Mehrotra-Type Predictor-Corrector Algorithms Revisited

Maziar Salahi* and Tamás Terlaky†

February 5, 2007

Abstract

Motivated by a numerical example which shows that a feasible version of Mehrotra's original predictor-corrector algorithm might be inefficient in practice, Salahi et al., proposed a so-called safeguard based variant of the algorithm that enjoys polynomial iteration complexity while its practical efficiency is preserved. In this paper we analyze the same Mehrotra's algorithm from a different perspective. We give a condition on the maximum step size in the predictor step, violation of which might imply a very small or zero step size in the corrector step of the algorithm. This might explain the reason for occasional ill behavior of the original algorithm.

We propose to cut the maximum step size in the predictor step if it is above a certain threshold. If this cut does not give a desirable step size, then we cut it for the second time that allows us to give a lower bound for the step size in the corrector step. This enables us to prove an $\mathcal{O}\left(n^{\frac{5}{2}} \log \frac{n}{\epsilon}\right)$ worst case iteration complexity bound for the new algorithm. By slightly modifying the Newton system in the corrector step we reduce the iteration complexity to $\mathcal{O}\left(n^{\frac{3}{2}} \log \frac{n}{\epsilon}\right)$. Finally, we report some illustrative computational results using the McIPM software package.

Keywords: Linear Optimization, Predictor-Corrector Methods, Interior-Point-Methods, Mehrotra-Type Algorithm, Polynomial Complexity.

*Corresponding author, Department of Mathematics, Faculty of Sciences, The University of Guilan, P.O. Box 1914, Rasht, Iran, salahim@guilan.ac.ir.

†Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada, L8S 4K1, terlaky@mcmaster.ca.

AMS Subject Classification: 90C05, 90C51

1 Introduction

Variants of Mehrotra's original predictor-corrector algorithm [6, 7] are among the most widely used algorithms in Interior-Point Methods (IPMs) based software packages [1, 3, 4, 14, 16, 18, 19]. The authors of [11] have shown by a numerical example that a feasible version of the algorithm may be forced to make many small steps that motivated them to introduce certain safeguards, what allowed them to prove polynomial iteration complexity. In this paper we analyze the same Mehrotra's algorithm from a different point of view, and subsequently we propose a new modification of the algorithm that also enjoys polynomial iteration complexity and maintains its efficiency in practice. Before going into the details of the algorithm, let us give a brief introduction to IPMs.

Throughout the paper we deal with the standard form LO problem:

$$(P) \qquad \min \{c^T x : Ax = b,\ x \geq 0\},$$

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

$$(D) \qquad \max \{b^T y : A^T y + s = c,\ s \geq 0\}.$$

Without loss of generality [9] we may assume that both (P) and (D) satisfy the interior point condition (IPC), i.e., there exists an (x^0, y^0, s^0) such that

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

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

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s &= c, \quad s \geq 0, \\ xs &= 0, \end{aligned} \tag{1}$$

where xs denotes the componentwise (Hadamard) product of the vectors x and s. The basic idea of primal-dual IPMs is to replace the third equation in (1) by the parameterized equation $xs = \mu e$, where e is the all one vector. This leads to the following system:

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s &= c, \quad s \geq 0, \\ xs &= \mu e. \end{aligned} \tag{2}$$

If the IPC holds, then system (2) has a unique solution for each $\mu > 0$. 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 gives *the central path* of (P) and (D) [5, 13]. It has been shown that the limit of the central path (as μ goes to zero) exists. Because the limit point satisfies the complementarity condition, it naturally yields optimal solutions for both (P) and (D), respectively [9].

Applying Newton's method to (2) from a given feasible interior point gives the following linear system of equations[1]:

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ x\Delta s + s\Delta x &= \mu e - xs, \end{aligned} \tag{3}$$

where $(\Delta x, \Delta y, \Delta s)$ is the Newton direction. For detailed information about classical IPMs and their iterations complexity the reader can consult [9, 15], and the references therein.

In what follows we briefly discuss a feasible version of Mehrotra's original algorithm. Variants of this algorithm has been analyzed in [10, 11, 12] and it will be considered for further developments in this paper. In the predictor step one solves the following system of equations, the so called affine scaling system:

$$\begin{aligned} A\Delta x^{\mathrm{a}} &= 0, \\ A^T\Delta y^{\mathrm{a}} + \Delta s^{\mathrm{a}} &= 0, \\ s\Delta x^{\mathrm{a}} + x\Delta s^{\mathrm{a}} &= -xs. \end{aligned} \tag{4}$$

Then the maximum feasible step size in this direction is computed i.e., the largest α_a such that

$$(x + \alpha_{\mathrm{a}}\Delta x^{\mathrm{a}}, s + \alpha_{\mathrm{a}}\Delta s^{\mathrm{a}}) \geq 0.$$

However, the algorithm does not make this step right away. Using the information from the predictor step it computes the corrector direction by solving the following system:

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs - \Delta x^{\mathrm{a}}\Delta s^{\mathrm{a}}, \end{aligned} \tag{5}$$

[1] We assume that one has an interior feasible starting point for the given problem, which can be obtained by using the self-dual embedding model [1, 9]. The infeasible case can also be carried out analogously [15].

where μ is defined adaptively as

$$\mu = \left(\frac{g_{\rm a}}{g}\right)^2 \frac{g_{\rm a}}{n},$$

where $g_{\rm a} = (x + \alpha_{\rm a}\Delta x^{\rm a})^T(s + \alpha_{\rm a}\Delta s^{\rm a})$ and $g = x^T s$. Since $(\Delta x^{\rm a})^T\Delta s^{\rm a} = 0$, the previous relation can be further simplified to

$$\mu = (1 - \alpha_{\rm a})^3 \mu_g, \tag{6}$$

where $\mu_g := \frac{x^T s}{n}$.

One can notice the major feature of this algorithm. In most predictor corrector algorithms one uses different coefficient matrices in the predictor and corrector steps, however here the same coefficient matrix is used both in the predictor and corrector steps. This leads to significant computational saving for large scale problems. Further, the corrector step of this algorithm uses some information from the predictor step, namely $\alpha_{\rm a}$ and $\Delta x^{\rm a}\Delta s^{\rm a}$.

Due to its superior practical performance, this algorithm is implemented with various heuristics in several software packages [1, 3, 4, 14, 16, 18, 19]. Most recently, in this framework, Mehrotra and Li [8] used a Krylov-subspace approach to generate several corrector directions and proved global convergence of the new variant and presented encouraging numerical results. Moreover, using the idea of multiple centrality steps Colombo and Gondzio [2] considered an infeasible variant of this algorithm while using a symmetric neighborhood. The neighborhood ensures that the coordinates of xs can neither be too small nor too big. The authors have also reported good numerical performance of their implementation.

As it is discussed in [11], the original Mehrotra's predictor corrector algorithm might make very small steps in order to keep the iterates in a certain neighborhood of the central path. In [11] the authors combined this algorithm with a simple large-update safeguard that guarantees polynomial iteration complexity, while the efficiency of the algorithm is preserved. This algorithm is also analyzed in [12] from a different point of view, which is called "postponing the choice of barrier parameter", and the authors proved similar complexity results as in [11], while the convergence results are stronger. In this paper, by straightforward analysis we give a criterion on $\alpha_{\rm a}$ that indicates when the algorithm risks very small or even zero step size in the corrector step. This motivates us to cut $\alpha_{\rm a}$ if it is above certain threshold. If this cut does not result in a desirable step size, then we cut $\alpha_{\rm a}$ for the second time which implies a theoretically warranted step size in the corrector step and subsequently polynomial iteration complexity.

The rest of the paper is organized as it follows. First, in Section 2, we present some technical lemmas and give a condition on α_{a}, violation of which might lead to a very small or zero step size for the corrector step. This motivates the construction of the new algorithm. Then, we establish a worst case iteration complexity bound for the new algorithm. A slightly modified version of the proposed algorithm is presented in Section 3 that enjoys even better iteration complexity. Finally, some illustrative numerical results are given in Section 4. For self completeness some technical results that are taken from [11] are presented in the Appendix. For simplicity we also use the following notations throughout the paper.

- $\mathcal{F} = \{(x, y, s) \in R^n \times R^m \times R^n |\ Ax = b,\ A^T y + s = c,\ x \geq 0,\ s \geq 0\}$.
- $\mathcal{F}^0 = \{(x, y, s) \in \mathcal{F} |\ x > 0,\ s > 0\}$.
- $\mathcal{I} = \{1, 2, \cdots, n\}$, $\mathcal{I}_+ = \{i \in \mathcal{I} |\ \Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}} > 0\}$, and $\mathcal{I}_- = \mathcal{I} \setminus \mathcal{I}_+$.
- $x(\alpha) = x + \alpha\Delta x,\ y(\alpha) = y + \alpha\Delta y,\ s(\alpha) = s + \alpha\Delta s$.

2 Safeguard Based Algorithm

To obtain an upper bound for the maximum number of iterations in IPMs based algorithms a certain neighborhood of the central path is considered in which the algorithms operate. The wider the neighborhood, the larger part of the feasible region is covered, thus there is more room for the algorithm to operate. Therefore, in this paper we consider the widely used neighborhood, namely the negative infinity norm neighborhood defined by

$$\mathcal{N}_\infty^-(\gamma) := \{(x, y, s) \in \mathcal{F}^0 :\ x_i s_i \geq \gamma \mu_g\ \forall i \in \mathcal{I}\}, \tag{7}$$

where $\gamma \in (0, 1)$ is a constant independent of n. The following theorem shows that there exist always a guarranteed positive step size in the predictor step of the algorithm.

Theorem 2.1 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and $(\Delta x^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ is the solution of (4). Then the maximum feasible step size, $\alpha_{\mathrm{a}} \in (0, 1]$, satisfies*

$$\alpha_{\mathrm{a}} \geq \sqrt{\frac{\gamma}{n}}. \tag{8}$$

Proof: See [11]. □

The following technical lemma and its corollary will be used in the derivation of the lower bound for the step size.

Lemma 2.2 *[11] Suppose that the current iterate* $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ *and let* $(\Delta x, \Delta y, \Delta s)$ *be the solution of (5), where* $\mu \geq 0$. *Then we have*

$$\|\Delta x \Delta s\| \leq 2^{\frac{-3}{2}} \left(\frac{1}{\gamma} \left(\frac{\mu}{\mu_g} \right)^2 - \left(2 - \frac{1}{2\gamma} \right) \frac{\mu}{\mu_g} + \frac{17\gamma + n}{16\gamma} \right) n\mu_g.$$

Proof: If we multiply the third equation of (5) by $(XS)^{-\frac{1}{2}}$, then by Lemma 5.3 we have

$$\begin{aligned}
\|\Delta x \Delta s\| &\leq 2^{\frac{-3}{2}} \left\| \mu (XSe)^{-\frac{1}{2}} - (XSe)^{\frac{1}{2}} - (XS)^{-\frac{1}{2}} \Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}} \right\|^2 \\
&= 2^{\frac{-3}{2}} \left(\mu^2 \sum_{i \in \mathcal{I}} \frac{1}{x_i s_i} + \sum_{i \in \mathcal{I}} x_i s_i + \sum_{i \in \mathcal{I}} \frac{(\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}})^2}{x_i s_i} - 2n\mu - 2\mu \sum_{i \in \mathcal{I}} \frac{\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}}}{x_i s_i} \right) \\
&\leq 2^{\frac{-3}{2}} \left(\frac{n\mu^2}{\gamma \mu_g} + n\mu_g + \frac{n\mu_g}{16} + \frac{n^2 \mu_g}{16\gamma} - 2n\mu + \frac{n\mu}{2\gamma} \right),
\end{aligned}$$

where the last inequality follows from Lemma 5.2 and the assumption that the previous iterate is in $\mathcal{N}_\infty^-(\gamma)$. By reordering we complete the proof of the lemma. □

The following corollary gives an explicit upper bound for the second order term when μ is chosen adaptively as given by (6).

Corollary 2.3 *If* $\mu = (1 - \alpha_{\mathrm{a}})^3 \mu_g$, *then*

$$\|\Delta x \Delta s\| \leq \frac{n^2}{\gamma} \mu_g.$$

The next theorem provides a bound for α_{a} that ensures a positive step size in the corrector step. It also indicates that a larger step size in the predictor step might result a very small or zero step size in the corrector step. For simplicity the following notation is used in the rest of our development:

$$t = \max_{i \in \mathcal{I}_+} \left\{ \frac{\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}}}{x_i s_i} \right\}. \tag{9}$$

From Lemma 5.1 it follows that $t \in [0, \ \frac{1}{4}]$. In the following lemma we give a bound for the t value when the duality gap is sufficiently small.

Lemma 2.4 *For sufficiently small* μ_g *one has* $t \leq \mathcal{O}(\mu_g)$.

Proof: By Theorem 7.4 of [15] for $(x, s) \in \mathcal{N}_\infty^-(\gamma)$ the relations

$$|\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}}| \le \mathcal{O}(\mu_g^2), \ \ i = 1, \cdots, n \tag{10}$$

hold. This implies the statement of the lemma seeing the definition of t. □

Corollary 2.5 *For sufficiently small* μ_g *one has* $\alpha_{\mathrm{a}} \ge 1 - \mathcal{O}(\mu_g)$.

Theorem 2.6 *Suppose that the current iterate* $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ *and let* $(\Delta x, \Delta y, \Delta s)$ *be the solution of (5) with* μ *as it is defined by (6). Then, for* $\alpha_{\mathrm{a}} \in (0, 1]$ *satisfying*

$$\alpha_{\mathrm{a}} < 1 - \left(\frac{\gamma t}{(1-\gamma)}\right)^{\frac{1}{3}} \tag{11}$$

the maximum step size in the corrector step is strictly positive.

Proof: Our goal is to find a lower bound for the maximal $\alpha \in (0, 1]$ such that

$$x_i(\alpha)s_i(\alpha) \ge \gamma \mu_g(\alpha), \ \ \forall i \in \mathcal{I}, \tag{12}$$

where

$$\mu_g(\alpha) = \frac{x(\alpha)^T s(\alpha)}{n} = \left(1 - \alpha + \frac{\mu}{\mu_g}\alpha\right)\mu_g. \tag{13}$$

After expanding (12) we have to have

$$x_i s_i + \alpha(\mu - x_i s_i - \Delta {x^{\mathrm{a}}}_i \Delta {s^{\mathrm{a}}}_i) + \alpha^2 \Delta x_i \Delta s_i \ge \gamma(1-\alpha)\mu_g + \gamma\alpha\mu$$

or

$$(1-\alpha)x_i s_i + \alpha(1-\gamma)\mu - \alpha \Delta {x^{\mathrm{a}}}_i \Delta {s^{\mathrm{a}}}_i + \alpha^2 \Delta x_i \Delta s_i \ge \gamma(1-\alpha)\mu_g. \tag{14}$$

Now, since $(\Delta x^{\mathrm{a}})^T \Delta s^{\mathrm{a}} = 0$, then we have $\mathcal{I}_+ \neq \emptyset$. Thus, in the worst case it is sufficient to consider those inequalities in (14) for which $i \in \mathcal{I}_+$ and $\Delta x_i \Delta s_i < 0$. To do so, by using Lemma 5.2 one has

$$\begin{aligned}(1-\alpha)x_i s_i \ &+ \alpha(1-\gamma)\mu - \alpha \Delta {x^{\mathrm{a}}}_i \Delta {s^{\mathrm{a}}}_i + \alpha^2 \Delta x_i \Delta s_i \\ &\ge (1-\alpha)x_i s_i + \alpha(1-\gamma)\mu - \alpha t x_i s_i + \alpha^2 \Delta x_i \Delta s_i \\ &= (1-\alpha(1+t))x_i s_i + \alpha(1-\gamma)\mu + \alpha^2 \Delta x_i \Delta s_i \\ &\ge \left((1-\alpha(1+t))\gamma + \alpha(1-\gamma)\frac{\mu}{\mu_g} + \alpha^2 \frac{\Delta x_i \Delta s_i}{\mu_g}\right)\mu_g,\end{aligned}$$

where the last inequality holds for $0 \le \alpha \le \frac{4}{5}$, since by Lemma 5.1 we have $0 \le t \le \frac{1}{4}$. Therefore the new iterate is certainly in the neighborhood $\mathcal{N}_\infty^-(\gamma)$ if

$$\left((1-\alpha(1+t))\gamma + \alpha(1-\gamma)\frac{\mu}{\mu_g} + \alpha^2 \frac{\Delta x_i \Delta s_i}{\mu_g}\right)\mu_g \ge \gamma(1-\alpha)\mu_g.$$

After simplifying, it is sufficient to have

$$\alpha \frac{\Delta x_i \Delta s_i}{\mu_g} - \gamma t + (1-\gamma)(1-\alpha_{\mathrm{a}})^3 \ge 0. \tag{15}$$

Now, since we assumed that $\Delta x_i \Delta s_i < 0$, then for α_{a} satisfying

$$-\gamma t + (1-\gamma)(1-\alpha_{\mathrm{a}})^3 > 0$$

or

$$\alpha_{\mathrm{a}} < 1 - \left(\frac{\gamma t}{1-\gamma}\right)^{\frac{1}{3}}$$

one always can guarantee a positive step size α in the corrector step. □

To have an explicit strictly positive lower bound for the maximum step size α_c in the corrector step, instead of (11), we use the following inequality:

$$\alpha_{\mathrm{a}} \le 1 - \left(\frac{2\gamma t}{1-\gamma}\right)^{\frac{1}{3}} := \alpha_1. \tag{16}$$

Corollary 2.7 *If α_{a} satisfies (16), then $\alpha_{\mathrm{c}} \ge \frac{\gamma^2 t}{n^2}$.*

Proof: It follows from (15) and Corollary 2.3. □

Remark 2.8 *By Theorem 2.6 we know that for sufficiently small μ_g, we can guarantee a positive step size in the corrector step for $\alpha_{\mathrm{a}} \le 1 - \mathcal{O}(\mu_g^{\frac{1}{3}})$. However, following the Corollary 2.5 $\alpha_{\mathrm{a}} \ge 1 - \mathcal{O}(\mu_g)$, which is greater than or equal to $1 - \mathcal{O}(\mu_g^{\frac{1}{3}})$ for sufficiently small μ_g. In other words, in asymptotic case we might need to cut α_{a}, but still having a reasonably big α_{a}.*

Remark 2.9 *From (16) and Lemma 2.4 it is obvious that when μ_g approaches to zero then α_1 approaches to one. In other words, our cut does not block the convergence of the affine scaling step size to one, it just reduces the speed of convergence in order to guarantee a positive step size for the corrector step.*

Corollary 2.10 *If $t = 0$ at a certain iteration, then the algorithm can make a full Newton step in the predictor step and stop with an optimal solution.*

Proof: From (15) it obvious that when $t = 0$, then $\Delta x^{\mathrm{a}}{}_i \Delta s^{\mathrm{a}}{}_i = 0 \;\forall i \in \mathcal{I}_+$ and subsequently $\Delta x^{\mathrm{a}}{}_i \Delta s^{\mathrm{a}}{}_i = 0 \;\forall i \in \mathcal{I}_-$. Therefore, a full Newton step in the predictor step leads to an optimal solution. □

Therefore, if α_{a} violates (16), then we let $\alpha_{\mathrm{a}} = \alpha_1$ and proceed with the corrector step. To ensure polynomial iteration complexity, if the maximum step size in the corrector step is still below a certain threshold depending only on the dimension, then we let $\alpha_{\mathrm{a}} = 1 - \left(\frac{\beta}{1-\beta}\right)^{\frac{1}{3}}$, where $\gamma \leq \beta < \frac{1}{4}$. Using (6) one can see that this choice further implies $\mu = \frac{\beta}{1-\beta}\mu_g$. By this choice one further can guarantee a lower bound for the maximum step size in the corrector step which is independent[2] of t. Subsequently the polynomial iteration complexity of the algorithm can be proved. The following corollary, which follows from Lemma 2.2, is useful to prove a lower bound for α_c in the next theorem.

Corollary 2.11 *If $\mu = \frac{\beta}{1-\beta}\mu_g$, where $\gamma \leq \beta < \frac{1}{4}$ and $\gamma \in (0, \frac{1}{4})$, then*

$$\|\Delta x \Delta s\| \leq \frac{n^2}{18\gamma}\mu_g.$$

Theorem 2.12 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and $(\Delta x, \Delta y, \Delta s)$ is the solution of (5) with $\mu = \frac{\beta}{1-\beta}\mu_g$. Then*

$$\alpha_{\mathrm{c}} \geq \frac{27\gamma^2}{2n^2}.$$

Proof: Following the proof of Theorem 2.6, in the worst case, as given by (15), it is sufficient to have

$$\alpha\frac{\Delta x_i \Delta s_i}{\mu_g} - \gamma t + (1-\gamma)\frac{\beta}{1-\beta} \geq 0.$$

Now, by Corollary 2.11 and using the fact that $0 \leq t \leq \frac{1}{4}$ by Lemma 5.1, the previous inequality holds for α satisfying

$$-\alpha\frac{n^2}{18\gamma} - \frac{\gamma}{4} + (1-\gamma)\frac{\beta}{1-\beta} \geq 0$$

or $\alpha \leq \frac{27\gamma^2}{2n^2}$. This implies $\alpha_{\mathrm{c}} \geq \frac{27\gamma^2}{2n^2}$. □

Based on the previous discussions we can outline our new algorithm as follows:

[2] Since t varies in the interval $[0, \frac{1}{4}]$, therefore one can not guarantee a unique lower bound from Corollary 2.7 which would further lead to polynomial iteration complexity.

Algorithm 1

Input:

A proximity parameters $\gamma \in (0, \frac{1}{4})$; a safeguard parameter $\beta \in [\gamma, \frac{1}{4})$;
an accuracy parameter $\epsilon > 0$; a starting point $(x^0, y^0, s^0) \in \mathcal{N}_\infty^-(\gamma)$.

begin
  while $x^T s \geq \epsilon$ **do**
  begin
    Predictor Step
    Solve (4) and compute the maximum step size α_a such that
    $(x(\alpha_\mathrm{a}), y(\alpha_\mathrm{a}), s(\alpha_\mathrm{a})) \in \mathcal{F}$; (the algorithm does not make this step).
    If $(1-\alpha_\mathrm{a})x(\alpha_\mathrm{a})^T s(\alpha_\mathrm{a}) \leq \epsilon$, **then**
    let $x = x(\alpha_\mathrm{a}),\ s = s(\alpha_\mathrm{a})$ and **stop.**
    end
  end
  begin
    Corrector step
    If $\alpha_\mathrm{a} > \alpha_1$ **then** let $\alpha_\mathrm{a} = \alpha_1$.
    end
    Solve (5) with μ defined by (6) and compute the
    maximum step size α_c such that $(x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c})) \in \mathcal{N}_\infty^-(\gamma)$;
    If $\alpha_\mathrm{c} < \frac{27\gamma^2}{2n^2}$, **then** solve (5) with $\mu = \frac{\beta}{1-\beta}\mu_g$ and compute
    the maximum step size α_c such that $(x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c})) \in \mathcal{N}_\infty^-(\gamma)$;
    end
    Set $(x, y, s) = (x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c}))$.
  end
end

The following theorem gives the maximum number of iterations for Algorithm 1 to find an $\epsilon-$approximate solution.

Theorem 2.13 *After at most*

$$O\left(n^{\frac{5}{2}} \log \frac{(x^0)^T s^0}{\epsilon}\right)$$

number of iterations Algorithm 1 stops with a solution for which $x^T s \leq \epsilon$.

Proof: If $\alpha_\mathrm{a} > \alpha_1$ and $\alpha_c \geq \frac{27\gamma^2}{2n^2}$, then using (13) we have

$$\mu_g(\alpha) \leq \left(1 - \frac{27\gamma^2(2-3\gamma)}{4(1-\gamma)n^2}\right)\mu_g.$$

If $\alpha_a > \alpha_1$ and $\alpha_c < \frac{27\gamma^2}{2n^2}$, then using (13) we have

$$\mu_g(\alpha) \leq \left(1 - \frac{27\gamma^2(1-2\beta)}{2(1-\beta)n^2}\right)\mu_g,$$

If $\alpha_a \leq \alpha_1$ and $\alpha_c \geq \frac{27\gamma^2}{2n^2}$, then using (13) and Theorem 2.1 we have

$$\mu_g(\alpha) \leq \left(1 - \frac{27\gamma^{\frac{5}{2}}}{2n^{\frac{5}{2}}}\right)\mu_g,$$

which completes the proof by Theorem 5.4. □

Remark 2.14 *If we use the* $\mu = \frac{\beta}{1-\beta}\mu_g$ *update when* $\alpha_a < 0.1$, *then analogous to the result of [10, 11] we further can reduce the iteration complexity to*

$$O\left(n^2 \log \frac{(x^0)^T s^0}{\epsilon}\right).$$

3 A Modified Version of Algorithm 1

In this section we slightly modify the Newton system in the corrector step of Algorithm 1 which allows us to improve on the iteration complexity of Algorithm 1 significantly. As one can see from the proof of Lemma 2.2, the bound for the negative components of $\Delta x^a \Delta s^a$ implies a factor of n^2 in the upper bound and subsequently in the iteration complexity of Algorithm 1. To sharpen the bound in Lemma 2.2, we change the Newton system (5) slightly, and use the result of the following lemma whenever the maximum step size in the affine scaling direction is less than a certain threshold, for example $0 < \alpha_a < 0.1$ and also when the maximum step size in the corrector step is below certain threshold.

Lemma 3.1 *Let* $\alpha_a \in (0, 1]$ *be the maximum step size in the predictor step. Then for any* $i \in \mathcal{I}_-$ *one has*

$$-\Delta x_i^a \Delta s_i^a \leq \frac{1}{\alpha_a}\left(\frac{1}{\alpha_a} - 1\right) x_i s_i. \tag{17}$$

Proof: For the maximum step size α_a in the predictor step one has

$$x_i(\alpha_a)s_i(\alpha_a) \geq 0, \quad \forall i \in \mathcal{I}.$$

This is equivalent to

$$(1-\alpha_a)x_i s_i + \alpha_a^2 \Delta x_i^a \Delta s_i^a \geq 0, \quad \forall i \in \mathcal{I}$$

or

$$(1-\alpha_{\rm a})x_is_i \ge -\alpha_{\rm a}^2\Delta x_i^{\rm a}\Delta s_i^{\rm a}, \quad \forall i \in \mathcal{I}$$

Finally

$$-\Delta x_i^{\rm a}\Delta s_i^{\rm a} \le \frac{1}{\alpha_{\rm a}}\left(\frac{1}{\alpha_{\rm a}}-1\right), \quad \forall i \in \mathcal{I}_-.$$

□

Now, when $0 < \alpha_{\rm a} < 0.1$ we modify the corrector step by damping the second order term in the right hand side of the third equation of system (5). The new system becomes

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs - \alpha_{\rm a}\Delta x^{\rm a}\Delta s^{\rm a}. \end{aligned} \tag{18}$$

Analogous to Lemma 2.2 we have the following upper bound for $\|\Delta x\Delta s\|$. Note that the bound given here is much stronger than the one in Lemma 2.2.

Lemma 3.2 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and let $(\Delta x, \Delta y, \Delta s)$ be the solution of (18) with $\mu \ge 0$. Then we have*

$$\|\Delta x\Delta s\| \le 2^{\frac{-3}{2}}\left(\frac{1}{\gamma}\left(\frac{\mu}{\mu_g}\right)^2 - \left(2-\frac{\alpha_{\rm a}}{2\gamma}\right)\frac{\mu}{\mu_g} + \frac{20-4\alpha_{\rm a}+\alpha_{\rm a}^2}{16}\right)n\mu_g.$$

Proof: The proof is analogous to the proof of Lemma 2.2. □

Corollary 3.3 *Let $\mu = (1-\alpha_{\rm a})^3\mu_g$, where $0 < \alpha_{\rm a} < 0.1$. Then*

$$\|\Delta x\Delta s\| \le \frac{1}{2\sqrt{2}\gamma}n\mu_g.$$

In the following theorem we give a lower bound for α_c for this specific case.

Theorem 3.4 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and let $(\Delta x, \Delta y, \Delta s)$ be the solution of (18) with $\mu = (1-\alpha_{\rm a})^3\mu_g$ and $0 < \alpha_{\rm a} < 0.1$. Then*

$$\alpha_{\rm c} \ge \frac{\gamma}{n}.$$

Proof: Following the proof of Theorem 2.12 it is sufficient to have

$$\gamma\left(1-\alpha(1+t\alpha_{\rm a})\right) + (1-\alpha_{\rm a})^3\alpha + \alpha^2\Delta x_i\Delta s_i \ge \gamma\left(1-\alpha+(1-\alpha_{\rm a})^3\alpha\right).$$

By Corollary 3.3 and using the fact that $0 \leq t \leq \frac{1}{4}$ by Lemma 5.1 and $\alpha_{\rm a} < 0.1$, this inequality certainly holds for any α satisfying

$$(1-\gamma)(1-\alpha_{\rm a})^3 - \frac{\gamma\alpha_{\rm a}}{4} - \frac{1}{2\sqrt{2}\gamma}\alpha n \geq 0.$$

This inequality holds for any $\alpha \leq \frac{\gamma}{n}$. Therefore, $\alpha_c \geq \frac{\gamma}{n}$. □

In the sequel we discuss the case when $\alpha_{\rm a} \in [0.1, 1]$.

Following the discussion of the previous section, when (16) holds, then one can guarantee $\alpha_c > 0$. In case of violating (16) we let $\alpha_{\rm a} = \alpha_1$ and proceed with the algorithm. If this results in a small step size α_c in the corrector step, then we cut $\alpha_{\rm a}$ for the second time as it is discussed in Section 2 and use system (18) instead of (5). In the next corollary and the subsequent theorem, analogously to Section 2, we discuss this particular case.

Corollary 3.5 *Let $\mu = \frac{\beta}{1-\beta}\mu_g$, where $\gamma \leq \beta < \frac{1}{4}$ and $\alpha_{\rm a} \in [0.1, 1]$, then*

$$\|\Delta x \Delta s\| \leq \frac{\beta}{\sqrt{2}\gamma(1-\beta)} n\mu_g.$$

Proof: It follows from Lemma 3.2. □

Theorem 3.6 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and let $(\Delta x, \Delta y, \Delta s)$ be the solution of (18) with $\mu = \dfrac{\beta}{1-\beta}\mu_g$. Then the maximum step size $\alpha_{\rm c}$, such that $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) \in \mathcal{N}_\infty^-(\gamma)$, satisfies*

$$\alpha_{\rm c} \geq \frac{\gamma}{\sqrt{2}n}.$$

Proof: Following the proof of Theorem 2.12 it is sufficient to have

$$\gamma\left(1-\alpha(1+\alpha_{\rm a}t)\right) + \frac{\beta}{1-\beta}\alpha + \alpha^2 \Delta x_i \Delta s_i \geq \gamma\left(1-\alpha+\frac{\beta}{1-\beta}\alpha\right).$$

By Corollary 3.5, using the fact that $0 \leq t \leq \frac{1}{4}$ by Lemma 5.1, and $\alpha_{\rm a} \in [0.1, 1]$, this inequality certainly holds for α satisfying

$$(1-\gamma)\frac{\beta}{1-\beta} - \frac{\gamma\alpha_{\rm a}}{4} - \frac{\beta}{\sqrt{2}\gamma(1-\beta)}\alpha n \geq 0.$$

This inequality holds for $\alpha \leq \frac{\gamma}{\sqrt{2}n}$. Therefore, $\alpha_c \geq \frac{\gamma}{\sqrt{2}n}$. □

Based on the previous discussions we can outline the scheme of Algorithm 2, the modified version of Algorithm 1, as follows:

Algorithm 2

Input:

A proximity parameters $\gamma \in (0, \frac{1}{4})$; a safeguard parameter $\beta \in [\gamma, \frac{1}{4})$;
an accuracy parameter $\epsilon > 0$; a starting point $(x^0, y^0, s^0) \in \mathcal{N}_\infty^-(\gamma)$.

begin
while $x^T s \geq \epsilon$ **do**
begin
Predictor Step
Solve (4) and compute the maximum step size α_a such that
$(x(\alpha_\mathrm{a}), y(\alpha_\mathrm{a}), s(\alpha_\mathrm{a})) \in \mathcal{F}$; (the algorithm does not make this step).
If $(1-\alpha_\mathrm{a}) x(\alpha_\mathrm{a})^T s(\alpha_\mathrm{a}) \leq \epsilon$, **then**
let $x = x(\alpha_\mathrm{a}),\ s = s(\alpha_\mathrm{a})$ and **stop.**
end
end
begin
Corrector step
If $\alpha_\mathrm{a} > \alpha_1$, **then** let $\alpha_\mathrm{a} = \alpha_1$.
end
If $\alpha_\mathrm{a} \geq 0.1$, **then** solve (5) with μ defined by (6) and compute the
maximum step size α_c such that $(x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c})) \in \mathcal{N}_\infty^-(\gamma)$;
end
If $\alpha_\mathrm{a} < 0.1$, **then** solve (18) with μ defined by (6) and compute the
maximum step size α_c such that $(x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c})) \in \mathcal{N}_\infty^-(\gamma)$;
end
If $\alpha_\mathrm{c} < \frac{\gamma}{\sqrt{2}n}$, **then** solve (18) with $\mu = \frac{\beta}{1-\beta}\mu_g$ and compute the
maximum step size α_c such that $(x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c})) \in \mathcal{N}_\infty^-(\gamma)$;
end
Set $(x, y, s) = (x(\alpha_\mathrm{c}), y(\alpha_\mathrm{c}), s(\alpha_\mathrm{c}))$.
end
end

Based on the previous results, the following theorem gives the worst case iterations complexity bound of Algorithm 2.

Theorem 3.7 *After at most*

$$\mathcal{O}\left(n^{\frac{3}{2}} \log \frac{(x^0)^T s^0}{\epsilon}\right)$$

number of iterations Algorithm 2 stops with a solution for which $x^T s \leq \epsilon$.

Proof: If $\alpha_a > \alpha_1$, and $\alpha_c \geq \frac{\gamma}{\sqrt{2n}}$, then using (13) we have

$$\mu_g(\alpha) \leq \left(1 - \frac{\gamma(2-3\gamma)}{2\sqrt{2}(1-\gamma)n}\right)\mu_g.$$

If $\alpha_a > \alpha_1$ and $\alpha_c < \frac{\gamma}{\sqrt{2n}}$, then using (13) we have

$$\mu_g(\alpha) \leq \left(1 - \frac{\gamma(1-2\beta)}{\sqrt{2}(1-\beta)n}\right)\mu_g.$$

If $0.1 \leq \alpha_a \leq \alpha_1$ and $\alpha_c \geq \frac{\gamma}{\sqrt{2n}}$, then using (13) we have

$$\mu_g(\alpha) \leq \left(1 - \frac{\gamma}{5\sqrt{2n}}\right)\mu_g.$$

Finally, if $\alpha_a < 0.1$, then using (13) and Theorem 2.1 we have

$$\mu_g(\alpha) \leq \left(1 - \frac{3\gamma^{\frac{3}{2}}}{n^{\frac{3}{2}}}\right)\mu_g,$$

which completes the proof by Theorem 5.4. □

Remark 3.8 *If we use the* $\mu = \frac{\beta}{1-\beta}\mu_g$ *update when* $\alpha_a < 0.1$, *then analogous to the result of [10] we further can reduce the iteration complexity to*

$$O\left(n \log \frac{(x^0)^T s^0}{\epsilon}\right).$$

4 Numerical Results

We have implemented Algorithm 2 using the McIPM software package. The motivation for using the modified algorithm are due to the following facts and observations:

- It enjoys better worst case iteration complexity than Algorithm 1.
- In practice a small α_a for a few problems leads to an increase in the number of iterations.

The implementation is done in MATLAB 7.1 on a Pentium M, 1.7 laptop with 1 GB RAM. In our new approach we let $\gamma = 0.0001$ and $\beta = 0.2$. In Table 1 we give a comparison between our version of the McIPM algorithm and the original McIPM, when McIPM is using the classical Newton search directions. We consider only relatively large problems in the NETLIB test set for linear optimization. The following abbreviations are used for the different versions of McIPM:

- **McIPM**: The implementation of McIPM which is using the classical search direction and taking advantages of several heuristics in determining the target μ value[3].

- **MMcIPM:** The implementation based on our new approach. This goal is achieved by modifying some of the McIPM's subroutines.

These computational results show that our theoretically rigorous IPM may compete with a state of the art software package that takes advantages of various heuristics in its implementation. Further investigation which may improve the computational power of this new approach is left for future research.

Acknowledgments: This research is supported by the NSERC Discovery Grant DG:5-48923, the Canada Research Chair program, and by MITACS.

References

[1] E.D. Andersen and K.D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, Kluwer Academic Publishers, pp. 197–232, 2000.

[2] M. Colombo and J. Gondzio. Further development of multiple centrality correctors for interior point methods, Technical Report MS-05-001, School of Mathematics, The University of Edinburgh, October 2005.

[3] CPLEX: ILOG Optimization. `http://www.ilog.com.`

[4] J. Czyzyk, S. Mehrtotra, M. Wagner, and S.J. Wright. PCx: An interior-point code for linear programming. *Optimization Methods and Software*, 11/12, pp. 397–430, 1999.

[5] N. Megiddo. Pathways to the optimal set in linear programming. In: N. Megiddo, editor, *Progress in Mathematical Programming: Interior Point and Related Methods*, pp. 131–158, 1989. Springer Verlag, New York. Identical version in: *Proceedings of the 6th Mathematical Programming Symposium of Japan, Nagoya, Japan*, pp. 1–35, 1986.

[3]The original implementation of McIPM is taking advantages of self-regular search directions based on a dynamic scheme [19].

Table 1: Comparison of the Number of Iterations

Problem	McIPM	MMcIPM	Problem	McIPM	MMcIPM
25fv47	27	28	pilotnov	26	27
80bau3b	42	43	shell	24	24
bnl1	32	33	ship04l	16	17
bnl2	37	41	ship04s	16	18
cycle	40	40	ship08l	19	20
czprob	32	32	ship08s	17	18
d2q06c	45	48	ship12l	27	29
e226	21	21	ship12s	22	23
forplan	31	30	sierra	18	20
ganges	20	20	stair	18	18
greenbeb	48	49	standata	18	17
maros	32	31	standmps	19	20
modszk1	30	29	stocfor1	14	14
perold	43	46	stocfor2	31	31
pilot	50	55	stocfor3	51	50
pilotja	43	49	truss	20	21
pilotwe	42	43	tuff	19	20
pilot4	37	38	wood1p	15	16
pilot87	78	76	woodw	24	25

[6] S. Mehrotra. On finding a vertex solution using inteior-point methods. *Linear Algebra and its Applications*, 152, pp. 233–253, 1991.

[7] S. Mehrotra. On the implementation of a (primal-dual) interior point method. *SIAM Journal on Optimization*, 2, pp. 575–601, 1992.

[8] S. Mehrotra and M. Li. Convergence conditions and Krylov subspace-based corrections for primal-dual interior point method. *SIAM Journal on Optimization*, 15, pp. 635–653, 2005.

[9] C. Roos, T. Terlaky, and J.-Ph.Vial. *Interior Point Methods for Linear Optimization.* Second ed., Springer Science, Boston, USA, 2006.

[10] M. Salahi. *New Adaptive Interior Point Algorithms for Linear Optimization.* Ph.D. Thesis, McMaster University, March 2006.

[11] M. Salahi, J. Peng, and T. Terlaky. On Mehrotra-Type Predictor-Corrector Algorithms. *Technical Report* 2005/4*, Advanced Optimization Lab. Department of Computing and Software, McMaster University, Hamilto, ON, Canada. http://www.cas.mcmaster.ca/∼oplab/publication.*

[12] M. Salahi and T. Terlaky. Postponing the choice of barrier parameter in Mehrotra-type predictor-corrector algorithms. To appear in *EJOR*.

[13] G. Sonnevend. An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In: A. Prékopa, J. Szelezsán, and B. Strazicky, editors, *System Modeling and Optimization: Proceeding of the 12th IFIP Conference*, Budapest, Hungary, September 1985, Volume 84, Lecture Notes in Control and Information Sciences, pp. 866–876. Springer Verlag, Berlin, 1986.

[14] R.J. Vanderbei. LOQO: An interior-point code for qudaratic programming. *Optimization Methods and Software,* 12, pp. 451–488, 1999.

[15] S.J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, USA, 1997.

[16] XpressMP: Dash Optimization. `http://www.dashoptimization.com.`

[17] Ye, Y., and Anstreicher, K.M., On Quadratic and $O(\sqrt{n}L)$ Convergence of a Predictor-Corrector Algorithm for LCP, *Mathematical Programming,* Vol. 62, pp. 537-551, 1993.

[18] Y. Zhang. Solving large-scale linear programms by inteior point methods under the Matlab enviroment. *Optimization Methods and Software,* 10, pp. 1-31, 1999.

[19] X. Zhu, J. Peng, T. Terlaky, and G. Zhang. On implementing self-regular proximity based feasible IPMs. *Technical Report* 2003/2*, Advanced Optimization Lab. Department of Computing and Software, McMaster University, Hamilton, ON, Canada.* `http://www.cas.mcmaster.ca/~oplab/publication`.

5 Appendix

In this section we present some technical lemmas that are used frequently during the analysis. The next two lemmas are quoted from [11].

Lemma 5.1 *Let* $(\Delta x^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ *be the solution of (4), then*

$$\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}} \le \frac{x_i s_i}{4}, \ \forall i \in \mathcal{I}_+.$$

Lemma 5.2 *Let* $(\Delta x^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ *be the solution of (4), then we have*

$$\sum_{i \in \mathcal{I}_+} \Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}} = \sum_{i \in \mathcal{I}_-} |\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}}| \le \frac{1}{4} \sum_{i \in \mathcal{I}_+} x_i s_i \le \frac{x^T s}{4}.$$

The following technical lemma is used in deriving bounds for $\|\Delta x \Delta s\|$. The proof can be found in [15].

Lemma 5.3 *Let* u *and* v *be any two vectors in* R^n *with* $u^T v \ge 0$. *Then*

$$\|UVe\| \le 2^{\frac{-3}{2}} \|u + v\|^2,$$

where $U = \operatorname{diag}(u)$ *and* $V = \operatorname{diag}(v)$.

The following theorem is used in deriving the final iterations complexity of our new algorithms. It is a slight modification of Theorem 3.2 in [15].

Theorem 5.4 *Let* $\epsilon \in (0, 1)$ *be given. Suppose that the algorithm for solving problem (P) generates a sequence of iterates that satisfy*

$$(x^{k+1})^T s^{k+1} \le \left(1 - \frac{\delta}{n^\omega}\right) (x^k)^T s^k$$

for some positive constants δ *and* ω. *Then one has* $(x^k)^T s^k \le \epsilon$ *for all* $k \ge K$, *where*

$$K = \frac{n^\omega}{\delta} \log \frac{(x^0)^T s^0}{\epsilon}.$$