}$. Then, we have the strict feasibility of full-Newton step, that is,*

$$x^+ > 0 \quad \text{and} \quad s^+ > 0.$$

Proof. First, for any $\alpha \in [0,1]$, we denote

$$x(\alpha) := x + \alpha\Delta x, \quad s(\alpha) := s + \alpha\Delta s.$$

Since $\delta < \frac{1}{\sqrt{1+4\kappa'}}$ and $v > \frac{e}{2}$, it follows from (6)–(8) and (16) that

$$\begin{aligned}\frac{x(\alpha)s(\alpha)}{\omega(t)} &= \frac{xs+\alpha(s\Delta x + x\Delta s)+\alpha^2\Delta x\Delta s}{\omega(t)}\\ &= v^2 + \alpha v p_v + \alpha^2 d_x d_s\\ &= (1-\alpha)v^2 + \alpha\left(v^2 + v\frac{2(v-v^2)}{2v-e}\right) + \alpha^2 d_x d_s\\ &= (1-\alpha)v^2 + \alpha\left(\frac{v^2}{2v-e} + \alpha d_x d_s\right).\end{aligned} \tag{18}$$

Note that $x(\alpha)s(\alpha) > 0$ holds if $\frac{v^2}{2v-e} + \alpha d_x d_s > 0$. Using Lemma 1 and the fact that $\alpha \in [0,1]$, we have

$$\begin{aligned}\frac{v^2}{2v-e} + \alpha d_x d_s &\ge \frac{v^2}{2v-e} - \alpha\|d_x d_s\|_\infty e \ge \frac{v^2}{2v-e} - (1+4\kappa')\delta^2 e\\ &> \frac{v^2}{2v-e} - e = \frac{(v-e)^2}{2v-e} > 0.\end{aligned}$$

This means that $x(\alpha)s(\alpha) > 0$ for each $\alpha \in [0,1]$. Because $x(0)$ and $s(0)$ are positive, and the functions $x(\alpha)$ and $s(\alpha)$ linearly depend on α, it further implies that $x(\alpha) > 0$ and $s(\alpha) > 0$ for all $\alpha \in [0,1]$. Hence, $x(1) = x^+ > 0$ and $s(1) = s^+ > 0$ hold. □

Next, we define $v_+ := \sqrt{\frac{x^+s^+}{\omega(t)}}$ for notational simplicity. The following lemma provides an upper bound on $\left\|e - v_+^2\right\|$, which is a key to analyzing the convergence of Algorithm 1.

Lemma 4. *Suppose that $\delta < \frac{1}{\sqrt{1+4\kappa}}$ and $v > \frac{e}{2}$. Then, we have*

$$\left\|e - v_+^2\right\| \le \left(1 + \sqrt{1+(1+4\kappa')^2}\right)\delta^2.$$

Proof. Considering (18) with $\alpha = 1$, we have

$$v_+^2 = \frac{x^+s^+}{\omega(t)} = \frac{v^2}{2v-e} + d_x d_s. \tag{19}$$

Thus, applying (17), (19) and Lemma 2, we verify

$$\begin{aligned}\left\|e - v_+^2\right\| &= \left\|e - \frac{v^2}{2v-e} - d_x d_s\right\|\\ &= \left\|-\frac{(v-e)^2}{2v-e} - d_x d_s\right\|\\ &\le \left\|\frac{(v-e)^2(2v-e)}{(2v-e)^2}\right\| + \|d_x d_s\|\\ &\le \left\|\frac{v(v-e)}{2v-e}\right\|^2 + \|d_x d_s\|\\ &\le \left(1 + \sqrt{1+(1+4\kappa')^2}\right)\delta^2,\end{aligned}$$

where the second inequality is due to the fact that

$$0 < 2v - e \le v^2 \tag{20}$$

for all $v > \frac{e}{2}$. Then, the proof is complete. □

Lemma 5 establishes an upper bound for the proximity function $\delta(x^+, s^+; t)$ after a full-Newton step with t fixed.

Lemma 5. *Suppose that $\delta < \frac{\sqrt{3}}{2\sqrt{1+4\kappa'}}$ and $v > \frac{e}{2}$. Then, we have $v_+ > \frac{e}{2}$ and*

$$\delta(x^+, s^+; t) \le \frac{\sqrt{1-(1+4\kappa')\delta^2}\left(1+\sqrt{1+(1+4\kappa')^2}\right)\delta^2}{1-2(1+4\kappa')\delta^2+\sqrt{1-(1+4\kappa')\delta^2}}.$$

Proof. Since $\delta < \frac{\sqrt{3}}{2\sqrt{1+4\kappa'}}$ and $v > \frac{e}{2}$, it follows from (19), (20) and Lemma 1 that

$$v_+^2 = \frac{v^2}{2v-e} + d_x d_s \ge e + d_x d_s \ge \left(1 - \|d_x d_s\|_\infty\right)e \ge \left[1-(1+4\kappa')\delta^2\right]e,$$

which implies

$$\min v_+ \ge \sqrt{1-(1+4\kappa')\delta^2} > \frac{1}{2}. \tag{21}$$

Thus, the inequality $v_+ > \frac{e}{2}$ holds. □

Next, we estimate an upper bound on the proximity function $\delta(x^+, s^+; t)$. Applying (17) yields

$$\begin{aligned}\delta(x^+, s^+; t) &= \left\|\frac{v_+ - v_+^2}{2v_+ - e}\right\| = \left\|\frac{v_+(e-v_+)}{2v_+ - e}\right\| = \left\|\frac{v_+(e-v_+)(e+v_+)}{(2v_+ - e)(e+v_+)}\right\| \\ &= \left\|\frac{v_+}{2v_+^2 + v_+ - e}(e - v_+^2)\right\|.\end{aligned} \tag{22}$$

For any $z > \frac{1}{2}$, we consider the function $f(z) = \frac{z}{2z^2+z-1}$. From $f'(z) < 0$, it follows that f is a monotonically decreasing function on the interval $\left(\frac{1}{2}, +\infty\right)$. Therefore, by (21), (22) and Lemma 4, we obtain

$$\begin{aligned}\delta(x^+, s^+; t) &\le f(\min v_+)\|e - v_+^2\| \\ &\le \frac{\sqrt{1-(1+4\kappa')\delta^2}}{1-2(1+4\kappa')\delta^2+\sqrt{1-(1+4\kappa')\delta^2}}\|e - v_+^2\| \\ &\le \frac{\sqrt{1-(1+4\kappa')\delta^2}\left(1+\sqrt{1+(1+4\kappa')^2}\right)\delta^2}{1-2(1+4\kappa')\delta^2+\sqrt{1-(1+4\kappa')\delta^2}}\end{aligned}$$

which is the desired result. □

Now, we recall the lower and upper bounds on the components of vector v, which was shown in Kheirfam [15, Lemma 2] for the semidefinite linear complementarity problem (SDLCP) and will be applied to the subsequent analysis of Algorithm 1 for WLCP.

Lemma 6. *Suppose that $v > \frac{e}{2}$. For $i = 1, 2, \cdots, n$, we have*

$$\frac{1}{2} + \frac{1}{4\rho(\delta)} \le v_i \le \frac{1}{2} + \rho(\delta),$$

where $\rho(\delta) = \delta + \sqrt{\frac{1}{4} + \delta^2}$.

Proof. By following the idea of Kheirfam [15, Lemma 2] for SDLCP, we prove the corresponding result for WLCP, which is a generalization of LCP. From the definition of $\delta(x, s; t)$, we get

$$\delta^2 = \left\|\frac{v - v^2}{2v - e}\right\|^2 = \sum_{i=1}^{n}\left|\frac{v_i - v_i^2}{2v_i - 1}\right|^2,$$

which implies that $\left|\frac{v_i - v_i^2}{2v_i - 1}\right| \le \delta$, for $i = 1, 2, \cdots, n$.

Then we have

$$-v_i^2 + (1+2\delta)v_i - \delta \ge 0$$

holds if

$$\frac{1+2\delta-\sqrt{1+4\delta^2}}{2} \le v_i \le \frac{1+2\delta+\sqrt{1+4\delta^2}}{2};$$

and

$$-v_i^2+(1-2\delta)v_i+\delta \le 0$$

holds if

$$v_i \le \frac{1-2\delta-\sqrt{1+4\delta^2}}{2} \quad \text{or} \quad v_i \ge \frac{1-2\delta+\sqrt{1+4\delta^2}}{2}.$$

Since $v_i > \frac{1}{2}$, the inequality $\left|\frac{v_i-v_i^2}{2v_i-1}\right| \le \delta$ holds if

$$\frac{1-2\delta+\sqrt{1+4\delta^2}}{2} \le v_i \le \frac{1+2\delta+\sqrt{1+4\delta^2}}{2}.$$

By letting $\rho(\delta)=\delta+\sqrt{\frac{1}{4}+\delta^2}$, we have

$$\frac{1}{2}+\frac{1}{4\rho(\delta)} \le v_i \le \frac{1}{2}+\rho(\delta).$$

□

To proceed, we deduce that

$$\rho(\delta)=\delta+\sqrt{\frac{1}{4}+\delta^2} \le \delta+\sqrt{\left(\frac{1}{2}+\delta\right)^2}=\frac{1}{2}+2\delta. \tag{23}$$

Suppose that t is reduced by the factor $(1-\theta)$, i.e., $t_+=(1-\theta)t$. Using (2) gives

$$\omega(t_+)=[1-(1-\theta)t]\omega+(1-\theta)tx^0s^0=\omega(t)+(\omega-x^0s^0)\theta t. \tag{24}$$

For convenience, we define

$$v^+ := \sqrt{\frac{x^+s^+}{\omega(t_+)}}.$$

Then, we build up the upper bound on the proximity function $\delta(x^+,s^+;t_+)$ for the new iterate (x^+,s^+).

Lemma 7. *Suppose that the starting point $(x^0,s^0)\in\mathcal{F}^0$ satisfies $x^0s^0>\omega$. If $\delta \le \frac{1}{2\sqrt{1+4\kappa'}}$ and $v>\frac{e}{2}$, then we have $v^+>\frac{e}{2}$ and*

$$\delta(x^+,s^+;t_+) \le \frac{\sqrt{3}\left\{\left(1+\sqrt{1+(1+4\kappa')^2}\right)\delta^2+\left[1+\left(5+4\kappa'\right)\delta^2\right]\beta\theta t\right\}}{1+\sqrt{3}},$$

where

$$\beta := \left\|\frac{x^0s^0-\omega}{\chi\omega+(e-\chi)\min(x^0s^0)}\right\|. \tag{25}$$

Proof. We first show that $v^+>\frac{e}{2}$. Since $v_+=\sqrt{\frac{x^+s^+}{\omega(t)}}$ and $v^+=\sqrt{\frac{x^+s^+}{\omega(t_+)}}$, it is easy to verify that

$$v^+=v_+\sqrt{\frac{\omega(t)}{\omega(t_+)}}=v_+\xi, \tag{26}$$

where we let $\xi := \sqrt{\frac{\omega(t)}{\omega(t_+)}}$. Since $x^0s^0>\omega$, it follows from (24) that

$$\xi=\sqrt{\frac{\omega(t)}{\omega(t_+)}}=\sqrt{\frac{\omega(t)}{\omega(t)+(\omega-x^0s^0)\theta t}} \ge e. \tag{27}$$

Thus, applying (26), (27) and Lemma 5 yields

$$v^+=v_+\xi \ge v_+ > \frac{e}{2}. \tag{28}$$

Now, we discuss the upper bound on $\delta(x^+, s^+; t_+)$. Using (28), we obtain

$$\begin{aligned}\delta(x^+, s^+; t_+) &= \left\| \frac{v^+ - (v^+)^2}{2v^+ - e} \right\| = \left\| \frac{v_+\xi - v_+^2\xi^2}{2v_+\xi - e} \right\| \\ &= \left\| \frac{v_+\xi(e - v_+\xi)}{2v_+\xi - e} \right\| = \left\| \frac{v_+\xi(e - v_+^2\xi^2)}{(2v_+\xi - e)(v_+\xi + e)} \right\|.\end{aligned} \tag{29}$$

Taking into account of the fact that $\delta \le \dfrac{1}{2\sqrt{1+4\kappa'}}$, it follows from (28) and (21) that

$$\min(v_+\xi) \ge \min v_+ \ge \sqrt{1 - (1+4\kappa')\delta^2} \ge \frac{\sqrt{3}}{2}. \tag{30}$$

Consider a monotonically decreasing function $f(z) = \dfrac{z}{(2z-1)(z+1)}$ for all $z > \dfrac{1}{2}$. Therefore, by (29) and (30), we have

$$\begin{aligned}\delta(x^+, s^+; t_+) &\le f\left(\min(v_+\xi)\right) \|e - v_+^2\xi^2\| \\ &\le f\left(\frac{\sqrt{3}}{2}\right) \|e - v_+^2\xi^2\| \\ &= \frac{\sqrt{3}\left\|e - v_+^2\xi^2\right\|}{1+\sqrt{3}}.\end{aligned} \tag{31}$$

Using the fact that $x^0 s^0 > \omega$, (2) and (9), we have

$$\beta = \left\| \frac{x^0 s^0 - \omega}{\chi\omega + (e-\chi)\min(x^0 s^0)} \right\| \ge \left\| \frac{x^0 s^0 - \omega}{\omega(t_+)} \right\|.$$

Then we obtain from (2), (24), (27) (25) and Lemma 4 that

$$\begin{aligned}\|e - v_+^2\xi^2\| &\le \|e - v_+^2\| + \|v_+^2 - v_+^2\xi^2\| \\ &\le \|e - v_+^2\| + \left\|v_+^2\right\|_\infty \left\| e - \frac{\omega(t)}{\omega(t_+)} \right\| \\ &= \|e - v_+^2\| + \left\|v_+^2\right\|_\infty \left\| \frac{\theta t(\omega - x^0 s^0)}{\omega(t_+)} \right\| \\ &\le \|e - v_+^2\| + \left\|v_+^2\right\|_\infty \left\| \frac{x^0 s^0 - \omega}{\omega(t_+)} \right\| \theta t \\ &\le \left(1 + \sqrt{1 + (1+4\kappa')^2}\right)\delta^2 + \left\|v_+^2\right\|_\infty \beta\theta t.\end{aligned} \tag{32}$$

Moreover, it follows from (19), Lemmas 6 and 1 that

$$\begin{aligned}\|v_+^2\|_\infty &= \left\| \frac{v^2}{2v - e} + d_x d_s \right\|_\infty = \left\| e + \frac{(v-e)^2}{2v-e} + d_x d_s \right\|_\infty \\ &\le 1 + \frac{\left(\frac{1}{2} + \rho(\delta) - 1\right)^2}{2\left(\frac{1}{2} + \rho(\delta)\right) - 1} + \|d_x d_s\|_\infty \le 1 + \frac{(2\rho(\delta) - 1)^2}{8\rho(\delta)} + (1+4\kappa')\delta^2 \\ &\le 1 + 4\delta^2 + (1+4\kappa')\delta^2 = 1 + \left(5 + 4\kappa'\right)\delta^2.\end{aligned} \tag{33}$$

The last inequality in (33) is due to the fact that by Lemma 6 and (23),

$$\frac{(2\rho(\delta) - 1)^2}{8\rho(\delta)} \le \frac{\left[2\left(\frac{1}{2} + 2\delta\right) - 1\right]^2}{8\left(\frac{1}{2} + 2\delta\right)} = \frac{16\delta^2}{4 + 16\delta} \le 4\delta^2.$$

Combining (31)–(33) gives

$$\begin{aligned}\delta(x^+, s^+; t_+) &\le \frac{\sqrt{3}\left[(1 + \sqrt{1 + (1+4\kappa')^2})\delta^2 + \left\|v_+^2\right\|_\infty \beta\theta t\right]}{1+\sqrt{3}} \\ &\le \frac{\sqrt{3}\left\{\left(1 + \sqrt{1 + (1+4\kappa')^2}\right)\delta^2 + \left[1 + \left(5 + 4\kappa'\right)\delta^2\right]\beta\theta t\right\}}{1+\sqrt{3}}.\end{aligned}$$

Then, the proof is complete. □

As denoted in the algorithm, $t_+ := (1-\theta)t$. In the following theorem, we see that by choosing the suitable value of the update parameter θ, the new iterate (x^+, s^+) always stays in the τt_+-neighborhood of the central path with $\tau = \frac{1}{2\sqrt{1+(1+4\kappa')^2}}$.

Theorem 1. *Suppose that*

$$\theta \le \frac{3\sqrt{1+(1+4\kappa')^2} - \left[1+\sqrt{1+(1+4\kappa')^2}\right]t}{\left[4+4(1+4\kappa')^2+(5+4\kappa')t^2\right]\beta + 3\sqrt{1+(1+4\kappa')^2}},$$

where β is given by (25). If $\delta \le \frac{t}{2\sqrt{1+(1+4\kappa')^2}}$, then we have

$$\delta(x^+, s^+; t_+) \le \frac{t_+}{2\sqrt{1+(1+4\kappa')^2}}.$$

Proof. Since $\delta \le \frac{t}{2\sqrt{1+(1+4\kappa')^2}}$, we obtain from Lemma 7 that

$$\begin{aligned}
&\delta(x^+, s^+; t_+)\\
&\le \frac{\sqrt{3}\left\{\left(1+\sqrt{1+(1+4\kappa')^2}\right)\delta^2 + \left[1+(5+4\kappa')\delta^2\right]\beta\theta t\right\}}{1+\sqrt{3}}\\
&\le \frac{2}{3}\left\{\frac{t^2}{4(1+(1+4\kappa')^2)} + \frac{t^2}{4\sqrt{1+(1+4\kappa')^2}} + \left[1+\frac{(5+4\kappa')t^2}{4(1+(1+4\kappa')^2)}\right]\beta\theta t\right\}
\end{aligned}$$

Thus, the inequality $\delta(x^+, s^+; t_+) \le \frac{t_+}{2\sqrt{1+(1+4\kappa')^2}}$ holds, if

$$\begin{aligned}
&\frac{2}{3}\left\{\frac{t^2}{4(1+(1+4\kappa')^2)} + \frac{t^2}{4\sqrt{1+(1+4\kappa')^2}} + \left[1+\frac{(5+4\kappa')t^2}{4(1+(1+4\kappa')^2)}\right]\beta\theta t\right\}\\
&\le \frac{t_+}{2\sqrt{1+(1+4\kappa')^2}} = \frac{(1-\theta)t}{2\sqrt{1+(1+4\kappa')^2}},
\end{aligned}$$

i.e.,

$$\begin{aligned}
&\frac{2}{3}\left\{\frac{t}{2\sqrt{1+(1+4\kappa')^2}} + \frac{t}{2} + 2\sqrt{1+(1+4\kappa')^2}\left[1+\frac{(5+4\kappa')t^2}{4(1+(1+4\kappa')^2)}\right]\beta\theta\right\}\\
&\le 1-\theta.
\end{aligned}$$

By simple calculations, we obtain that when

$$\theta \;\le\; \frac{3\sqrt{1+(1+4\kappa')^2} - \left[1+\sqrt{1+(1+4\kappa')^2}\right]t}{\left[4+4(1+4\kappa')^2+(5+4\kappa')t^2\right]\beta + 3\sqrt{1+(1+4\kappa')^2}},$$

the new iterate (x^+, s^+) always satisfies $\delta(x^+, s^+; t_+) \le \tau t_+$ with $\tau = \frac{1}{2\sqrt{1+(1+4\kappa')^2}}$ after a full-Newton step. Then, the desired result follows. $\square$

4. Polynomial complexity

For $t \in (0, t_0]$ with $t_0 = 1$, we choose a strictly feasible initial point $(x^0, s^0) \in \mathcal{F}^0$ such that $x^0 s^0 > \omega$. Take

$$\theta \;\le \theta(t) := \frac{3\sqrt{1+(1+4\kappa')^2} - \left[1+\sqrt{1+(1+4\kappa')^2}\right]t}{\left[4+4(1+4\kappa')^2+(5+4\kappa')t^2\right]\beta + 3\sqrt{1+(1+4\kappa')^2}}, \tag{34}$$

where β is given in (25). From (2), (6) and (17), we have $\omega(t_0) = x^0 s^0$ and $v^0 = e$ which satisfies $\delta(v^0) = 0 \le \frac{t_0}{2\sqrt{1+(1+4\kappa')^2}}$. With this and previous discussions, we are in a position to show that Algorithm 1 has polynomial complexity.

Theorem 2. *Let (x^0, s^0) be a strictly feasible point of $P_*(\kappa)$-WLCP (1) such that $x^0 s^0 > \omega$. If the update parameter*

$$\theta = \theta_{\min} := \frac{4-\sqrt{2}}{6+5\sqrt{2}\beta+8\beta\sqrt{1+(1+4\kappa')^2}}, \tag{35}$$

then Algorithm 1 provides an ε-approximate solution for $P_*(\kappa)$-WLCP (1) after at most

$$\left\lceil \frac{1}{\theta_{\min}} \log\left(\frac{\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\|}{\varepsilon} \right) \right\rceil + 1$$

iterations, where β is given in (25).

Proof. Since $v_+ = \sqrt{\frac{x^+ s^+}{\omega(t)}}$ and $x^0 s^0 > \omega$, we can conclude from (2) and Lemma 4 that

$$\begin{aligned} \|x^+ s^+ - \omega\| &\le \|x^+ s^+ - \omega(t)\| + \|\omega(t) - \omega\| \\ &= \left\| \left(v_+^2 - e\right)\omega(t) \right\| + \left\| x^0 s^0 - \omega \right\| t \\ &\le \left\| e - v_+^2 \right\| \cdot \|\omega(t)\|_\infty + \left\| x^0 s^0 - \omega \right\| t \\ &\le \left(1 + \sqrt{1+(1+4\kappa')^2}\right)\delta^2 \max(x^0 s^0) + \left\| x^0 s^0 - \omega \right\| t. \end{aligned} \tag{36}$$

Since $\delta \le \frac{t}{2\sqrt{1+(1+4\kappa')^2}}$ and $v > \frac{e}{2}$, we obtain from (36) that

$$\begin{aligned} \|x^+ s^+ - \omega\| &\le \left[\frac{1+\sqrt{1+(1+4\kappa')^2}}{4\left[1+(1+4\kappa')^2\right]} \max(x^0 s^0) t + \|x^0 s^0 - \omega\| \right] t \\ &\le \left[\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\| \right] t. \end{aligned} \tag{37}$$

Moreover, it follows from that (34) and (35) that for $t \in (0,1]$

$$\begin{aligned} \theta_{\min} &:= \frac{4-\sqrt{2}}{\left[5\sqrt{2} + 8\sqrt{1+(1+4\kappa')^2}\right]\beta + 6} \\ &\le \frac{4-\sqrt{2}}{\left[\frac{2(5+4\kappa')}{\sqrt{1+(1+4\kappa')^2}} + 8\sqrt{1+(1+4\kappa')^2}\right]\beta + 6} \\ &\le \frac{\left(2 - \frac{1}{\sqrt{2}}\right)\sqrt{1+(1+4\kappa')^2}}{\left[9 + 4\kappa' + 4(1+4\kappa')^2\right]\beta + 3\sqrt{1+(1+4\kappa')^2}} \\ &\le \frac{2\sqrt{1+(1+4\kappa')^2} - 1}{\left[9 + 4\kappa' + 4(1+4\kappa')^2\right]\beta + 3\sqrt{1+(1+4\kappa')^2}} \\ &\le \frac{3\sqrt{1+(1+4\kappa')^2} - \left[1 + \sqrt{1+(1+4\kappa')^2}\right]t}{\left[4 + 4(1+4\kappa')^2 + (5+4\kappa')t^2\right]\beta + 3\sqrt{1+(1+4\kappa')^2}}. \end{aligned} \tag{38}$$

Taking into account of the fact that $t_0 = 1$, $t_k = (1-\theta_{\min})t_{k-1}$ and using (37) and (38), we have

$$\begin{aligned} &\left\| x^k s^k - \omega \right\| \\ &\le \left[\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\| \right] t_{k-1} \\ &\le \left[\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\| \right] (1-\theta_{\min})^{k-1} \end{aligned}$$

after k iterations. Therefore, the stopping criterion $\|x^k s^k - \omega\| \le \varepsilon$ holds if

$$\left[\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\| \right] (1-\theta_{\min})^{k-1} \le \varepsilon.$$

By taking logarithm, we obtain

$$\begin{aligned} &(k-1)\log(1-\theta_{\min}) \\ &\le \log\varepsilon - \log\left(\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\| \right). \end{aligned}$$

Using the inequality $\log(1-\theta_{\min}) \le -\theta_{\min}$ for $\theta_{\min} \in (0,1)$, we know that Algorithm 1 stops if

$$k \ge \frac{1}{\theta_{\min}} \log\left(\frac{\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} \max(x^0 s^0) + \|x^0 s^0 - \omega\|}{\varepsilon} \right) + 1.$$

□

Corollary 1. *Let (x^0, s^0) be a strictly feasible point of $P_*(\kappa)$-WLCP (1) such that $x^0 s^0 > \omega$, and θ be given by (34). Then, Algorithm 1 requires at most*

$$O\left(\gamma \log\left(\frac{\frac{1}{2(1+4\kappa')} \max(x^0 s^0) + \|x^0 s^0 - \omega\|}{\varepsilon} \right)\right)$$

iterations to find an ε-approximate solution for $P_*(\kappa)$-WLCP (1), where

$$\gamma := 3 + \left(8 + 13\kappa'\right) \left\| \frac{x^0 s^0 - \omega}{\chi\omega + (e-\chi)\min(x^0 s^0)} \right\|.$$

Proof. By (35), we know that

$$\begin{aligned}
\frac{1}{\theta_{\min}} &= \frac{6 + 5\sqrt{2}\beta + 8\beta\sqrt{1+(1+4\kappa')^2}}{4-\sqrt{2}} \\
&\le \frac{2}{5}\left[6 + 5\sqrt{2}\beta + 8\beta\sqrt{1+(1+4\kappa')^2}\right] \\
&\le \frac{2}{5}\left[6 + 5\sqrt{2}\beta + 8\beta\left(\sqrt{2} + 4\kappa'\right)\right] \\
&= \frac{2}{5}\left(6 + 13\sqrt{2}\beta + 32\beta\kappa'\right) \\
&\le 3 + (8+13\kappa')\beta \\
&= 3 + (8+13\kappa')\left\| \frac{x^0 s^0 - \omega}{\chi\omega + (e-\chi)\min(x^0 s^0)} \right\| := \gamma.
\end{aligned} \tag{39}$$

Moreover, we compute that

$$\begin{aligned}
\frac{1+\sqrt{1+(1+4\kappa')^2}}{4+4(1+4\kappa')^2} &= \frac{1 + \frac{1}{\sqrt{1+(1+4\kappa')^2}}}{4\sqrt{1+(1+4\kappa')^2}} \\
&\le \frac{1+\frac{1}{\sqrt{2}}}{4\sqrt{1+(1+4\kappa')^2}} \\
&< \frac{1}{2\sqrt{1+(1+4\kappa')^2}} \\
&< \frac{1}{2(1+4\kappa')}.
\end{aligned} \tag{40}$$

Then, combining (39), (40) and Theorem 2 yields the desired result. □

By Theorem 2, we know that the complexity of Algorithm 1 is polynomially dependent on the size of $P_*(\kappa)$-WLCP, because $\|x^0 s^0 - \omega\|$ and $\left\| \frac{x^0 s^0 - \omega}{\chi\omega + (e-\chi)\min(x^0 s^0)} \right\|$ involve n-dimensional variables, leading to calculations that grow polynomially with n. The IPM for $P_*(\kappa)$–WLCP in Potra [37] has a polynomial iteration complexity with

$$O\left((1+\kappa) \frac{\frac{(x^0)^T s^0}{n} + \left\|x^0 s^0 - \omega\right\|}{\min x^0 s^0} \log \frac{\frac{(x^0)^T s^0}{n} + \left\|x^0 s^0 - \omega\right\|}{\varepsilon} \right).$$

Our complexity results are similar to the results of Potra [37]. Especially, for $P_*(\kappa)$-LCP, i.e., in the case of $\omega = 0$, we have by (9) that $\chi = 0$ and then

$$\begin{aligned}
\left\| \frac{x^0 s^0 - \omega}{\chi\omega + (e-\chi)\min(x^0 s^0)} \right\| &= \left\| \frac{x^0 s^0}{\min(x^0 s^0) e} \right\| \\
&= \sqrt{\sum_i^n \left(\frac{x_i^0 s_i^0}{\min(x^0 s^0)} \right)^2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\min(x^0 s^0)} \sqrt{\sum_i^n \left(x_i^0 s_i^0\right)^2} \\
&= \frac{\left\|x^0 s^0\right\|}{\min(x^0 s^0)} \\
&\geq \sqrt{n}.
\end{aligned}
$$

According to Corollary 1, the computational complexity of Algorithm 1 is

$$
O\left((1+2\kappa')\frac{\left\|x^0 s^0\right\|}{\min(x^0 s^0)} \log\left(\frac{\frac{1}{2(1+4\kappa')}\max(x^0 s^0) + \left\|x^0 s^0\right\|}{\varepsilon}\right)\right),
$$

which is slightly bigger than the known complexity of the IPM for $P_*(\kappa)$-LCP [30]

$$
O\left((1+\kappa)\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon}\right).
$$

5. Numerical experiments

In this section, we test Algorithm 1 for $P_*(\kappa)$-WLCPs and compare it with Asadi's algorithm [32] and Kheirfam's algorithm [31]. All experiments are performed on AMD Ryzen 9 6900HX with Radeon Graphics 3.30 GHz, 32 GB RAM by using MATLAB R2021a. The linear system is solved using "linsolve" in MATLAB in order to obtain the search direction (d_x, d_s). Problem 1, Problem 2, Problem 3 and Problem 6 are constructed according to the methods summarized in reference [41]. We indicate the value of the running time (in seconds), iteration number and $\|xs-\omega\|$ by CPU, Iter and Gap, respectively. Let the stopping criterion be $\|xs-\omega\| \leq \varepsilon$ with $\varepsilon = 10^{-5}$.
Problem 1 Consider the $P_*(\kappa)$-WLCP with

$$
M = \begin{pmatrix}
-3 & 0 & -5 & 1 & -4 & 0 & 4 & -2 & -4 & 1 \\
0 & 4 & 25 & -5 & 0 & 0 & -20 & 10 & 0 & 0 \\
10 & -25 & -3 & 0 & -20 & -5 & 0 & 0 & -20 & 5 \\
-10 & 25 & 0 & 1 & 20 & 5 & 0 & -10 & 20 & -5 \\
-10 & 0 & 25 & -5 & 118 & 0 & -20 & 10 & 20 & -5 \\
0 & 0 & 20 & -4 & 0 & 20 & -16 & 8 & 0 & 0 \\
-10 & 25 & 0 & 0 & 20 & 5 & 16 & 0 & 20 & -5 \\
8 & -20 & 0 & -4 & -16 & -4 & 0 & 53 & -16 & 4 \\
-2 & 0 & 5 & -1 & 4 & 0 & -4 & 2 & 2 & -1 \\
4 & 0 & -10 & 2 & -8 & 0 & 8 & -4 & -8 & 0
\end{pmatrix},\quad
q = \begin{pmatrix} 14 \\ -12 \\ 60 \\ -44 \\ -131 \\ -26 \\ -69 \\ -3 \\ -3 \\ 18 \end{pmatrix},\quad
\omega = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.
$$

For Problem 1, the $P_*(\kappa)$-matrix $M = M_0 + Q$, where M_0 is a $P_*(\kappa)$-matrix (denoted by IM_SU_10_01 from Nagy [42]) and $Q \in \mathbb{R}^{10\times 10}$ is a diagonal matrix with integer entries in the interval $[-5, 5]$. The starting point is chosen as $x^0 = e,\ s^0 = 2e$ with $e \in \mathbb{R}^{10}$. We solve Problem 1 using the parameters $\theta = 0.8,\ 0.99$, respectively, and depict the iteration points x, s produced by each iteration in Figs. 1–4. From Figs. 1 and 2, each component of the iteration point x, s remains strictly positive during each iteration, which indicates that the algorithm is feasible and effective for Problem 1 by using $\theta = 0.8$. Observing Figs. 3 and 4, when solving Problem 1 by taking $\theta = 0.99$, a negative component of the iteration point x appears at the third iteration and is lifted to a positive value at the fourth iteration. Thus, Algorithm 1 with $\theta = 0.99$ for Problem 1 produces occasionally an infeasible point with a negative component and gives rise to numerical instability, but does not affect its convergence. This example illustrates that even if there is an occasional infeasible iteration point due to smaller numerical inaccuracies, the algorithm can still achieve convergence. Although smaller errors usually do not prevent Algorithm 1 from reaching the termination condition, larger numerical errors may prevent the algorithm from taking a full step.
Problem 2 Consider the $P_*(\kappa)$-WLCP with

$$
M = \begin{pmatrix} M_1 & M_1 \\ M_1 & M_1 \end{pmatrix}, M_1 = \begin{pmatrix} M_0 & M_0 \\ M_0 & M_0 \end{pmatrix}, \quad q = -0.9Me + 0.8e,
$$

$$
\omega = \begin{pmatrix}
0.7 & 0.7 & 0.5 & 0.7 & 0.3 & 0.5 & 0.7 & 0.4 & 0.7 & 0.6 \\
0.4 & 0.5 & 0.7 & 0.2 & 0.3 & 0.7 & 0.3 & 0.5 & 0.7 & 0.7 \\
0.5 & 0.1 & 0.1 & 0.5 & 0.4 & 0.7 & 0.6 & 0.5 & 0.4 & 0.2 \\
0.7 & 0.4 & 0.1 & 0.5 & 0.4 & 0.1 & 0.7 & 0.3 & 0.2 & 0.1
\end{pmatrix}^T.
$$

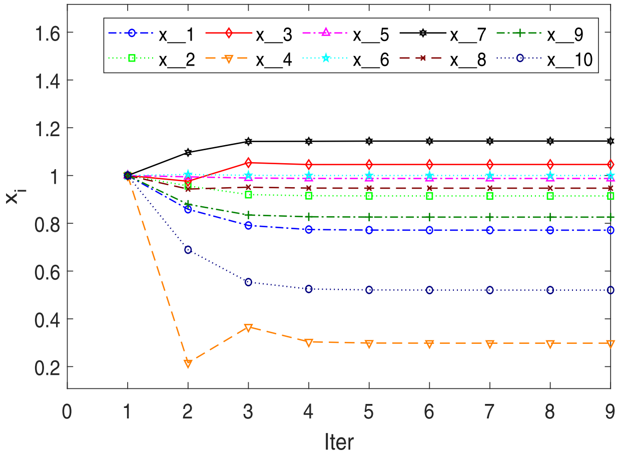

Fig. 1. The values of x_i with $\theta = 0.8$ for Problem 1.

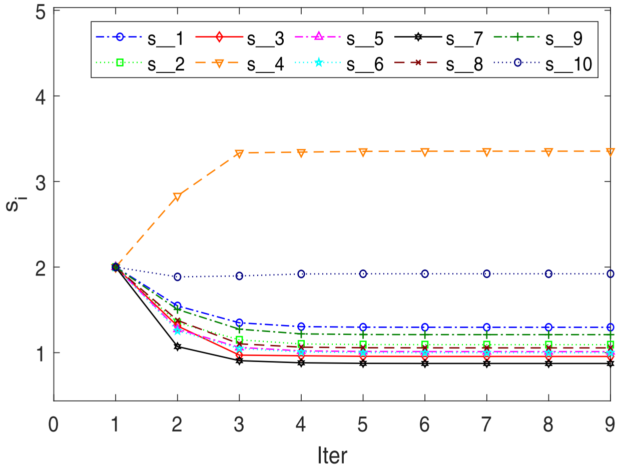

Fig. 2. The values of s_i with $\theta = 0.8$ for Problem 1.

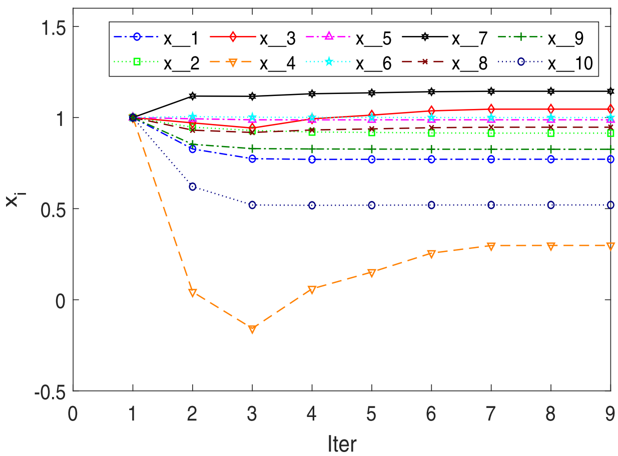

Fig. 3. The values of x_i with $\theta = 0.99$ for Problem 1.

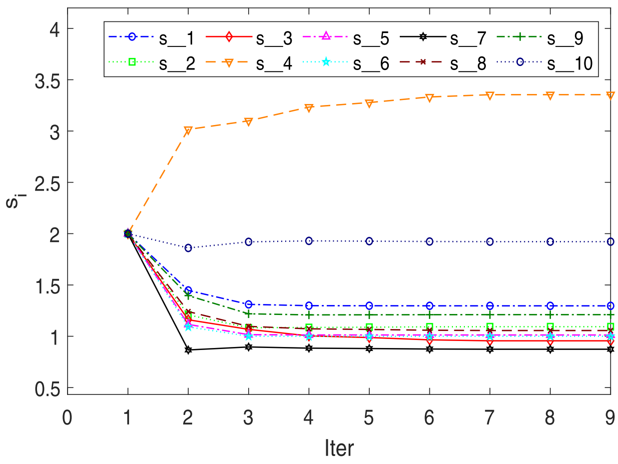

Fig. 4. The values of s_i with $\theta = 0.99$ for Problem 1.

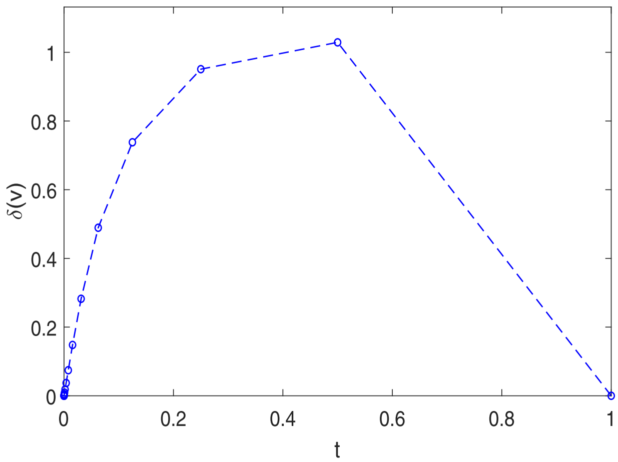

Fig. 5. The values of $\delta(v)$ with $\theta = 0.5$ for Problem 2.

where

$$M_0 = \begin{pmatrix} 144 & -16 & -72 & -24 & 48 & 60 & 20 & -96 & 120 & -32 \\ -80 & 220 & 60 & -75 & 90 & -105 & 5 & -60 & -200 & 20 \\ -64 & 96 & 180 & 72 & 60 & -12 & 28 & 96 & -80 & 24 \\ -16 & 8 & 6 & 84 & -42 & 18 & -10 & -48 & -60 & -4 \\ -64 & -32 & 60 & 24 & 144 & -48 & -12 & -32 & 20 & 32 \\ -24 & 56 & 24 & 30 & -42 & 42 & -6 & 8 & -50 & 20 \\ 0 & -24 & 36 & -48 & 36 & 30 & 24 & -48 & -20 & -4 \\ 80 & -64 & -48 & -12 & 36 & 24 & 8 & 160 & -160 & 24 \\ 36 & 60 & 36 & 9 & 36 & 45 & 18 & 0 & 90 & -48 \\ -60 & -84 & 9 & 0 & -72 & -54 & -9 & -84 & 90 & 78 \end{pmatrix}, e \in \mathbb{R}^{40}.$$

For Problem 2, the $P_*(\kappa)$-matrix $M_0 = PM_2Q$, where $M_2 \in \mathbb{R}^{10\times 10}$ is a $P_*(\kappa)$-matrix (denoted by ENM_SU_10_08 from Nagy [42]) and P, Q are both 10-dimensional diagonal matrices with integer entries in the interval $[1, 5]$. The starting point is chosen as $x^0 = 0.9e$, $s^0 = 0.8e$ with $e \in \mathbb{R}^{40}$. The update parameter is $\theta = 0.5$ and the explicit theoretical value of the parameter is $\theta_{\min} = \frac{4-\sqrt{2}}{6+13\sqrt{2\beta}}$ with $\beta = \left\| \frac{x^0 s^0 - \omega}{\omega} \right\|$ given by (25) and (35). We plot the values of $\delta(v)$ with t for two values of the parameter θ in Figs. 5 and 6. From Figs. 5 and 6, we can see that as the value of t decreases, $\delta(v)$ decreases and eventually converges to 0. When $\theta = 0.5$, Algorithm 1 produces a large δ, which indicates that the iteration points x, s remain in a large neighborhood of the central path, which may result in fluctuating values and cause the algorithm not to converge stably. In contrast, Algorithm 1 with the theoretical parameter $\theta_{\min}$ results in a smaller δ. When an appropriate value of θ is chosen, the iteration points x, s generated by Algorithm 1 remain within a smaller neighborhood of the central path, avoiding the generation of negative components and thus ensuring stable convergence.

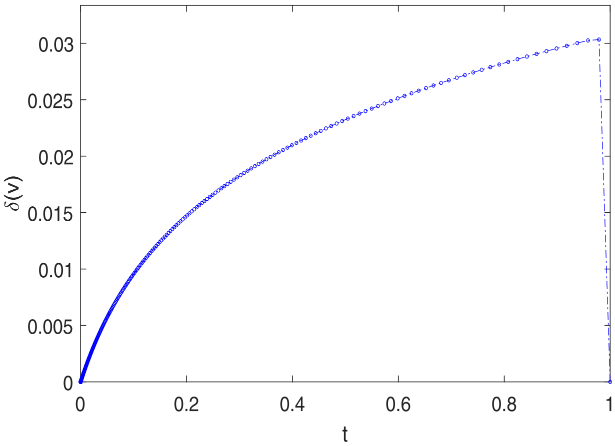

Fig. 6. The values of $\delta(v)$ with $\theta = \theta_{\min}$ for Problem 2.

Table 2
Numerical results for Problem 3 with different initial points.

$x^0 \in \mathbb{R}^{50}$	$s^0 \in \mathbb{R}^{50}$	Gap	CPU	Iter
e	$5e$	4.5340E-06	0.01628	36
e	$10e$	8.3873E-06	0.00458	28
e	$20e$	8.0081E-06	0.00410	25
e	$100e$	5.2657E-06	0.00498	28
e	$500e$	6.5848E-06	0.00541	30

Problem 3 Consider the $P_*(\kappa)$-WLCPs with

$$M = \begin{pmatrix} M_0 & O \\ M_1 & M_0 \end{pmatrix}, \quad q = -Mx^0 + s^0, \quad \omega = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

where

$$M_0 = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ -1 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & 1 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & \cdots & 6 \\ 2 & 6 & 9 & \cdots & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 6 & 10 & \cdots & 2n-3 \end{pmatrix}.$$

For Problem 3, we set the update parameter $\theta = 0.5$ and dimension $n = 50$. We choose different initial points $x^0 = e$ and $s^0 \in \{0.1e, 5e, 10e, 100e, 500e\}$, where $e \in \mathbb{R}^{50}$. The average of CPU, Iter and Gap for these initial points are given in Table 2. Moreover, for Problem 3, we choose the initial points $(x^0, s^0) = (e, 8e)$, where $e \in \mathbb{R}^{50}$. The numerical results for Problem 3 with various dimensions $n \in \{10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$ and update parameters $\theta \in \{0.1, 0.3, 0.5\}$ are presented in Table 3.

Problem 4 (Harker's problem in Harker and Pang [43]) Consider the $P_*(0)$-WLCPs with

$$M = \begin{pmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \cdots & 0 \\ 0 & -1 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 4 \end{pmatrix}, \quad q = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \omega = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

For Problem 4, we set the update parameter $\theta = 0.5$ and dimension $n = 50$. We choose different initial points $x^0 \in \{e, 2e, 5e, 10e, 10^2 e\}$, where $e \in \mathbb{R}^{50}$ and $s^0 = Mx^0 + q$. The average of CPU, Iter and Gap for these initial points are given in Table 4. Moreover, we choose the initial points $(x^0, s^0) = (e, Me + e)$, and the numerical results for Problem 4 with various update parameters $\theta \in \{0.1, 0.3, 0.5\}$ and dimensions $n \in \{10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$ are presented in Table 5.

From Tables 2 and 4, we observe that the choice of initial points has a certain influence on running time and iteration number. In Tables 3 and 5, it can be seen that running time and iteration number decrease as θ increases. For Problem 3, taking $\theta = 0.1$, the running time required to solve the 1000×1000 problem instances is less than 20 seconds, and taking $\theta = 0.5$ gives better numerical

Table 3
Numerical results for Problem 3 with different θ and n.

n	$\theta = 0.1$		$\theta = 0.3$		$\theta = 0.5$	
	CPU	Iter	CPU	Iter	CPU	Iter
10	0.00449	129	0.00127	39	0.00069	18
20	0.00791	132	0.00202	40	0.00105	18
50	0.03117	136	0.00740	41	0.00418	19
100	0.09484	139	0.02159	42	0.01130	19
200	0.40984	143	0.12353	43	0.06276	20
300	1.13669	144	0.37213	44	0.37774	21
400	2.31830	146	0.60487	44	0.34786	21
500	4.36144	147	1.15435	45	0.65813	22
600	7.56975	148	2.22277	45	3.49840	22
700	10.70742	149	2.84216	46	1.69987	23
800	13.11426	150	3.75040	47	2.08271	24
900	16.43551	151	6.67633	48	3.51493	24
1000	18.81505	152	8.50218	49	5.89956	25

Table 4
Numerical results for Problem 4 with different initial points.

$x^0 \in \mathbb{R}^{50}$	$s^0 \in \mathbb{R}^{50}$	Gap	CPU	Iter
e	$(4 \quad 3 \quad \cdots \quad 3 \quad 4)$	9.0474E-06	0.00139	21
$2e$	$(7 \quad 5 \quad \cdots \quad 5 \quad 7)$	5.0520E-06	0.00135	23
$5e$	$(16 \quad 11 \quad \cdots \quad 11 \quad 16)$	7.5964E-06	0.00147	26
$10e$	$(31 \quad 21 \quad \cdots \quad 21 \quad 31)$	7.3652E-06	0.00165	28
$100e$	$(301 \quad 201 \quad \cdots \quad 201 \quad 301)$	5.5476E-06	0.00198	34

Table 5
Numerical results for Problem 4 with different θ and n.

n	$\theta = 0.1$		$\theta = 0.3$		$\theta = 0.5$	
	CPU	Iter	CPU	Iter	CPU	Iter
10	0.00216	115	0.00084	35	0.00039	19
20	0.00321	118	0.00102	36	0.00055	19
50	0.01580	123	0.00278	37	0.00148	20
100	0.02897	126	0.00778	38	0.00404	20
200	0.18781	129	0.06073	39	0.03016	21
300	0.43216	131	0.14323	40	0.08407	21
400	0.87437	133	0.26834	40	0.13747	21
500	1.36757	134	0.40624	41	0.21305	22
600	2.40681	135	0.67889	41	0.35597	22
700	3.44338	135	1.02962	41	0.54707	22
800	5.05017	136	1.47352	41	0.80652	22
900	6.68859	136	1.95822	43	1.02425	23
1000	8.33534	137	2.54317	45	1.37030	24

results. For Problem 4, even when handling the largest 1000×1000 problem instances, the running time is less than 10 seconds. Moreover, running time of Algorithm 1 grows slightly as the dimension goes up, which reflects the stability of Algorithm 1. To sum up, Problem 3 and Problem 4 with varying sizes ($n \leq 1000$) can be solved with ease by Algorithm 1.

Problem 5 (Watson's problem in Watson [44]) Consider the $P_*(0)$-WLCPs with

$$M = \begin{pmatrix} 6 & -4 & 2 & \cdots & 0 \\ -4 & 6 & -4 & \cdots & 0 \\ 2 & -4 & 6 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 6 \end{pmatrix}, \quad q = -Me + 6e, \quad \omega = rand(n, 1).$$

Problem 6 Consider the $P_*(\kappa)$-WLCPs with

$$M = \begin{pmatrix} 3 & 0 & 0 & \cdots & 0 \\ -2 & 3 & 0 & \cdots & 0 \\ -2 & -2 & 3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -2 & -2 & -2 & \cdots & 3 \end{pmatrix}, \quad q = -Me + 8e, \quad \omega = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

For convenience we let Algorithm 2 and Algorithm 3 represent the IPM for monotone WLCP given in Asadi et al. [32] and the IPM for $P_*(\kappa)$-LCP proposed in Kheirfam and Haghighi [31], respectively. The monotone WLCP in Asadi et al. [32] is formulated as:

$$Px + Qs + Ry = a, \quad xs = \omega, \quad x \geq 0, \quad s \geq 0, \tag{41}$$

where $P\Delta x + Q\Delta s + R\Delta y = 0$ implies $\Delta x^T \Delta s \geq 0$. The $P_*(\kappa)$-LCP in Kheirfam and Haghighi [31] is formulated as:

$$s = Mx + q, \quad xs = 0, \quad x \geq 0, \quad s \geq 0, \tag{42}$$

where M is a $P_*(\kappa)$-matrix. Note that when $\kappa = 0$, in the case of $P = M, Q = -I, R = O$, and $a = -q$, then the monotone WLCP solved in Asadi et al. [32] is the $P_*(0)$-WLCP (1) addressed in our paper. When $\omega = 0$, the $P_*(\kappa)$-WLCP (1) becomes $P_*(\kappa)$-LCP (42) in Kheirfam and Haghighi [31]. Therefore, in different experimental settings, we evaluate the performance of these algorithms as follows

- Solve Problem 5 ($P_*(0)$-WLCPs) using Algorithms 1 and 2, respectively.
- Solve Problem 6 ($P_*(\kappa)$-LCPs) using Algorithms 1 and 3, respectively.

The experiments for Problem 5 are implemented based on the same feasible initial points $(x^0, s^0) = (e, 6e)$, the dimensions $n \in \{40, 80, 100, 200, 300, 400, 500, 600\}$ and updated parameter $\theta \in \{0.2, 0.5\}$. The experiments for Problem 6 use the initial points $(x^0, s^0) = (e, 8e)$, with $\theta \in \{0.2, 0.5\}$ and dimensions $n \in \{50, 80, 100, 120, 160, 200, 300, 400\}$. Due to the randomness of ω, we repeat the calculation 10 times to reduce the experimental error.

In addition, Algorithm 1 is different from the other two algorithms in the following aspects.

1) Algorithm 2 could solve the monotone-WLCPs with $\omega \geq 0$ and $\kappa = 0$, Algorithm 3 could solve the $P_*(\kappa)$-LCPs with $\omega = 0$ and $\kappa \geq 0$, and Algorithm 1 could solve the $P_*(\kappa)$-WLCPs with $\omega \geq 0$ and $\kappa \geq 0$.
2) The key distinction among these algorithms lies in the linear systems used to determine the search direction, described as follows
   - Algorithm 1: The search direction is obtained by solving system (8), which uses the AET function $\varphi(t) = t - \sqrt{t}$ to transform equation of the central path system.
   - Algorithm 2: The system shares the same first equation as system (5), but the second equation is $s\Delta x + x\Delta s = \omega(t) - xs$, where $\omega(t) = \left(1 - \frac{nt}{(x^0)^T s^0}\right)\omega + \frac{nt}{(x^0)^T s^0} x^0 s^0$, $t \in (0, 1]$. The equation of the central path system is not transformed, namely, the AET function here is $\varphi(t) = t$.
   - Algorithm 3: Similarly, its first equation matches system (8), whereas the second equation is $d_x + d_s = e - v^2$, since they use the AET function $\varphi(t) = \frac{\sqrt{t}}{2(1+\sqrt{t})}$.
3) Although both Algorithm 1 and Algorithm 2 could solve WLCPs, they differ in the linear system of search direction, the problem to be solved, and $\omega(t)$. Additionally, they employ different proximity measures δ, scaled vectors v, and threshold parameters for the neighbourhood τ. For Algorithm 1, there are $\delta = \left\| \frac{v-v^2}{2v-e} \right\|$, $v = \sqrt{\frac{xs}{\omega(t)}}$, $\delta \leq \tau t$ with $\tau = \frac{1}{2\sqrt{1+(1+4\kappa')^2}}$, $t \in (0, 1]$. For Algorithm 2, there are $\delta = \left\| \frac{\omega(t)}{t} - v^2 \right\|$, $v = \sqrt{\frac{xs}{t}}$ with $t \in \left(0, \frac{(x^0)^T s^0}{n}\right)$, $\delta \leq \tau$ with $\tau = \frac{n \min x^0 s^0}{2(x^0)^T s^0}$.

Through comparative experiments in different problem contexts, we analyze the performance of each algorithm. Detailed results and comparisons in terms of the number of iterations and running time can be found in Tables 6 and 7. To be more specific, Table 6 shows the comparison results of Algorithms 1 and 2, and Table 7 collects the results of Algorithms 1 and 3.
From observing Tables 6 and 7, some numerical findings can be drawn:

- In comparison to Algorithm 2, Algorithm 1 outperforms it with respect to both numbers of iteration and running time. Specifically, Algorithm 1 requires a smaller running time than Algorithm 2. Algorithm 1 requires much fewer iterations than Algorithm 2, which is especially noticeable in solving high-dimensional $n = 600$ problems using small parameters $\theta = 0.2$.
- The total running time of Algorithm 1 is not significantly different from that of Algorithm 3 when $\theta = 0.5$, but its total iterations are fewer than that of Algorithm 2.
- Algorithm 1 has the fewest numbers of iteration and smallest running time regardless of whether sets up parameter $\theta = 0.2$ or $\theta = 0.5$, indicating that it is more effective at solving Problem 5 than Algorithm 2. Moreover, Algorithm 1 gives slightly better results than Algorithm 3 when solving Problem 6 in lower dimensions.

From Table 6, it can be seen that there is a noticeable difference between Algorithms 1 and 2 in terms of running time. To visualize this difference, we compare the running time (in seconds) for a single iteration of Algorithms 1 and 2 when solving Problem 5 based on $(x^0, s^0) = (e, 6e)$ and $\theta = 0.5$ in Fig. 7 and Fig. 8. As shown in Figs. 7 and 8, a single iteration of Algorithm 1 has less running time than Algorithm 2 and has a smaller fluctuation range. Since the single iteration of Algorithm 1 takes less time, the overall running time is less than that of Algorithm 2.

Moreover, Algorithm 1 utilizes the same AET function $\varphi(t) = t - \sqrt{t}$ as Algorithm 4 (the algorithm in Darvay et al. [30]) to compute the search direction. However, Algorithm 1 solves the $P_*(\kappa)$-WLCP with the weight vector $\omega \geq 0$; while Algorithm 4 solves the $P_*(\kappa)$-LCP, which is a special case of $P_*(\kappa)$-WLCP with $\omega = 0$. We solve Problem 5 by Algorithm 1 and Algorithm 4 based on the initial points $(x^0, s^0) = (e, 6e)$. The numerical results for Problem 5 obtained by Algorithm 1 and Algorithm 4 are shown in Table 8, where the theoretical values of parameters $\theta(t)$ and $\theta_{\min}$ for Algorithm 1 are given by (34) and (35), respectively. The parameter $\theta = \frac{1}{(9\kappa+8)\sqrt{n}}$ is the theoretical parameters for Algorithm 4 in Darvay et al. [30]. Observing Table 8, with a fixed parameter $\theta = 0.8$, there is little difference in the number of iterations between Algorithms 1 and 4, but Algorithm 4 has a more obvious advantage in running time. However, Algorithm 1 demonstrates significantly better performance than Algorithm 4 when theoretical parameters are used.

In summary, the aforementioned numerical results demonstrate that Algorithm 1 can solve the $P_*(\kappa)$-WLCP efficiently.

Table 6
Comparison of Algorithm 1 and Algorithm 2 for Problem 5 with different n and θ.

		Algorithm 1		Algorithm 2	
		$\theta = 0.2$	$\theta = 0.5$	$\theta = 0.2$	$\theta = 0.5$
40	CPU	0.00606	0.00292	0.05990	0.02136
	Iter	64.3	21.0	72.3	24.3
80	CPU	0.02378	0.01050	0.08888	0.03067
	Iter	65.7	22.0	74.3	24.7
100	CPU	0.03819	0.01237	0.15793	0.05485
	Iter	66.3	22.7	75.0	25.0
200	CPU	0.14203	0.11562	1.11629	0.38454
	Iter	67.7	23.0	76.3	25.7
300	CPU	0.47790	0.22344	3.22503	1.08730
	Iter	69.3	24.7	77.3	25.7
400	CPU	1.14429	0.41286	6.63892	2.30643
	Iter	70.0	26.3	77.3	26.3
500	CPU	1.96725	0.84345	12.82270	4.28734
	Iter	70.3	27.3	78.3	28.7
600	CPU	3.23566	1.28296	27.50821	9.18333
	Iter	71.3	29.0	78.7	31.0
Total	CPU	7.03516	2.90411	51.61787	17.35583
	Iter	545.0	196.0	609.7	211.3

Table 7
Comparison of Algorithms 1 and 3 for Problem 6 with different n and θ.

		Algorithm 1		Algorithm 3	
		$\theta = 0.2$	$\theta = 0.5$	$\theta = 0.2$	$\theta = 0.5$
50	CPU	0.02047	0.00612	0.05126	0.01779
	Iter	92.0	31.0	100.0	32.0
80	CPU	0.04179	0.01513	0.07186	0.02248
	Iter	96.0	32.0	104.0	34.0
100	CPU	0.07157	0.02321	0.10289	0.02827
	Iter	97.0	32.0	106.0	34.0
120	CPU	0.10433	0.04132	0.13268	0.05747
	Iter	98.0	39.0	107.0	47.0
150	CPU	0.30682	0.09366	0.32949	0.11712
	Iter	100.0	37.0	110.0	41.0
200	CPU	0.44237	0.15838	0.50616	0.20108
	Iter	102.0	39.0	112.0	42.0
300	CPU	1.00906	0.38632	1.26720	0.50342
	Iter	104.0	42.0	116.0	47.0
400	CPU	1.95364	1.09957	2.40285	1.87884
	Iter	106.0	50.0	118.0	54.0
Total	CPU	3.95004	1.82371	4.86438	2.82646
	Iter	795.0	302.0	873.0	331.0

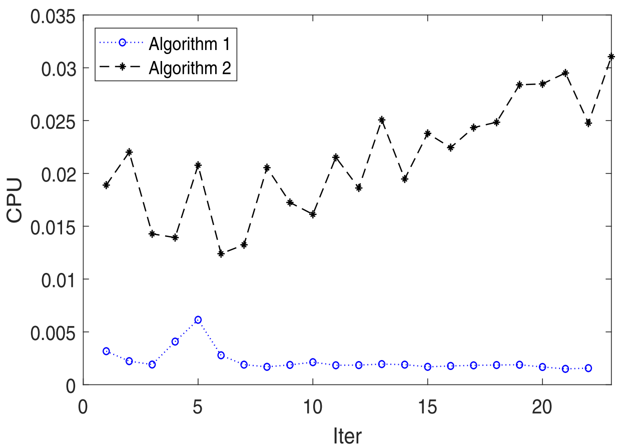

Fig. 7. CPU of Algorithms 1 and 2 for Problem 5 with $n = 40$.

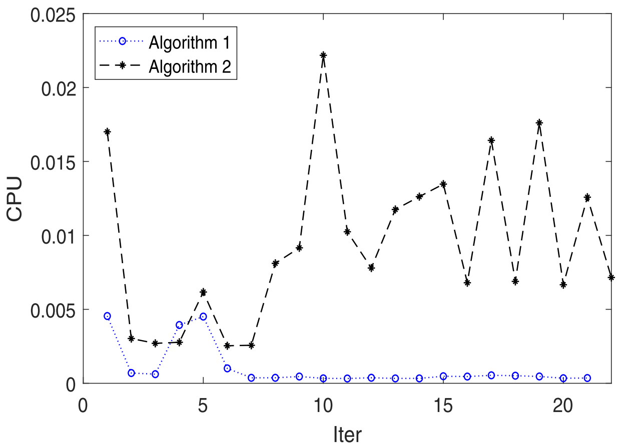

Fig. 8. CPU of Algorithms 1 and 2 for Problem 5 with $n = 100$.

Table 8
Comparison of Algorithm 1 and Algorithm 4 for Problem 5 with different n and θ.

	n	Algorithm 1 $\theta = \theta(t)$	$\theta = \theta_{\min}$	$\theta = 0.8$	Algorithm 4 $\theta = \frac{1}{(9\kappa+8)\sqrt{n}}$	$\theta = 0.8$
10	CPU	0.00173	0.00555	0.00046	0.00821	0.00019
	Iter	104	353	11	387	10
20	CPU	0.00707	0.02302	0.00067	0.02330	0.00039
	Iter	147	499	11	575	11
40	CPU	0.01461	0.04777	0.00163	0.07877	0.00084
	Iter	207	710	11	852	11
70	CPU	0.12291	0.44202	0.00494	0.32200	0.00196
	Iter	276	947	11	1167	11
80	CPU	0.20269	0.72973	0.00734	0.75515	0.00460
	Iter	331	1140	12	1424	12
Total	CPU	0.34900	1.24808	0.01503	1.18744	0.00797
	Iter	1065	3649	56	4405	55

6. Conclusions

In light of considering the kernel function $\varphi(t) = t - \sqrt{t}$, we extend a feasible IPM with full-Newton step for $P_*(\kappa)$-LCP [30] to $P_*(\kappa)$-WLCP. The analysis of our algorithm is more complicated than $P_*(\kappa)$-LCP case because of the nonnegative weight vector ω in $P_*(\kappa)$-WLCP. Using AET technology and Newton's method, we obtain new search directions. By choosing the appropriate initial point and parameters, we verify the feasibility and convergence of Algorithm 1. With the suitable choices of the update parameter θ, we prove that the iteration points always stay in our τt-neighborhood. Then, Algorithm 1 is shown to have polynomial complexity. Finally, some preliminary numerical results illustrate the efficiency of our algorithm.

CRediT authorship contribution statement

Xiaoni Chi: Supervision, Methodology, Formal analysis, Conceptualization; **Lin Gan:** Writing – original draft, Methodology; **Zhuoran Gao:** Writing – original draft, Methodology; **Jein-Shan Chen:** Writing – review & editing, Conceptualization.

Acknowledgements

The first three authors are grateful to the editor and the anonymous referees for their valuable comments and suggestions which have greatly improved this paper. This research is supported by the National Natural Science Foundation of China (no. 12361064), Guangxi Natural Science Foundation (no 2021GXNSFAA220034) and the Science and Technology Project of Guangxi (Guike AD23023002), China. The 4th author's work is supported by National Science and Technology Council (NSTC 112-2115-M-003-012-MY2, NSTC 113-2124-M-003-002); Higher Education Sprout Project-Center for Optimal Intelligent Data Analytics and Prediction.

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.apnum.2025.10.003

References

[1] F.A. Potra, Weighted complementarity problems-a new paradigm for computing equilibria, SIAM J. Optim. 22 (4) (2012) 1634–1654.
[2] K.G. Murty, Linear Complementarity, Linear and Nonlinear Programming, Helderman-Verlag, 1988.
[3] Y.Y. Ye, A path to the Arrow-Debreu competitive market equilibrium, Math. Program. 111 (1–2) (2008) 315–348.
[4] K.M. Anstreicher, Interior-point algorithms for a generalization of linear programming and weighted centring, Optim. Method. Softw. 27 (4–5) (2012) 605–612.
[5] G. Goel, V.V. Vazirani, A perfect price discrimination market model with production, and a rational convex program for it, Math. Oper. Res 36 (4) (2011) 762–782.
[6] K. Jain, V.V. Vazirani, Eisenberg-Gale markets: algorithms and game-theoretic properties, Game. Econ. Behav. 70 (1) (2010) 84–106.
[7] M. Kojima, N. Megiddo, T. Noma, A. Yoshise, A Unified Approach to Interior-Point Algorithms for Linear Complementarity Problems, 538, Springer, Berlin, 1991.
[8] S.M. Guu, R.W. Cottle, On a Subclass of P_0, Linear Algebr. Appl. 223–224 (1995) 325–335.
[9] H. Väliaho, P_*-matrices are just sufficient, Linear Algebr. Appl. 239 (1–3) (1996) 103–108.
[10] S.J. Chung, NP-completeness of the linear complementarity problem, J. Optim. Theory Appl. 60 (1989) 393–399.
[11] G. Lesaja, C. Roos, Unified analysis of kernel-based interior-point methods for $P_*(\kappa)$-linear complementarity problems, SIAM J. Optim. 20 (6) (2010) 3014–3039.
[12] Y.Q. Bai, M.E. Ghami, C. Roos, A new efficient large-update primal-dual interior-point method based on a finite barrier, SIAM J. Optim. 13 (3) (2003) 766–782.
[13] Y.Q. Bai, G. Lesaja, C. Roos, G.Q. Wang, M.E. Ghami, A class of large-update and small-update primal-dual interior-point algorithms for linear optimization, J. Optim. Theory Appl. 138 (3) (2008) 341–359.
[14] L.P. Zhang, Y.H. Xu, A full-Newton step infeasible interior point algorithm and its parameters analysis, J. Ind. Manag. Optim. 20 (5) (2024) 1916–1933.
[15] B. Kheirfam, An infeasible interior point method for the monotone SDLCP based on a transformation of the central path, Appl. Math. Comput. 57 (1) (2018) 685–702.
[16] M. Pirhaji, M. Zangiabadi, H. Mansouri, A corrector-predictor arc search interior-point algorithm for symmetric optimization, Acta Math. Sci. 38 (4) (2018) 1269–1284.
[17] B. Alzalg, K. Badarneh, A. Ababneh, An infeasible interior-point algorithm for stochastic second-order cone optimization, J. Optim. Theory Appl. 181 (2019) 324–346.
[18] J.-S. Chen, J. Ye, J. Zhang, J.C. Zhou, Exact formula for the second-order tangent set of the second-order cone complementarity set, SIAM J. Optim. 29 (4) (2019) 2986–3011.
[19] J. Miao, A quadratically convergent $O((1+\kappa)\sqrt{n}L)$-iteration algorithm for the $P_*(\kappa)$-matrix linear complementarity problem, Math. Program. 69 (1–3) (1995) 355–368.
[20] T. Illés, J. Peng, C. Roos, T. Terlaky, A strongly polynomial rounding procedure yielding a maximally complementary solution for $P_*(\kappa)$ linear complementarity problems, SIAM J. Optim. 11 (2) (2000) 320–340.
[21] K. Amini, M.R. Peyghami, Exploring complexity of large update interior-point methods for $P_*(\kappa)$ linear complementarity problem based on kernel function, Appl. Math. Comput. 207 (2) (2009) 501–513.
[22] F. Gurtuna, C. Petra, F.A. Potra, O. Shevchenko, A. Vancea, Corrector-predictor methods for sufficient linear complementarity problems, Comput. Optim. Appl. 48 (3) (2011) 453–485.
[23] Y.H. Lee, Y.Y. Cho, G.M. Cho, Interior-point algorithms for $P_*(\kappa)$-LCP based on a new class of kernel functions, Glob. Optim. 58 (1) (2014) 137–149.
[24] G.Q. Wang, C.J. Yu, K.L. Teo, A full-Newton step feasible interior-point algorithm for $P_*(\kappa)$-linear complementarity problems, J. Glob. Optim. 59 (1) (2014) 81–99.
[25] C. Roos, T. Terlaky, J.-P. Vial, Theory and algorithms for Linear Optimization. An Interior Point Approach, John Wiley & Sons, Chichester, UK, 1997.
[26] G. Lesaja, F.A. Potra, Adaptive full Newton-step infeasible interior-point method for sufficient horizontal LCP, Optim. Method. Softw. 34 (5) (2019) 1014–1034.
[27] Z. Darvay, New interior point algorithms in linear programming, Adv. Model Optim. 5 (1) (2003) 51–92.
[28] Z. Darvay, I.M. Papp, P.R. Takács, Complexity analysis of a full-Newton step interior-point method for linear optimization, Period Math. Hung. 73 (1) (2016) 27–42.
[29] Z. Darvay, T. Illés, J. Povh, P.R. Rigó, Feasible corrector-predictor interior-point algorithm for $P_*(\kappa)$-linear complementarity problems based on a new search direction, SIAM J. Optim. 30 (3) (2020) 2628–2658.
[30] Z. Darvay, T. Illés, C. Majoros, Interior-point algorithm for sufficient LCPs based on the technique of algebraically equivalent transformation, Optim. Lett. 15 (1) (2021) 357–376.
[31] B. Kheirfam, M. Haghighi, A full-Newton step feasible interior-point algorithm for $P_*(\kappa)$-LCP based on a new search direction, Croat. Oper. Res. Rev. 7 (1) (2016) 277–290.
[32] A. Asadi, Z. Darvay, G. Lesaja, N. Mahdavi-Amiri, F. Potra, A full-Newton step interior-point method for monotone weighted linear complementarity problems, J. Optim. Theory Appl. 186 (115) (2020) 864–878.
[33] X.N. Chi, G.Q. Wang, G. Lesaja, Kernel-based full-Newton step feasible interior-point algorithm for $P_*(\kappa)$-weighted linear complementarity problem, J. Optim. Theory Appl. 202 (1) (2024) 108–132.
[34] N. Lu, Z.H. Huang, A smoothing Newton algorithm for a class of non-monotonic symmetric cone linear complementarity problems, J. Optim. Theory Appl. 161 (2014) 446–464.
[35] J.Y. Tang, A modified damped Gauss-Newton method for non-monotone weighted linear complementarity problems, Optim. Method. Softw. 2 (2021) 1145–1164.
[36] J. Zhang, A smoothing Newton algorithm for weighted linear complementarity problem, Optim. Lett. 10 (3) (2016) 499–509.
[37] F.A. Potra, Sufficient weighted complementarity problem, Comput. Optim. Appl. 64 (2) (2016) 467–488.
[38] X.N. Chi, M.S. Gowda, J.Y. Tao, The weighted horizontal linear complementarity problem on a Euclidean Jordan algebra, J. Glob. Optim. 73 (1) (2019) 153–169.
[39] J.K. Lee, Y.Y. Cho, J.H. Jin, G.M. Cho, A new full-Newton step infeasible interior-point method for $P_*(\kappa)$-linear complementarity problem, Optim. Lett. 18 (4) (2024) 943–964.
[40] X.N. Chi, Y.P. Yang, J.-S. Chen, Complexity analysis of a predictor-corrector interior-point algorithm for $P_*(\kappa)$-weighted linear complementarity problems, J. Ind. Manag. Optim. 21 (1) (2025) 731–750.
[41] M. Nagy, T. Illés, J. Povh, A. Varga, J. Žerovnik, Sufficient matrices: properties, generating and testing, J. Optim. Theory Appl. 202 (2024) 204–236.
[42] M. Nagy, A collection of small size sufficient matrices, 2019. https://sites.google.com/view/menagy/research/sufficient-matrices.
[43] P.T. Harker, J.S. Pang, A damped-Newton method for the linear complementarity problem, in: Allgower, G., Georg, K. (eds.) Computational Solution of Nonlinear Systems of Equations, Lectures in Applied Mathematics, vol. 26, 1990, pp. 265–284.
[44] L.T. Watson, Solving the nonlinear complementarity problem by a homotopy method, SIAM J. Control Optim. 17 (1) (1979) 36–46.
[45] G.Q. Wang, Y.Q. Bai, A class of polynomial interior point algorithms for the Cartesian P-matrix linear complementarity problem over symmetric cones, J. Optim. Theory Appl. 152 (3) (2012) 739–772.