

An $O(\sqrt{n}L)$ iteration primal-dual path-following method, based on wide neighborhoods and large updates, for monotone LCP

Wenbao Ai ^{*} Shuzhong Zhang [†]

February 18, 2005

Abstract

In this paper we propose a new class of primal-dual path-following interior point algorithms for solving monotone linear complementarity problems. At each iteration, the method would select a target on the central path with a large update from the current iterate, and then the Newton method is used to get the search directions, followed by adaptively choosing the step sizes, which are e.g. the largest possible steps before leaving a neighborhood that is as wide as the \mathcal{N}_{∞}^- neighborhood. The only deviation from the classical approach is that we treat the classical Newton direction as the sum of two other directions, corresponding to respectively the negative part and the positive part of the right-hand-side. We show that if these two directions are equipped with different and appropriate step sizes then the method enjoys the low iteration bound of $O(\sqrt{n} \log L)$, where n is the dimension of the problem and $L = \frac{(x^0)^T s^0}{\varepsilon}$ with ε the required precision and (x^0, s^0) the initial interior solution. For a predictor-corrector variant of the method, we further prove that, besides the predictor steps, each corrector step also reduces the duality gap by a rate of $1 - 1/O(\sqrt{n})$. Additionally, if the problem has a strict complementarity solution then the predictor steps converge Q-quadratically.

Keywords: Monotone LCP, primal-dual interior point method, wide neighborhood.

Mathematics Subject Classification: 90C33, 90C51, 90C05.

^{*}School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China.
Email: wenbaoai@vip.sina.com

[†]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong. Email: zhang@se.cuhk.edu.hk. Research supported by Hong Kong RGC Earmarked Grant CUHK4174/03E.

1 Introduction

In this paper we consider the following monotone linear complementary problem (LCP):

$$(LCP) \quad \begin{cases} s = Mx + q \\ x \geq 0, s \geq 0, x^T s = 0, \end{cases}$$

where $q \in \mathfrak{R}^n$ and $M \in \mathfrak{R}^{n \times n}$ is a monotone matrix, i.e. $M + M^T$ is positive semidefinite, or equivalently, $x^T Mx \geq 0$ for any $x \in \mathfrak{R}^n$.

A particular choice of M is a block skew symmetric matrix, namely $M = \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix}$. In that case, the corresponding monotone LCP problem is nothing but a linear programming problem.

The primal-dual interior point method for linear programming was first introduced by Kojima, Mizuno and Yoshise [5] and Megiddo [7], which essentially aims at solving the following parameterized problem by Newton's method, for shrinking values of the parameter $\mu > 0$,

$$(LCP)_\mu \quad \begin{cases} s = Mx + q \\ x_i > 0, s_i > 0, x_i s_i = \mu, i = 1, \dots, n. \end{cases}$$

The exact solution of the above problem is known as the analytic central path, with the varying path parameter $\mu > 0$. At each iteration, the method would choose a target on the central path and apply the Newton method to move closer to the target, while confining the iterate to stay within a certain neighborhood of the analytic central path. This method was found to be not only elegant in its simplicity and symmetricity, but also extremely efficient in practical implementations. There has been, however, an inconsistency between theory and practice: fast algorithms in practice may actually render worse complexity bounds. In their first paper [5], Kojima, Mizuno and Yoshise proposed that the iterates reside in a *wide* neighborhood of the central path, known as the \mathcal{N}_∞^- -neighborhood (details of the notion will be discussed later), and the targets on the central path are shifted towards the origin by a large update (percentage reduction) at each iteration. The worst case iteration bound was proved to be $O(nL)$, where n is the larger dimension of a standard linear programming problem, and L is its input-length. Then, in a subsequent paper, [6], the same authors proposed a variant of the method, where the iterates are restricted to a much smaller neighborhood, known as the \mathcal{N}_2 -neighborhood, and at each step the target is shifted with a small update. The algorithm became too conservative to be efficient in practice. However, the worst case iteration bound of the variant was improved to $O(\sqrt{n}L)$. In fact, many early primal-dual interior point methods either use a narrow neighborhood, or take small step sizes; see e.g. the primal-dual method by Monteiro and Adler [9, 10]. The first practically efficient $O(\sqrt{n}L)$ primal-dual interior point algorithm was the celebrated predictor-corrector algorithm of Mizuno, Todd and Ye [8]. In the predictor step of the algorithm, an adaptive step size is taken, ensuring its practical efficiency. The iteration bound is still

retained to be $O(\sqrt{n}L)$ since the \mathcal{N}_2 small neighborhoods are used to control the centrality of the iterates. Gonzaga [2] proposed to compute and combine the predictor and corrector steps based on the information of the same iterate, thus reducing the effort required by the Cholesky factorization. Along a related but different line, Ye, Güler, Tapia and Zhang [18] proved that the predictor step in the predictor-corrector scheme reduces the duality gap with a quadratic convergence rate. This result was extended by Ye and Anstreicher in [17] to the monotone LCP problem, assuming a strict complementary solution exists. We refer to Wright [14] for an excellent exposition on the primal-dual interior point method for linear programming and LCP problems.

The issue of the neighborhood size in the method has generated some research interests on its own. It is believed that in its original form, the primal-dual interior point algorithm of Kojima, Mizuno and Yoshise [5] may indeed not enjoy the low iteration bound of $O(\sqrt{n}L)$. However, it is possible to modify the algorithm to gain both the theoretical and the practical advantages. A first such attempt was made by Xu [15], who proposed an $O(\sqrt{n}L)$ method, in which the small neighborhood was only used as a safeguard, and the iterates are allowed to go far beyond. However, the new neighborhood, though much larger than the small neighborhood, does not necessarily contain the wide neighborhood: the neighborhood is still much narrower than the \mathcal{N}_∞^- wide neighborhood. Hung and Ye [4] proposed to use higher-order corrections on the Newton method, and showed that the iteration bound of their high order primal-dual interior point method with the \mathcal{N}_∞^- wide neighborhood can be reduced to $O(n^{\frac{n+1}{2n}}L)$. Sturm and Zhang [13] proposed to follow a central region, instead of the central path. The central region is defined to be precisely an \mathcal{N}_∞^- wide neighborhood of the central path. Then, they introduced a (narrow) neighborhood of the central region (the whole area is thus wider than the central region which is a wide neighborhood itself) in which all the iterates reside. By choosing the direction towards a target in the central region properly, Sturm and Zhang [13] managed to show that their algorithm has an iteration bound of $O(\sqrt{n}L)$. Since the iterates are required to take adaptive (thus long) steps, maximum possible within the wide neighborhood, the algorithm is highly efficient in practice. Later, Sturm generalized the method to solve Semidefinite Programming (SDP) problems, and the method has become one of the pillars for his famous SDP solver SeDuMi [12]. Ai [1] proposed a new wide neighborhood interior point algorithm with $O(\sqrt{n}L)$ iteration bound. The current paper is inspired by [1], in the definition of the new wide neighborhood; however, they differ greatly in both the scope and the results to be achieved. Recently, Peng, Terlaky and Zhao [11] introduced a variant of the corrector-predictor approach, based on a self-regular function to define the neighborhood, which is wide. They showed that their algorithm enjoys the iteration bound of $O(\sqrt{n} \log nL)$ for linear programming problems.

As far as we know, in the context of path-following approach, none had succeeded in retaining the $O(\sqrt{n}L)$ complexity while allowing a large update (meaning a reduction by a universal percentage, independent of the problem parameters) of the target along the central path at all iterations, even if one is allowed to stay within narrow neighborhoods. Indeed, deriving a path-following interior point

method with the $O(\sqrt{n}L)$ iteration bound while working with large-updates and wide neighborhood at each iteration is one of the objectives to be achieved in this paper. In other words, the current paper aims at modifying the original primal-dual interior point method with minimum changes, maintaining large updates and working with wide neighborhood at all iterations, to gain the $O(\sqrt{n}L)$ iteration bound and retain practical efficiency. We organize the paper as follows. Our new methodology will be introduced in Section 2, where the main underlying ideas will be explained. In Section 3, we present the technical lemmas that will be important for the subsequent analysis, and in Section 4 we discuss some easily implementable variants of the general method and show the low computational complexity status. In a similar spirit, we discuss another variant in Section 5, based on the predictor-corrector methodology. Novel properties of the algorithm will be discussed, including its progressive property even for the corrector steps. The low complexity bound will be proven for this variant, and the superlinear convergence property will be shown, provided that a strictly complementary solution exists.

The notation used in this paper is fairly standard: the i -th component of vector $x \in \Re^n$ is denoted by x_i ; e is the all one vector with an appropriate dimension; if $d \in \Re^n$ then we denote D to be an $n \times n$ diagonal matrix with d as the diagonal components; for $x, y \in \Re^n$, xy is the component product in \Re^n , and so is true for other operations, e.g. $1/(xy)$ and $(xy)^{-0.5}$; $x \geq (>) y$ means that the inequality holds component-wisely; for any $a \in \Re$, a^+ denotes its nonnegative part, i.e. $a^+ := \max\{a, 0\}$, and a^- denotes its nonpositive part, i.e. $a^- := \min\{a, 0\}$; the same notation is used for vector $x \in \Re^n$, namely x^+ is the nonnegative part of x and x^- is the nonpositive part of x ; the L_p -norm of $x \in \Re^n$ is denoted by $\|x\|_p$, and in particular we write $\|x\|$ for $\|x\|_2$ — the Euclidean norm.

2 Separating large and small components: a new paradigm

Let us denote

$$\mathcal{F}_{++} := \{(x, s) \mid s = Mx + q, x > 0, s > 0\},$$

which is assumed to be nonempty throughout this paper.

The central path for (LCP) is defined as

$$\mathcal{C} := \{(x, s) \in \mathcal{F}_{++} \mid xs = \mu e\}$$

and its small neighborhood is defined as

$$\mathcal{N}_2(\beta) := \{(x, s) \in \mathcal{F}_{++} \mid \|xs - \mu e\| \leq \beta\mu\}$$

where $\beta \in (0, 1)$ is a given constant and $\mu := x^T s / n$. The so-called wide neighborhood is defined as follows:

$$\mathcal{N}_\infty^-(1 - \tau_2) := \{(x, y, s) \in \mathcal{F}_{++} \mid xs \geq \tau_2 \mu e\}$$

where $0 < \tau_2 < 1$.

Before proceeding, let us recall the classical primal-dual interior point method with wide neighborhood and large update of the targets. Let $0 \leq \tau \leq 1$ and $0 < \tau_2 < 1$ be two given parameters. Suppose that the current iterate is $(x, s) \in \mathcal{N}_\infty^-(1 - \tau_2)$. The search direction $(\Delta x, \Delta s)$ is the solution of the following system of linear equations:

$$\begin{cases} \Delta s = M \Delta x, \\ s \Delta x + x \Delta s = \tau \mu e - xs, \end{cases} \quad (1)$$

where

$$\mu = x^T s / n.$$

Then the next iterate will be given by

$$(x + \bar{\alpha} \Delta x, s + \bar{\alpha} \Delta s)$$

where $\bar{\alpha}$ is the solution of the subproblem

$$\begin{aligned} & \text{minimize} && (x + \alpha \Delta x)^T (s + \alpha \Delta s) \\ & \text{subject to} && (x + \alpha \Delta x, s + \alpha \Delta s) \in \mathcal{N}_\infty^-(1 - \tau_2) \\ & && \alpha \in [0, 1]. \end{aligned}$$

Naturally, all the iterates are contained in the wide neighborhood $\mathcal{N}_\infty^-(1 - \tau_2)$.

An important ingredient of this paper is to introduce a new neighborhood for the central path, defined as

$$\mathcal{N}(\tau_1, \tau_2, \eta) := \mathcal{N}_\infty^-(1 - \tau_2) \cap \left\{ (x, s) \in \mathcal{F}_{++} \mid \|(\tau_1 \mu e - xs)^+\| \leq \eta(\tau_1 - \tau_2)\mu \right\}, \quad (2)$$

where $\eta \geq 1$ and τ_1 satisfying $0 < \tau_2 < \tau_1 < 1$, are two more parameters.

The above defined neighborhood is itself a wide neighborhood, since one can easily verify that

$$\mathcal{N}_\infty^-(1 - \tau_1) \subseteq \mathcal{N}(\tau_1, \tau_2, \eta) \subseteq \mathcal{N}_\infty^-(1 - \tau_2).$$

Moreover, if $\tau_1 - \eta(\tau_1 - \tau_2)/\sqrt{n-1} > \tau_2$, then

$$\mathcal{N}_\infty^-(1 - \tau_1 + \eta(\tau_1 - \tau_2)/\sqrt{n-1}) \subseteq \mathcal{N}(\tau_1, \tau_2, \eta),$$

and if $\tau_1 - \eta(\tau_1 - \tau_2)/\sqrt{n-1} \leq \tau_2$, then

$$\mathcal{N}(\tau_1, \tau_2, \eta) = \mathcal{N}_\infty^-(1 - \tau_2).$$

Specially, if we choose $\eta = 1$, the neighborhood can be expressed more simply as follows.

$$\mathcal{N}(\tau_1, \tau_2, 1) = \{(x, s) \in \mathcal{F}_{++} \mid \|(\tau_1 \mu e - xs)^+\| \leq (\tau_1 - \tau_2)\mu\} \supseteq \mathcal{N}_\infty^-(1 - \tau_1). \quad (3)$$

In this paper, the newly introduced neighborhood $\mathcal{N}(\tau_1, \tau_2, \eta)$ will play an important role. The reason for working with $\mathcal{N}(\tau_1, \tau_2, \eta)$ is that the measure for the components in xs that are ‘dangerously’ close to zero is captured by the quantity $\|(\tau_1\mu e - xs)^+\|$, and we are less concerned about the ‘large’ components of xs present in $(\tau_1\mu e - xs)^-$. In fact, we will see later that this separation is crucial. The part $(\tau_1\mu e - xs)^+$ is used to control the centrality, and the other part $(\tau_1\mu e - xs)^-$ is important for the progress towards optimality.

Suppose that our current iterate is (x, s) . Let $0 \leq \tau \leq 1$. Another key ingredient of our method is to decompose the Newton step, from xs to the target on the central path $\tau\mu e$ (large update), into two equations:

$$\begin{cases} \Delta s_- = M \Delta x_- \\ s \Delta x_- + x \Delta s_- = (\tau\mu e - xs)^- \end{cases} \quad (4)$$

and

$$\begin{cases} \Delta s_+ = M \Delta x_+ \\ s \Delta x_+ + x \Delta s_+ = (\tau\mu e - xs)^+ \end{cases} \quad (5)$$

Since $\tau\mu e - xs = (\tau\mu e - xs)^- + (\tau\mu e - xs)^+$, the usual Newton direction is simply $(\Delta x_-, \Delta s_-) + (\Delta x_+, \Delta s_+)$. In this paper, however, we propose to treat these two directions separately. Essentially those are what we need to modify the original large update and wide neighborhood path-following method. The payoff for the changes will become clear later. At this stage, we only remark that the extra computational effort is very marginal, compared to the computation of a single Newton direction. For reference, we shall call $(\Delta x_-, \Delta s_-)$ and $(\Delta x_+, \Delta s_+)$ the *Newton constituent directions*.

Let $\alpha := (\alpha_1, \alpha_2) \in \mathfrak{R}_+^2$ be the step sizes taken along $(\Delta x_-, \Delta s_-)$ and $(\Delta x_+, \Delta s_+)$ respectively. The step is

$$(x(\alpha), s(\alpha)) := (x, s) + \alpha_1(\Delta x_-, \Delta s_-) + \alpha_2(\Delta x_+, \Delta s_+).$$

The best α can be obtained by solving the following two-dimensional optimization problem:

$$\begin{aligned} & \text{minimize} && x(\alpha)^T s(\alpha) \\ & \text{subject to} && (x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta) \\ & && 0 \leq \alpha_1 \leq 1, 0 \leq \alpha_2 \leq 1. \end{aligned} \quad (6)$$

Remark that due to the monotonicity, the above objective function $x(\alpha)^T s(\alpha)$ is convex in α .

Below we describe a generic framework for our wide-neighborhood and large-update primal-dual path-following method.

Algorithm 2.1.

Input parameters: required precision $\varepsilon > 0$, neighborhood parameters $\eta^k \geq 1$, $0 < \tau_2^k < \tau_1^k < 1$, and target parameters $0 \leq \tau^k \leq 1$, $k = 0, 1, \dots$, and the initial solution $(x^0, s^0) \in \mathcal{N}(\tau_1^0, \tau_2^0, \eta^0)$.

Output: a sequence of iterates $\{(x^k, s^k) \mid k = 0, 1, 2, \dots\}$.

Step 0 Set $k = 0$.

Step 1 If $(x^k)^T s^k \leq \varepsilon$ then stop.

Step 2 Solve $(\Delta x_-^k, \Delta s_-^k)$ and $(\Delta x_+^k, \Delta s_+^k)$ based on (4) and (5).

Find step size vector $\alpha^k \in \mathfrak{R}_{++}^2$, such that $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1^k, \tau_2^k, \eta^k)$.

Step 3 Set $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$.

Let $k := k + 1$ and go to **Step 1**.

We remark here that the optimal step sizes according to (6) may be used in **Step 2**, and the parameters η^k , τ_1^k , τ_2^k and τ^k may be set to constants. It is however convenient to allow for the flexibilities at this stage.

The main result of this paper is to prove that the above generic method can be specified into easy implementable variants with given parameters, in such a way that the iteration bound will be $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$. Moreover, the method can also be implemented in the predictor-corrector style. In that case, in addition to the above iteration bound one also obtains a quadratic convergence rate for the predictor steps, provided that a strict complementary solution exists. These specific implementations will be discussed in Sections 4 and 5 respectively. To facilitate the analysis, we need to study the properties of the two separated Newton constituent directions. This will be the topic of the next section.

3 Technical lemmas

In this section we choose to set $\tau = \tau_1$.

First, it is useful for our subsequent analysis to note the following triangle inequalities for the ‘minus’ and ‘plus’ operations on the vectors.

Proposition 3.1. *For any $u, v \in \mathfrak{R}^n$ and $p \geq 1$, we have*

$$\|(u + v)^+\|_p \leq \|u^+\|_p + \|v^+\|_p$$

and

$$\|(u+v)^-\|_p \leq \|u^-\|_p + \|v^-\|_p.$$

Proof. As $0 \leq (u+v)^+ \leq u^+ + v^+$ we conclude that

$$\|(u+v)^+\|_p \leq \|u^+ + v^+\|_p \leq \|u^+\|_p + \|v^+\|_p.$$

Similarly, we have

$$\|(u+v)^-\|_p \leq \|u^-\|_p + \|v^-\|_p.$$

□

The next proposition is concerned with the feasibility of the iterates along given Newton directions. It would be undesirable if the iterates would leave the feasible region and then return to it again.

Proposition 3.2. *Suppose that $(x, s) \in \mathcal{F}_{++}$ and $z + xs \geq 0$. Let $(\Delta x, \Delta s)$ be the solution of $\Delta s = M\Delta x$, $s\Delta x + x\Delta s = z$. If $(x + t_0\Delta x)(s + t_0\Delta s) > 0$ for some $0 < t_0 \leq 1$, then $x + t\Delta x > 0$ and $s + t\Delta s > 0$ for all $0 \leq t \leq t_0$.*

Proof. Let $(\bar{x}, \bar{s}) := (x + t_0\Delta x, s + t_0\Delta s)$.

We have

$$\begin{aligned} & (x + \delta t_0\Delta x)(s + \delta t_0\Delta s) \\ &= xs + \delta t_0(s\Delta x + x\Delta s) + \delta^2 t_0^2 \Delta x \Delta s \\ &= xs + \delta t_0 z + \delta^2 (\bar{x}\bar{s} - t_0 z - xs) \\ &= (1 - \delta)xs + \delta(1 - \delta)(t_0 z + xs) + \delta^2 \bar{x}\bar{s} > 0 \end{aligned} \tag{7}$$

for all $0 \leq \delta \leq 1$.

If there are $0 < t_1 \leq t_0$ and $1 \leq i \leq n$ with either $(x + t_1\Delta x)_i < 0$ or $(s + t_1\Delta s)_i < 0$, then, since $(x, s) > 0$, by continuity there must exist $0 < t_2 < t_1 \leq t_0$, such that $(x + t_2\Delta x)_i (s + t_2\Delta s)_i = 0$. Letting $\delta = t_2/t_0$, we have $(x + \delta t_0\Delta x)_i (s + \delta t_0\Delta s)_i = 0$, which would contradict (7). The proposition is thus proven. □

The term $z + xs$ is sometimes called the *target* to be tracked, and it is naturally nonnegative for most interior point methods. In particular, for Algorithm 2.1, this property boils down to verifying $xs + \alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+ \geq 0$.

Let us denote

$$h(\alpha) := xs + \alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+ \tag{8}$$

and

$$I^+ := \{i \mid \tau_1 \mu - x_i s_i > 0\}. \quad (9)$$

Since $(x, s) \in \mathcal{F}_{++}$ we have

$$h_i(\alpha) = \begin{cases} x_i s_i + \alpha_2(\tau_1 \mu - x_i s_i) = (1 - \alpha_2)x_i s_i + \alpha_2 \tau_1 \mu > 0, & i \in I^+, \\ x_i s_i + \alpha_1(\tau_1 \mu - x_i s_i) \geq x_i s_i + \tau_1 \mu - x_i s_i = \tau_1 \mu > 0, & i \notin I^+, \end{cases} \quad (10)$$

for all $\alpha \in [0, 1]^2$. Proposition 3.2 thus asserts that $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$ if and only if $x(\alpha)s(\alpha) \geq \tau_2 \mu(\alpha)e$ and $\|(\tau_1 \mu(\alpha)e - x(\alpha)s(\alpha))^+\| \leq \eta(\tau_1 - \tau_2)\mu(\alpha)$, where

$$\mu(\alpha) := x(\alpha)^T s(\alpha)/n. \quad (11)$$

We further have

$$\begin{aligned} \mu(\alpha) &:= (x + \Delta x(\alpha))^T (s + \Delta s(\alpha))/n \\ &= (x^T s + s^T \Delta x(\alpha) + x^T \Delta s(\alpha) + \Delta x(\alpha)^T \Delta s(\alpha))/n \\ &= \mu + \alpha_1 e^T (\tau_1 \mu e - xs)^- /n + \alpha_2 e^T (\tau_1 \mu e - xs)^+ /n + \Delta x(\alpha)^T \Delta s(\alpha)/n, \end{aligned} \quad (12)$$

where $\Delta x(\alpha) = \alpha_1 \Delta x_- + \alpha_2 \Delta x_+$ and $\Delta s(\alpha) = \alpha_1 \Delta s_- + \alpha_2 \Delta s_+$.

Lemma 3.3. *It holds that*

$$\mu(\alpha) \geq (1 - \alpha_1)\mu.$$

Proof. By the monotonicity we have $\Delta x(\alpha)^T \Delta s(\alpha) \geq 0$. Therefore from (12) we have

$$\begin{aligned} \mu(\alpha) &\geq \mu + \alpha_1 e^T (\tau_1 \mu e - xs)^- /n \\ &\geq \mu + \alpha_1 e^T (-xs)/n \\ &= (1 - \alpha_1)\mu. \end{aligned}$$

□

We note the following simple but useful relationships:

$$\begin{cases} e^T (\tau_1 \mu e - xs)^- = -(1 - \tau_1)x^T s - e^T (\tau_1 \mu e - xs)^+, \\ \| (xs)^{-0.5} (\tau_1 \mu e - xs)^- \| = \| (\sqrt{xs} - \tau_1 \mu e / \sqrt{xs})^+ \| \leq \| \sqrt{xs} \| = x^T s, \\ e^T (\tau_1 \mu e - xs)^+ \leq \sqrt{n} \| (\tau_1 \mu e - xs)^+ \|. \end{cases} \quad (13)$$

For convenience we set

$$\beta := \frac{\tau_1 - \tau_2}{\tau_1}. \quad (14)$$

Obviously we have $\beta \in (0, 1)$, $\tau_1 - \tau_2 = \beta\tau_1$ and $\tau_2 = (1 - \beta)\tau_1$. Let

$$\hat{\eta} = \max \left\{ \frac{\|(\tau_1\mu e - xs)^+\|}{\beta\tau_1\mu}, 1 \right\} \quad (15)$$

It follows that $\hat{\eta} \leq \eta$ if $(x, s) \in \mathcal{N}(\tau_1, \tau_2, \eta)$.

Lemma 3.4. *If $\mu(\alpha) \leq \mu$, then it holds that*

$$\|(\tau_1\mu(\alpha)e - h(\alpha))^+\| \leq (1 - \alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha).$$

Proof. As $\mu(\alpha) \leq \mu$ it follows from (10) that

$$\tau_1\mu(\alpha) - h_i(\alpha) \leq \begin{cases} \tau_1\mu(\alpha) - \frac{\mu(\alpha)}{\mu}h_i(\alpha) = \frac{\mu(\alpha)}{\mu}(1 - \alpha_2)(\tau_1\mu - x_i s_i), & \text{if } i \in I^+, \\ 0, & \text{else,} \end{cases}$$

which implies that

$$\|(\tau_1\mu(\alpha)e - h(\alpha))^+\| \leq \frac{\mu(\alpha)}{\mu}(1 - \alpha_2)\|(\tau_1\mu - xs)^+\| \leq (1 - \alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha).$$

□

Lemma 3.5. *Let $u, v \in \mathfrak{R}^n$ be such that $u^T v \geq 0$, and let $r = u + v$. Then, we have*

$$\|(uv)^-\|_1 \leq \|(uv)^+\|_1 \leq \frac{1}{4}\|r\|^2.$$

Proof. Let the index set J be

$$J := \{i \mid u_i v_i > 0\}.$$

As $u^T v \geq 0$ we have

$$\|(uv)^-\|_1 \leq \|(uv)^+\|_1 = \sum_{i \in J} u_i v_i \leq \frac{1}{4} \sum_{i \in J} (u_i + v_i)^2 = \frac{1}{4} \sum_{i \in J} (r_i)^2 \leq \frac{1}{4} \|r\|^2.$$

□

Lemma 3.6. *Suppose $\beta \leq \frac{1}{2}$ and $\alpha_1 = t\alpha_2\hat{\eta}\sqrt{\frac{\beta\tau_1}{n}}$ for some $t \geq 0$. Then we have*

$$\|(\Delta x(\alpha)\Delta s(\alpha))-\|_1 \leq \|(\Delta x(\alpha)\Delta s(\alpha))^+\|_1 \leq (t^2 + 1)\alpha_2^2\hat{\eta}^2\beta\tau_1\mu/4.$$

Proof. We have

$$s\Delta x(\alpha) + x\Delta s(\alpha) = \alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+.$$

Multiply both sides of the above equality by $(xs)^{-0.5}$. Denote $u := x^{-0.5}s^{0.5}\Delta x(\alpha)$, $v := x^{0.5}s^{-0.5}\Delta x(\alpha)$ and $r := (xs)^{-0.5}(\alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+)$. So we have $u + v = r$. Notice that $u^T v = \Delta x(\alpha)^T \Delta s(\alpha) \geq 0$. Therefore, by Lemma 3.5 we have

$$\begin{aligned}
& \|(\Delta x(\alpha)\Delta s(\alpha))^- \|_1 \\
& \leq \|(\Delta x(\alpha)\Delta s(\alpha))^+ \|_1 \\
& \leq \frac{1}{4} \|(xs)^{-0.5}\alpha_1(\tau_1\mu e - xs)^- + (xs)^{-0.5}\alpha_2(\tau_1\mu e - xs)^+\|^2 \\
& = \frac{1}{4} (\alpha_1^2 \|(\sqrt{xs} - \tau_1\mu e/\sqrt{xs})^+\|^2 + \alpha_2^2 \|(xs)^{-0.5}(\tau_1\mu e - xs)^+\|^2) \\
& \leq \frac{1}{4} (\alpha_1^2 \|\sqrt{xs}\|^2 + \alpha_2^2 \|(\tau_1\mu e - xs)^+\|^2 / (\tau_2\mu)) \\
& \leq \frac{1}{4} (t^2\alpha_2^2\hat{\eta}^2\beta\tau_1\mu + \alpha_2^2\hat{\eta}^2\beta\tau_1\mu) \\
& = (t^2 + 1)\alpha_2^2\hat{\eta}^2\beta\tau_1\mu/4.
\end{aligned} \tag{16}$$

□

Lemma 3.7. *Suppose that $\tau_1 \leq 1/4$ and $\beta \leq 1/2$. If $\alpha_1 = \alpha_2\hat{\eta}\sqrt{\frac{\beta\tau_1}{n}}$ and $\alpha_2 \leq 1/\hat{\eta}^2$, then we have*

$$\mu(\alpha) \leq (1 - \frac{\hat{\eta}\sqrt{\beta\tau_1}}{10\sqrt{n}}\alpha_2)\mu.$$

Proof. By (12), (13) and Lemma 3.6 we have

$$\begin{aligned}
\mu(\alpha) & \leq \mu + \alpha_1 e^T(\tau_1\mu e - xs)^-/n + \alpha_2 e^T(\tau_1\mu e - xs)^+/n + \|(\Delta x(\alpha)\Delta s(\alpha))^+\|_1/n \\
& \leq \mu - \alpha_1(1 - \tau_1)\mu + \alpha_2 \|e\| \|(\tau_1\mu e - xs)^+\|/n + \|(\Delta x(\alpha)\Delta s(\alpha))^+\|_1/n \\
& \leq \mu - \alpha_1(1 - \tau_1)\mu + \alpha_2\hat{\eta}\beta\tau_1\mu/\sqrt{n} + \alpha_2^2\hat{\eta}^2\beta\tau_1\mu/2n \\
& \leq \mu - 3\alpha_1\mu/4 + 3\alpha_2\hat{\eta}\beta\tau_1\mu/2\sqrt{n} \\
& = \mu - \alpha_2 \frac{3\hat{\eta}\sqrt{\beta\tau_1}(1 - 2\sqrt{\beta\tau_1})\mu}{4\sqrt{n}} \\
& \leq \mu - \alpha_2 \frac{3\hat{\eta}\sqrt{\beta\tau_1}(1 - 1/\sqrt{2})\mu}{4\sqrt{n}} \\
& \leq (1 - \alpha_2 \frac{\hat{\eta}\sqrt{\beta\tau_1}}{10\sqrt{n}})\mu.
\end{aligned}$$

□

Lemma 3.8. *Suppose that $(x, s) \in \mathcal{N}(\tau_1, \tau_2, \eta)$, $\tau_1 \leq 1/4$ and $\beta \leq 1/2$. If $\alpha_1 = \alpha_2\hat{\eta}\sqrt{\frac{\beta\tau_1}{n}}$ and $\alpha_2 \leq 1/\hat{\eta}^2$, then $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$.*

Proof. By Lemma 3.7 it follows that $\mu(\alpha) \leq \mu$. Further, it follows from (10) and Lemma 3.6 that

$$\begin{aligned}
x(\alpha)s(\alpha) &= h(\alpha) + \Delta x(\alpha)\Delta s(\alpha) \\
&\geq (\tau_2\mu + \alpha_2(\tau_1 - \tau_2)\mu)e - \|(\Delta x(\alpha)\Delta s(\alpha))^- \|e \\
&\geq (\tau_2\mu + \alpha_2\beta\tau_1\mu)e - (\alpha_2^2\hat{\eta}^2\beta\tau_1\mu/2)e \\
&\geq (\tau_2\mu + \alpha_2\beta\tau_1\mu - \alpha_2\beta\tau_1\mu/2)e \\
&\geq \tau_2\mu e \\
&\geq \tau_2\mu(\alpha)e,
\end{aligned}$$

which also implies $(x(\alpha), s(\alpha)) > 0$ according to Proposition 3.2. Therefore, $(x(\alpha), s(\alpha)) \in \mathcal{N}_\infty^-(1-\tau_2)$.

At the same time, by Lemma 3.3 we have

$$\mu(\alpha) \geq (1 - \alpha_1)\mu \geq (1 - \frac{\hat{\eta}\sqrt{\beta\tau_1}}{\sqrt{n}})\mu \geq (1 - \sqrt{\beta\tau_1})\mu \geq \mu/2.$$

Using Lemmas 3.4 and 3.6 we obtain

$$\begin{aligned}
&\|(\tau_1\mu(\alpha)e - x(\alpha)s(\alpha))^+\| \\
&= \|(\tau_1\mu(\alpha)e - h(\alpha) - \Delta x(\alpha)\Delta s(\alpha))^+\| \\
&\leq \|(\tau_1\mu(\alpha)e - h(\alpha))^+ + (-\Delta x(\alpha)\Delta s(\alpha))^+\| \\
&\leq \|(\tau_1\mu(\alpha)e - h(\alpha))^+\| + \|(\Delta x(\alpha)\Delta s(\alpha))^- \| \\
&\leq (1 - \alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha) + \alpha_2^2\hat{\eta}^2\beta\tau_1\mu/2 \\
&\leq (1 - \alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha) + \alpha_2\hat{\eta}\beta\tau_1\mu(\alpha) \\
&= \hat{\eta}\beta\tau_1\mu(\alpha) \\
&\leq \eta\beta\tau_1\mu(\alpha),
\end{aligned}$$

proving that $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$. □

4 The iteration bound and an implementation

Now we are in a position to present our complexity results.

First, let us consider the generic Algorithm 2.1.

Theorem 4.1. *Suppose that $\eta \geq 1$, $\tau = \tau_1 \leq 1/4$ and $\beta \leq 1/2$ are fixed for all iterations. Furthermore, suppose that the plane-search procedure (6) is applied at each iteration of Algorithm 2.1. Then, Algorithm 2.1 terminates in $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iterations.*

Proof. By Lemma 3.8, at each iteration, if we let $\hat{\alpha} = (\sqrt{\beta\tau_1/n}/\hat{\eta}, 1/\hat{\eta}^2)$ then we have

$$(x(\hat{\alpha}), s(\hat{\alpha})) \in \mathcal{N}(\tau_1, \tau_2, \eta).$$

Furthermore, according to Lemma 3.7 we also have

$$\mu(\hat{\alpha}) \leq \left(1 - \frac{\sqrt{\beta\tau_1}}{10\hat{\eta}\sqrt{n}}\right)\mu \leq \left(1 - \frac{\sqrt{\beta\tau_1}}{10\eta\sqrt{n}}\right)\mu.$$

Therefore, the exact plane search would lead to at least the same amount of reduction in $\mu(\alpha)$, and hence the theorem is proven. \square

The plane-search subproblem being an optimization problem with a convex objective and only two variables can be solved relatively easily. However, it is also possible to reduce the number of search variables in the subproblem to only one without sacrificing the practical efficiency too much.

The main observation here is that under some mild conditions, the objective function in the subproblem (6) is monotone with respect to α_1 for any fixed $\alpha_2 \in [0, 1]$.

More precisely, we have the following result.

Theorem 4.2. *Suppose $(x, s) \in \mathcal{N}_\infty^-(1 - \tau_2)$, $\tau = \tau_1 \leq 1/4$ and $\tau_1 \leq 2\tau_2$ (i.e. $\beta \leq 1/2$). For any fixed $\alpha_2 \in [0, 1]$, $x(\alpha)^T s(\alpha)$ is a monotonically decreasing function in α_1 for $\alpha_1 \in [0, 1]$.*

Proof. We have

$$\begin{aligned} x(\alpha)^T s(\alpha) &= (x + \alpha_1 \Delta x_- + \alpha_2 \Delta x_+)^T (s + \alpha_1 \Delta s_- + \alpha_2 \Delta s_+) \\ &= x^T s + \alpha_1 (x^T \Delta s_- + s^T \Delta x_-) + \alpha_2 (x^T \Delta s_+ + s^T \Delta x_+) \\ &\quad + \alpha_1^2 \Delta x_-^T \Delta s_- + \alpha_1 \alpha_2 (\Delta x_-^T \Delta s_+ + \Delta x_+^T \Delta s_-) + \alpha_2^2 \Delta x_+^T \Delta s_+. \end{aligned}$$

Therefore, since $0 \leq \alpha_1 \leq 1$ and $0 \leq \alpha_2 \leq 1$,

$$\begin{aligned} \frac{\partial(x(\alpha)^T s(\alpha))}{\partial\alpha_1} &= e^T (\tau_1 \mu e - xs)^- + 2\alpha_1 \Delta x_-^T \Delta s_- + \alpha_2 (\Delta x_-^T \Delta s_+ + \Delta x_+^T \Delta s_-) \\ &\leq e^T (\tau_1 \mu e - xs)^- + 2\Delta x_-^T \Delta s_- + |(D^{-1} \Delta x_-)^T (D \Delta s_+) + (D \Delta s_-)^T (D^{-1} \Delta x_+)| \\ &\leq e^T (\tau_1 \mu e - xs)^- + 2\Delta x_-^T \Delta s_- + \|(D^{-1} \Delta x_-, D \Delta s_-)\| \|(D^{-1} \Delta x_+, D \Delta s_+)\| \end{aligned} \quad (17)$$

where $D = X^{1/2} S^{-1/2}$, and we also used the monotonicity of M (thus $\Delta x_-^T \Delta s_- \geq 0$) in the second step.

By Lemma 3.6 and Lemma 3.5, we have

$$\begin{aligned}
\Delta x_-^T \Delta s_- &\leq \|(\Delta x_- \Delta s_-)^+\|_1 \\
&= \|((D^{-1} \Delta x_-)(D \Delta s_-))^+\|_1 \\
&\leq \|(D^{-1} \Delta x_-)(D \Delta s_-)\|_1 \\
&\leq \frac{1}{4} (\|D^{-1} \Delta x_-\|^2 + \|D \Delta s_-\|^2) \\
&\leq \frac{1}{4} \|D^{-1} \Delta x_- + D \Delta s_-\|^2 \\
&= \frac{1}{4} \|(\tau_1 \mu e - xs)^- / \sqrt{xs}\|^2 \\
&= \frac{1}{4} \|\sqrt{(xs - \tau_1 \mu e)^+} \sqrt{(xs - \tau_1 \mu e)^+ / xs}\|^2 \\
&\leq \frac{1}{4} \|\sqrt{(xs - \tau_1 \mu e)^+}\|^2 \\
&= e^T (xs - \tau_1 \mu e)^+ / 4,
\end{aligned} \tag{18}$$

where we used the monotonicity $(D^{-1} \Delta x_-)^T D \Delta s_- = \Delta x_-^T \Delta s_- \geq 0$ in the fifth step, and the fact that $0 \leq (xs - \tau_1 \mu e)^+ / xs \leq e$ in the eighth step. In fact, one concludes from the chain of inequalities in (18) that

$$\|(D^{-1} \Delta x_-, D \Delta s_-)\|^2 \leq e^T (xs - \tau_1 \mu e)^+. \tag{19}$$

Similarly,

$$\begin{aligned}
\|(D^{-1} \Delta x_+, D \Delta s_+)\|^2 &\leq \|D^{-1} \Delta x_+ + D \Delta s_+\|^2 \\
&= \|(\tau_1 \mu e - xs)^+ / \sqrt{xs}\|^2 \\
&\leq \|\sqrt{(\tau_1 \mu e - xs)^+}\|^2 = e^T (\tau_1 \mu e - xs)^+,
\end{aligned}$$

where the third step is due to the fact that since $\tau_1 \leq 2\tau_2$ and $(x, s) \in \mathcal{N}_\infty^-(1 - \tau_2)$ and so

$$(\tau_1 \mu e - xs)^+ \leq (2\tau_2 \mu e - xs)^+ = (2(\tau_2 \mu e - xs) + xs)^+ \leq xs.$$

Furthermore,

$$\begin{aligned}
e^T (\tau_1 \mu e - xs)^+ &\leq n(\tau_1 - \tau_2)\mu \\
&\leq n\mu/8 \\
&\leq (1 - \tau_1)n\mu/6 \\
&= e^T (xs - \tau_1 \mu e)/6 \\
&\leq e^T (xs - \tau_1 \mu e)^+ / 6.
\end{aligned}$$

Therefore we have

$$\|(D^{-1} \Delta x_+, D \Delta s_+)\|^2 \leq e^T (xs - \tau_1 \mu e)^+ / 6. \tag{20}$$

Substituting (18),(19) and (20) into (17) finally yields that

$$\frac{\partial(x(\alpha)^T s(\alpha))}{\partial\alpha_1} \leq -\left(\frac{1}{2} - \frac{1}{\sqrt{6}}\right) e^T (xs - \tau_1 \mu e)^+ < 0.$$

□

In view of Theorem 4.1 and Theorem 4.2, we may solve subproblem (6) approximately in the following way. First, set $\alpha_2 = 1/\hat{\eta}^2$. Second, find the greatest $\bar{\alpha}_1$ in $[0, 1]$ such that $(x(\bar{\alpha}_1), 1/\hat{\eta}^2), s(\bar{\alpha}_1, 1/\hat{\eta}^2)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$. One may for instance use bisection on α_1 for this purpose. Then, set $\alpha_1 = \bar{\alpha}_1$. Theorem 4.2 and Lemma 3.8 guarantee that $\bar{\alpha}_1 \geq \hat{\eta}^{-1} \sqrt{\beta\tau_1/n}$. It is clear that if the plane search procedure as described in Theorem 4.1 is replaced by this line search procedure then the $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iteration bound still holds. Particularly, if we let $\eta^k \equiv 1$, then $\hat{\eta} \equiv 1$ and so we can always choose $\alpha_2 \equiv 1$. Another benefit of $\eta^k \equiv 1$ is that the corresponding neighborhoods $\mathcal{N}(\tau_1^0, \tau_2^0, 1)$ are simply expressed by (21) and (3). A concrete practical implementation is recommended as follows. Its numerical performance will be discussed in Section 6.

Algorithm 4.3.

Input parameters: required precision $\varepsilon > 0$, target parameter and neighborhood parameters $0 < \tau = \tau_1 < 1$ and $\tau_2 = 0.5\tau_1$ (i.e. $\beta = 0.5$), and the initial solution $(x^0, s^0) \in \mathcal{N}(\tau_1^0, \tau_2^0, 1)$.

Step 0 Set $k = 0$.

Step 1 If $(x^k)^T s^k \leq \varepsilon$ then stop.

Step 2 Solve $(\Delta x_-^k, \Delta s_-^k)$ and $(\Delta x_+^k, \Delta s_+^k)$ based on (4) and (5).

Set $\alpha_2^k = 1$ and find the largest α_1^k on the closed interval $[\sqrt{\beta\tau_1/n}, 1]$, such that $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1, \tau_2, 1)$.

Step 3 Set $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$.

Let $k := k + 1$ and go to **Step 1**.

5 A predictor-corrector scheme

In this section, we shall slightly change the notation. For simplicity, we shall always choose $\eta = 1$, i.e. we consider the neighborhood $\mathcal{N}(\tau_1, \tau_2, 1)$. We introduce a new notation $\mathcal{N}(\tau_1; \beta)$ to indicate the set $\mathcal{N}(\tau_1, \tau_2, 1)$ (see (3)), i.e.,

$$\mathcal{N}(\tau_1; \beta) = \{(x, s) \in \mathcal{F}_{++} \mid \|(\tau_1 \mu e - xs)^+\| \leq \beta \tau_1 \mu\} \quad (21)$$

where $\beta = (\tau_1 - \tau_2)/\tau_1$, as given in (14).

Below we describe another variant of Algorithm 2.1, which is essentially a predictor-corrector type algorithm.

Algorithm 5.1.

Input parameters: required precision $\varepsilon > 0$, neighborhood parameters $0 < \tau_1 \leq 1/4$, $0 < \beta \leq 1/2$, and the initial solution $(x^0, s^0) \in \mathcal{N}(\tau_1; \beta/2)$.

Output: a sequence of iterates $\{(x^k, s^k) \mid k = 0, 1, 2, \dots\}$.

Step 0 Set $k = 0$.

Step 1 If $(x^k)^T s^k \leq \varepsilon$ then stop. Otherwise, if k is even (including 0), go to **Step 2**; if k is odd, go to **Step 3**.

Step 2 (Predictor Step). Set $\tau^k = 0$. Solve $(\Delta x_-^k, \Delta s_-^k)$ based on (4).

Find largest step size $0 < \alpha_1^k \leq 1$, such that $(x(\alpha_1^k), s(\alpha_1^k)) \in \mathcal{N}(\tau_1; \beta)$. Go to **Step 4**.

Step 3 (Corrector Step). Set $\tau^k = \tau_1$. Solve $(\Delta x_-^k, \Delta s_-^k)$ and $(\Delta x_+^k, \Delta s_+^k)$ based on (4) and (5).

Find step size vector $\alpha^k = (\alpha_1^k, 1) \in [0, 1]^2$, such that $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1; \beta/2)$ and α_1^k is maximum. Go to **Step 4**.

Step 4 Set $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$.

Let $k := k + 1$ and go to **Step 1**.

Remark that in the predictor step, since τ^k is set to be 0, we have $(\tau^k \mu_k e - x^k s^k)^+ = 0$, and so the Newton constituent direction with respect to the positive part is simply zero. In both the corrector and the predictor steps, we only need to search for a single step size. An important feature of the above algorithm is that in the corrector step we also aim at a large update of the target. In other words, the gap function is expected to be reduced for the corrector steps as well.

We shall now prove that Algorithm 5.1 indeed works correctly.

Let us denote

$$\lambda := \|(\tau_1 \mu e - xs)^+\|_1 / \|(\tau_1 \mu e - xs)^-\|_1, \quad (22)$$

which means

$$\lambda e^T (\tau_1 \mu e - xs)^- + e^T (\tau_1 \mu e - xs)^+ = 0.$$

So when we choose $\alpha_1 \geq \lambda$ we have

$$\mu(\alpha_1, 1) \leq \mu + \Delta x(\alpha_1, 1)^T \Delta s(\alpha_1, 1) / n. \quad (23)$$

If $(x, s) \in \mathcal{N}(\tau_1; \beta)$, $\tau_1 \leq 1/4$ and $\beta \leq 1/2$, then we derive from (13) that

$$\begin{aligned}
\lambda &\leq \sqrt{n} \|(\tau_1 \mu e - xs)^+\| / ((1 - \tau_1)x^T s) \\
&\leq \frac{\sqrt{\beta \tau_1}}{1 - \tau_1} \sqrt{\frac{\beta \tau_1}{n}} \\
&\leq \frac{\sqrt{2}}{3} \sqrt{\frac{\beta \tau_1}{n}}.
\end{aligned} \tag{24}$$

If $(x, s) \in \mathcal{N}(\tau_1; \beta)$, then computing $\hat{\eta}$ from (15) would yield

$$\hat{\eta} = 1, \tag{25}$$

and so by (10) we obtain immediately

$$h(\alpha_1, 1) \geq \tau_1 \mu e \tag{26}$$

for all $\alpha_1 \in [0, 1]$ and for all $(x, s) \in \mathcal{F}_{++}$.

Lemma 5.2. *Suppose $(x, s) \in \mathcal{N}(\tau_1; \beta)$, $\tau = \tau_1 \leq 1/4$ and $\beta \leq 1/2$. Then, for any $\alpha_1 \in \left[\lambda, \sqrt{\frac{\beta \tau_1}{2n}}\right]$ we have $(x(\alpha_1, 1), s(\alpha_1, 1)) \in \mathcal{N}(\tau_1; \beta/2)$.*

Proof. First of all, observe that (24) guarantees that the interval $\left[\lambda, \sqrt{\frac{\beta \tau_1}{2n}}\right]$ is not empty. Due to (23), (26), (25) and Lemma 3.6, we have

$$\begin{aligned}
&\|(\tau_1 \mu(\alpha_1, 1)e - x(\alpha_1, 1)s(\alpha_1, 1))^+\| \\
&= \|(\tau_1 \mu(\alpha_1, 1)e - h(\alpha_1, 1) - \Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^+\| \\
&\leq \left\| \left(\tau_1 \mu e - h(\alpha_1, 1) + \frac{\tau_1 \Delta x(\alpha_1, 1)^T \Delta s(\alpha_1, 1)}{n} e - \Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1) \right)^+ \right\| \\
&\leq \left\| \left(\frac{\tau_1 \Delta x(\alpha_1, 1)^T \Delta s(\alpha_1, 1)}{n} e - \Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1) \right)^+ \right\| \\
&\leq \left\| \frac{\tau_1 \Delta x(\alpha_1, 1)^T \Delta s(\alpha_1, 1)}{n} e + (-\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^+ \right\| \\
&\leq \left\| \frac{\tau_1 \Delta x(\alpha_1, 1)^T \Delta s(\alpha_1, 1)}{n} e \right\| + \|(-\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^+\| \\
&\leq \Delta x(\alpha_1, 1)^T \Delta s(\alpha_1, 1) + \|(\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^- \| \\
&= \|(\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^+\|_1 - \|(\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^- \|_1 + \|(\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^- \| \\
&\leq \|(\Delta x(\alpha_1, 1)\Delta s(\alpha_1, 1))^+\|_1 \\
&\leq (\beta/2)\tau_1(3\mu/4).
\end{aligned}$$

Applying Lemma 3.3 yields

$$\mu(\alpha_1, 1) \geq (1 - \sqrt{\beta\tau_1/2n})\mu \geq (1 - 1/4)\mu = 3\mu/4.$$

Therefore, $(x(\alpha_1, 1), s(\alpha_1, 1)) \in \mathcal{N}(\tau_1; \beta/2)$. \square

Lemma 5.3. *Suppose that $(x, s) \in \mathcal{N}(\tau_1; \beta)$, $\tau = \tau_1 \leq 1/4$ and $\beta \leq 1/2$. Then we have*

$$\mu\left(\sqrt{\frac{\beta\tau_1}{2n}}, 1\right) \leq \left(1 - \frac{\sqrt{2\beta\tau_1}}{32\sqrt{n}}\right)\mu.$$

Proof. Let us denote $\alpha^0 := (\sqrt{\frac{\beta\tau_1}{2n}}, 1)$. Notice that $\hat{\eta} = 1$. Due to (13) and Lemma 3.6 we obtain

$$\begin{aligned} \mu\left(\sqrt{\frac{\beta\tau_1}{2n}}, 1\right) &= \mu + \sqrt{\frac{\beta\tau_1}{2n}} \frac{e^T(\tau_1\mu e - xs)^-}{n} + \frac{e^T(\tau_1\mu e - xs)^+}{n} + \frac{\Delta x(\alpha^0)^T \Delta s(\alpha^0)}{n} \\ &\leq \mu - \sqrt{\frac{\beta\tau_1}{2n}} (1 - \tau_1)\mu + \|(\tau_1\mu e - xs)^+\|/\sqrt{n} + \|(\Delta x(\alpha^0)^T \Delta s(\alpha^0))^+\|_1/n \\ &\leq \mu - \frac{3\sqrt{2}}{8} \sqrt{\frac{\beta\tau_1}{n}} \mu + \frac{\beta\tau_1\mu}{\sqrt{n}} + \frac{3\beta\tau_1\mu}{8n} \\ &\leq \mu - \frac{3\sqrt{2}}{8} \sqrt{\frac{\beta\tau_1}{n}} \mu + \sqrt{\frac{\beta\tau_1}{8n}} \mu + \frac{3}{8\sqrt{8}} \sqrt{\frac{\beta\tau_1}{n}} \mu \\ &= \mu - \frac{\sqrt{2}}{32} \sqrt{\frac{\beta\tau_1}{n}} \mu. \end{aligned}$$

The lemma is proven. \square

A remarkable feature of Algorithm 5.1 as revealed by Lemma 5.3 is that, the gap measurement μ is reduced by a rate of $1 - 1/O(\sqrt{n})$ even at the corrector steps.

Lemma 5.4. *Let $(\Delta x^a, \Delta s^a)$ be the search direction of a predictor step in Algorithm 5.1, and $\bar{\alpha}$ be the actual step size taken in that predictor step. Then,*

$$\bar{\alpha} \geq \frac{2}{1 + \sqrt{1 + 4\delta/\beta\tau_1}}$$

where $\delta = \|\Delta x^a \Delta s^a\|/\mu$.

Proof. We have

$$\mu(\alpha) = x(\alpha)^T s(\alpha)/n = (1 - \alpha)\mu + \alpha^2(\Delta x^a)^T \Delta s^a/n.$$

Note that $(x, s) \in \mathcal{N}_2^-(\tau_1, \beta)$ and

$$\begin{aligned} \left\|(\tau_1((\Delta x^a)^T \Delta s^a/n)e - \Delta x^a \Delta s^a)^+\right\|^2 &\leq \left\|\tau_1((\Delta x^a)^T \Delta s^a/n)e - \Delta x^a \Delta s^a\right\|^2 \\ &= \|\Delta x^a \Delta s^a\|^2 - \tau_1(2 - \tau_1)((\Delta x^a)^T \Delta s^a)^2/n \\ &\leq \|\Delta x^a \Delta s^a\|^2. \end{aligned}$$

Therefore,

$$\begin{aligned}
& \|(\tau_1\mu(\alpha)e - x(\alpha)s(\alpha))^+\| \\
= & \left\| \left((1-\alpha)(\tau_1\mu e - xs) + \alpha^2\tau_1((\Delta x^a)^T \Delta s^a/n)e - \alpha^2\Delta x^a \Delta s^a \right)^+ \right\| \\
\leq & (1-\alpha) \|(\tau_1\mu e - xs)^+\| + \alpha^2 \left\| \left(\tau_1((\Delta x^a)^T \Delta s^a/n)e - \Delta x^a \Delta s^a \right)^+ \right\| \\
\leq & (1-\alpha)\beta\tau_1\mu + \alpha^2 \|\Delta x^a \Delta s^a\|.
\end{aligned}$$

Applying similar reasoning as in Lemma 4.17 of [16], we see that for each α with

$$0 \leq \alpha \leq \frac{2}{1 + \sqrt{1 + 4\delta/\beta\tau_1}}$$

we will have

$$\begin{aligned}
\|(\tau_1\mu(\alpha)e - x(\alpha)s(\alpha))^+\| & \leq (1-\alpha)\beta\tau_1\mu + \alpha^2 \|\Delta x^a \Delta s^a\| \\
& \leq 2\beta\tau_1(1-\alpha)\mu \\
& \leq 2\beta\tau_1\mu(\alpha),
\end{aligned}$$

and therefore, $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1; 2\beta)$. Differently put, we have $\bar{\alpha} \geq \frac{2}{1 + \sqrt{1 + 4\delta/\beta\tau_1}}$ as the lemma claims. \square

Since $\delta \leq n/2$, together with Lemma 5.3, the next theorem follows immediately (see also the proof of Theorem 4.18 in [16]).

Theorem 5.5. *Let $\beta = 1/4$. Then Algorithm 5.1 will terminate in $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iterations.*

Assume additionally that the LCP problem has a strictly complementary solution; that is, there is a partition B and N , and solution (x^*, s^*) , such that $B \cup N = \{1, 2, \dots, n\}$, $B \cap N = \emptyset$, $x_B^* > 0$, $x_N^* = 0$, and $s_B^* = 0$, $s_N^* > 0$. Since the sequence generated by Algorithm 5.1 is contained in a wide neighborhood, it satisfies

$$(1 - \beta)\tau_1\mu_k \leq x^k s^k \leq n\mu^k. \quad (27)$$

Using Lemma 2 of [3], we know that there exists some constant $0 < \xi < 1$ such that

$$\xi \leq x_j^k \leq 1/\xi \quad \text{for } j \in B, \quad \text{and} \quad \xi \leq s_j^k \leq 1/\xi \quad \text{for } j \in N. \quad (28)$$

For simplicity, we drop the index k . We apply the same proof as for Theorem 3.6 in [17], and so relations (27) and (28) give rise to

$$\|\Delta x^a\| = O(\mu) \quad \text{and} \quad \|\Delta x^a\| = O(\mu).$$

Then due to Lemma 5.4, we have the following result:

Theorem 5.6. *Let $\{(x^k, s^k) \mid k = 0, 1, 2, \dots\}$ be the sequence generated by Algorithm 5.1. Suppose that (LCP) has a strictly complementary solution. Then, $(x^k)^T s^k \rightarrow 0$ Q -quadratically for the predictor steps.*

6 Preliminary numerical tests

We shall test our algorithms on some randomly generated instances, in order to get a feel of how the method might perform in practice.

To achieve this, we wrote simple Matlab codes for four algorithms: (1) the Mizuno-Todd-Ye type predictor-corrector algorithm [8]; (2) the classical wide-neighborhood path-following algorithm of Kojima-Mizuno-Yoshise [5]; (3) Algorithm 4.3; and (4) Algorithm 5.1. These four algorithms will be denoted, respectively, by: (1) Algorithm PC; (2) Algorithm WN; (3) Algorithm New-WN; and (4) Algorithm New-PC. All algorithms do not use Mehrotra's higher order correction technique. The neighborhoods are taken to be $\mathcal{N}_2(1/2)$ in predictor step and $\mathcal{N}_2(1/4)$ in corrector step for Algorithm PC, and $\mathcal{N}_\infty^-(1 - \tau/2)$ for Algorithm WN, and $\beta = 1/2$ for Algorithm New-WN and Algorithm New-PC. To test the role of the parameter τ , we tried three different values of τ : $\tau = \tau^1 = 0.005$, $\tau = \tau^2 = 0.001$, and $\tau = \tau^3 = 0.0005$, respectively. All algorithms terminate after the relative duality gap satisfies

$$\frac{x^T s}{(x^0)^T s^0 + 1} \leq 10^{-8}.$$

For each dimension n , the entry in the column 'iter' is the average number of iterations of 10 randomly generated monotone LCPs with the same n , and the number in the bracket is the standard deviation of these 10 runs. In case a Matlab numerical warning occurred in the procedure, then we mark a superscript * next to that corresponding entry.

The first set of testing monotone LCP problems are generated as follows. After one inputs any positive integer n , Matlab generates an $n \times n$ matrix $A = \text{rand}(n)$ randomly. Then we take $M = A^T A$ and $b = e - Me$ to obtain a monotone LCP and its initial feasible solution (e, e) . The numerical results of this set of problems are showed in Table 1.

To test the influence from the skewness of matrix M , in the next set of test problems we let $M = A^T A + m(B - B^T)$, where $B = \text{rand}(n)$ and $m = 1$. The numerical results are showed in Table 2. It turns out that the number of iterations actually decreases on average as compared with the case when M is purely positive semidefinite. In our experiences, we found that the numbers of iterations for all algorithms tested always decrease if m increases.

Based on the numerical results we have generated so far, Algorithm New-WN is the fastest, and Algorithm New-WN and Algorithm New-PC are faster than Algorithm WN, while Algorithm PC

| n | PC | WN | | | New-WN | | | New-PC | | |
|------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | | $\tau = \tau^1$ | $\tau = \tau^2$ | $\tau = \tau^3$ | $\tau = \tau^1$ | $\tau = \tau^2$ | $\tau = \tau^3$ | $\tau = \tau^1$ | $\tau = \tau^2$ | $\tau = \tau^3$ |
| | iter | iter | iter | iter | iter | iter | iter | iter | iter | iter |
| 100 | 34.0 (1.70) | 16.8 (0.70) | 16.7 (0.87) | 16.7 (1.33) | 10.7 (0.43) | 10.6 (0.45) | 10.6 (0.45) | 12.5 (0.47) | 12.2 (0.49) | 12.1 (0.50) |
| 200 | 36.8 (0.61) | 16.1 (0.57) | 16.9 (0.37) | 16.0 (0.66) | 10.6 (0.20) | 10.7 (0.22) | 10.7 (0.22) | 12.9 (0.23) | 12.9 (0.23) | 12.8 (0.22) |
| 300 | 38.6 (0.61) | 17.1 (0.30) | 17.9 (0.49) | 19.4 (0.64) | 11.0 (0.12) | 11.1 (0.10) | 11.2 (0.11) | 13.2 (0.18) | 13.3 (0.16) | 13.3 (0.16) |
| 400 | 41.6 (0.65) | 20.0 (0.42) | 19.0 (0.45) | 21.3 (0.57) | 11.7 (0.20) | 11.7 (0.19) | 11.7 (0.19) | 13.6 (0.23) | 13.7 (0.21) | 13.7 (0.21) |
| 500 | 40.6* (0.50) | 19.4 (0.54) | 20.8 (0.40) | 19.8 (0.63) | 11.2 (0.12) | 11.4 (0.13) | 11.4 (0.13) | 13.5 (0.17) | 13.4 (0.18) | 13.4 (0.18) |
| 600 | 43.2 (0.45) | 20.6 (0.34) | 21.2 (0.44) | 23.8 (0.45) | 11.3 (0.10) | 11.5 (0.12) | 11.5 (0.12) | 14.1 (0.16) | 14.2 (0.17) | 14.2 (0.17) |
| 700 | 43.6 (0.30) | 22.5 (0.49) | 22.8 (0.54) | 22.1 (0.40) | 12.0 (0.10) | 12.0 (0.10) | 12.1 (0.10) | 14.8 (0.13) | 14.9 (0.15) | 14.8 (0.16) |
| 800 | 44.0* (0.34) | 22.3 (0.41) | 23.3 (0.47) | 23.0 (0.26) | 12.3 (0.07) | 12.4 (0.09) | 12.4 (0.09) | 15.0 (0.14) | 15.0 (0.14) | 14.9 (0.15) |
| 900 | 43.4* (0.32) | 22.3 (0.29) | 23.1 (0.34) | 21.3 (0.41) | 12.0 (0.12) | 12.0 (0.12) | 12.0 (0.12) | 14.4 (0.18) | 14.4 (0.18) | 14.5 (0.17) |
| 1000 | 44.4 (0.28) | 25.0 (0.41) | 23.3 (0.49) | 23.8 (0.60) | 12.2 (0.10) | 12.1 (0.10) | 12.1 (0.10) | 15.3 (0.13) | 15.4 (0.13) | 15.4 (0.13) |

Table 1: Iteration Numbers on monotone LCPs with $M = A^T A$

| n | PC | WN | | | New-WN | | | New-PC | | |
|------|-------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | | $\tau = \tau^1$ | $\tau = \tau^2$ | $\tau = \tau^3$ | $\tau = \tau^1$ | $\tau = \tau^2$ | $\tau = \tau^3$ | $\tau = \tau^1$ | $\tau = \tau^2$ | $\tau = \tau^3$ |
| | iter | iter | iter | iter | iter | iter | iter | iter | iter | iter |
| 100 | 11.0 (0.00) | 6.2 (0.46) | 5.7 (0.28) | 5.1 (0.22) | 4.8 (0.13) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) |
| 200 | 9.0 (0.00) | 5.0 (0.17) | 4.9 (0.21) | 5.1 (0.19) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) |
| 300 | 9.0 (0.00) | 4.5 (0.12) | 5.3 (0.18) | 4.9 (0.19) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) |
| 400 | 9.0 (0.00) | 4.6 (0.10) | 5.4 (0.16) | 5.3 (0.16) | 4.0 (0.00) | 4.0 (0.00) | 3.8 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) |
| 500 | 9.0 (0.00) | 4.7 (0.09) | 4.7 (0.11) | 4.4 (0.09) | 4.0 (0.00) | 4.0 (0.00) | 3.2 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) |
| 600 | 9.0 (0.00) | 4.5 (0.06) | 4.6 (0.12) | 4.5 (0.09) | 4.0 (0.00) | 3.4 (0.06) | 3.1 (0.04) | 4.0 (0.00) | 4.0 (0.00) | 3.8 (0.05) |
| 700 | 9.0 (0.00) | 4.7 (0.08) | 3.8 (0.07) | 4.3 (0.09) | 4.0 (0.00) | 3.0 (0.00) | 3.0 (0.00) | 4.0 (0.00) | 4.0 (0.00) | 3.3 (0.05) |
| 800 | 9.0 (0.00) | 4.5 (0.06) | 3.2 (0.07) | 4.4 (0.11) | 4.0 (0.00) | 3.0 (0.00) | 3.0 (0.00) | 4.0 (0.00) | 3.7 (0.05) | 3.1 (0.03) |
| 900 | 9.0 (0.00) | 4.4 (0.05) | 3.7 (0.07) | 4.0 (0.16) | 4.0 (0.00) | 3.0 (0.00) | 3.0 (0.00) | 4.0 (0.00) | 3.5 (0.05) | 3.0 (0.00) |
| 1000 | 9.0 (0.00) | 4.0 (0.00) | 3.4 (0.05) | 3.9 (0.08) | 4.0 (0.00) | 3.0 (0.00) | 3.0 (0.00) | 4.0 (0.00) | 3.5 (0.05) | 3.0 (0.00) |

Table 2: Iteration Numbers on monotone LCPs with $M = A^T A + m(B - B^T)$

appears to be the slowest. Moreover, Algorithm WN, Algorithm New-WN, and Algorithm New-PC had always run smoothly, but Algorithm PC got 3 warnings due to badly scaled matrices. Certainly, our implementations are very coarse. For instance, we did not fine-tune the parameters, nor did we use any higher order corrections. In the future, we plan to study the performance of the method for practical problems with more refined linear algebras and careful implementations.

References

- [1] W. Ai, *Neighborhood-following interior-point algorithm for linear programming*, Proceedings on the 5th International Conference on Optimization: Techniques and Applications (ICOTA), Vol. 2, 795–801 (December 15-17, 2001. Hong Kong). To appear in *Science in China*.
- [2] C.C. Gonzaga, *The largest step path following algorithm for monotone linear complementarity problems*, *Mathematical Programming* 76, 309–332, 1997.
- [3] O. Güler and Y. Ye, *Convergence behavior of interior-point algorithms*, *Mathematical Programming* 60, 215–228, 1993.

- [4] P. Hung and Y. Ye, *An asymptotical $O(\sqrt{n}L)$ -iteration path-following linear programming algorithm that use wide neighborhoods*, SIAM Journal on Optimization 6, 570–586, 1996.
- [5] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in *Progress in Mathematical Programming: Interior point and related methods* pp. 29–47, (ed. N. Megiddo), Springer Verlag, New York, 1989.
- [6] M. Kojima, S. Mizuno, and A. Yoshise, *A polynomial-time algorithm for a class of linear complementarity problems* Mathematical Programming 44, 1–26, 1989.
- [7] N. Megiddo, *Pathways to the optimal set in linear programming*, in *Progress in Mathematical Programming: Interior point and related methods* pp. 131–158, (ed. N. Megiddo), Springer Verlag, New York, 1989.
- [8] S. Mizuno, M.J. Todd, and Y. Ye, *On adaptive step primal-dual interior-point algorithms for linear programming*, Mathematics of Operations Research 18, 964–981, 1993.
- [9] R.D.C. Monteiro, I. Adler, *Interior path following primal-dual algorithms, Part I: linear programming*, Mathematical Programming 44, 27–41, 1989.
- [10] R.D.C. Monteiro, I. Adler, *Interior path following primal-dual algorithms, Part II: convex quadratic programming*, Mathematical Programming 44, 43–66, 1989.
- [11] J.M. Peng, T. Terlaky, and Y.B. Zhao, *A predictor-corrector algorithm for linear optimization based on a specific self-regular proximity function*, Technical Report, McMaster University, Ontario, Canada, 2003.
- [12] J.F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, Volumes 11-12, pp. 625–653, 1999. Special issue on Interior Point Methods (CD supplement with software).
- [13] J.F. Sturm and S. Zhang, *On a wide region of centers and primal-dual interior point algorithms for linear programming*, Mathematics of Operations Research 22, 408–431, 1997.
- [14] S.J. Wright, *Primal-dual interior-point methods*, SIAM Publications, Philadelphia, 1997.
- [15] X. Xu, *An $O(\sqrt{n}L)$ -iteration large-step infeasible path-following algorithm for linear programming*, Technical Report, Department of Management Sciences, University of Iowa, Iowa City, 1994.
- [16] Y. Ye, *Interior point algorithms: theory and analysis*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1997.

- [17] Y. Ye and K. Anstreicher, *On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for LCP* Mathematical Programming 62., 537–551, 1993.
- [18] Y. Ye, O. Güler, R.A. Tapia and Y. Zhang, *A quadratically convergent $O(\sqrt{n}L)$ -iteration algorithm for linear programming*, Mathematical Programming 59, 151–162, 1993.