AMO - Advanced Modeling and Optimization Volume 1, Number 2, 1999

Extending Mehrotra's Corrector for Linear Programs

Florian Jarre and Martin Wechs
Institut für Angewandte Mathematik und Statistik
Universität Würzburg, Am Hubland
D–97074 Würzburg, Federal Republic of Germany

Dec. 1996, revised Oct. 1997, May 1999

Abstract. In this article a primal-dual interior-point method for solving linear programs is proposed. A new approach for generating higher-order search directions, and a new method for an efficient higher-order subspace search along several search directions are the basis of the proposed extension. The subspace search is reduced to a linear program in several variables. The method using the simplest (two-dimensional) subprograms is a slight variation of Mehrotra's predictor-corrector method, and is thus known to be practically very efficient. The higher-dimensional subproblems are guaranteed to be at least as effective — with respect to a certain measure — as the two-dimensional ones. Numerical experiments with the PCx-package indicate that also in practice the higher order subspace search is very cheap and efficient.

AMS 1991 subject classification: 90C05.

Key words. Linear program, infeasible-interior-point method.

1. Introduction

We analyze the Mehrotra corrector for linear programs, and propose a modification to allow for several repetitions of the corrector step. A higher dimensional search along the directions generated by the Mehrotra corrector is reformulated as a linear subprogram that is solved at each iteration of the interior-point method. The resulting method bears similarities with the GMRES method for solving linear systems of equations: By solving a sequence of subproblems of the same type but of lower dimensions, the solution of the original problem is approximated. Here, the subproblems are formulated such that they yield an increasingly better search direction for an interior-point method. When the dimension of the subproblem reaches $n+1$ where n is the dimension of the initial program, the subproblem solves the initial program. However, similar as in the case of GMRES, the approach is only attractive when the dimension of the subproblems remains very small.

Numerical experiments using a modification of the program PCx [1] for solving the NETLIB problems indicate that the new approach is about 20% faster in running time than standard second or third order methods, and the number of iterations is reduced by about 30% in the average.

The theoretical investigation of this method includes a proof of global convergence that covers a long-step variant of Mehrotra's predictor corrector method [8, 7, 10] as a special case.

1.1 Notation

Throughout the paper, we use the following notation. Lower-case letters are used for vectors, and upper-case letters are used for matrices. For a vector denoted by a lower-case letter, the upper-case version means the diagonal matrix whose diagonal elements are the components of the vector. For example,

$$X = \operatorname{diag}(x_1, x_2, \ldots, x_n)$$

is the diagonal matrix associated with the vector $x = [\,x_1 \quad x_2 \quad \cdots \quad x_n\,]^T \in I\!R^n$. By e we denote the vector $e = [\,1 \quad 1 \quad \cdots \quad 1\,]^T \in I\!R^n$. The vector norm $\|x\| = \sqrt{x^T x}$ is always the Euclidean norm, and $\|M\| = \max_{\|x\|=1} \|Mx\|$ denotes the corresponding Euclidean matrix norm. The notation $x > 0$ (respectively, $x \geq 0$) implies that all components of the vector x are positive (respectively, nonnegative). Instead of $x \geq 0$, we also write $x \in I\!R^n_+$. The same symbol, 0, will be used to denote the number zero, the zero vector, and the zero matrix. When 0 denotes the zero vector or the zero matrix, then its dimension will always be clear from the context. The capital letter I is used to denote the (square) identity matrix; its size will always be apparent from the context as well.

1.2 Problem formulation and optimality conditions

We consider linear programs in the standard form

$$\min_{x \in I\!R^n} \{ \; c^T x \quad | \quad Ax = b, \quad x \geq 0 \; \}. \tag{1}$$

We assume that A is a given real $m \times n$ matrix, and b and c are given real vectors of length m and n, respectively. In the sequel, we always assume that A has full row rank. The dual problem to (1) can be written as

$$\max_{s \in I\!R^n, \; y \in I\!R^m} \{ \; b^T y \quad | \quad A^T y + s = c, \quad s \geq 0 \; \}. \tag{2}$$

The Karush-Kuhn-Tucker optimality conditions for the primal-dual problems (1) and (2) can be stated as follows:

$$Ax \; = \; b, \tag{3}$$

$$A^T y + s \; = \; c, \tag{4}$$

$$Xs \; = \; 0, \tag{5}$$

$$x, \, s \; \geq \; 0. \tag{6}$$

Equations (3) and (4) are referred to as primal and dual feasibility, respectively, and the nonlinear equation (5) is called the complementarity condition. Points $(x, y, s) \in I\!R^n_+ \times I\!R^m \times I\!R^n_+$ satisfying (3), (4) and $Xs = \mu e$ are called points on the central path and are denoted by $(x(\mu), y(\mu), s(\mu))$.

2. A Standard Interior-Point Method

Most interior-point algorithms can be viewed as variants of damped Newton's method applied to the system of equations

$$Ax \; = \; b, \tag{7}$$

$$A^T y + s = c, \tag{8}$$

$$Xs = \tilde{r}. \tag{9}$$

Here, $\tilde{r}$ is a suitably chosen vector to be specified later.

Applying Newton's method to (7)–(9) starting at some initial point (x, y, s) with $x > 0$, $s > 0$ yields the system of linear equations

$$A\Delta x = b - Ax, \tag{10}$$

$$A^T \Delta y + \Delta s = c - A^T y - s, \tag{11}$$

$$S\Delta x + X\Delta s = \tilde{r} - Xs, \tag{12}$$

for the Newton correction $(\Delta x, \Delta y, \Delta s)$. We now denote by

$$p := b - Ax, \quad q := c - A^T y - s, \quad \text{and} \quad r := \tilde{r} - Xs \tag{13}$$

the right-hand sides in (10)–(12).

Then, after a reordering, the linear system (10)–(12) can be written in the form

$$\begin{bmatrix} X & S & 0 \\ I & 0 & A^T \\ 0 & A & 0 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ q \\ p \end{bmatrix}, \tag{14}$$

with a 3×3-block matrix. Using the diagonal structure of X, S and I the factorization of the matrix in (14) can be reduced to the factorization of the (lower-dimensional) matrix $AXS^{-1}A^T$, see e.g. [6].

The update

$$x \longrightarrow x + \alpha_x \Delta x, \quad y \longrightarrow y + \alpha_s \Delta y, \quad s \longrightarrow s + \alpha_s \Delta s,$$

is always performed such that strict positivity $x > 0$, $s > 0$ is maintained.

A method that proved to be very efficient in practice is based on Mehrotra's corrector. For the computation of the search directions $(\Delta x, \Delta y, \Delta s)$, Mehrotra's method uses the initial choice $\tilde{r}^{(1)} = 0$, i.e. $r = r^{(1)} = -Xs$. With this choice of r, a prediction $(\Delta x^{(1)}, \Delta y^{(1)}, \Delta s^{(1)})$ is obtained by solving (14). One then sets $\tilde{r}^{(2)} = \mu e - \Delta X^{(1)} \Delta s^{(1)}$, i.e.

$$r^{(2)} = \mu e - Xs - \Delta X^{(1)} \Delta s^{(1)} \tag{15}$$

for some small $\mu \geq 0$ to obtain a correction $(\Delta x, \Delta y, \Delta s) = (\Delta x^{(2)}, \Delta y^{(2)}, \Delta s^{(2)})$ by solving (14) again, but now with $r = r^{(2)}$. A more detailed explanation is given in Section 3.2.

The practical advantages of Mehrotras corrector are that typically the computed correction allows a much longer step length along Δx and Δs without violating the positivity of x and s, and that the computation of the corrector step can use the same factorization of the matrix in (14) as was used for the computation of the prediction $(\Delta x^{(1)}, \Delta y^{(1)}, \Delta s^{(1)})$.

We now summarize a conceptual interior-point algorithm based on Mehrotras predictor-corrector step. The algorithm is a minor modification of the one analyzed in [5].

It either generates an approximate solution satisfying

$$x^T s \leq \varepsilon, \quad \|Ax - b\| \leq \varepsilon_p \quad \text{and} \quad \|A^T y + s - c\| \leq \varepsilon_d \tag{16}$$

where $\varepsilon > 0$, $\varepsilon_p > 0$ and $\varepsilon_d > 0$ are three user specified quantities, or it stops with an iterate satisfying

$$\|(x, s)\| \geq \omega^*, \tag{17}$$

where ω^* is a (very large) user specified number.

Algorithm 1 (Infeasible-interior-point algorithm)

INPUT: *A parameter* $\epsilon > 0$ *and an initial point* (x^0, y^0, s^0) *with* x^0, $s^0 > 0$.

For $k = 0, 1, \ldots,$ *do* :

1) *If* (16) *or* (17) *holds then stop.*
2) *Compute* $(\Delta x, \Delta y, \Delta s)$ *by solving the linear system* (14) *with* $x = x^k$, $y = y^k$, $s = s^k$, $\tilde{r} = \mu e$ *and right-hand side* (13). *If a corrector step is used,* r *may be chosen as in* (15).
3) *Choose* α_x^k, $\alpha_s^k \in (0, 1)$.
4) *Set*

$$\begin{aligned} x^{k+1} &= x^k + \alpha_x^k \Delta x, \\ y^{k+1} &= y^k + \alpha_s^k \Delta y, \\ s^{k+1} &= s^k + \alpha_s^k \Delta s. \end{aligned}$$

In the above algorithm we do not specify the choice of the parameters α_x^k, α_s^k, or μ. We first investigate how to modifiy the search direction to obtain a more efficient method, and then examine convergence of the modified method.

3. Improving the Search Direction

In this section we discuss a new approach to improve the search direction. We start by looking "backwards" at this task and ask what kind of right-hand-side r resp. $\tilde{r}$ should we have chosen to obtain an efficient search direction.

3.1 Finding a better residual

We assume for the moment that the linear program (1) has an optimal solution x^{opt} and its dual (2) has an optimal solution $y^{\text{opt}}, s^{\text{opt}}$. By letting $\Delta x^{\text{opt}} = x^{\text{opt}} - x$, $\Delta s^{\text{opt}} = s^{\text{opt}} - s$, and $\Delta y^{\text{opt}} = y^{\text{opt}} - y$ we see that $\Delta x^{\text{opt}}, \Delta y^{\text{opt}}, \Delta s^{\text{opt}}$ satisfy (14) with p and q as in (13), and $r = r^{\text{opt}} = X\Delta s^{\text{opt}} + S\Delta x^{\text{opt}}$. Hence, there is a "right" choice of $r = r^{\text{opt}}$, and if we knew this choice, the method would converge in one step. By lack of better notation we will call r a residual in the sequel[1].

For any given x, y, s the right choice obviously satisfies

$$r^{\text{opt}} = X\Delta s^{\text{opt}} + S\Delta x^{\text{opt}} \geq -2Xs \tag{18}$$

because $\Delta s^{\text{opt}} \geq -s$ and $\Delta x^{\text{opt}} \geq -x$. We thus obtain a simple lower bound for r^{opt}. A (reasonably small) upper bound for r^{opt}, however, is quite difficult. When Δx^{opt} or Δs^{opt} are large, r^{opt} will

[1] Closely related to the residual vector is the "target vector" defined in Roos et. al. [11], but for our purposes, the concept of a residual vector is more appropriate.

be large as well. If x and s are strictly feasible points on the central path, then $(\Delta x^{\rm opt}, \Delta s^{\rm opt}) \le (n-1)(x,s)$ as shown in [12], Lemma 16. The proof of Lemma 16 in [12] in fact shows that $r^{\rm opt} \le (n-2)Xs$. From (5) and (18) also follows that

$$0 = (X + \Delta X^{\rm opt})(s + \Delta s^{\rm opt}) = r^{\rm opt} + Xs + \Delta X^{\rm opt}\Delta s^{\rm opt},$$

and hence,

$$r^{\rm opt} = -Xs - \Delta X^{\rm opt}\Delta s^{\rm opt}.$$

Next, we review the corrector step.

3.2 Iterating the Mehrotra corrector

We may view Mehrotra's corrector step as the first step in a fixed point iteration. Observe that solving the system (3) – (5) is equivalent to solving the nonlinear system

$$A\Delta x = p, \quad A^T\Delta y + \Delta s = q, \quad \text{and} \quad X\Delta s + S\Delta x = -Xs - \Delta X\Delta s \tag{19}$$

where p and q are as in (13) and $(\Delta x, \Delta y, \Delta s)$ are the correction for (x, y, s). Mehrotra's corrector defines $\Delta x^{(2)}$ and $\Delta s^{(2)}$ by solving the system

$$A\Delta x^{(2)} = p, \quad A^T\Delta y^{(2)} + \Delta s^{(2)} = q, \quad \text{and} \quad X\Delta s^{(2)} + S\Delta x^{(2)} = r^{(2)} := \mu e - Xs - \Delta X^{(1)}\Delta s^{(1)} \tag{20}$$

where $\mu > 0$ is typically chosen rather small. Note that in (20), the right-hand-side of the third equation is changed compared to (19), thus perturbing (19) to a linear system.

In the following we derive an explicit expression for the vectors generated by a repeated recursion of (20), and investigate their convergence properties. As the set of vectors Δx, Δs generated by a repeated recursion of (20) lies in an n-dimensional subspace of $\mathbb{R}^{2n}$ satisfying the first 2 equations of (20), it is convenient to express this recursion in terms of $r = r^{(2)}$.

Let

$$\Lambda := X^{-1/2}S^{-1/2}, \quad D := X^{1/2}S^{-1/2}, \quad \text{and} \quad \Pi_R := DA^T(AD^2A^T)^{-1}AD$$

be the orthogonal projection onto the range of DA^T. By $\Pi_N := I - \Pi_R$ we denote the orthogonal projection onto the null space of AD. Let further be $\bar{x}$ such that $A\bar{x} = b$ (possibly $\bar{x} \not\ge 0$), and $\bar{p}$ such that $A\bar{p} = p$. To compute a vector $\bar{p}$ for some p given by (13) we may set $\bar{p} = \bar{x} - x$. Given some p, q, r in (13), it is straightforward to verify that

$$D^{-1}\Delta x = \Pi_R D^{-1}\bar{p} - \Pi_N Dq + \Pi_N \Lambda r, \tag{21}$$

$$DA^T\Delta y = \Pi_R D^{-1}\bar{p} + \Pi_R Dq - \Pi_R \Lambda r,$$

and

$$D\Delta s = -\Pi_R D^{-1}\bar{p} + \Pi_N Dq + \Pi_R \Lambda r, \tag{22}$$

where $(\Delta x, \Delta y, \Delta s)$ are given by (14). Let $v := \Pi_R D^{-1}\bar{p} - \Pi_N Dq$ and $V = \mathrm{diag}(v)$ be the associated diagonal matrix. Note that

$$\Delta X\Delta s = (D^{-1}\Delta X)(D\Delta s) = \mathrm{diag}(v + \Pi_N\Lambda r)(-v + \Pi_R\Lambda r). \tag{23}$$

Iterating the process (20) yields

$$r^{(j+1)} := \mu e - Xs - \Delta X^{(j)} \Delta s^{(j)}, \tag{24}$$

and hence generates the sequence satisfying the recursion

$$r^{(j+1)} = \Phi(r^{(j)}) := \mu e - Xs + Vv + V(\Pi_N - \Pi_R)\Lambda r^{(j)} - \mathrm{diag}(\Pi_N \Lambda r^{(j)})\Pi_R \Lambda r^{(j)}.$$

The derivative of Φ is given by

$$D\Phi(r) = V(\Pi_N - \Pi_R)\Lambda - \mathrm{diag}(\Pi_N \Lambda r)\Pi_R \Lambda - \mathrm{diag}(\Pi_R \Lambda r)\Pi_N \Lambda.$$

If the spectral radius of $D\phi(r^{\mathrm{opt}})$ is less than one, the fixed point theorem guarantees that, locally, r^{opt} is an attractor. Unfortunately, for degenerate problems, the fixed point r^{opt} is not unique, but belongs to some (small dimensional) convex polyhedron, and hence, it is not surprising that $D\Phi(r^{\mathrm{opt}})$ may have the eigenvalue 1. While the effects of degeneracy may be less important for the recursion (24), there are also fixed points r corresponding to infeasible solutions $x \not\geq 0$ or $s \not\geq 0$ of (3) – (5). Finally, the global behavior of the above iteration for general positive starting points is typically[2] rapidly diverging. Thus, it is difficult to justify a repeated application of the corrector, and to our knowledge, a plain recursion of the corrector as in (24) is not used in practice either.

We therefore modify the Mehrotra corrector and consider the rescaled quantities

$$\widetilde{\Delta x} := D^{-1}\Delta x = \Lambda^{-1} X^{-1} \Delta x \quad \text{and} \quad \widetilde{\Delta s} := D\Delta s = \Lambda^{-1} S^{-1} \Delta s.$$

Let

$$\tilde{v}_x = \Pi_R D^{-1}\bar{p} - \Pi_N Dq - \Pi_N \Lambda X s \quad \text{and} \quad \tilde{v}_s = -\Pi_R D^{-1}\bar{p} + \Pi_N Dq - \Pi_R \Lambda X s.$$

Let Ψ be the function describing the Mehrotra correction. From (21) and (22) follows that the Mehrotra corrector to $(\widetilde{\Delta x},\ \widetilde{\Delta s})$ is given by

$$\begin{bmatrix} \widetilde{\Delta x}^{+} \\ \widetilde{\Delta s}^{+} \end{bmatrix} := \Psi\left(\begin{bmatrix} \widetilde{\Delta x} \\ \widetilde{\Delta s} \end{bmatrix}\right) := \begin{bmatrix} \tilde{v}_x \\ \tilde{v}_s \end{bmatrix} - \begin{bmatrix} \Pi_N \\ \Pi_R \end{bmatrix} \Lambda \widetilde{\Delta X} \widetilde{\Delta s}.$$

The derivative of Ψ is

$$D\Psi\left(\begin{bmatrix} \widetilde{\Delta x} \\ \widetilde{\Delta s} \end{bmatrix}\right) = -\begin{bmatrix} \Pi_N \\ \Pi_R \end{bmatrix} \left(\Lambda \widetilde{\Delta S},\ \Lambda \widetilde{\Delta X}\right) = -\begin{bmatrix} \Pi_N \\ \Pi_R \end{bmatrix} \left(S^{-1}\Delta S,\ X^{-1}\Delta X\right).$$

Obviously,

$$\left\| \begin{bmatrix} \Pi_N \\ \Pi_R \end{bmatrix} \right\| = 1, \quad \text{and} \quad \left\| \left(S^{-1}\Delta S,\ X^{-1}\Delta X\right) \right\| = \max_{1 \le i \le n} \left\{ \frac{\Delta s_i^2}{s_i^2} + \frac{\Delta x_i^2}{x_i^2} \right\}^{1/2}. \tag{25}$$

This establishes a sufficient criterion for the convergence of the Mehrotra corrector; the corrector is locally convergent if $\frac{\overline{\Delta s_i}^2}{s_i^2} + \frac{\overline{\Delta x_i}^2}{x_i^2} < 1$ for all i, where $\overline{\Delta x}$ and $\overline{\Delta s}$ are the (unknown) steps from x to $x(\mu)$ and from s to $s(\mu)$. For the full step with $\mu = 0$ this sufficient condition is never satisfied since either $\Delta s_i^{\mathrm{opt}} = -s_i$ or $\Delta x_i^{\mathrm{opt}} = -x_i$.

[2] This remark refers to a few small size randomly generated linear programs and the starting points $x = s = e$. For starting points near the central path and close to the optimal solution the scheme converged in these examples.

The first step of the Mehrotra corrector proved to be most efficient in practice: For each component i there are two possibilities: Either $\Delta x_i^{(1)}\Delta s_i^{(1)} < 0$ or $\Delta x_i^{(1)}\Delta s_i^{(1)} \geq 0$. In the first case, when for example $\Delta s_i^{(1)} < 0$ and $\Delta x_i^{(1)} > 0$ it follows from $x_i\Delta s_i^{(1)} + s_i\Delta x_i^{(1)} = -x_i s_i$ that $\Delta s_i^{(1)} < -s_i$, the (affine scaling) step is "too long". Setting $r_i^{(2)} = \mu - x_i s_i - \Delta x_i^{(1)}\Delta s_i^{(1)}$ effects that the weighted sum $x_i\Delta s_i^{(2)} + s_i\Delta x_i^{(2)} > x_i\Delta s_i^{(1)} + s_i\Delta x_i^{(1)}$ of Δx_i and Δs_i increases—which can also be seen as a simple heuristics to increase $\Delta s_i^{(1)}$, and thus to improve the affine scaling step. Likewise, if $\Delta x_i^{(1)}\Delta s_i^{(1)} > 0$ it follows $(\Delta x_i^{(1)}, \Delta s_i^{(1)}) > -(x_i, s_i)$. In this case, the weighted sum of Δx_i and Δs_i is reduced by the correction $-\Delta x_i^{(1)}\Delta s_i^{(1)}$ added to r_i. The above heuristics is supported by the following lemma:

Lemma 1 (Monotonicity principle)
The corrections Δx_i and Δs_i are monotone ascending functions of r_i. If A does not contain empty columns or "fixed variables" x_i (such that $Ax = b$ implies $x_i = const$) then Δx_i and Δs_i are strictly monotone ascending functions of r_i.

Proof:
In (14) we replace r by $r + \epsilon e_i$, and denote the solution by $\Delta x(\epsilon)$ and $\Delta s(\epsilon)$. From (21) and (22) follows

$$\Delta s(\epsilon)_i = \Delta s_i + \epsilon\, e_i^T D^{-1}\Pi_R \Lambda e_i,$$
$$\Delta x(\epsilon)_i = \Delta x_i + \epsilon\, e_i^T D \Pi_N \Lambda e_i,$$

where e_i is the i-th unit vector. The diagonal matrices D and Λ are positive. Thus, the lemma follows from the simple fact that any orthogonal projection Π is symmetric positive semidefinite, and hence the relation $e_i^T \Pi e_i \geq 0$ holds true. Moreover, the case that $\Pi_R e_i = 0$ implies that the i-th column of A is zero. Likewise, the case that $\Pi_N e_i = 0$ implies that x_i is a fixed variable. Thus, if empty columns and fixed variables are eliminated in a preprocessing, the dependence of Δx_i and Δs_i on r_i is strictly monotone. #

For the second and later corrections the situation is different: The corrections $\Delta X \Delta s$ in (20) are added to $\mu e - Xs$ and only in the first correction, "$-Xs$" corresponds to the right-hand side of the previous iterate of (24). Furthermore, the three cases distinguished above (namely $\Delta x > 0, \Delta s < 0$, and $\Delta x > 0, \Delta s > 0$, and $\Delta x < 0, \Delta s > 0$) are complemented by a fourth case. In a simple example with $\mu = 0$, we observed the case $\Delta s_i < -s_i$ and $\Delta x_i < -x_i$. (This, of course, implies $r_i < -2x_i s_i$.) The correction $\Delta x_i \Delta s_i > 0$ subtracted from $-x_i s_i$ then resulted in an even smaller value of r_i and also in smaller values of Δx_i and Δs_i. Iterating this process led to a quadratically diverging sequence.
We use two remedies in this situation to modify $r^+ = r^{(j+1)}$ given $r = r^{(j)}$ and $\Delta x = \Delta x^{(j)}$ and $\Delta s = \Delta s^{(j)}$.

First, as observed in (18), the residual can be increased to $r_i^+ \geq -2x_i s_i$. Second, the monotonicity principle can be considered in successive updates: Whenever $\Delta x_i < -x_i$ or $\Delta s_i < -s_i$ we require that $r_i^+ > r_i$ by setting

$$r_i^+ = \begin{cases} \max\{\mu - x_i s_i - \Delta x_i \Delta s_i,\ r_i - 2\min\{(x_i + \Delta x_i)s_i,\ (s_i + \Delta s_i)x_i\}\} & \text{if } \Delta x_i < -x_i \text{ or } \Delta s_i < -s_i, \\ \max\{\mu - x_i s_i - \Delta x_i \Delta s_i,\ -2x_i s_i\} & \text{else.} \end{cases}$$

We note that modifying the residual in a nonconvergent fixed point iteration is essentially different from perturbing the negative components of the Newton direction in an interior-point method. We do *not* propose to change the directions $(\Delta x, \Delta y, \Delta s)$ in the interior-point method to allow for a longer step length at the cost of violating the equality constraints.

3.3 A linear subprogram

To further improve the repeated iteration of the Mehrotra corrector we first discuss how to judge the quality of some given r or the quality of the correction $(\Delta x, \Delta y, \Delta s)$ corresponding to a certain value r: On the one side a step of length (nearly) one without violating the positivity constraints on x and s is desirable to reduce the residuals in (3) and (4), and on the other hand a large reduction in the complementarity $x^T s$ is desired.

Maximizing the step length along the search direction starting from (x, y, s) can simply be written as

$$\max_{t\in[0,1]}\{\ t \ \mid\ x + t\Delta x \geq 0, \quad s + t\Delta s \geq 0\ \}. \tag{26}$$

The maximal value of t in (26) forms a first quality certificate for $\Delta x, \Delta s$ satisfying (10) and (11). Now let

$$\bar{\mu} := \max\{c^T x - b^T y,\ x^T s\}$$

Rather then minimizing the nonlinear quantity $x^T s$ we minimize the linear quantity $c^T x - b^T y$. If the initial point x, y, s satisfies (3) and (4), then both are the same. Reducing the duality gap along $(\Delta x, \Delta y, \Delta s)$ can hence be written as

$$\max_{t\in[0,1]}\{\ t \ \mid\ c^T(x + \Delta x) - b^T(y + \Delta y) \leq \bar{\mu} - t\bar{\mu}\ \}. \tag{27}$$

The maximal value of t in (27) provides a second quality certificate for $\Delta x, \Delta s$. We now observe that $\Delta x, \Delta y, \Delta s$ are affine functions of the vector r. The above programs (26) and (27) are "almost" linear programs with respect to t and the vector r, the nonlinear terms $t\Delta x$ and $t\Delta s$ can easily be eliminated. This is done next.

Let $(\Delta x^{(1)}, \Delta y^{(1)}, \Delta s^{(1)}),\ \dots\ , (\Delta x^{(j)}, \Delta y^{(j)}, \Delta s^{(j)})$ be j directions generated from (14) with different (affinely independent[3]) vectors $r^{(1)},\ \dots\ , r^{(j)}$. Any search step

$$\begin{aligned}
\Delta x = \Delta x(\rho) &= \rho_1 \Delta x^{(1)} + \rho_2 \Delta x^{(2)} + \ldots + \rho_j \Delta x^{(j)},\\
\Delta y = \Delta y(\rho) &= \rho_1 \Delta y^{(1)} + \rho_2 \Delta y^{(2)} + \ldots + \rho_j \Delta y^{(j)},\\
\Delta s = \Delta s(\rho) &= \rho_1 \Delta s^{(1)} + \rho_2 \Delta s^{(2)} + \ldots + \rho_j \Delta s^{(j)},
\end{aligned}$$

with $\rho_1 + \rho_2 + \ldots + \rho_j = 1$ satisfies the equations (10) and (11). Given any j directions generated from (14) with different residuals r in (13), consider the linear program

$$\max_{t\in[0,1]}\left\{\ t \ \left|\ \begin{array}{c} x + \Delta x(\rho) \geq 0 \\ s + \Delta s(\rho) \geq 0 \end{array},\quad \begin{array}{c} c^T(x + \Delta x(\rho)) - b^T(y + \Delta y(\rho)) \leq \bar{\mu} - t\bar{\mu} \\ \rho_1 + \rho_2 + \ldots + \rho_j = t \end{array}\right.\right\} \tag{28}$$

This program combines (26) and (27). It is easy to see that the vector $(\Delta x(\rho), \Delta y(\rho), \Delta s(\rho))$ for which program (28) attains the optimal value $t = 1$ yields the solution to the primal and dual

[3]Here, we call j vectors $r^{(1)},\ \ldots,\ r^{(j)}$ affinely independent if they contain $j-1$ linearly independent vectors. (If $Ax = b$ and $A^T y + s = c$ then the vectors $r^{(l)}$ are chosen linearly independent.)

linear programs (1) and (2). If $r^{(1)} = r^{\rm opt}$ then $\rho_1 = t$, $\rho_2 = \ldots = \rho_j = 0$ is feasible for (28) for all $t \in [0,1]$. Hence, the quantities $(\Delta x^{\rm opt}, \Delta y^{\rm opt}, \Delta s^{\rm opt})$ as in (18) solve (28) with $t = 1$ if, for example, $r^{(1)} = r^{\rm opt}$. Thus, given $n+1$ affinely independent vectors $r^{(l)}$ in $I\!R^n$, and the corresponding quantities $(\Delta x^{(l)}, \Delta y^{(l)}, \Delta s^{(l)})$ (which can all be obtained using the same factorization of (14)), the original linear program can be "reduced"[4] to a linear program of the form (28).

Program (28) can be defined in terms of any j directions $(\Delta x^{(l)}, \Delta y^{(l)}, \Delta s^{(l)})$ $(1 \le l \le j)$ satisfying (10) and (11). The particular choice of the residuals $r^{(l)}$ is interesting only when trying to obtain a value t close to 1 with $j \ll n$ directions in (28).

After eliminating the variable t from $\rho_1 + \rho_2 + \ldots + \rho_j = t$, program (28) is essentially of the form (2). Program (28) has an initial feasible solution $t = 0$ and $\rho = 0$, yet it is about twice the size of (2), and the sparsity structure that may exist in (2) is likely to be lost in (28).

Remark: We point out that the program (28) can be seen as a simple way to perform a search for the next iterate along the subspace spanned by the vectors

$$(\Delta x^{(1)}, \Delta y^{(1)}, \Delta s^{(1)}),\ \ldots\ ,(\Delta x^{(j)}, \Delta y^{(j)}, \Delta s^{(j)}).$$

The merit function is a nonsmooth function. The nonsmooth choice is taken for two reasons: A distance function using the (smooth) Euklidean norm of the deviation from the central path is used in many short step algorithms and allows a proof of polynomial convergence. Yet the Euklidean norm did not prove to be effective in numerical implementations. The program (28) allows to fit the method into the framework of Kojima et al [5] who analyzed a method that proved to be practically efficient. Moreover, solving a suitable modification of (28) is very cheap:

For the subspace search procedures along the j directions $(\Delta x^{(1)}, \Delta y^{(1)}, \Delta s^{(1)})$, $\ldots$,$(\Delta x^{(j)}$, $\Delta y^{(j)}$, $\Delta s^{(j)})$ that are performed at each step of the algorithm in the next section, the constraints $0 \le \rho_l \le 1$ will be added. For this modification, and for given vectors $(\Delta x^{(l)}, \Delta y^{(l)}, \Delta s^{(l)})$, it is likely that (28) can be solved quickly by the simplex method. Assume that (28) is solved by the simplex method for a certain number j of different directions. Then j is increased by 1. As starting vertex for the simplex method the better one, either the previous optimal solution, or t times the newly added direction with maximal t maintaining feasibility is taken. In the context of the simplex method we would like to observe that for small j and some "random" point (x, y, s) generated by the interior-point method it is likely that (28) is nondegenerate even if (1) is not.

3.4 Using the linear subprogram for the generation of the search direction

If the iterates generated from (24) are used for (28), a possible divergence of the $r^{(j)}$ is guaranteed not to harm—except that each additional $r^{(j)}$ costs one extra backsolve. Rather than using (24) for the definition of $r^{(j+1)} = \mu e - Xs - \Delta X \Delta s$, it seems more sensible to use a correction based on the solution $\Delta x(\rho)$ and $\Delta s(\rho)$ from (28). In this case, Δx and Δs are the best approximation to the desired step $\overline{\Delta x}$ and $\overline{\Delta s}$ known at this stage of the algorithm. Note that the "best" approximation $\Delta x, \Delta s$ for the iterative scheme may be one with a step length in $(\bar t, 1]$, where $\bar t$ is the optimal value in (28). Indeed, in our numerical examples it was most efficient to use the correction $\frac{\Delta X(\rho)\Delta s(\rho)}{(\rho^T e)^2}$ for computing the next Mehrotra correction. Here, $\rho^T e = \rho_1 + \ldots + \rho_j$ is a damping factor of the

[4]In the sense that the solution of the original program can be recovered from the solution of the transformed program

search direction in (28). The above scaling by $\frac{1}{(\rho^T e)^2}$ corresponds to the solution $\frac{\Delta x(\rho)}{\rho^T e}$, $\frac{\Delta s(\rho)}{\rho^T e}$ with the value $t = 1$ in (28) but violating the nonnegativity constraints $x + \frac{\Delta x(\rho)}{\rho^T e} \geq 0$, $s + \frac{\Delta s(\rho)}{\rho^T e} \geq 0$.

The modification of Algorithm 1 to include (28) is straightforward. At Step 2) of each iteration, a sequence of vectors $r^{(j)}$ is generated. When the optimal value t of (28) is larger than some value $\bar{t} \in (0, 1)$ or when j exceeds a certain number $\bar{j} \geq 2$, the generation of new vectors $r^{(j)}$ is stopped, and the vector $(\Delta x, \Delta y, \Delta s)$ obtained from (28) is used as result of Step 2). The value $\bar{j}$ will depend on the linear program to be solved; if the factorizations of the matrices in (14) is very cheap, $\bar{j} = 2$ or 3 will be most efficient, for problems with a more expensive factorization larger values of $\bar{j}$ will be more efficient.

4. A Globally Convergent Long Step Method

We add the modifications of Section 3 to Algorithm 1 of Section 2, and specify the range of certain parameters as to guarantee its convergence.

The method analyzed in [5] generates a sequence of iterates in a neighborhood $\mathcal{N}$ that depends on two quantities ε_p, ε_d, and on three parameters $\gamma \in (0, 1)$, $\gamma_p > 0$, and $\gamma_d > 0$. The set $\mathcal{N}$ consists of points $(x, y, s) \in I\!R^n_+ \times I\!R^m \times I\!R^n_+$ satisfying

$$x_i s_i \geq \gamma x^T s / n \quad \text{for} \quad 1 \leq i \leq n \tag{29}$$

$$x^T s \geq \gamma_p \|Ax - b\| \quad \text{or} \quad \|Ax - b\| \leq \varepsilon_p \tag{30}$$

$$x^T s \geq \gamma_d \|A^T y + s - c\| \quad \text{or} \quad \|A^T y + s - c\| \leq \varepsilon_d. \tag{31}$$

Below we show that the method in [5] also converges when using the search directions defined in the previous sections. We modify the rules for the choice of the step length in [5] to allow for the linear subprograms (28). The method in [5] depends on three parameters $0 < \beta_1 < \beta_2 < \beta_3 < 1$. For the method in this paper, the quantities β_2 and β_3 are not needed—their role is absorbed by the linear subprograms that are solved at each iteration—and β_1 is replaced by two quantities β_{11} and β_{12} with $\gamma < \beta_{11} < \beta_{12} < 1$. Instead of $r = \beta_1 \frac{x^T s}{n} e - Xs$ in [5], we choose the initial r as $r^{(1)} = \beta_{11} \frac{x^T s}{n} e - Xs$. The quantity μ in (24) is chosen from $(\beta_{11} \frac{x^T s}{n},\ \beta_{12} \frac{x^T s}{n})$. This modification forms only a mild restriction to the conceptual method of the previous section if $\beta_{11} > 0$ is chosen sufficiently small, and $\beta_{12} < 1$ is large.

In the algorithm below, the number $\tilde{\mu}$ is a bound on the primal and dual infeasibility needed to ensure the first conditions of (30) and (31), and $\bar{\mu} \geq \tilde{\mu}$ is an additional upper bound for the "duality gap". Finally, μ is the current "duality gap" divided by n. We now give a formal statement of the algorithm, and then explain the details of the algorithm, in particular concerning the changes in the linear subprogram compared to (28). The algorithm performs a (very cheap) 'inner loop' at Step 3) of each iteration. To faciliate the notation in the inner loop we omitt the iteration indices of the outer loop.

Algorithm 2

START: Let an initial point $x = x^0 > 0$, $y = y^0$ and $s = s^0 > 0$ and some stopping criteria $\omega^\star, \epsilon^\star, \epsilon_p, \epsilon_d > 0$ be given.
Set $\tilde{\mu} = \tilde{\mu}_0 = \max\{\gamma_p \|Ax - b\|,\ \gamma_d \|A^T y + s - c\|\}$ with $\gamma_p, \gamma_d > 0$.
Define $1 > \hat{\gamma} > \gamma > 0$ where $\gamma \ll n \min_i x_i s_i / \max\{x^T s,\ \tilde{\mu}\}$.
Choose $\gamma < \beta_{11} < \beta_{12} < 1$ with $\beta_{11} \le \sqrt{\gamma/(2n)}$.
(This is possible for $\gamma < 1/(2n)$.)

1) For the current iterate $x > 0$, y and $s > 0$ define

$$\mu = \frac{x^T s}{n}, \qquad \bar{\mu} = \max\{c^T x - b^T y, x^T s, \tilde{\mu}\}.$$

If $\bar{\mu} < \epsilon^\star$, $\|Ax - b\| < \epsilon_p$, and $\|A^T y + s - c\| < \epsilon_d$, or if $\|(x, s)\| > \omega^\star$ then STOP, else

2) Set

$$r^{(1)} = \beta_{11} \mu e - X s$$

3) For $j = 1, 2, \ldots, \bar{j}$ iterate:
a) Compute the solution $\Delta x^{(j)}, \Delta s^{(j)}$ of the linear program LP

$$\begin{array}{rcll}
 & \text{maximize} & t & \\
c^T(x + \Delta x(\rho)) - b^T(y + \Delta y(\rho)) & \le & (1-t)\bar{\mu} & (32)\\
\rho_1 e^T r^{(1)} + \rho_2 e^T r^{(2)} + \ldots + \rho_j e^T r^{(j)} & \le & (\beta_{12} - 1) t\, x^T s & (33)\\
1 \ge \rho_1 + \rho_2 + \ldots + \rho_j & \ge & t & (34)\\
x + \Delta x(\rho) & \ge & (1-t)\hat{x} & (35)\\
s + \Delta s(\rho) & \ge & (1-t)\hat{s} & (36)\\
\rho_1, \rho_2, \ldots, \rho_j & \ge & 0. & (37)
\end{array}$$

where

$$\hat{x}_i := \begin{cases} 0 & \text{if } x_i s_i \ge \hat{\gamma} \max\{\mu, \tilde{\mu}/n\} \\ \frac{1}{2} x_i & \text{else} \end{cases}, \qquad \hat{s}_i := \begin{cases} 0 & \text{if } x_i s_i \ge \hat{\gamma} \max\{\mu, \tilde{\mu}/n\} \\ \frac{1}{2} s_i & \text{else} \end{cases}.$$

b) Set

$$r^{(j+1)} = u^{(j)} + \Delta r^{(j)}$$

with

$$u_i^{(j)} := \begin{cases} \max\left\{ \sigma\mu - x_i s_i - \frac{\Delta x_i^{(j)} \Delta s_i^{(j)}}{(\rho^T e)^2},\ r_i^{(j)} - 2 s_i \left(x_i + \frac{\Delta x_i^{(j)}}{\rho^T e} \right) \right\} & \text{if } \frac{\Delta x_i^{(j)}}{\rho^T e} + x_i < 0 \\ \max\left\{ \sigma\mu - x_i s_i - \frac{\Delta x_i^{(j)} \Delta s_i^{(j)}}{(\rho^T e)^2},\ r_i^{(j)} - 2 x_i \left(s_i + \frac{\Delta s_i^{(j)}}{\rho^T e} \right) \right\} & \text{if } \frac{\Delta s_i^{(j)}}{\rho^T e} + s_i < 0 \\ \max\left\{ \sigma\mu - x_i s_i - \frac{\Delta x_i^{(j)} \Delta s_i^{(j)}}{(\rho^T e)^2},\ -2 x_i s_i \right\} & \text{else} \end{cases}$$

for some $\sigma \in [0, \beta_{12})$ and

$$\Delta r_i^{(j)} = \begin{cases} 0 & \text{if } x_i s_i \geq \hat{\gamma} \max\{\mu, \tilde{\mu}/n\} \\ \max\{\beta_{11}\mu - x_i s_i - u_i^{(j)}, 0\} & \text{else} \end{cases}$$

for $1 \leq i \leq n$.

4) Set

$$\Delta x = \Delta x^{(\bar{j})}, \qquad \Delta y = \Delta y^{(\bar{j})}, \qquad \Delta s = \Delta s^{(\bar{j})},$$

and compute

$$\bar{\alpha} := \max_{\alpha \in [0,1]} \left\{ \alpha \,|\, (X + \alpha \Delta X)(s + \alpha \Delta s) \geq \frac{\gamma}{n} \max\left\{(1 - \alpha\rho^T e)\tilde{\mu},\ (x + \alpha \Delta x)^T (s + \alpha \Delta s)\right\} e \right\}$$

5) Set

$$x^+ := x + \bar{\alpha}\Delta x, \qquad y^+ := y + \bar{\alpha}\Delta y, \qquad s^+ := s + \bar{\alpha}\Delta s, \qquad \tilde{\mu}^+ = \tilde{\mu}(1 - \bar{\alpha}\rho^T e)$$

and go to Step 1) with $x, y, s, \tilde{\mu}$ replaced by $x^+, y^+, s^+, \tilde{\mu}^+$.

Comments

In program LP, condition (34) guarantees that the residuals (10), (11) are reduced by a factor of at least αt if a step of length α along $\Delta x(\rho), \Delta y(\rho), \Delta s(\rho)$ is taken, and

$$\tilde{\mu} = \max\{\gamma_p \|Ax - b\|,\ \gamma_d \|A^T y + s - c\|\} \tag{38}$$

holds for all iterations by the update of $\tilde{\mu}$ in Step 5). Condition (33) is used to guarantee that $x^T s$ is locally reduced along the direction $(\Delta x, \Delta s)$. Because of (34), this condition is always satisfied if, for example, all $r^{(j)}$ are restricted to satisfy

$$(\beta_{11} - 1)x^T s \leq e^T r^{(j)} \leq (\beta_{12} - 1)x^T s. \tag{39}$$

Note that $r^{(j)} + Xs$ is what is often referred to as centering component of the search direction, and thus the "-1" in $(\beta_{11} - 1)$ is a suitable shift. We observe that (39) is certainly satisfied for $r^{(1)}$. For $j > 1$, the first inequality may be enforced by adding a "centering component" ϵe to $r^{(j)}$ or by raising components of $r^{(j)}$ with small $x_i s_i$. Likewise, the second inequality can be enforced by reducing components of $r^{(j)}$ with large $x_i s_i$. When $\rho_i \in [0, 1]$, then, by convexity, condition (39) also holds for the solution of LP.

The definition of $\hat{x}_i$, and $\hat{s}_i$ and $\Delta r_i^{(j)}$ can be understood as follows. Whenever condition (29), or the first part of conditions (30) or (31) are close to being violated, the small components of Xs are "increased" by the choice of $\hat{x}_i$, $\hat{s}_i$ and $\Delta r_i^{(j)}$. (The definition $\hat{x} = \hat{s} = 0$ also allows a proof of convergence; in our experiments, however, a definition with $\hat{x}, \hat{s} \neq 0$ was more efficient.)

To show the global convergence of the algorithm we state some basic relations in Section 4.1 that may be of some interest in other contexts as well. In Section 4.2 we then state the main theoretical result and present a proof thereof.

4.1 Basic properties

We denote the corrections by $(\Delta x(\rho), \Delta s(\rho))$. Here, the dependence on ρ is linear, more precisely, the corrections are convex combinations of the iterates generated in Step 3) above. By $\bar{\rho} = \rho^T e$ we denote the 'net' step length of the correction $(\Delta x(\rho), \Delta s(\rho))$. Here, the name 'net' step length is motivated by the fact

$$\begin{aligned} A(x + \Delta x(\rho)) - b &= -(1-\bar{\rho})p = (1-\bar{\rho})(Ax - b), \\ A^T(y + \Delta y(\rho)) + s + \Delta s(\rho) - c &= -(1-\bar{\rho})q = (1-\bar{\rho})(A^T y + s - c), \end{aligned}$$

where p and q are defined by (13). The last two equations also imply

$$\begin{aligned} (y + \Delta y(\rho))^T b &= (1-\bar{\rho})(y + \Delta y(\rho))^T p + (x + \Delta x(\rho))^T A^T (y + \Delta y(\rho)) \\ &= (1-\bar{\rho})(y + \Delta y(\rho))^T p + (x + \Delta x(\rho))^T (c - (1-\bar{\rho})q - s - \Delta s(\rho)). \end{aligned} \tag{40}$$

Rearranging the terms in (40) yields

$$\begin{aligned} (x + \Delta x(\rho))^T c - (y + \Delta y(\rho))^T b = \quad & (1-\bar{\rho})(x + \Delta x(\rho))^T q \\ & -(1-\bar{\rho})(y + \Delta y(\rho))^T p + (x + \Delta x(\rho))^T (s + \Delta s(\rho)). \end{aligned} \tag{41}$$

For $\rho = 0$ the above equation reduces to

$$y^T p - x^T q = s^T x + y^T b - x^T c. \tag{42}$$

Defining the scaled quantities

$$\Delta x = \bar{\rho}^{-1} \Delta x(\rho), \qquad \Delta s = \bar{\rho}^{-1} \Delta s(\rho), \qquad \Delta y = \bar{\rho}^{-1} \Delta y(\rho), \tag{43}$$

we see that $\Delta x, \Delta y, \Delta s$ satisfy (10), (11), and obtain

$$q^T \Delta x = \Delta x^T (A^T \Delta y + \Delta s) = p^T \Delta y + \Delta x^T \Delta s.$$

It follows

$$q^T \Delta x - p^T \Delta y = \Delta s^T \Delta x.$$

Inserting this and (42) in (41) we get

$$\begin{aligned} c^T(x + \Delta x(\rho)) - b^T(y + \Delta y(\rho)) &= \\ (x + \bar{\rho}\Delta x)^T c - (y + \bar{\rho}\Delta y)^T b &= s^T x + \bar{\rho}(\Delta s^T x + \Delta x^T s) + \bar{\rho}^2 \Delta x^T \Delta s \\ &\qquad -(1-\bar{\rho})(s^T x - c^T x + b^T y - \bar{\rho}\Delta s^T \Delta x) \\ &= \bar{\rho} x^T s + (1-\bar{\rho})(c^T x - b^T y) + \bar{\rho}\Delta s^T \Delta x + \bar{\rho} e^T r \\ &= \bar{\rho}(x + \Delta x)^T (s + \Delta s) + (1-\bar{\rho})(c^T x - b^T y). \end{aligned} \tag{44}$$

Hence, the left hand side of (44) is a convex combination of $(x + \Delta x)^T (s + \Delta s)$ and $(c^T x - b^T y)$. We show that $(\Delta x, \Delta y)$ is a descent direction for $x^T s$, and hence that (44) is reduced at each step. From (21) and (22) follows

$$\begin{aligned} (x + \bar{\rho}\Delta x)^T (s + \bar{\rho}\Delta s) &= x^T s + \bar{\rho} r^T e + \bar{\rho}^2 \Delta x^T \Delta s \\ &= x^T s + \bar{\rho}(e - \Lambda^2 u - \bar{\rho}\Lambda(I - 2\Pi_R)v)^T r - \bar{\rho}^2 v^T v + \bar{\rho} r^T \Lambda^2 u, \end{aligned}$$

where u can be chosen arbitrarily and v is the vector used in (23).

Let $u = r + Xs = X\Delta s + S\Delta x + Xs$ with $\Delta x, \Delta s$ as in (43), then $r = u - Xs = \Lambda^{-2}(\Lambda^2 u - e)$. Using the identity $(I - 2\Pi_R)^T(I - 2\Pi_R) = I$ and the orthogonality of Π_N and Π_R we obtain

$$\begin{aligned}(x + \Delta x(\rho))^T(s + \Delta s(\rho)) \quad &= \quad x^T s - \frac{\bar{\rho}}{2}(e - \Lambda^2 u - \bar{\rho}\Lambda(I - 2\Pi_R)v)^T \Lambda^{-2}(e - \Lambda^2 u - \bar{\rho}\Lambda(I - 2\Pi_R)v) \\ &\quad -\bar{\rho}^2(1 - \frac{\bar{\rho}}{2})v^T v - \frac{\bar{\rho}}{2}(e - \Lambda^2 u)^T \Lambda^{-2}(e - \Lambda^2 u) + \bar{\rho} r^T \Lambda^2 u. \qquad (45)\end{aligned}$$

This can be verified when expanding the nested terms in (45). Using

$$(e - \Lambda^2 u)^T \Lambda^{-2}(e - \Lambda^2 u) = x^T s - 2e^T u + u^T \Lambda^2 u,$$

and

$$r^T \Lambda^2 u = u^T \Lambda^2 u - e^T u,$$

the new duality gap can be bounded by omitting the second and third term in the right hand side of (45):

$$\begin{aligned}(x + \Delta x(\rho))^T(s + \Delta s(\rho)) \quad &\leq \quad (1 - \frac{\bar{\rho}}{2})x^T s + \frac{\bar{\rho}}{2}u^T \Lambda^2 u \qquad (46) \\ &\leq \quad (1 - \frac{\bar{\rho}}{2})\bar{\mu} + \frac{\bar{\rho}}{2}u^T \Lambda^2 u.\end{aligned}$$

In (46) we now set $\bar{\rho} = 1$ and insert this in the right hand side of (44) to obtain

$$\begin{aligned}(x + \Delta x(\rho))^T c - (y + \Delta y(\rho))^T b \quad &\leq \quad \frac{\bar{\rho}}{2}(x^T s + u^T \Lambda^2 u) + (1 - \bar{\rho})(c^T x - b^T y) \\ &\leq \quad (1 - \frac{\bar{\rho}}{2})\bar{\mu} + \frac{\bar{\rho}}{2}u^T \Lambda^2 u. \qquad (47)\end{aligned}$$

These relations are used now to establish global convergence.

4.2 Global Convergence

In this section we first state the main results and then present their proof.

Theorem 1
Algorithm 2 stops after finitely many steps, either with an approximate solution $(x^\star, s^\star)$ and $y^\star$ of (1) and (2) satisfying the stopping criteria, or $(x^\star, s^\star)$ satisfies $\|(x^\star, s^\star)\| > \omega^\star$.

The quantity $\omega^\star$ in the stopping criterion in Algorithm 1 is related to the existence of an optimal solution by the following observation:

Lemma 2 *Let $\underline{\varrho} := \min_i\{x_i^0, s_i^0\}$ and $\overline{\varrho} := \max_i\{x_i^0, s_i^0\}$ be bounds on the components of the starting point. If the linear programs (1), (2) have an optimal solution $(x^{\text{opt}}, y^{\text{opt}}, s^{\text{opt}})$ and if*

$$Ax = b + \theta(Ax^0 - b), \qquad A^T y + s = c + \theta(A^T y^0 + s^0 - c)$$

holds for some $\theta \in [0, 1]$, then

$$\|(x, s)\|_1 \leq \frac{\overline{\varrho}}{\underline{\varrho}}\|(x^{\text{opt}}, s^{\text{opt}})\|_1 + \frac{\theta}{\underline{\varrho}}(x^0)^T s^0 + \frac{n\mu}{\theta\underline{\varrho}},$$

where $\mu = x^T s/n$ as before.

Note that the assumptions of Lemma 2 are always satisfied throughout the algorithm, and that the quantity $\frac{n\mu}{\theta\varrho}$ can be bounded by $\frac{\bar\mu}{\tilde\mu}\frac{n\mu}{\underline{\varrho}}$. The proof of Lemma 2 is given given in Lemma 15 in [9] for the case $\underline{\varrho} = \overline{\varrho}$; the modification to the above formulation is straightforward.

We now prove Theorem 1, and show a slightly stronger convergence result, namely: If Algorithm 2 converges to an approximate solution $(x^\star, s^\star)$ then the convergence rate of Algorithm 2 is linear in the sense that there exist constants $M > 0$ and $c_M < 1$ such that $\bar\mu_k \le M(c_M)^k$ for all k, and $\bar\mu_k$ is the value of $\bar\mu$ of Step 1) at iteration k. The proof of Theorem 1 consists of four parts. The four headers state the topics of the corresponding parts.

Proof of Theorem 1: Assume that $\|(x,s)\| \le \omega^\star$ throughout the algorithm. Else the algorithm terminates and the theorem holds true.

1. **Boundedness of Δx and Δs**
   By (21) and (22) it holds

$$\begin{aligned} X^{-1}\Delta x &= X^{-1}D(v + \Pi_N \Lambda r) \\ S^{-1}\Delta s &= S^{-1}D^{-1}(\Pi_R \Lambda r - v) \end{aligned}$$

   where $\Delta x, \Delta s$ are defined in (43) and v is defined in (23). Observe that

$$X^{-1}D = S^{-1}D^{-1} = \Lambda.$$

   From Step 4) of the algorithm we get

$$\|\Lambda^2\| \le \frac{1}{\gamma \max\{\mu, \tilde\mu/n\}}.$$

   On the other hand, by definition, $v = \Pi_R D^{-1}\bar p - \Pi_N D q$. Let $\bar p$ be of minimal norm such that $A\bar p = p$. From $\tilde\mu = \max\{\gamma_P\|p\|, \gamma_D\|q\|\}$ and

$$\left.\begin{aligned} \|D^{-1}\| = \|D^{-1}S^{-1}S\| \le \|\Lambda\|\|S\| \\ \|D\| = \|DX^{-1}X\| \le \|\Lambda\|\|X\| \end{aligned}\right\} \le \frac{\omega^\star}{\sqrt{\gamma \max\{\mu, \tilde\mu/n\}}}$$

   then follows

$$\|v\| \le \frac{c_1 \omega^\star \tilde\mu}{n\sqrt{\gamma \max\{\mu, \tilde\mu/n\}}}$$

   for a suitably chosen $c_1 > 0$. Here, c_1 depends on the singular values of A, and the factor n in the denominator is added to simplify the notation later on. We may also bound

$$\|\Pi_N \Lambda r\|, \|\Pi_R \Lambda r\| \le \|\Lambda\|\|r\| \le \frac{\beta_{11} n^{1/2}\mu + n^{3/2}\mu}{\sqrt{\gamma \max\{\mu, \tilde\mu/n\}}}$$

   for $r = r^{(1)}$.
   As $r^{(j)}$ is determined recursively by the iteration in Step 3) and $\mu = x^T s/n \le (\omega^\star)^2/n$ is bounded, for fixed $\bar j$ the existence of a constant $c_2 > 0$ follows with

$$\|\Lambda r^{(j)}\| \le \frac{c_2 \mu}{\sqrt{\gamma \max\{\mu, \tilde\mu/n\}}}$$

for $j \le \bar{j}$. (Theoretically, c_2 may become very large when $\bar{j}$ is large, and this "growth of c_2" was in fact observed in numerical experiments where the Mehrotra corrector diverged quadratically with growing $\bar{j}$. However, when truncating the Mehrotra corrector to $r \ge -2Xs$, we could no longer observe such fast divergence.) Summarizing, above estimates imply that

$$\begin{aligned} \|X^{-1}\Delta x\|, \|S^{-1}\Delta s\| &\le \|\Lambda\|(\|v\| + \|\Lambda r\|) \\ &\le \frac{c_1\omega^\star\tilde{\mu}/n + c_2\mu}{\gamma\max\{\mu, \tilde{\mu}/n\}} \\ &\le \frac{c_1\omega^\star + c_2}{\gamma} =: c_3. \end{aligned} \tag{48}$$

2. **Construction of a feasible solution of the linear program in Step 3)**
Recall that by LP we denote the linear sub-program solved at Step 3) of the interior-point method. We consider the constraint (35) in LP. If only the predictor step is used then (35) is satisfied if

$$e + \rho_1 X^{-1}\Delta x^{(1)} \ge \frac{1}{2}e \ge (1-t)X^{-1}\hat{x}.$$

and this is true for $1 - 2\rho_1 c_3 \ge 0$; (the analogous argument, of course, is true for x replaced by s). Hence,

$$\rho_1 = \frac{1}{2c_3}, \qquad t = \rho_2 = \rho_3 = \ldots = \rho_j = 0$$

satisfies (35) and (36) of LP. With (47) and the definition of $r^{(1)}$ follows

$$\begin{aligned} c^T(x + \rho_1\Delta x^{(1)}) - b^T(y + \rho_1\Delta y^{(1)}) &\le (1 - \frac{\rho_1}{2})\bar{\mu} + \frac{\rho_1}{2}\beta_{11}^2\mu^2 e^T\Lambda^2 e \\ &\le (1 - \frac{\rho_1}{2})\bar{\mu} + \frac{\rho_1}{2}\frac{\gamma}{2n}\frac{n\mu^2}{\gamma\max\{\mu, \tilde{\mu}/n\}} \\ &\le \left(1 - \frac{\rho_1}{4}\right)\bar{\mu}. \end{aligned}$$

The treatment of (33) and (34) is straightforward. Hence, with the above choice of ρ, the value $t = \frac{1}{8c_3}$ does not violate the feasibility conditions of LP. The optimal solution of LP therefore satisfies

$$\bar{\rho} := \rho^T e \ge t \ge \frac{1}{8c_3} > 0.$$

3. **Magnitude of $\bar{\alpha}$ in Step 4)**
First, we consider the "small" components i with $x_i s_i < \hat{\gamma}\max\{\mu, \tilde{\mu}/n\}$. We ignore the remaining components, and for brevity of notation, we assume that all i are "small". By the definition of $r^{(j)}$ in Step 3b): $r^{(j)} \ge \beta_{11}\mu e - Xs$ resp. $r(\rho) \ge \bar{\rho}(\beta_{11}\mu e - Xs)$ follows. We distinguish two cases in Step 4). First, the maximum in the constraint of the definition of $\bar{\alpha}$ is given by $(1 - \alpha\bar{\rho})\tilde{\mu}$, and second it is given by $(x + \alpha\Delta x)^T(s + \alpha\Delta s)$.
1) In the first case, with $r = u - Xs = r(\rho)/\bar{\rho}$ we get

$$Xs + \alpha\underbrace{(X\Delta s(\rho) + S\Delta x(\rho))}_{=\bar{\rho}r} + \alpha^2\Delta X(\rho)\Delta s(\rho) - \frac{\gamma}{n}(1 - \alpha\bar{\rho})\tilde{\mu}e$$

$$= (1-\alpha\bar\rho)\underbrace{\left(Xs - \frac{\gamma}{n}\tilde\mu e\right)}_{\geq 0} + \alpha\bar\rho u + \alpha^2\bar\rho^2 \Delta X\Delta s$$
$$\geq \alpha\bar\rho\beta_{11}\mu e + \alpha^2\bar\rho^2 X^{-1}\Delta X(SX)S^{-1}\Delta s.$$

By (48), this is nonnegative whenever

$$\beta_{11}\mu - \alpha\bar\rho n c_3^2\mu \geq 0$$

is true. The latter condition holds when

$$0 \leq \alpha \leq \frac{\beta_{11}}{c_3^2 n} \leq \frac{\beta_{11}}{\bar\rho c_3^2 n},$$

since $\bar\rho \leq 1$.
2) In the second case, a similar estimate holds. Observe that $u^T e \leq \beta_{12} x^T s$ because of the constraint (33) of LP, and $X\Delta s(\rho) + S\Delta x(\rho) = \bar\rho r = \bar\rho(u - Xs)$. Thus,

$$Xs + \alpha(X\Delta s(\rho) + S\Delta x(\rho)) + \alpha^2\Delta X(\rho)\Delta s(\rho)$$
$$-\frac{\gamma}{n}(x^T s + \alpha(x^T\Delta s(\rho) + s^T\Delta x(\rho)) + \alpha^2\Delta x(\rho)^T\Delta s(\rho))e$$
$$= (1-\alpha\bar\rho)\underbrace{(Xs - \gamma\mu e)}_{\geq 0} + \alpha\bar\rho u e - \alpha\bar\rho\frac{\gamma}{n}(u^T e)e + \alpha^2\bar\rho^2\left(\Delta X\Delta s - \frac{\gamma}{n}(\Delta x^T\Delta s)e\right)$$
$$\geq \alpha\bar\rho\beta_{11}\mu e - \alpha\bar\rho\frac{\gamma}{n}\beta_{12}(x^T s)e + \alpha^2\bar\rho^2(\Delta X\Delta s - \frac{\gamma}{n}(X^{-1}\Delta x)^T(XS)(S^{-1}\Delta s)e)$$
$$\geq 0$$

is satisfied if $(\beta_{11} - \gamma\beta_{12})\mu - \alpha\bar\rho(n+\gamma)c_3^2\mu \geq 0$. This implies that

$$0 < \alpha = \frac{\beta_{11} - \gamma\beta_{12}}{(n+\gamma)c_3^2} \leq \frac{\beta_{11} - \gamma\beta_{12}}{\bar\rho(n+\gamma)c_3^2}$$

is a suitable choice for α, because of $\beta_{11} - \gamma\beta_{12} > 0$ since $\beta_{12} < 1 < \beta_{11}/\gamma$ follows from the choice of the constants.

The second (more simple) case $x_i s_i \geq \hat\gamma \max\{\mu, \tilde\mu/n\}$ follows by a similar argument.

4. **Convergence**
With the above results, for each iteration follows

$$\alpha \geq \hat\alpha := \frac{\beta_{11} - \gamma\beta_{21}}{(n+\gamma)c_3^2}, \qquad \bar\rho \geq t \geq \hat\rho := \frac{1}{8c_3}.$$

Thus, $\tilde\mu_k$ is bounded above by

$$\tilde\mu_k \leq (1 - \hat\alpha\hat\rho)^k\tilde\mu_0 \tag{49}$$

and LP guarantees that

$$c^T x^{(k+1)} - b^T y^{(k+1)} \leq (1 - \hat\alpha\hat\rho)\bar\mu_k. \tag{50}$$

From

$$Ax = b - p, \qquad A^T y + s = c - q$$

follows $x^T s = (c-q)^T x - (b-p)^T y$, or more precisely,

$$(x^{(k+1)})^T s^{(k+1)} = c^T x^{(k+1)} - b^T y^{(k+1)} - (q^{(k+1)})^T x^{(k+1)} + (p^{(k+1)})^T y^{(k+1)}. \qquad (51)$$

As a consequence of (38) and (49) we obtain

$$\|p^k\|,\ \|q^k\| \le \frac{\tilde{\mu}^k}{\min\{\gamma_P, \gamma_D\}} \le (1-\hat{\alpha}\hat{\rho})^k \frac{\tilde{\mu}_0}{\min\{\gamma_P,\ \gamma_D\}}.$$

Note that $\|y\| = \|(AA^T)^{-1}A(c-q-s)\| \le \|(AA^T)^{-1}A\|(\|c\| + \tilde{\mu}_0/\gamma_D + \omega^\star) \le c_4\omega^\star$ and $\|x\| \le \omega^\star$ are valid for a $c_4 > 1$ large enough. Inserting these bounds and (50) in (51) yields

$$\bar{\mu}_{k+1} \le \bar{\mu}_k(1-\hat{\alpha}\hat{\rho}) + c_5(1-\hat{\alpha}\hat{\rho})^{k+1}$$

for $c_5 = \frac{2c_4\omega^\star}{\min\{\gamma_P,\gamma_D\}}$. By Proposition 1 below, there exists a constant $c_M < 1$, and a constant $M > 0$ such that

$$\bar{\mu}_k \le M(c_M)^k$$

for all k, i.e. the method is globally linearly convergent. #

Proposition 1 *If for all $k \in I\!N$*

$$0 \le a_{k+1} \le \sigma a_k + \sigma^k \tilde{M}$$

holds true for some fixed values $\sigma < 1$ and $\tilde{M} > 0$, then there exist constants $c_M < 1$ and $M > 0$ such that

$$a_k \le M(c_M)^k$$

for all k.

In the above proposition, of course we associate a_k with $\bar{\mu}_k$. The proof is straighforward:

Proof: If $a_k \le \frac{2\tilde{M}}{1-\sigma}\sigma^k$ for all k then, the statement is trivially true with $c_M = \sigma$ and $M = \frac{2\tilde{M}}{1-\sigma}$. If not, let k_0 be the smallest k with $a_k > \frac{2\tilde{M}}{1-\sigma}\sigma^k$. Then, for all $k \ge k_0$

$$a_k \le \left(\frac{1+\sigma}{2}\right)^{k-k_0} a_{k_0}$$

holds true. This can be seen by induction. Indeed,

$$a_{k+1} \le \sigma a_k + \sigma^k \tilde{M} = \sigma a_k + \sigma^{k-k_0}\sigma^{k_0}\tilde{M}.$$

The last term $\sigma^{k_0}\tilde{M}$ is bounded by $\frac{1-\sigma}{2}a_{k_0}$, and by induction hypothesis, $a_k \le (\frac{1+\sigma}{2})^{k-k_0}a_{k_0}$. Thus, since $\sigma < (1+\sigma)/2$ it follows

$$a_{k+1} \le \left(\sigma\left(\frac{1+\sigma}{2}\right)^{k-k_0} + \sigma^{k-k_0}\frac{1-\sigma}{2}\right)a_{k_0} \le \left(\frac{1+\sigma}{2}\right)^{k-k_0+1} a_{k_0},$$

and the induction is complete. By choosing $c_M = \frac{1+\sigma}{2}$ and $M \ge a_{k_0}(\frac{1+\sigma}{2})^{-k_0}$ large enough, the statement of the proposition also follows for the first k_0 iterates a_j with $j \le k_0$. #

4.3 Implementation

In the previous chapters we have analyzed a globally convergent interior point method. In our implementation we added several modifications to this method that we did not analyze theoretically, but that resulted in better running times of the algorithm for all examples that we tested.

Today, all efficient numerical implementations of interior-point methods for solving linear programs use different steps in the primal and dual space. We have integrated the idea of using different step lengths in the program (28) resulting in the following program

$$
\begin{array}{rcl}
\text{maximize} & t & \\
c^T(x+\Delta x(\rho^x)) - b^T(y+\Delta y(\rho^s)) & \leq & (1-t)\bar{\mu} \\
s^T\Delta x(\rho^x) + x^T\Delta s(\rho^s) & \leq & (\beta_{12}-1)t\, x^T s \\
\lambda\, e^T\rho^x + (1-\lambda)e^T\rho^s & \geq & t \\
e^T\rho^x & \leq & 1 \\
e^T\rho^s & \leq & 1 \\
x+\Delta x(\rho^x) & \geq & (1-t)\hat{x} \\
s+\Delta s(\rho^s) & \geq & (1-t)\hat{s} \\
\rho_1^x, \rho_2^x, \ldots, \rho_k^x, \rho_1^s, \rho_2^s, \ldots, \rho_k^s & \geq & 0.
\end{array} \tag{52}
$$

The parameter λ in the third constraint is chosen in $[0,1]$. It is a "weight factor" which directs the reduction of the primal and dual infeasibility. For $\lambda > 1/2$ a reduction of the primal infeasibility (large $e^T\rho_x$ resulting in small $\|p\|$) is "rewarded" more, and for $\lambda < 1/2$ a reduction of the dual infeasibility is rewarded more. If p and q are small compared to $x^T s$, the right hand side t of this constraint may be replaced by some smaller value like $\frac{t}{2}$, or $\min\{1, \tilde{\mu}/\bar{\mu}\}\ t$. If $p=0$ and $q=0$ the constraint may be omitted. We have defined λ as

$$\lambda := \frac{\gamma_P\|p\|}{\gamma_p\|p\| + \gamma_D\|q\|}.$$

We point out that numerically, it is more efficient (see Czyzyk et.al. [1]) to set $p=0$ and $q=0$ for the corrector steps with $k \geq 2$, and to add the corrector steps to the predictor step (instead of replacing the predictor step with the corrector steps). Mathematically, both is equivalent, but the latter option requires slightly less computations. The changes to adapt (28) or the above program to this approach are straightforward.

In practice, it is not necessary to solve the LP in Step 3) exactly. One might, for example, restrict the positivity constraints to a few components i for which $-\Delta x_i^{(j)}/x_i$ or $-\Delta s_i^{(j)}/s_i$ are large, and thus greatly reduce the size the linear program LP in step 3). Only in the last cycle of Step 3), when $j=\bar{j}$, the exact solution of the linear program is computed. By suitable choice of the indices in LP that are "likely" to be active, and by using efficient updates of these guessed indices in case that the guess was wrong, we succeeded to keep the solution of the linear program LP very fast (for large problems and $j=3$ it took about 1%-2% of the time needed for a back solve, and for the small problems it took up to at most 30% of the back solve time).

For most iterations we did not use any centering component, i.e. we have set $\beta_{11} = 0$. Instead[5], for certain "critical" components of x and s, namely for those components which are nearly active in the linear subprograms we added an extra correction to the Mehrotra corrector. By this we tried to increase components which are at the boundary of the γ-neighborhood $\mathcal{N}$.

We also changed the right-hand side of the linear subprograms and used a formula for $\hat{x}$ and $\hat{s}$ that allowed for larger step lengths at each iteration.

Finally, the choice of parameters describing the neighborhood and the linear subprograms greatly influenced the running time of the algorithm. As the current parameter settings are still subject to modifications based on further numerical experiments we omitt the technical details.

Below we list the CPU time and the number of iterations for various NETLIB problems. The computations were performed on an SGI Origin2000 with the specifications 8 MIPS R10000, 250 MHz CPU's, 1GB=4x256MB RAM and 32MB=8x4MB L2 Cache. Although the SGI Origin2000 is a multi processor machine, our code is using only one CPU with 256MB RAM and 4MB L2 Cache.

The table lists the number of iterations and the running times for different versions of PCx. By 'PCx 0' we denote the version PCx release 1.1, without rescaling, and only using one Mehrotra corrector at each iteration. By 'PCx 1' we denote the current version PCx release 1.1 using higher order corrections proposed by J. Gondzio [2]. The order of the correction is determined adaptively, and depends on the ratio 'cost for back solve' and 'cost for factorization'. The order ranges between 2 (usual Mehrotra corrector) for simple problems and 4-5 (three to four Mehrotra corrector steps) for the more difficult problems *dfl001, pds-10, pilot87*, and for *seba*. Finally, by 'PCx 2' we denote our new modification using $\bar{j} = 3$ for all examples. The overall trend for 'PCx 2' is that the savings in iteration count amount to 30%, and the savings in running time to about 20% in the average. We point out problems *ship12l, fit1p* for which the Cholesky factorization of ADA^T is cheaper than forming the residuals and computing $\Delta x, \Delta y, \Delta s$ (!!). Such problems do not lend themselves to higher-order methods, and for these problems 'PCx 1' uses just one corrector step. Nevertheless, even for these problems our code with a third order correction and solving the associated LP at each cycle of Step 3) still resulted in a slight improvement in the overall running time! For some problems which are nearly infeasible, or which have have a poorly chosen starting point like *bnl1, greenbea,* or *pilot4* our code performs slightly worse than PCx 1; modifications that produce more robust versions of the algorithm are in preparation. Nevertheless we believe that at this point already, the results clearly indicate that the approach of adaptively determiniming the search directions at each step with the help of small-size linear sub-programs is numerically attractive.

We note that all problems are rescaled in the current version PCx release 1.1 used for 'PCx 1' and for our modification 'PCx 2', and for Problem *woodw*, the rescaling made the problem more difficult. For this problem 'PCx 1' uses just the plain Mehrotra corrector for the rescaled problem.

[5]This choice is not covered by our analysis. We therefore include a safeguard, and in case that the value t in LP becomes too small, the quantity β_{11} is raised to some positive value. However, in our examples, the safeguard was never activated. We also note that the use of different step lengths in primal and dual space is not analyzed either; such safeguards for different step lengths in (52) can be used similarly as the one in [5], but are tedious to analyze.

Program	PCx 0		PCx 1		PCx 2	
Problem	It.	sec.	It.	sec.	It.	sec.
25fv47	27	1.28	22	1.17	17	0.86
80bau3b	36	4.39	37	4.17	21	2.99
adlittle	11	0.02	12	0.02	8	0.03
afiro	7	0.01	8	0.01	5	0.01
agg	32	0.46	17	0.29	13	0.22
agg2	24	0.63	20	0.59	16	0.48
agg3	23	0.61	19	0.58	14	0.42
bnl1	44	0.96	39	0.79	43	0.90
bnl2	35	4.41	31	4.47	23	2.88
box1	4	0.03	4	0.02	4	0.03
brandy	19	0.13	16	0.12	15	0.12
cycle	28	2.25	21	1.89	15	1.25
d2q06c	29	6.90	24	6.50	20	4.82
dfl001	58	698.52	46	587.99	36	412.77
e226	21	0.18	16	0.15	12	0.13
fit1d	22	0.51	17	0.48	11	0.37
fit1p	25	0.71	17	0.50	10	0.38
galenet	5	0.00	5	0.00	4	0.00
greenbea	50	4.88	42	4.79	48	4.28
greenbeb	37	3.59	37	4.21	23	2.34
israel	24	0.44	19	0.40	12	0.27
nemsafm	17	0.18	17	0.17	13	0.23
nesm	28	1.27	25	1.34	18	1.01
nl	37	22.01	35	23.81	26	16.14
pds-10	41	382.47	37	384.10	27	268.99
pilot	44	20.55	31	16.40	22	11.01
pilot4	55	1.52	46	1.45	48	1.33
pilot87	43	64.38	30	50.48	24	37.57
pilotja	54	3.91	29	2.45	21	1.69
recipe	9	0.02	9	0.02	7	0.03
seba	16	2.15	12	1.85	9	1.37
ship12l	16	0.49	16	0.46	11	0.39
stocfor2	23	1.11	20	0.84	15	0.72
stocfor3	33	15.25	31	12.81	22	9.64
wood1p	16	1.93	18	2.42	15	1.87
woodw	22	1.35	29	2.15	19	1.18
world	79	188.40	61	178.59	56	122.23

In some examples a further reduction in running time and iteration numbers was possible by setting $\bar{j} = 5$. Overall however, the above value $\bar{j} = 3$ was most efficient. Further refinements of the higher-order corrector steps including an adaptive choice of the order $\bar{j}$ of the corrector step are still in preparation.

---

*Terminates with status “UNKNOWN”.

4.4 Concluding Remarks

The method proposed in this paper generates search directions that are at least as good (measured in terms of (26) and (27)) as the primal-dual direction or as Mehrotra's corrector. Numerical experiments indicate that the method is fast and efficient in practice as well. For interior-point methods using self-dual formulations of the linear proram (1), (2), the computation of the affine scaling direction "costs" about two back solves, while the Mehrora corrector can be written such that it amounts to just one additional back solve. In this case the use of higher order corrections seems particularly promising. Moreover the linear programs (28) can be simplified for methods with feasible starting points. Modifications of the approach in this paper tailored to such self-dual approaches are subject of future research.

Acknowledgements

The autors would like to express their thankfulness to Steve Wright and Joe Czycyk for making the source code of PCx available and for their support when working with PCx. The problem description in sections 1.1 and 1.2 of this paper is taken from [3].

References

[1] Czycyk, J., S. Mehrotra, S.J. Wright (1996). PCx User Guide. Technical Report OTC 96/01, Optimization Technology Center, Argonne National Laboratory, Argonne, Ill.

[2] Gondzio, J. (1996). Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications* **6**, 137–156.

[3] Freund, R. W., F. Jarre (1996). A QMR-based interior-point algorithm for solving linear programs. *Math. Programming* **76** 183–210.

[4] Karmarkar, N.K. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica* **4** 373–395.

[5] Kojima, M., N. Megiddo, S. Mizuno (1993). A primal-dual infeasible-interior-point algorithm for linear programming. *Math. Programming* **61** 263–280.

[6] Lustig, I.J., R.E. Marsten, D.F. Shanno (1991). Computational experience with a primal-dual interior point method for linear programming. *Linear Algebra Appl.* **152** 191–222.

[7] ——, (1992). On implementing Mehrotra's predictor-corrector interior-point method for linear programming. *SIAM J. Optim.* **2** 435–449.

[8] Mehrotra, S., (1992). On the implementation of a primal-dual interior-point method. *SIAM J. Optim.* **2** 575–601.

[9] Mizuno, S., F. Jarre (1996). An infeasible-interior-point algorithm using projections onto a convex set. *Ann. Oper. Res.* **62** 59–80.

[10] Monteiro, R.D.C., I. Adler, M.G.C. Resende (1990). A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research* **15** 191–214.

[11] Roos, C., Terlaky, T., Vial, J.-Ph. (1997). *Theory and Algorithms for Linear Optimization, an Interior Point Approach*, J. Wiley & Sons.

[12] Vavasis, S.A., Y. Ye (1996). A primal-dual interior point method whos running time depends only on the constraint Matrix. *Math. Progr.* **74** 79–120.