Optim Lett (2011) 5:729–743
DOI 10.1007/s11590-010-0242-6

ORIGINAL PAPER

An $O(\sqrt{n}L)$ iteration primal-dual second-order corrector algorithm for linear programming

Changhe Liu · Hongwei Liu · Weijie Cong

Received: 4 December 2009 / Accepted: 21 September 2010 / Published online: 8 October 2010

Abstract In this paper, we propose a primal-dual second-order corrector interior point algorithm for linear programming problems. At each iteration, the method computes a corrector direction in addition to the Ai–Zhang direction [Ai and Zhang in SIAM J Optim 16:400–417 (2005)], in an attempt to improve performance. The corrector is multiplied by the square of the step-size in the expression of the new iterate. We prove that the use of the corrector step does not cause any loss in the worst-case complexity of the algorithm. To our best knowledge, this is the first wide neighborhood second-order corrector algorithm enjoyed the low iteration bound of $O(\sqrt{n}L)$, the same as the best known complexity results for interior point methods.

Keywords Linear programming · Interior-point methods · Second-order methods · Polynomial complexity

1 Introduction

In the past 20 years, interior point methods (IPMs) have become highly successful in solving linear programming (LP), especially large-scale ones, while enjoying good theoretical convergence and complexity properties [6,8,13,16,17,22]. In addition, some recent related articles can be found in [5]. Examples of IPMs that are reliable

C. Liu · H. Liu (✉) · W. Cong
Department of Applied Mathematics, Xidian University, Xi'an, China
e-mail: hwliu@mail.xidian.edu.cn

W. Cong
e-mail: cwjhfl@126.com

C. Liu
Department of Applied Mathematics, Henan University of Science and Technology, Luoyang, China
e-mail: changheliu@163.com

both in theory and in practice include the primal-dual path-following method of Kojima et al. [10] with some long-step linesearch procedure [22], and an infeasible formulation of this algorithm [9]. The majority of commercial and public IPM codes implement a variant of the latter, Mehrotra's predictor–corrector (MPC) algorithm [11,12], and some of them employ in addition, Gondzio's higher-order corrections [7]. Since its first implementations and testing on the standard set of LP test problems (the Netlib test set [14]), the MPC algorithm proved to be, especially on large-scale problems, much faster than the infeasible primal-dual path-following algorithm, in terms of both the number of iterations and the computational time. Its past and present practical successes, however, have not been enhanced by equally praiseworthy theoretical guarantees of good performance: no global convergence or polynomial complexity results are known for this method. It is, in fact, acknowledged among practitioners that there are examples on which the MPC algorithm fails to converge (see [15], page 407). Simple safeguards could be incorporated into the method to force it into the convergence framework of existing methods [19,20]. In [21] the authors proposed a different approach, which is based on postponing the choice of the barrier parameter. Most recently, in [3] the author provided an example that shows that MPC 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. Therefore, in [4,19,24] the corrector is multiplied by the square of the step-size in the expression of the new iterate. Moreover, in [4] the author showed that the quadratic expression of each new iterate can be interpreted as a second order Taylor approximation at the current iterate of a local path from the current iterate to a target point on the primal-dual central path of the problems.Therefore, the corrector director provides curvature information of this path and posits the new iterate closer to the primal-dial central path.

Recently, based on Ai's original idea [1], an important result was given by Ai and Zhang [2] for monotone linear complementarity problems (LCPs). Their algorithm uses a wide neighborhood and decomposes the classical Newton direction into two orthogonal directions using different step-length for each of them. They proved that the algorithm stops after at most $O(\sqrt{n}L)$ iterations, where n is the number of variables and L is the input data length. This result yields the first wide neighborhood path-following algorithm having the same theoretical complexity as a small neighborhood path-following algorithm for monotone LCPs, which include LP as a special case. The current paper aims at modifying the Ai–Zhang's primal-dual interior point method to gain a class of second order MPC algorithm with $O(\sqrt{n}L)$ iteration bound for linear programming problems.

The rest of the paper is organized as follows. In Sect. 2 we recall the key features of Ai–Zhang's primal-dual path-following method. In Sect. 3 we present a new class of primal-dual second order corrector algorithm and establish its worst case iteration complexity. Some numerical results are provided in Sect. 4. Finally, we finish the paper with some conclusions and considerations about future work.

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 the Euclidean norm $\|\cdot\|_2$. We also use the notations $\mathcal{I} =: \{1, 2, \ldots, n\}$.

2 Ai–Zhang's primal-dual path-following method

Before going in the details of the algorithm, we briefly review the basics and unique results of IPMs.

We consider primal-dual IPMs for solving the following LP problem

$$(P) \quad \min c^T x, \quad s.t.\ Ax = b, \quad x \geq 0, \tag{2.1}$$

where $A \in R^{m\times n}$ satisfies $rank(A) = m$, $c \in R^n$, $b \in R^m$ and its dual problem

$$(D) \quad \max b^T y, \quad s.t.\ A^T y + s = c, \quad s \geq 0. \tag{2.2}$$

It is common in IPMs theory to assume that both (P) and (D) satisfy the interior point condition (IPC) [18], i.e., there exists an (x^0, y^0, s^0) such that

$$Ax^0 = b, \quad A^T y^0 + s^0 = c, \quad x^0 > 0, \quad s^0 > 0.$$

Note

$$\mathcal{F}^0 := \{(x, y, s) : Ax = b, A^T y + s = c, (x, s) > 0\}.$$

Finding optimal solutions of (P) and (D) is equivalent to solving the following system:

$$\begin{cases} Ax = b, & x \geq 0, \\ A^T y + s = c, & s \geq 0, \\ xs = 0. \end{cases} \tag{2.3}$$

The basic idea of primal-dual IPMs is to replace the third equation in (2.3) by the parameterized equation $xs = \mu e$, where e is the all one vector. This leads to the following system:

$$\begin{cases} Ax = b, & x \geq 0, \\ A^T y + s = c, & s \geq 0, \\ xs = \mu e. \end{cases} \tag{2.4}$$

If the IPC holds, then for each $\mu > 0$, system (2.4) has a unique solution. This solution, denoted by $(x(\mu), y(\mu), s(\mu))$, is called the μ-center of the primal-dual pair (P) and

(D). The set of μ-centers with all $\mu > 0$ gives the central path of (P) and (D), i.e.,

$$\mathcal{C} := \{(x, y, s) \in \mathcal{F}^0 : xs = \mu e\}.$$

It has been shown that the limit of the central path (as μ goes to zero) exists and it is a optimal solution of (P) and (D) [18].

Applying Newton's method to (2.4) for a given feasible point (x, y, s) gives the following linear system of equations

$$\begin{cases} A\triangle x = 0, \\ A^T\triangle y + \triangle s = 0, \\ s\triangle x + x\triangle s = \mu e - xs, \end{cases} \tag{2.5}$$

where $(\triangle x, \triangle y, \triangle s)$ 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.

One of the popular neighborhood is the so-called small neighborhood, defined as

$$\mathcal{N}_2(\beta) := \{(x, y, s) \in \mathcal{F}^0 : \|xs - \mu e\| \le \beta\mu\},$$

where $\beta \in (0, 1)$ is a given constant and $\mu = x^T s/n$ is associated with the actual duality gap. Another popular alternative is called negative infinity neighborhood that is a wide neighborhood, defined as

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

where $\tau \in (0, 1)$.

Ai and Zhang [1,2] have introduced a new neighborhood for the central path, defined as

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

where $0 < \tau_2 < \tau_1 < 1$ are two parameters.

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

$$\mathcal{N}_2(1 - \tau_1) \subseteq \mathcal{N}_\infty^-(1 - \tau_1) \subseteq \mathcal{N}(\tau_1, \tau_2) \subseteq \mathcal{N}_\infty^-(1 - \tau_2),\ \forall\, 0 < \tau_2 < \tau_1 < 1. \tag{2.6}$$

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

$$\begin{cases} A\triangle x_- = 0, \\ A^T\triangle y_- + \triangle s_- = 0, \\ s\triangle x_- + x\triangle s_- = (\tau\mu e - xs)^- \end{cases} \tag{2.7}$$

and

$$\begin{cases} A\triangle x_+ = 0, \\ A^T\triangle y_+ + \triangle s_+ = 0, \\ s\triangle x_+ + x\triangle s_+ = (\tau\mu e - xs)^+. \end{cases} \tag{2.8}$$

Then,they treat these two directions separately and let $\alpha := (\alpha_1, \alpha_2) \in R^2_+$ be the step-sizes taken along $(\triangle x_-, \triangle y_-, \triangle s_-)$ and $(\triangle x_+, \triangle y_+, \triangle s_+)$ respectively. The new step is

$$(x(\alpha), y(\alpha), s(\alpha)) := (x, y, s) + \alpha_1(\triangle x_-, \triangle y_-, \triangle s_-) + \alpha_2(\triangle x_+, \triangle y_+, \triangle s_+). \tag{2.9}$$

The best α can be obtained by the plane-search procedure, i.e., solving the following two-dimensional optimization problem:

$$\begin{aligned} &\min\ x(\alpha)^T s(\alpha) \\ &s.t.\ \ (x(\alpha), y(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2), \\ &\qquad 0 \le \alpha_1, \alpha_2 \le 1. \end{aligned} \tag{2.10}$$

Ai–Zhang's wide-neighborhood and large update primal-dual path-following method can be outlined as follows.

Algorithm 2.1 (*Ai-Zhang's primal-dual path-following method*)

Input parameters: an accuracy parameter $\varepsilon > 0$, neighborhood parameters $0 < \tau_2 < \tau_1 < 1$, a centering parameter $0 \le \tau \le 1$ and an initial point $(x^0, y^0, s^0) \in \mathcal{N}(\tau_1, \tau_2)$, $\mu_0 = (x^0)^T s^0/n$.

Step0 Set $k := 0$.

Step1 If $(x^k)^T s^k \le \varepsilon$, then stop.

Step2 Compute the directions $(\triangle x^k_-, \triangle y^k_-, \triangle s^k_-)$ by (2.7) and $(\triangle x^k_+, \triangle y^k_+, \triangle s^k_+)$ by (2.8). Find a step length vector $\alpha^k = (\alpha^k_1, \alpha^k_2) > 0$ giving a sufficient reduction of the duality gap and assuring $(x(\alpha^k), y(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1, \tau_2)$.

Step3 Let $(x^{k+1}, y^{k+1}, s^{k+1}) := (x(\alpha^k), y(\alpha^k), s(\alpha^k))$ and $\mu_{k+1} = (x^{k+1})^T s^{k+1}/n$. Set $k := k + 1$ and go to Step 1.

After proving some technical lemmas, the authors show that the above method can be specified into easy implementable variants with given parameters, and has an iteration bound of $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$. Moreover, the authors implement the method in the predictor–corrector style. In that case, in addition to the above iteration bound they also obtains a quadratic convergence rate for the predictor steps, provided that a strict complementary solution exists. Preliminary numerical experiments show that the performance of the two algorithms are promising.

3 Primal-dual second-order corrector algorithm

In this section, we present a new primal-dual second-order corrector algorithm for liner programming. We use the information from (2.7) to compute the corrector direction given by

$$\begin{cases} A\triangle x_-^c = 0, \\ A^T\triangle y_-^c + \triangle s_-^c = 0, \\ s\triangle x_-^c + x\triangle s_-^c = -\triangle x_-\triangle s_-. \end{cases} \tag{3.1}$$

Finally, the new iterate is

$$\begin{aligned} &(x(\alpha), y(\alpha), s(\alpha)) \\ &\quad := (x, y, s) + (\triangle x(\alpha), \triangle y(\alpha), \triangle s(\alpha)) \\ &\quad := (x, y, s) + \alpha_1(\triangle x_-, \triangle y_-, \triangle s_-) + \alpha_1^2(\triangle x_-^c, \triangle y_-^c, \triangle s_-^c) \\ &\qquad +\alpha_2(\triangle x_+, \triangle y_+, \triangle s_+). \end{aligned} \tag{3.2}$$

The method in this paper is similar to the methods as in [4,24]. However, instead of trying to correct all Newton direction, we correct only the "bad" components. Namely, we only try to correct the direction $(\triangle x_-, \triangle y_-, \triangle s_-)$ by (3.1); the direction $(\triangle x_+, \triangle y_+, \triangle s_+)$, along which the step-size α_2 can be always chosen one according to Ai and Zhang, is already reasonably good, and do not need to be changed.

Now, after all the previous discussions we may outline our new primal-dual second-order corrector algorithm as follows.

Algorithm 3.1 (*Primal-dual second-order corrector algorithm*)

Input parameters: an accuracy parameter $\varepsilon > 0$, neighborhood parameters $0 < \tau_2 < \tau_1 < 1$, a centering parameter $0 \le \tau \le 1$ and an initial point $(x^0, y^0, s^0) \in \mathcal{N}(\tau_1, \tau_2)$, $\mu_0 = (x^0)^T s^0/n$.

Step0 Set $k := 0$.
Step1 If $(x^k)^T s^k \le \varepsilon$, then stop.
Step2 Compute the directions $(\triangle x_-^k, \triangle y_-^k, \triangle s_-^k)$ by (2.7) and $(\triangle x_+^k, \triangle y_+^k, \triangle s_+^k)$ by (2.8).
Step3 Compute the directions $(\triangle x_-^{c,k}, \triangle y_-^{c,k}, \triangle s_-^{c,k})$ by (3.1). Find a step length vector $\alpha^k = (\alpha_1^k, \alpha_2^k) > 0$ giving a sufficient reduction of the duality gap and assuring $(x(\alpha^k), y(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1, \tau_2)$.
Step4 Let $(x^{k+1}, y^{k+1}, s^{k+1}) := (x(\alpha^k), y(\alpha^k), s(\alpha^k))$ and $\mu_{k+1} = (x^{k+1})^T s^{k+1}/n$. Set $k := k+1$ and go to Step 1.

In this paper, we prove that Algorithm 3.1 can be specified into easy implementable variants with given parameters, and has an iteration bound of $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$. In this section we choose to set $\tau = \tau_1$. First, we give some technical lemmas that will be used frequently during the analysis.

Lemma 3.1 *Let* $(x, y, s) \in \mathcal{F}^0$, $(\triangle x_-, \triangle y_-, \triangle s_-)$ *be the solution of* (2.7). *Note* $\mathcal{I}_+ = \{i \in \mathcal{I} : (\triangle x_-)_i(\triangle s_-)_i > 0\}$ *and* $\mathcal{I}_- = \mathcal{I} \setminus \mathcal{I}_+$. *Then*

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

Proof By Eq. (2.7) for $i \in \mathcal{I}_+$ we have

$$s_i(\triangle x_-)_i + x_i(\triangle s_-)_i = (\tau_1\mu - x_i s_i)^- \le 0.$$

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

$$2\sqrt{x_i s_i(\triangle x_-)_i(\triangle s_-)_i} \leq -s_i(\triangle x_-)_i - x_i(\triangle s_-)_i = (x_i s_i - \tau_1\mu)^+ \leq x_i s_i$$

we get

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

□

Lemma 3.2 *Let* $(x, y, s) \in \mathcal{F}^0$, $(\triangle x_-, \triangle y_-, \triangle s_-)$ *be the solution of* (2.7)*, then we have*

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

Proof Since $\triangle x_-{}^T \triangle s_- = 0$, the proof is a direct consequence of Lemma 3.1. □

The next lemma comes from Lemma 3.5 in [2].

Lemma 3.3 *Let* $u, v \in R^n$ *be such that* $u^T v \geq 0$*, and let* $r = u + v$*. Then, we have*

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

The next lemma is concerned with the feasibility of the iterates.

Lemma 3.4 *Suppose that* $(x, y, s) \in \mathcal{F}^0$ *and* $z + 2xs \geq 0$*. Let* $(\triangle x, \triangle y, \triangle s)$ *be the solution of*

$$\begin{cases} A\triangle x = 0, \\ A^T\triangle y + \triangle s = 0, \\ s\triangle x + x\triangle s = z. \end{cases}$$

If $(x + t_0\triangle x)(s + t_0\triangle s) > 0$ *for some* $0 < t_0 \leq 1$*, then* $(x + t\triangle x, s + t\triangle s) > 0$ *for all* $0 \leq t \leq t_0$.

Proof Let $(\bar{x}, \bar{s}) = (x + t_0\triangle x, s + t_0\triangle s)$. For all $\delta \in [0, 1]$, we have

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

By using continuity, it follows that $(x + t\triangle x, s + t\triangle s) > 0$ for all $0 \le t \le t_0$, since $(x, s) > 0$. □

In Algorithm 3.1, this property boils down to verifying $\alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+ - \alpha_1^2(\triangle x_- \triangle s_-) + 2xs \ge 0$. Let us denote

$$h(\alpha) := xs + \alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+ - \alpha_1^2(\triangle x_- \triangle s_-).$$

Since $(x, y, s) \in \mathcal{F}^0$, by Lemma 3.1, if $\tau_1\mu - x_i s_i \le 0$, then it holds that

$$\begin{aligned} h(\alpha)_i &= x_i s_i + \alpha_1(\tau_1\mu - x_i s_i) - \alpha_1^2(\triangle x_-)_i(\triangle s_-)_i \\ &\ge x_i s_i + \alpha_1\tau_1\mu - \alpha_1 x_i s_i - \frac{1}{4}\alpha_1^2 x_i s_i \\ &\ge \left(1 - \alpha_1 - \frac{1}{4}\alpha_1^2\right) x_i s_i + \alpha_1\tau_1\mu, \end{aligned} \tag{3.3}$$

and if $\tau_1\mu - x_i s_i > 0$, then

$$\begin{aligned} h(\alpha)_i &= x_i s_i + \alpha_2(\tau_1\mu - x_i s_i) - \alpha_1^2(\triangle x_-)_i(\triangle s_-)_i \\ &\ge x_i s_i + \alpha_2\tau_1\mu - \alpha_2 x_i s_i - \frac{1}{4}\alpha_1^2 x_i s_i \\ &\ge \left(1 - \alpha_2 - \frac{1}{4}\alpha_1^2\right) x_i s_i + \alpha_2\tau_1\mu. \end{aligned} \tag{3.4}$$

Therefore, for all $\alpha \in [0, 1]^2$, we have $h(\alpha) + xs \ge 0$. Lemma 3.4 thus asserts that $(x(\alpha), (y(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2)$ if and only if $\|(\tau_1\mu(\alpha)e - x(\alpha)s(\alpha))^+\| \le (\tau_1 - \tau_2)\mu(\alpha)$, where

$$\begin{aligned} \mu(\alpha) &:= (x + \triangle x(\alpha))^T(s + \triangle s(\alpha))/n \\ &= \mu + \alpha_1 e^T(\tau_1\mu e - xs)^-/n + \alpha_2 e^T(\tau_1\mu e - xs)^+/n. \end{aligned} \tag{3.5}$$

We note the following simple but useful relationships:

$$-x^T s \le e^T(\tau_1\mu e - xs)^- \le -(1 - \tau_1)x^T s, \tag{3.6}$$

$$e^T(\tau_1\mu e - xs)^+ \le \sqrt{n}\|(\tau_1\mu e - xs)^+\|. \tag{3.7}$$

For convenience we set

$$\beta := \frac{\tau_1 - \tau_2}{\tau_1}.$$

Obviously we have $\beta \in (0, 1)$, $\tau_1 - \tau_2 = \beta\tau_1$, and $\tau_2 = (1 - \beta\tau_1)$. We introduce a new notation $\mathcal{N}(\tau_1, \beta)$ to indicate the set $\mathcal{N}(\tau_1, \tau_2)$, i.e.,

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

The following lemma will be used frequently in the sequel.

Lemma 3.5 *Suppose* $(x, y, s) \in \mathcal{N}(\tau_1, \beta)$, $\beta \le 1/2$, *and* $\alpha_1 \le \alpha_2\sqrt{\frac{\beta\tau_1}{2n}}$. *Then we have*

$$\|(\triangle x(\alpha)\triangle s(\alpha))^-\|_1 \le \|(\triangle x(\alpha)\triangle s(\alpha))^+\|_1 \le \frac{1}{2}\alpha_2^2\beta\tau_1\mu.$$

Proof Denote $u := (x^{-1/2}s^{1/2})\triangle x(\alpha)$, $v := (x^{1/2}s^{-1/2})\triangle s(\alpha)$ and

$$r := (xs)^{-1/2}(\alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+) - \alpha_1^2(xs)^{-1/2}(\triangle x_-\triangle s_-).$$

So by Eqs. (2.7), (2.8) and (3.1) we have $u^T v = \triangle x(\alpha)^T\triangle s(\alpha) = 0$ and $u + v = r$. Therefore, by Lemma 3.3 we have

$$\begin{aligned}
&\|(\triangle x(\alpha)\triangle s(\alpha))^-\|_1\\
&\quad\le \|(\triangle x(\alpha)\triangle s(\alpha))^+\|_1\\
&\quad\le \frac{1}{4}\|(xs)^{-1/2}(\alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+) - \alpha_1^2(xs)^{-1/2}(\triangle x_-\triangle s_-)\|^2\\
&\quad\le \frac{1}{4}(\|(xs)^{-1/2}(\alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+)\|\\
&\qquad+\alpha_1^2\|(\triangle x_-\triangle s_-)/\sqrt{xs}\|)^2.
\end{aligned}$$

If we multiply the third equation of (2.7) by $(xs)^{-1/2}$, then by Lemma 5.3 of [22] we have

$$\|\triangle x_-\triangle s_-\| \le 2^{-3/2}\|(xs)^{-1/2}(\tau_1\mu - xs)^-\|^2 \le 2^{-3/2}\|(xs)^{1/2}\|^2 = 2^{-3/2}n\mu. \quad (3.8)$$

Moreover, by the relationships (2.6) we have

$$\begin{aligned}
&\|(\alpha_1(xs)^{-1/2}(\tau_1\mu e - xs)^- + \alpha_2(xs)^{-1/2}(\tau_1\mu e - xs)^+)\|^2\\
&\quad= \alpha_1^2\|(xs)^{-1/2}(\tau_1\mu e - xs)^-\|^2 + \alpha_2^2\|(xs)^{-1/2}(\tau_1\mu e - xs)^+\|^2\\
&\quad\le \alpha_1^2 n\mu + \alpha_2^2(\beta\tau_1\mu)^2/(\tau_2\mu)\\
&\quad\le \alpha_2^2\frac{\beta\tau_1}{2n}(n\mu) + \alpha_2^2\beta\tau_1\mu\\
&\quad= \frac{3}{2}\alpha_2^2\beta\tau_1\mu.
\end{aligned}$$

Therefore

$$\begin{aligned}\|(\triangle x(\alpha)\triangle s(\alpha))^+\|_1 &\le \frac{1}{4}\left(\sqrt{\frac{3}{2}\alpha_2^2\beta\tau_1\mu} + 2^{-3/2}\alpha_1^2 n\mu/\sqrt{\tau_2\mu}\right)^2 \\ &= \frac{1}{4}\left(\sqrt{\frac{3}{2}} + \sqrt{\frac{\beta\tau_1}{32\tau_2}}\right)^2 \alpha_2^2\beta\tau_1\mu \\ &\le \frac{1}{2}\alpha_2^2\beta\tau_1\mu.\end{aligned}$$

The proof is complete. □

Suppose $(x, y, s) \in \mathcal{N}(\tau_1, \beta)$, then by (3.5), (3.6), (3.7) we have

$$\begin{aligned}\mu(\alpha) &\le \mu - \alpha_1(1-\tau_1)\mu + \alpha_2\|(\tau_1\mu e - xs)^+\|/\sqrt{n} \\ &\le \mu - \alpha_1(1-\tau_1)\mu + \alpha_2\beta\tau_1\mu/\sqrt{n}\end{aligned} \tag{3.9}$$

and

$$\mu(\alpha) \ge \mu + \alpha_1 e^T(\tau_1\mu e - xs)^-/n \ge \mu - \alpha_1 x^T s/n = (1-\alpha_1)\mu. \tag{3.10}$$

Lemma 3.6 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}(\tau_1, \beta)$, $\tau_1 \le 1/5$, and $\beta \le 1/2$. If $\alpha_1 = \alpha_2\sqrt{\frac{\beta\tau_1}{2n}}$, then we have*

$$\|(\tau_1\mu(\alpha)e - h(\alpha))^+\| \le (1-\alpha_2)\beta\tau_1\mu.$$

Proof If $\tau_1\mu - x_i s_i \le 0$, by (3.3) and (3.9) we have

$$\begin{aligned}&\tau_1\mu(\alpha) - h(\alpha)_i \\ &\le \tau_1(\mu - \alpha_1(1-\tau_1)\mu + \alpha_2\beta\tau_1\mu/\sqrt{n}) - \left(1 - \alpha_1 - \frac{1}{4}\alpha_1^2\right)x_i s_i - \alpha_1\tau_1\mu \\ &\le \tau_1(\mu - \alpha_1(1-\tau_1)\mu + \alpha_2\beta\tau_1\mu/\sqrt{n}) - \left(1 - \alpha_1 - \frac{1}{4}\alpha_1^2\right)\tau_1\mu - \alpha_1\tau_1\mu \\ &= \left(-1 + \tau_1 + \frac{1}{4}\alpha_1 + \sqrt{2\beta\tau_1}\right)\alpha_1\tau_1\mu \\ &\le \frac{9\sqrt{5} - 32}{40}\alpha_1\tau_1\mu \\ &\le 0\end{aligned}$$

and if $\tau_1\mu - x_i s_i > 0$, by (3.4), (3.9) and above result we have

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

$$
\begin{aligned}
&= \left(1 - \alpha_2 - \frac{1}{4}\alpha_1^2\right)(\tau_1\mu - x_i s_i) + \left(-1 + \tau_1 + \frac{1}{4}\alpha_1 + \sqrt{2\beta\tau_1}\right)\alpha_1\tau_1\mu \\
&\le \left(1 - \alpha_2 - \frac{1}{4}\alpha_1^2\right)(\tau_1\mu - x_i s_i) \\
&\le (1 - \alpha_2)(\tau_1\mu - x_i s_i),
\end{aligned}
$$

which implies that

$$
\|(\tau_1\mu(\alpha)e - h(\alpha))^+\| \le (1 - \alpha_2)\|(\tau_1\mu e - xs)^+\| \le (1 - \alpha_2)\beta\tau_1\mu.
$$

The proof is complete. □

The next lemma gives out a sufficient condition which guarantees all the iterates in the neighborhood $\mathcal{N}(\tau_1, \beta)$.

Lemma 3.7 *Suppose that the current iterate* $(x, y, s) \in \mathcal{N}(\tau_1, \beta)$, $\tau_1 \le 1/5$, *and* $\beta \le 1/2$. *If* $\alpha_1 = \alpha_2\sqrt{\frac{\beta\tau_1}{2n}}$, *then* $(x(\alpha), y(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \beta)$.

Proof Using Lemmas 3.5 , Lemmas 3.6 and (3.10) we obtain

$$
\begin{aligned}
\|(\tau_1\mu(\alpha)e - x(\alpha)s(\alpha))^+\| &= \|(\tau_1\mu(\alpha)e - h(\alpha) - \triangle x(\alpha)\triangle s(\alpha))^+\| \\
&\le \|(\tau_1\mu(\alpha)e - h(\alpha))^+\| + \|(-\triangle x(\alpha)\triangle s(\alpha))^+\| \\
&\le (1 - \alpha_2)\beta\tau_1\mu + \frac{1}{2}\alpha_2^2\beta\tau_1\mu \\
&= (1 - \alpha_1)\beta\tau_1\mu + \left(\alpha_1 - \alpha_2 + \frac{1}{2}\alpha_2^2\right)\beta\tau_1\mu \\
&\le \beta\tau_1\mu(\alpha),
\end{aligned}
$$

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

Now we are in a position to present our main complexity result.

Theorem 3.1 *Suppose that* $\tau = \tau_1 \le 1/5$ *and* $\beta \le 1/2$ *are fixed for all iterations. Furthermore, suppose that the plane-search procedure* (2.10) *is applied at each iteration. Then, Algorithm 3.1 terminates in* $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$ *iterations.*

Proof By Lemma 3.7, at each iteration, if we let $\hat{\alpha}_1 = \hat{\alpha}_2\sqrt{\frac{\beta\tau_1}{2n}}$ and $\hat{\alpha}_2 = 1$, then we have

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

Furthermore, from (3.9) we also have

$$\begin{aligned}\mu(\hat{\alpha}) &\le \left(1-\hat{\alpha}_1(1-\tau_1)+\hat{\alpha}_2\beta\tau_1/\sqrt{n}\right)\mu \\ &\le \left(1-\hat{\alpha}_2(1-\tau_1-\sqrt{2\beta\tau_1})\sqrt{\beta\tau_1/(2n)}\right)\mu \\ &\le \left(1-\left(1-\frac{1}{5}-\frac{\sqrt{5}}{5}\right)\sqrt{\beta\tau_1/(2n)}\right)\mu \\ &\le \left(1-\frac{\sqrt{\beta\tau_1}}{3\sqrt{2n}}\right)\mu.\end{aligned}$$

Therefore, the exact plane search (2.10) would lead to at least the same amount of reduction in $\mu(\alpha)$, and hence the theorem is proven. By the proof of Theorem 3.2 in [22], Algorithm 3.1 stops after at most $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$ iterations. □

The plane-search subproblem (2.10) 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 the objective function in the subproblem (2.10) is monotone with respect to α_1 for any fixed $\alpha_2 \in [0, 1]$ from the expression (3.5). So, we can solve subproblem (2.10) approximately in the following way. First, set $\alpha_2 = \theta \in (0, 1]$. Second, find the greatest $\alpha_1 \in [0, 1]$ such that $(x(\alpha), y(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \beta)$. Lemma 3.7 guarantee that $\alpha_1 \ge \theta\sqrt{\beta\tau_1/(2n)}$. It is clear that if the plane search procedure in Theorem 3.1 is replaced by this line search procedure, then the $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$ iteration bound still holds.

4 Numerical results

In this section, we compare the proposed primal-dual second-order corrector algorithm with Ai–Zhang's primal-dual path-following algorithm [2] and Mehrotra's original predictor–corrector algorithm [12]. These three algorithms will be denoted , respectively, by Algorithm PDSOC, Algorithm PDPF and Algorithm MPC. We consider only relatively small problems in the NETLIB test set for linear programming. We adopt the homogeneous self-dual formulation of Ye, Todd and Mizuno [23]. So it is easy to find a strictly feasible starting point. Numerical results were obtained by using MATLAB R2007b on an Intel Core 2 PC (1.86 Ghz) with 1 GB RAM. We stop the iteration of each algorithm if the relative duality gap satisfies

$$\frac{x^T s}{1+|c^T x|} \le 10^{-8}.$$

Table 1 lists the names of the test problems, the number of iterations (iter), the total CPU time (time), the relative duality gap (relgap) and the optimal value $c^T x$ when the

Table 1 Computational performance of the proposed algorithm

Problem	Rows	Columns	Iter	Time	relgap	Optimal value
adlittle	57	97	13	0.5696	1.3598e−11	2.2549495758e+05
afiro	28	32	9	0.0368	2.5346e−10	−4.6475313744e+02
bandm	306	472	20	18.9870	8.1633e−11	−1.5862801840e+02
blend	75	83	11	0.3066	1.7318e−09	−3.0812149773e+01
brandy	221	249	22	5.9716	1.3845e−11	1.5185098965e+03
degen2	445	534	13	48.3770	1.3932e−10	−1.4351780006e+03
e226	224	282	21	15.0970	5.9062e−09	−1.8751928971e+01
israel	175	142	22	7.1984	6.2222e−11	−8.9664450570e+05
lotfi	154	308	22	4.6812	1.4726e−09	−2.5264479077e+01
sc105	106	103	13	0.5170	2.0803e−09	−5.2202050527e+01
sc205	206	203	11	1.9954	2.6743e−10	−5.2202054970e+01
sc50a	51	48	11	0.1125	2.0537e−11	−6.4575077013e+01
sc50b	51	48	11	0.0894	4.3458e−11	−6.9999999895e+01
scagr25	472	500	16	45.1710	1.3472e−09	−1.4752987007e+07
scagr7	130	140	12	1.4241	8.8295e−09	−2.3311269207e+06
scfxm1	331	457	26	28.1670	7.9130e−11	1.8416759082e+04
scsd1	78	760	10	16.4480	5.2516e−10	8.6666666870e+00
scsd6	148	1350	12	94.4180	2.1895e−09	5.0500000286e+01
sctap1	301	480	19	33.3830	1.3602e−10	1.4122500011e+03
share1b	118	225	30	5.2419	9.4396e−09	−7.6585055520e+04
share2b	97	79	12	0.6466	1.9195e−09	−4.1573223775e+02
stocfor1	118	111	17	0.7055	1.5390e−11	−4.1131976014e+04

Algorithm 3.1 terminates. The computational experiments reported in Table 1 confirm that Algorithm 3.1, even with a relatively naive implementation, is efficient and accurate.

In Table 2 we compare the number of iterations and the CPU time required by our proposed algorithm to those required by Ai–Zhang's algorithm and Mehrotra's algorithm. From the results in Table 2 we find that, on the average, the number of iterations and the computational time required by our algorithm are about 53.61% and 48.05%, respectively, less than those required by Ai–Zhang's algorithm. Although our implementations are very coarse, the proposed algorithm is comparable to Mehrotra's algorithm as a whole.

5 Conclusions

In this paper, we propose a primal-dual second-order corrector interior point algorithm with $O(\sqrt{n}L)$ iteration bound for linear programming problems. It adds a corrector step to Ai–Zhang's primal-dual path-following algorithm, in an attempt to improve performance. Our preliminary implementations show that this algorithm is promising

Table 2 Comparison with Ai–Zhang's algorithm and Mehrotra's algorithm

Problem	PDSOC		PDPF		MPC	
Name	Iter	Time	Iter	Time	Iter	Time
adlittle	13	0.5696	23	0.8798	13	0.4943
afiro	9	0.0368	16	0.0563	9	0.0326
bandm	20	18.9870	47	46.1090	22	19.1301
blend	11	0.3066	19	0.4560	11	0.2651
brandy	22	5.9716	44	9.9977	20	4.5628
degen2	13	48.3770	24	82.8378	12	41.1407
e226	21	15.0970	52	39.2270	21	13.5303
israel	22	7.1984	55	15.9984	20	5.7834
lotfi	22	4.6812	55	9.6648	17	3.2413
sc105	13	0.5170	24	0.8716	12	0.4298
sc205	11	1.9954	21	3.5359	12	2.0189
sc50a	11	0.1125	21	0.1788	11	0.1013
sc50b	11	0.0894	19	0.1580	10	0.0933
scagr25	16	45.1710	35	94.9330	15	38.0652
scagr7	12	1.4241	25	2.5747	12	1.2366
scfxm1	26	28.1670	58	57.9045	24	23.7125
scsd1	10	16.4480	17	25.1739	10	14.6971
scsd6	12	94.41800	22	158.6378	12	86.1777
sctap1	19	33.3830	44	66.0888	20	31.6501
share1b	30	5.2419	78	16.4964	30	3.6996
share2b	12	0.6466	28	1.2948	11	0.5168
stocfor1	17	0.7055	34	1.2457	15	0.5513

for solving linear programming. However, there are still some unsettled issues for implementation. For example, an adaptive update of the centering parameter and efficient strategies to choose step lengths deserve more work for real applications of the algorithm. Thus, the gap between theory and practice still exists. We hope that further research inspired by this paper will close this gap.

Acknowledgments The authors are very grateful to the editor and the anonymous referees for their valuable suggestions which helped to improve the paper. This work was partially supported by the National Natural Science Foundation of China (Grant No. 61072144) and the Fundamental Research Funds for the Central Universities (Grant No. JY10000970004).

References

1. Ai, W.: Neighborhood-following algorithms for linear programming. Sci. China Ser. A. **47**, 812–820 (2004)
2. 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)

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. April 2005. http://www.optimization-online.org/DB_HTML/2005/03/1097.html
5. Floudas, C.A., Pardalos, P.M.: Encyclopedia of optimization, 2nd edn. Springer, New York (2009)
6. Freund, R.M., Mizuno, S.: Interior point methods: current status and future directions. In: Frenk, J.B.G., Roos, C., Terlaky, T., Zhang, S. (eds.) High Performance Optimization, pp. 441–446. Kluwer, Dordrecht (2000)
7. Gondzio, J.: Multiple centrality corrections in a primal-dual method for linear programming. Comput. Optim. Appl. **6**, 137–156 (1996)
8. Gonzaga, C.C.: Path-following methods for linear programming. SIAM Rev. **34**, 167–224 (1992)
9. Kojima, M., Megiddo, N., Mizuno, S.: A primal-dual infeasible-interior-point algorithm for linear programming. Math. Program. **61**, 263–280 (1993)
10. 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)
11. Mehrotra, S.: On finding a vertex solution using interior point methods. Linear Algebra Appl. **152**, 233–253 (1991)
12. Mehrotra, S.: On the implementation of a primal-dual interior point method. SIAM J. Optim. **2**, 575–601 (1992)
13. Nematollahi, E., Terlaky, T.: A simpler and tighter redundant Klee-Minty construction. Optim. Lett. **2**, 403–414 (2008)
14. NETLIB: A Repository of Linear Programming Test Problems. http://www.netlib.org/
15. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
16. Pardalos, P.M., Wolkowicz, H.: Topics in Semidefinite and Interior-Point Methods. American Math Society, Providence (1998)
17. Potra, F.A., Wright, S.J.: Interior-point methods. J. Comput. Appl. Math. **124**, 281–302 (2000)
18. Roos, C., Terlaky, T., Vial, J.-Ph.: Theory and Algorithms for Linear Optimization: An Interior Point Approach. Wiley, Chichester (1997)
19. Salahi, M., Mahdavi-Amiri, N.: Polynomial time second order Mehrotra-type predictor–corrector algorithms. Appl. Math. Comput. **183**, 646–658 (2006)
20. Salahi, M., Peng, J., Terlaky, T.: On Mehrotra-type predictor–corrector algorithms. SIAM J. Optim. **18**, 1377–1397 (2007)
21. Salahi, M., Terlaky, T.: Postponing the choice of the barrier parameter in Mehrotra-type predictor–corrector algorithms. Eur. J. Oper. Res. **182**, 502–513 (2007)
22. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)
23. Ye, Y., Todd, M.J., Mizuno, S.: An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. Math. Oper. Res. **19**, 53–67 (1994)
24. Zhang, Y., Zhang, D.T.: On polynomiality of the Mehrotra-type predictor–corrector interior-point algorithms. Math. Program. **68**, 303–318 (1995)