Optim Lett (2016) 10:619–634
DOI 10.1007/s11590-015-0889-0

ORIGINAL PAPER

An $O(\sqrt{n}L)$ iteration Mehrotra-type predictor-corrector algorithm for monotone linear complementarity problem

Changhe Liu[1] · **Youlin Shang**[1] · **Hongwei Liu**[2]

Received: 21 June 2014 / Accepted: 1 April 2015 / Published online: 11 April 2015

Abstract In this paper we propose a new class of Mehrotra-type predictor-corrector algorithm for the monotone linear complementarity problems (LCPs). At each iteration, the method computes a corrector direction in addition to the Ai–Zhang direction (SIAM J Optim 16:400–417, 2005), in an attempt to improve performance. Starting with a feasible point (x^0, s^0) in the wide neighborhood $\mathcal{N}(\tau, \beta)$, the algorithm enjoys the low iteration bound of $O(\sqrt{n}L)$, where n is the dimension of the problem and $L = \log \frac{(x^0)^T s^0}{\varepsilon}$ with ε the required precision. We also prove that the new algorithm can be specified into an easy implementable variant for solving the monotone LCPs, in such a way that the iteration bound is still $O(\sqrt{n}L)$. Some preliminary numerical results are provided as well.

Keywords Linear complementarity problem · Interior-point methods · Wide neighborhood · Mehrotra-type predictor-corrector algorithm · Polynomial complexity

✉ Changhe Liu
changheliu@163.com

Hongwei Liu
hwliu@mail.xidian.edu.cn

1 Department of Applied Mathematics, Henan University of Science and Technology, Luoyang, People's Republic of China

2 Department of Applied Mathematics, Xidian University, Xi'an, People's Republic of China

1 Introduction

Consider the following monotone linear complementary problem (LCP):

$$
\begin{aligned}
s &= Mx + q, \\
x^T s &= 0, \quad x \geq 0, \quad s \geq 0,
\end{aligned}
\tag{1.1}
$$

where $q \in R^n$ and $M \in R^{n\times n}$ is a monotone matrix, i.e., $x^T Mx \geq 0$ for any $x \in R^n$. A particular choice of M is a block skew symmetric matrix, $M = \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix}$. In that case, the corresponding monotone LCP is nothing but a linear programming (LP) problem.

The primal-dual interior-point method was first introduced by Kojima et al. [6] and Megiddo [9], which essentially aims at solving the following parameterized problem by Newton's method, for shrinking values of the parameter $\mu > 0$:

$$
\begin{aligned}
s &= Mx + q, \\
xs &= \mu e, \quad x > 0, \quad s > 0.
\end{aligned}
\tag{1.2}
$$

The exact solution of the above problem is known as the analytic central path, with the varying path parameter $\mu > 0$. Applying Newton's method to (1.2) for a given feasible point (x, s) gives the following linear system of equations

$$
\begin{aligned}
M\triangle x - \triangle s &= 0, \\
s\triangle x + x\triangle s &= \mu e - xs,
\end{aligned}
\tag{1.3}
$$

where $(\triangle x, \triangle s)$ give the Newton step. 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 [6], Kojima et al. proposed that the iterates reside in a wide neighborhood of the central path, known as the $\mathcal{N}_\infty^-$-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 (meaning a reduction by a universal percentage, independent of the problem parameters) at each iteration. The worst case iteration bound was proved to be $O(nL)$, where n is the dimension of a standard LP problem, and L is its input-length. Then, in a subsequent paper [7], 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 $O(\sqrt{n}L)$. The first practically efficient $O(\sqrt{n}L)$ primal-dual interior-point algorithm for LP was the predictor-corrector algorithm of Mizuno, Todd and Ye (MTY) [11]. In the predictor step, the MTY algorithm uses the so-called primal-dual affine scaling step with $\mu = 0$ in (1.3) and the iterate moves to a slightly larger neighborhood. Then, in the corrector step, it uses $\mu = x^T s/n$ to bring the iterate towards the central path, back to the smaller neighborhood. 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. Recently, Ai and Zhang [1] introduced a new class of primal-dual path-following interior-point algorithms for solving monotone LCPs, working with large update and wide neighborhood at each iteration. They showed that their algorithm enjoys the iteration bound of $O(\sqrt{n}L)$. This result yields the first wide neighborhood path-following algorithm having the same theoretical complexity as a small neighborhood algorithm for monotone LCPs, which include LP as a special case. They also proposed a predictor-corrector variant of the method in [1], and proved that the predictor steps converge Q-quadratically if the problem has a strictly complementary solution.

A more practical predictor-corrector algorithm whose variants are the backbone of most interior-point algorithms based software packages is Mehrotra's predictor-corrector algorithm (MPC) [10]. The use of the same coefficient matrix of the Newton system in both the predictor and corrector steps and an adaptive update of the centering parameter led to superior practical performance compared to all the aforementioned polynomial time algorithms. However, no convergence theory is available for Mehrotra's original algorithm. In fact, there are examples for which the MPC algorithm diverges. Simple safeguards could be incorporated into the method to force it into the convergence framework of existing methods [12,13,15]. Recently, in [8], the authors propose a primal-dual second-order corrector interior-point algorithm for LP, which is a special case of the monotone LCPs. It adds a corrector step to Ai–Zhang's path-following algorithm [1], in an attempt to improve performance. The preliminary implementations show that this algorithm is promising for solving LP. The current paper aims at modifying Ai–Zhang's path-following algorithm in [1] to gain a class of Mehrotra-type predictor-corrector algorithm for monotone LCPs. The algorithm has the low iteration bound of $O(\sqrt{n}L)$ and a superior practical performance for solving LCPs. It must be pointed out that the corrector direction and the step length of the proposed algorithm are different from the algorithm in [8]. The new schemes in this paper will enable us to prove the monotonicity of the duality gap with respect to the step length for the monotone LCPs. In fact, the main result of this paper is to prove that the proposed generic method can be specified into an easy implementable variant with given parameters for the monotone LCPs, in such a way that the iteration bound is $O(\sqrt{n}L)$.

The rest of the paper is organized as follows. In Sect. 2, we recall the key features of Ai–Zhang's method and then present a new class of Mehrotra-type predictor-corrector algorithm. In Sect. 3, we establish the worst case iteration complexity of the proposed algorithm and discuss an easy implementable variant of the general method. In Sect. 4, some preliminary numerical results are reported. In Sect. 5 some discussions and conclusions are given.

Throughout the paper, all of the vectors are column vectors and e denotes the vector with all components equal to one. For vector $x \in R^n$, x_i denotes the ith component of vector x and X denotes the $n \times n$ diagonal matrix with x as the diagonal components. For any two vectors x and s, xs denotes the componentwise product of the two vectors, and so is true for other operations, e.g., $1/(xs)$ and $(xs)^{-0.5}$. For any $a \in R$, 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 $x \in R^n$, namely x^+ is the nonnegative part of x and x^- is the nonpositive part of x. The L_p-norm of $x \in R^n$ is denoted by $\|\cdot\|_p$, and in particular we write $\|\cdot\|$ for Euclidean norm $\|\cdot\|_2$.We also use the notations $\mathcal{I} =: \{1, 2, \ldots, n\}$, $\mathcal{F} := \{(x, s) \in R^n \times R^n : s = Mx + q, x \geq 0, s \geq 0\}$, and $\mathcal{F}^0 := \{(x, s) \in R^n \times R^n : s = Mx + q, x > 0, s > 0\}$.

2 Mehrotra-type predictor-corrector algorithm

The central path for the LCP is defined as $\mathcal{C} := \{(x, s) \in \mathcal{F}^0 : xs = \mu e\}$, and its so-called wide neighborhood is defined as follows:

$$\mathcal{N}_\infty^-(1-\tau) := \{(x, s) \in \mathcal{F}^0 : xs \geq \tau\mu\},$$

where $\tau \in (0, 1)$ is a given constant and $\mu = x^T s/n$. Ai and Zhang [1] have introduced a new neighborhood for the central path, defined as

$$\mathcal{N}(\tau_1, \tau_2) := \{(x, s) \in \mathcal{F}^0 : \|(\tau_1\mu e - xs)^+\| \leq (\tau_1 - \tau_2)\mu\},$$

where $0 < \tau_2 < \tau_1 < 1$ are two parameters. For convenience we set $\beta = (\tau_1 - \tau_2)/\tau_1$ and introduce a new notation to indicate the neighborhood in this paper, i.e.,

$$\mathcal{N}(\tau, \beta) := \{(x, s) \in \mathcal{F}^0 : \|(\tau\mu e - xs)^+\| \leq \beta\tau\mu\}.$$

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

$$\mathcal{N}_\infty^-(1-\tau) \subseteq \mathcal{N}(\tau, \beta) \subseteq \mathcal{N}_\infty^-(1-(1-\beta)\tau), \ \forall\, 0 < \tau, \beta < 1. \tag{2.1}$$

Another key ingredient of Ai–Zhang's method is to decompose the Newton step into the following two equations:

$$\begin{aligned} M\triangle x_- - \triangle s_- &= 0, \\ s\triangle x_- + x\triangle s_- &= (\tau\mu e - xs)^- \end{aligned} \tag{2.2}$$

and

$$\begin{aligned} M\triangle x_+ - \triangle s_+ &= 0, \\ s\triangle x_+ + x\triangle s_+ &= (\tau\mu e - xs)^+. \end{aligned} \tag{2.3}$$

As pointed out in [1], the part $(\tau\mu e - xs)^-$ is important for the progress towards optimality, and the other part $(\tau\mu e - xs)^+$ is used to control the centrality. To make this point clear, let us move the iterate from (x, s) to $(x + \alpha\triangle x, s + \alpha\triangle s)$, where $(\triangle x, \triangle s) = (\triangle x_-, \triangle s_-) + (\triangle x_+, \triangle s_+)$ is the usual Newton direction since

$$s\triangle x + x\triangle s = (\tau\mu e - xs)^- + (\tau\mu e - xs)^+ = \tau\mu e - xs. \tag{2.4}$$

At each iteration we expect the duality gap to decrease by

$$\begin{aligned} x^T s - (x + \alpha\triangle x)^T (s + \alpha\triangle s) &= -\alpha(x^T\triangle s + s^T\triangle x) \\ &= -\alpha e^T(\tau\mu e - xs) \\ &= -\alpha\sum[\tau\mu e - xs]_i^- - \alpha\sum[\tau\mu e - xs]_i^+, \end{aligned}$$

from which we see that the negative components of $\tau\mu e - xs$ are responsible for reducing the duality gap. On the other hand, the positive components of $\tau\mu e - xs$, i.e., $x_i s_i < \tau\mu = \tau x^T s/n$ indicate that the iterate is "close" to the boundary of the positive orthant. For these coordinates, using (2.4) we have

$$\begin{aligned} (x_i + \alpha\triangle x_i)(s_i + \alpha\triangle s_i) &= x_i s_i + \alpha(x_i\triangle s_i + s_i\triangle x_i) + \alpha^2(\triangle x_i\triangle s_i) \\ &= x_i s_i + \alpha(\tau\mu - x_i s_i) + \alpha^2(\triangle x_i\triangle s_i). \end{aligned}$$

Because the coefficient of the first order term of α is bigger than zero, then $(x_i + \alpha\triangle x_i)(s_i + \alpha\triangle s_i)$ increases locally and pushes the iterate to the interior of the first orthant, i.e., keeping the centrality. Ai and Zhang [1] suggested to treat negative and positive components of $\tau\mu e - xs$ separately to obtain a better iteration complexity bound for wide neighborhood IPMs.

Our new Mehrotra-type predictor-corrector algorithm uses the information from (2.2) to compute the corrector direction by

$$\begin{aligned} M\triangle x_-^c - \triangle s_-^c &= 0, \\ s\triangle x_-^c + x\triangle s_-^c &= -\alpha_a\triangle x_-\triangle s_-, \end{aligned} \tag{2.5}$$

where $\alpha_a \in [0, 1]$ is the maximum feasible step length that ensures $(x + \alpha_a\triangle x_-, s + \alpha_a\triangle s_-) \geq 0$, i.e.,

$$\alpha_a := \arg\max\{\alpha_a \in [0, 1] : x + \alpha_a\triangle x_- \geq 0, s + \alpha_a\triangle s_- \geq 0\}. \tag{2.6}$$

Finally, the new iterate is

$$\begin{aligned} (x(\alpha), s(\alpha)) &:= (x, s) + (\triangle x(\alpha), \triangle s(\alpha)) \\ &:= (x, s) + \alpha_1((\triangle x_-, \triangle s_-) + (\triangle x_-^c, \triangle s_-^c)) + \alpha_2(\triangle x_+, \triangle s_+). \end{aligned} \tag{2.7}$$

To get the best step lengths we expect to solve the following subproblem:

$$\begin{aligned} &\min x(\alpha)^T s(\alpha) \\ &s.t.\ (x(\alpha), s(\alpha)) \in \mathcal{N}(\tau, \beta), \\ &0 \leq \alpha_1, \alpha_2 \leq 1. \end{aligned} \tag{2.8}$$

Most recently, in [3] the author provided an example that shows that Mehrotra-type predictor-corrector algorithms may fail to converge to an optimal solution. The cause of the bad performance of the algorithm on the problem is that the corrector direction had too much influence on the resulting search direction. Indeed, in [8,15] the new iterate is constructed as a quadratic function of the step length from the current iterate, where the predictor direction and the corrector are multiplied by the step length and the square of the step length, respectively. In a similar way, Salahi et al. [13] propose finding the corrector by weighting the term $\triangle x_- \triangle s_-$ in (2.5) by the allowed step length α_a for the predictor direction. Moreover, [12] combines above two methods and Colombo and Gondzio [5] give the general method with weighting the corrector by a parameter $\omega \in (0, 1]$. Our method in this paper is similar to [13]. However, instead of trying to correct all Newton direction, we only try to correct the direction $(\triangle x_-, \triangle s_-)$, along which the iterate approaches to the boundary of the positive orthant. The direction $(\triangle x_+, \triangle s_+)$, along which the iterate is pushed to the interior of the positive orthant, is already reasonably good, and does not need to be changed. In fact, the step size taken along the direction $(\triangle x_+, \triangle s_+)$ can always be chosen one according to Ai and Zhang [1]. On the other hand, instead of adding the centering term to the corrector step, we move it to the predictor step as in [4,8,15]. We point out that the corrector direction and the step lengths in the present paper are different from the algorithm in [8]. The new schemes will enable us to prove the monotonicity of the duality gap with respect to the step length for the monotone LCPs. Then, the proposed generic method can be specified into an easy implementable variant with given parameters and the iteration bound is still $O(\sqrt{n}L)$.

Now, after all the previous discussions we outline the Mehrotra-type predictor-corrector algorithm as follows.

Algorithm 2.1 (*Mehrotra-type predictor-corrector algorithm*)

Input parameters $\varepsilon > 0,\ 0 < \tau, \beta < 1$, and an initial point $(x^0, s^0) \in \mathcal{N}(\tau, \beta)$ with $\mu_0 = (x^0)^T s^0/n$.

Step 0 Set $k := 0$.

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

Step 2 (Predictor step) Compute the directions $(\triangle x_-^k, \triangle s_-^k)$ by (2.2) and $(\triangle x_+^k, \triangle s_+^k)$ by (2.3). Find the maximum feasible step length α_a by (2.6).

Step 3 (Corrector step) Compute the directions $(\triangle x_-^{c,k}, \triangle s_-^{c,k})$ by (2.5). Find the step lengths $\alpha^k = (\alpha_1^k, \alpha_2^k)$ giving a sufficient reduction of the duality gap and assuring $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau, \beta)$.

Step 4 Let $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$ and $\mu_{k+1} = (x^{k+1})^T s^{k+1}/n$. Set $k := k+1$ and go to Step 1.

There are three comments we would like to address about the presented algorithm. First of all, the algorithm is a feasible algorithm, and an initial point (x^0, s^0) in $\mathcal{N}(\tau, \beta)$ is needed. It is easy to obtain such a point by employing homogeneous and self-dual embedding method for the monotone LCP of Ye [2,14]. Second, although we suggest to solve the subproblem (2.8) to decide the best step lengths, solving this problem could be expensive. Hence, a "sufficient" duality gap decrease obtained at a low computational cost is preferred against the "maximal possible" duality gap decrease at a high computational cost. Even if we do not use the optimal solution of problem (2.8) as the step lengths, we are still able to achieve polynomial convergence. In fact, we will prove in Sect. 3 that the above algorithm can be specified into an easy implementable variant with given parameters, and has an iteration bound of $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$. Third, in spite of the fact that three linear systems (2.2), (2.3), and (2.5) have to be solved, the additional cost is very marginal, since they have the same coefficient matrix.

Before proceeding to the complexity result, we have to give some technical lemmas that will be used frequently during the analysis.

Lemma 2.1 *Let* $(x, s) \in \mathcal{F}^0$, $(\triangle x_-, \triangle s_-)$ *be the solution of* (2.2)*, and* α_a *is the maximum feasible step length defined by* (2.6)*. Let* $\mathcal{I}_+ = \{i \in \mathcal{I} : (\triangle x_-)_i(\triangle s_-)_i > 0\}$ *and* $\mathcal{I}_- = \{i \in \mathcal{I} : (\triangle x_-)_i(\triangle s_-)_i < 0\}$. *Then*

$$(\triangle x_-)_i(\triangle s_-)_i \leq \frac{x_i s_i}{4}, \quad \forall i \in \mathcal{I}_+$$

and

$$-(\triangle x_-)_i(\triangle s_-)_i \leq \frac{x_i s_i}{\alpha_a^2}, \quad \forall i \in \mathcal{I}_-.$$

Proof By Eq. (2.2) for $i \in \mathcal{I}_+$ we have

$$s_i(\triangle x_-)_i + x_i(\triangle s_-)_i = (\tau\mu - x_i s_i)^- \leq 0.$$

Since $(\triangle x_-)_i(\triangle s_-)_i > 0$, this inequality implies that both $s_i(\triangle x_-)_i < 0$ and $s_i(\triangle s_-)_i < 0$. Then, from

$$2\sqrt{x_i s_i(\triangle x_-)_i(\triangle s_-)_i} \leq -s_i(\triangle x_-)_i - x_i(\triangle s_-)_i = (x_i s_i - \tau\mu)^+ \leq x_i s_i$$

we get $(\triangle x_-)_i(\triangle s_-)_i \leq \frac{x_i s_i}{4}$.

For the maximum feasible step length α_a, one has

$$(x_i + \alpha_a(\triangle x_-)_i)(s + \alpha_a(\triangle s_-)_i) \geq 0.$$

This is equivalent to

$$x_i s_i + \alpha_a(\tau\mu - x_i s_i)^- + \alpha_a^2(\triangle x_-)_i(\triangle s_-)_i \geq 0.$$

For $i \in \mathcal{I}_-$ the statement of the lemma follows. □

Lemma 2.2 *Let* $(x, s) \in \mathcal{F}^0$, $(\triangle x_-, \triangle s_-)$ *be the solution of* (2.2). *Then, we have*

$$\sum_{i \in \mathcal{I}_-} |(\triangle x_-)_i(\triangle s_-)_i| \leq \sum_{i \in \mathcal{I}_+} (\triangle x_-)_i(\triangle s_-)_i \leq \frac{x^T s}{4}.$$

Proof By the monotonicity we have $\triangle x_-{}^T \triangle s_- \geq 0$. Therefore, the result of the lemma is a direct consequence of Lemma 2.1. □

Lemma 2.3 [1, Lemma 3.5] *Let $u, v \in R^n$ be such that $u^T v \geq 0$, and let $r = u + v$. Then, we have*

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

We also note the following simple but useful relationships:

$$-x^T s \leq e^T(\tau\mu e - xs)^- \leq -(1-\tau)x^T s, \tag{2.9}$$

$$e^T(\tau\mu e - xs)^+ \leq \sqrt{n}\|(\tau\mu e - xs)^+\|. \tag{2.10}$$

Let us denote

$$h(\alpha) := xs + \alpha_1(\tau\mu e - xs)^- + \alpha_2(\tau\mu e - xs)^+ - \alpha_1\alpha_a(\triangle x_- \triangle s_-).$$

Lemma 2.4 *Suppose $(x, s) \in \mathcal{N}(\tau, \beta)$, $\beta \leq 1/2$, and $\alpha_1 \leq \alpha_2\sqrt{\frac{\beta\tau}{2n}}$. Then we have*

$$\|(\triangle x(\alpha)\triangle s(\alpha))^-\|_1 \leq \|(\triangle x(\alpha)\triangle s(\alpha))^+\|_1 \leq \frac{2}{3}\alpha_2^2\beta\tau\mu.$$

Proof Denote $u := (x^{-0.5}s^{0.5})\triangle x(\alpha)$, $v := (x^{0.5}s^{-0.5})\triangle s(\alpha)$ and

$$r := (xs)^{-0.5}(\alpha_1(\tau\mu e - xs)^- + \alpha_2(\tau\mu e - xs)^+) - \alpha_1\alpha_a(xs)^{-0.5}(\triangle x_- \triangle s_-).$$

So by Eqs. (2.2), (2.3) and (2.5) we have $u^T v = \triangle x(\alpha)^T \triangle s(\alpha) \geq 0$ and $u + v = r$. Therefore, by Lemma 2.3 we have

$$\begin{aligned}
\|(\triangle x(\alpha)\triangle s(\alpha))^-\|_1 &\leq \|(\triangle x(\alpha)\triangle s(\alpha))^+\|_1 \\
&\leq \frac{1}{4}\|(xs)^{-0.5}(\alpha_1(\tau\mu e - xs)^- + \alpha_2(\tau\mu e - xs)^+) \\
&\quad - \alpha_1\alpha_a(xs)^{-0.5}(\triangle x_- \triangle s_-)\|^2 \\
&\leq \frac{1}{4}(\|(\alpha_1(\tau\mu e - xs)^- + \alpha_2(\tau\mu e - xs)^+)/\sqrt{xs}\| \\
&\quad + \alpha_1\alpha_a\|(\triangle x_- \triangle s_-)/\sqrt{xs}\|)^2.
\end{aligned}$$

By the relationships (2.1) we have

$$\begin{aligned}
&\|(\alpha_1(\tau\mu e - xs)^- + \alpha_2(\tau\mu e - xs)^+)/\sqrt{xs}\|^2 \\
&\quad = \alpha_1^2\|(\tau\mu e - xs)^-/\sqrt{xs}\|^2 + \alpha_2^2\|(\tau\mu e - xs)^+/\sqrt{xs}\|^2 \\
&\quad \leq \alpha_1^2\|(\sqrt{xs} - \tau\mu/\sqrt{xs})^+\|^2 + \alpha_2^2\|(\tau\mu e - xs)^+\|^2/((1-\beta)\tau\mu) \\
&\quad \leq \alpha_1^2\|\sqrt{xs}\|^2 + \alpha_2^2(\beta\tau\mu)^2/((1-\beta)\tau\mu)
\end{aligned}$$

$$
\begin{aligned}
&\leq \alpha_2^2 \frac{\beta\tau}{2n}(n\mu) + \alpha_2^2 \beta\tau\mu \\
&= \frac{3}{2}\alpha_2^2 \beta\tau\mu.
\end{aligned}
$$

By Lemma 2.1 and Lemma 2.2 we have

$$
\begin{aligned}
&(\alpha_1\alpha_a \|(\triangle x_- \triangle s_-)/\sqrt{xs}\|)^2 \\
&= \alpha_1^2 \left(\alpha_a^2 \sum_{i \in \mathcal{I}_+} \frac{((\triangle x_-)_i (\triangle s_-)_i)^2}{x_i s_i} + \alpha_a^2 \sum_{i \in \mathcal{I}_-} \frac{((\triangle x_-)_i (\triangle s_-)_i)^2}{x_i s_i} \right) \\
&\leq \alpha_1^2 \left(\frac{1}{4} \sum_{i \in \mathcal{I}_+} (\triangle x_-)_i (\triangle s_-)_i - \sum_{i \in \mathcal{I}_-} (\triangle x_-)_i (\triangle s_-)_i \right) \\
&\leq \alpha_1^2 \left(\frac{1}{16} n\mu + \frac{1}{4} n\mu \right) \\
&\leq \frac{5}{32}\alpha_2^2 \beta\tau\mu. \qquad (2.11)
\end{aligned}
$$

Therefore

$$
\|(\triangle x(\alpha) \triangle s(\alpha))^+\|_1 \leq \frac{1}{4}\left(\sqrt{\frac{3}{2}\alpha_2^2\beta\tau\mu} + \sqrt{\frac{5}{32}\alpha_2^2\beta\tau\mu} \right)^2 \leq \frac{2}{3}\alpha_2^2\beta\tau\mu.
$$

The proof is complete. □

By Lemma 2.1, if $\tau\mu - x_i s_i \leq 0$, then it holds that

$$
\begin{aligned}
h(\alpha)_i &= x_i s_i + \alpha_1(\tau\mu - x_i s_i) - \alpha_1\alpha_a (\triangle x_-)_i (\triangle s_-)_i \\
&\geq x_i s_i + \alpha_1 \tau\mu - \alpha_1 x_i s_i - \frac{1}{4}\alpha_1\alpha_a x_i s_i \\
&\geq \left(1 - \frac{5}{4}\alpha_1\right) x_i s_i + \alpha_1 \tau\mu, \qquad (2.12)
\end{aligned}
$$

and if $\tau\mu - x_i s_i > 0$, then

$$
\begin{aligned}
h(\alpha)_i &= x_i s_i + \alpha_2(\tau\mu - x_i s_i) - \alpha_1\alpha_a (\triangle x_-)_i (\triangle s_-)_i \\
&\geq x_i s_i + \alpha_2 \tau\mu - \alpha_2 x_i s_i - \frac{1}{4}\alpha_1\alpha_a x_i s_i \\
&\geq \left(1 - \frac{1}{4}\alpha_1 - \alpha_2\right) x_i s_i + \alpha_2 \tau\mu. \qquad (2.13)
\end{aligned}
$$

Since $(x, s) \in \mathcal{N}(\tau, \beta)$, by (2.9), (2.10) and Lemma 2.4 we have

$$
\begin{aligned}
\mu(\alpha) &= (x + \triangle x(\alpha))^T (s + \triangle s(\alpha))/n \\
&= x^T s/n + \alpha_1 e^T (\tau\mu e - xs)^-/n + \alpha_2 e^T (\tau\mu e - xs)^+/n \\
&\quad - \alpha_1\alpha_a \triangle x_-^T \triangle s_-/n + \triangle x(\alpha)^T \triangle s(\alpha)/n
\end{aligned}
$$

$$\begin{aligned}
&\le \mu - \alpha_1(1-\tau)\mu + \alpha_2\|(\tau\mu e - xs)^+\|/\sqrt{n} + \|(\triangle x(\alpha)\triangle s(\alpha))^+\|_1/n \\
&\le \mu - \alpha_1(1-\tau)\mu + \alpha_2\beta\tau\mu/\sqrt{n} + 2\alpha_2^2\beta\tau\mu/(3n) \\
&\le \mu - \alpha_1(1-\tau)\mu + 5\alpha_2\beta\tau\mu/(3\sqrt{n}). \qquad (2.14)
\end{aligned}$$

By above result and Lemma 2.2 we have

$$\begin{aligned}
\mu(\alpha) &\ge \mu + \alpha_1 e^T(\tau\mu e - xs)^-/n - \alpha_1\alpha_a\triangle x_-^T\triangle s_-/n \\
&\ge \mu - \alpha_1 x^T s/n - \frac{1}{n}\alpha_1\alpha_a\sum_{i\in\mathcal{I}_+}(\triangle x_-)_i(\triangle s_-)_i \\
&\ge \mu - \alpha_1\mu - \frac{1}{4}\alpha_1\alpha_a\mu \\
&\ge \left(1 - \frac{5}{4}\alpha_1\right)\mu. \qquad (2.15)
\end{aligned}$$

Lemma 2.5 *Suppose that the current iterate* $(x, s) \in \mathcal{N}(\tau, \beta)$, $\tau \le 1/8$, *and* $\beta \le 1/2$. *If* $\alpha_1 = \alpha_2\sqrt{\frac{\beta\tau}{2n}}$, *then we have* $\|(\tau\mu(\alpha)e - h(\alpha))^+\| \le (1-\alpha_2)\beta\tau\mu$.

Proof If $\tau\mu - x_i s_i \le 0$, by (2.12) and (2.14) we have

$$\begin{aligned}
&\tau\mu(\alpha) - h(\alpha)_i \\
&\quad\le \tau(\mu - \alpha_1(1-\tau)\mu + 5\alpha_2\beta\tau\mu/(3\sqrt{n})) - \left(1 - \frac{5}{4}\alpha_1\right)x_i s_i - \alpha_1\tau\mu \\
&\quad\le \tau(\mu - \alpha_1(1-\tau)\mu + 5\alpha_2\beta\tau\mu/(3\sqrt{n})) - \left(1 - \frac{5}{4}\alpha_1\right)\tau\mu - \alpha_1\tau\mu \\
&\quad= -\frac{3}{4}\alpha_1\tau\mu + \alpha_1\tau_1^2\mu + 5\alpha_2\beta\tau_1^2\mu/(3\sqrt{n}) \\
&\quad= \left(-\frac{3}{4} + \tau + \frac{5}{3}\sqrt{2\beta\tau}\right)\alpha_1\tau\mu \\
&\quad\le 0,
\end{aligned}$$

and if $\tau\mu - x_i s_i > 0$, by (2.13) and above result we have

$$\begin{aligned}
&\tau\mu(\alpha) - h(\alpha)_i \\
&\quad\le \tau(\mu - \alpha_1(1-\tau)\mu + 5\alpha_2\beta\tau\mu/(3\sqrt{n})) - \left(1 - \frac{1}{4}\alpha_1 - \alpha_2\right)x_i s_i - \alpha_2\tau\mu \\
&\quad= \left(1 - \frac{1}{4}\alpha_1 - \alpha_2\right)(\tau\mu - x_i s_i) - \frac{3}{4}\alpha_1\tau\mu + \alpha_1\tau_1^2\mu + 5\alpha_2\beta\tau_1^2\mu/(3\sqrt{n}) \\
&\quad\le \left(1 - \frac{1}{4}\alpha_1 - \alpha_2\right)(\tau\mu - x_i s_i) \\
&\quad\le (1-\alpha_2)(\tau\mu - x_i s_i),
\end{aligned}$$

which implies that

$$\|(\tau\mu(\alpha)e - h(\alpha))^+\| \le (1-\alpha_2)\|(\tau\mu e - xs)^+\| \le (1-\alpha_2)\beta\tau\mu.$$

The proof is complete. □

Lemma 2.6 *Suppose that the current iterate* $(x, s) \in \mathcal{N}(\tau, \beta)$, $\tau \le 1/8$, *and* $\beta \le 1/2$. *If* $\alpha_1 = \alpha_2\sqrt{\frac{\beta\tau}{2n}}$, *then we have* $\|(\tau\mu(\alpha)e - x(\alpha)s(\alpha))^+\| \le \beta\tau\mu(\alpha)$.

Proof Using Lemmas 2.4 , Lemmas 2.5 and (2.15) we obtain

$$\begin{aligned}
\|(\tau\mu(\alpha)e - x(\alpha)s(\alpha))^+\| &= \|(\tau\mu(\alpha)e - h(\alpha) - \triangle x(\alpha)\triangle s(\alpha))^+\| \\
&\le \|(\tau\mu(\alpha)e - h(\alpha))^+\| + \|(-\triangle x(\alpha)\triangle s(\alpha))^+\| \\
&\le (1-\alpha_2)\beta\tau\mu + \frac{2}{3}\alpha_2^2\beta\tau\mu \\
&= \left(1 - \frac{5}{4}\alpha_1\right)\beta\tau\mu + \left(\frac{5}{4}\alpha_1 - \frac{1}{3}\alpha_2\right)\beta\tau\mu \\
&\le \left(1 - \frac{5}{4}\alpha_1\right)\beta\tau\mu \\
&\le \beta\tau\mu(\alpha).
\end{aligned}$$

The proof is complete. □

Lemma 2.7 *Suppose that* $(x, s) \in \mathcal{N}(\tau, \beta)$. *If* $x(\alpha)s(\alpha) > 0$ *for some* $\alpha \in [0, 1]^2$, *then* $(x(\alpha), s(\alpha)) > 0$.

Proof For all $t \in [0, 1]$, we have

$$\begin{aligned}
x(t\alpha)s(t\alpha) &= (x + \triangle x(t\alpha))(s + \triangle s(t\alpha)) \\
&= xs + t(s\triangle x(\alpha) + x\triangle s(\alpha)) + t^2\triangle x(\alpha)\triangle s(\alpha) \\
&= (1-t^2)xs + (t-t^2)(s\triangle x(\alpha) + x\triangle s(\alpha)) + t^2 x(\alpha)s(\alpha) \\
&= (1-t^2)xs + (t-t^2)(\alpha_1(\tau\mu e - xs)^- + \alpha_2(\tau\mu e - xs)^+ \\
&\quad - \alpha_1\alpha_a(\triangle x_-\triangle s_-)) + t^2 x(\alpha)s(\alpha) \\
&\ge (1-t^2)xs + (t-t^2)\left(-\alpha_1 xs - \frac{\alpha_1\alpha_a}{4}xs\right) + t^2 x(\alpha)s(\alpha) \\
&\ge (1-t^2)xs - \frac{5}{4}(t-t^2)xs + t^2 x(\alpha)s(\alpha) \\
&= (1-t)(1-t/4)xs + t^2 x(\alpha)s(\alpha) \\
&> 0.
\end{aligned}$$

By using continuity, it follows that $(x(\alpha), s(\alpha)) > 0$, since $(x, s) > 0$. □

By Lemma 2.6 and Lemma 2.7, we get a sufficient condition to keep all the iterates in the neighborhood $\mathcal{N}(\tau, \beta)$.

Lemma 2.8 *Suppose that the current iterate* $(x, s) \in \mathcal{N}(\tau, \beta)$, $\tau \le 1/8$, *and* $\beta \le 1/2$. *If* $\alpha_1 = \alpha_2\sqrt{\frac{\beta\tau}{2n}}$, *then* $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau, \beta)$.

3 The iteration bound and an implementation

The following key result shows that the objective function in the optimization problem with two variables (2.8) is monotone with respect to α_1 for any fixed α_2.

Theorem 3.1 *Suppose* $(x, s) \in \mathcal{N}(\tau, \beta)$, $\tau \le 1/16$, *and* $\beta \le 1/5$. *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

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

Therefore, by (2.2) and (2.5) we have

$$\begin{aligned} &\frac{\partial (x(\alpha)^T s(\alpha))}{\partial \alpha_1} \\ &\quad = 2\alpha_1 (\triangle x_- + \triangle x_-^c)^T (\triangle s_- + \triangle s_-^c) \\ &\qquad + (x + \alpha_2 \triangle x_+)^T (\triangle s_- + \triangle s_-^c) + (s + \alpha_2 \triangle s_+)^T (\triangle x_- + \triangle x_-^c) \\ &\quad = 2\alpha_1 (\triangle x_- + \triangle x_-^c)^T (\triangle s_- + \triangle s_-^c) + x^T (\triangle s_- + \triangle s_-^c) \\ &\qquad + s^T (\triangle x_- + \triangle x_-^c) + \alpha_2((\triangle x_+)^T (\triangle s_- + \triangle s_-^c) + (\triangle s_+)^T (\triangle x_- + \triangle x_-^c)) \\ &\quad \le 2\alpha_1 \|((\triangle x_- + \triangle x_-^c)(\triangle s_- + \triangle s_-^c))^+\|_1 + e^T (\tau \mu e - xs)^- \\ &\qquad - \alpha_a \triangle x_-^T \triangle s_- + \alpha_2((\triangle x_+)^T (\triangle s_- + \triangle s_-^c) + (\triangle s_+)^T (\triangle x_- + \triangle x_-^c)) \\ &\quad \le 2\|((\triangle x_- + \triangle x_-^c)(\triangle s_- + \triangle s_-^c))^+\|_1 - e^T (xs - \tau \mu e)^+ \\ &\qquad + |(\triangle x_+)^T (\triangle s_- + \triangle s_-^c) + (\triangle s_+)^T (\triangle x_- + \triangle x_-^c)|. \end{aligned} \tag{3.1}$$

Denote

$$D = X^{0.5} S^{-0.5}, \quad u = D^{-1}(\triangle x_- + \triangle x_-^c), \quad v = D(\triangle s_- + \triangle s_-^c)$$

and

$$r = (xs)^{-0.5} (\tau \mu e - xs)^- - \alpha_a (xs)^{-0.5} (\triangle x_- \triangle s_-).$$

So by (2.2) and (2.5) we have $u^T v \ge 0$ and $u + v = r$. Therefore, by Lemma 2.3 we have

$$\|((\triangle x_- + \triangle x_-^c)(\triangle s_- + \triangle s_-^c))^+\|_1 = \|(uv)^+\|_1 \le \frac{1}{4}\|r\|^2. \tag{3.2}$$

By Lemma 2.1, Lemma 2.2 and (2.11) we have

$$\begin{aligned} \|u + v\|^2 &= \|r\|^2 \\ &= \|(xs)^{-0.5}(\tau \mu e - xs)^- - \alpha_a (xs)^{-0.5}(\triangle x_- \triangle s_-)\|^2 \end{aligned}$$

$$
\begin{aligned}
&= \|(xs)^{-0.5}(\tau\mu e - xs)^-\|^2 + \alpha_a^2\|(xs)^{-0.5}(\triangle x_-\triangle s_-)\|^2\\
&\quad - 2\alpha_a e^T((\tau\mu e - xs)^-(\triangle x_-\triangle s_-)/(xs))\\
&= \|(xs-\tau\mu e)^+/\sqrt{xs}\|^2 + \alpha_a^2\|(xs)^{-0.5}(\triangle x_-\triangle s_-)\|^2\\
&\quad + 2\alpha_a e^T((xs-\tau\mu e)^+(\triangle x_-\triangle s_-)/(xs))\\
&\le \|\sqrt{(xs-\tau\mu e)^+}\|^2 + \left(\frac{1}{16}n\mu + \frac{1}{4}n\mu\right) + \frac{1}{2}e^T(xs-\tau\mu e)^+\\
&= \frac{5}{16}n\mu + \frac{3}{2}e^T(xs-\tau\mu e)^+.
\end{aligned}\tag{3.3}
$$

Since $(x,s)\in\mathcal{N}(\tau,\beta)$,

$$
\|D^{-1}\triangle x_+ + D\triangle s_+\|^2 = \|(xs)^{-0.5}(\tau\mu e - xs)^+\|^2 \le \frac{(\beta\tau)^2\mu}{(1-\beta)\tau}. \tag{3.4}
$$

By (3.3), (3.4) and the monotonicity of M, we have

$$
\begin{aligned}
&|(\triangle x_+)^T(\triangle s_- + \triangle s_-^c) + (\triangle s_+)^T(\triangle x_- + \triangle x_-^c)|\\
&\quad = |(D^{-1}\triangle x_+)^T v + (D\triangle s_+)^T u|\\
&\quad \le \|(D^{-1}\triangle x_+, D\triangle s_+)\|\|(u,v)\|\\
&\quad \le \|D^{-1}\triangle x_+ + D\triangle s_+\|\|u+v\|\\
&\quad \le \sqrt{\frac{(\beta\tau)^2\mu}{(1-\beta)\tau}\left(\frac{5}{16}n\mu + \frac{3}{2}e^T(xs-\tau\mu e)^+\right)}\\
&\quad \le \sqrt{\frac{29n}{16(1-\beta)\tau}}\beta\tau\mu.
\end{aligned}\tag{3.5}
$$

Substituting (3.2), (3.3) and (3.5)into (3.1) finally yields that

$$
\begin{aligned}
\frac{\partial(x(\alpha)^T s(\alpha))}{\partial\alpha_1} &\le \frac{5}{32}n\mu - \frac{1}{4}e^T(xs-\tau\mu e)^+ + \sqrt{\frac{29n}{16(1-\beta)\tau}}\beta\tau\mu\\
&\le \frac{5}{32}n\mu - \frac{1}{4}(1-\tau)n\mu + \sqrt{\frac{29n}{16(1-\beta)\tau}}\beta\tau\mu\\
&= \frac{1}{4}n\mu\left(-\frac{3}{8} + \tau + \sqrt{\frac{29\beta^2\tau}{(1-\beta)n}}\right)\\
&< 0.
\end{aligned}\tag{3.6}
$$

The proof is complete. □

In view of Theorem 3.1, we may solve subproblem (2.8) approximately in the following way. First, set $\alpha_2 = \theta \in (0,1]$. Second, find the greatest $\alpha_1 \in [0,1]$

such that $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau, \beta)$. Then, we have $\alpha_1 \geq \theta\sqrt{\beta\tau/(2n)}$ by Lemma 2.8. Moreover, from (2.14) we also have

$$\begin{aligned}
\mu(\alpha) &\leq \left(1 - \alpha_1(1-\tau) + 5\alpha_2\beta\tau/(3\sqrt{n})\right)\mu \\
&\leq \left(1 - \theta\left(1 - \tau - \frac{5}{3}\sqrt{2\beta\tau}\right)\sqrt{\beta\tau/(2n)}\right)\mu \\
&\leq \left(1 - \theta\left(1 - \frac{1}{8} - \frac{5\sqrt{2}}{12}\right)\sqrt{\beta\tau/(2n)}\right)\mu \\
&\leq \left(1 - \frac{\theta\sqrt{\beta\tau}}{4\sqrt{2n}}\right)\mu.
\end{aligned}$$

Thus, at each iteration, we have

$$(x^k)^T s^k \leq \left(1 - \frac{\theta\sqrt{\beta\tau}}{4\sqrt{2n}}\right)^k (x^0)^T s^0,$$

which implies that $(x^k)^T s^k \leq \varepsilon$ when $k \geq \frac{4\sqrt{2n}}{\theta\sqrt{\beta\tau}} \log \frac{(x^0)^T s^0}{\varepsilon}$.

It is clear that if the plane-search procedure (2.8) is replaced by this line search procedure in Algorithm 2.1, the $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iteration bound holds. At the same time, the exact plane-search (2.8) would lead to at least the same amount of reduction of the duality gap, and hence the iteration bound still holds. Now it is easy to present the complexity result of the algorithm.

Theorem 3.2 *Algorithm 2.1 will terminate in $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iterations with a solution $x^T s \leq \varepsilon$.*

4 Numerical results

In order to get a feel of how the method might perform in practice, we shall test our algorithms on some randomly generated instances. We have implemented the three algorithms (Algorithm 2.1, Ai and Zhang's path-following algorithm and predictor-corrector algorithm in [1]) in MATLAB on an Intel Core 2 PC (1.86 Ghz) with 1 GB RAM, and the results are summarized in Tables 1 and 2. These three algorithms will be denoted, respectively, by New-MPC, Ai–Zhang's PF and Ai–Zhang's PC. We used test problems generated as in [1]. The parameters were chosen as follows: $\alpha_2 = 1$, $\tau = 0.001$ and $\beta = 1/2$. We use bisection in closed interval $[\sqrt{\beta\tau/(2n)}, 1]$ to find the greatest α_1 such that $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau, \beta)$. All algorithms terminate after the relative duality gap satisfies

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

We first test the monotone LCPs which are generated as follows: A = rand(n), M = A' * A, q = ones(n,1) − M * ones(n,1). Then the monotone LCP has an initial feasible point (e, e). Table 1 lists the average number of iterations (iter) and

Table 1 Numerical results on monotone LCPs with $M = A^T A$

n	Ai–Zhang's PF		Ai–Zhang's PC		New-MPC	
	Iter	Time	Iter	Time	Iter	Time
100	9.7	0.0092	11.9	0.0106	7.5	0.0081
200	10.9	0.0366	12.8	0.0423	7.9	0.0270
300	10.5	0.1055	12.9	0.1304	7.9	0.0831
400	11.1	0.2279	13.4	0.2709	8.3	0.1795
500	10.9	0.4007	13.1	0.4793	8.0	0.3063
600	11.2	0.6670	14.0	0.8256	8.1	0.5148
700	11.6	1.0414	13.8	1.2202	8.4	0.7868
800	12.4	1.5841	14.5	1.8201	8.4	1.1159
900	12.5	2.1878	14.1	2.4264	8.4	1.5269
1000	12.0	2.7687	14.7	3.3216	8.3	1.9732

Table 2 Numerical results on monotone LCPs with $M = A^T A + (B - B^T)$

n	Ai–Zhang's PF		Ai–Zhang's PC		New-MPC	
	Iter	Time	Iter	Time	Iter	Time
100	4.0	0.0032	4.0	0.0048	3.7	0.0030
200	4.0	0.0077	4.0	0.0078	3.0	0.0078
300	4.0	0.0280	3.0	0.0233	3.0	0.0219
400	4.0	0.0626	3.0	0.0486	3.0	0.0545
500	3.9	0.1203	3.0	0.0908	3.0	0.0890
600	3.4	0.1717	3.0	0.1421	3.0	0.1550
700	3.0	0.2452	3.0	0.2188	3.0	0.2439
800	3.0	0.3485	3.0	0.3439	3.0	0.3392
900	3.0	0.4844	3.0	0.5204	3.0	0.5030
1000	3.0	0.6343	3.0	0.6609	3.0	0.6719

the average CPU time (time) per iteration of ten randomly generated problems with the same n. To test the influence from the skewness of matrix M, we also test the LCPs generated as follows: A = rand(n), B = rand(n), M = A' * A + (B − B'), q = ones(n,1) − M * ones(n,1). The numerical results are shown 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.

Our preliminary implementations show that this algorithm is promising. From the results presented in Table 1, we see that, in the number of iterations and the computational time, the proposed new algorithm saves about 28.19 and 27.77 %, respectively, compared to Ai–Zhang's path-following algorithm, and about 40.09 and 38.17 %, respectively, compared to Ai–Zhang's predictor-corrector algorithm.

5 Conclusions

The current paper aims at modifying Ai and Zhang's primal-dual path-following interior-point method [1] to gain a class of Mehrotra-type predictor-corrector algorithm for monotone LCPs. We have shown that the use of the corrector step does not cause any loss in the worst-case complexity of the algorithm. We prove that the proposed generic method can be specified into an easy implementable variant with given parameters and the iteration bound is still $O(\sqrt{n}L)$. Our preliminary implementation also provides us an encouraging evidence that the new algorithm may also perform well in practice. Certainly, our implementations are very coarse. More refined linear algebras and efficient strategies to choose step lengths deserve more work for real applications of the algorithm. A theoretical direction for the future research is generalization the algorithm to the $P_*(\kappa)$ LCPs or general conic optimization.

Acknowledgments This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 11471102 and 11426091) and the Natural Science Foundation of Henan University of Science and Technology (Grant No. 2014QN039).

References

1. Ai, W., Zhang, S.: An $O(\sqrt{n}L)$ iteration primal-dual path-following method, based on wide neighborhoods and large updates, for monotone LCP. SIAM J. Optim. **16**, 400–417 (2005)
2. Andersen, E.D., Ye, Y.: On a homogeneous algorithm for the monotone complementarity problem. Math. Program. **84**, 375–399 (1999)
3. Cartis, C.: Some disadvantages of a Mehrotra-tpye primal-dual corrector interior point algorithm for linear programming. Appl. Numer. Math. **59**, 1110–1119 (2009)
4. Cartis, C.: On the convergence of a primal-dual second-order corrector interior point algorithm for linear programming. Technical Report, Numerical Analysis Group, Computing Laboratory, Oxford University (2005). http://www.optimization-online.org/DB_HTML/2005/03/1097.html. Accessed April 2005
5. Colombo, M., Gondzio, J.: Further development of multiple centrality correctors for interior point methods. Comput. Optim. Appl. **41**, 277–305 (2008)
6. Kojima, M., Mizuno, S., Yoshise, A.: A primal-dual interior point algorithm for linear programming. In: Megiddo, N. (ed.) Progress in Mathematical Programming: Interior Point and Related Methods, pp. 29–47. Springer, New York (1989)
7. Kojima, M., Mizuno, S., Yoshise, A.: A polynomial-time algorithm for a class of linear complementarity problems. Math. Program. **44**, 1–26 (1989)
8. Liu, C., Liu, H., Cong, W.: An $O(\sqrt{n}L)$ iteration primal-dual second-order corrector algorithm for linear programming. Optim. Lett. **5**, 729–743 (2011)
9. Megiddo, N.: Pathways to the optimal set in linear programming. In: Megiddo, N. (ed.) Progress in Mathematical Programming: Interior Point and Related Methods, pp. 131–158. Springer, New York (1989)
10. Mehrotra, S.: On the implementation of a primal-dual interior point method. SIAM J. Optim. **2**, 575–601 (1992)
11. Mizuno, S., Todd, M.J., Ye, Y.: On adaptive step primal-dual interior-point algorithms for linear programming. Math. Oper. Res. **18**, 964–981 (1993)
12. Salahi, M., Mahdavi-Amiri, N.: Polynomial time second order Mehrotra-type predictor-corrector algorithms. Appl. Math. Comput. **183**, 646–658 (2006)
13. Salahi, M., Peng, J., Terlaky, T.: On Mehrotra-type predictor-corrector algorithms. SIAM J. Optim. **18**, 1377–1397 (2007)
14. Ye, Y.: On homogeneous and self-dual algorithms for LCP. Math. Program. **76**, 211–221 (1996)
15. Zhang, Y., Zhang, D.T.: On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms. Math. Program. **68**, 303–318 (1995)