# PREDICTOR-CORRECTOR INTERIOR-POINT METHODS FOR LINEAR CONSTRAINED OPTIMIZATION

Neculai Andrei
Research Institute for Informatics
8-10, Averescu Avenue, Bucharest - Romania

September 23, 1997

**Abstract**

Predictor-corrector interior-point algorithms for linear constrained optimization problem are described. The algorithms are based on the Mehrotra's predictor-corrector method, in context of linear programming, modified to take into accout the nonlinearity of the objective function. The convergence of the algorithms, as well as some other properties are proved. Computational results are provided for a number of linear constrained optimization problems in comparison with some codes as TOLMIN (Powell), MINOS (Murtagh & Saunders), SPENBAR (Andrei), NLPQL (Schittkowski).

## 1. Introduction

Since the publication of the article of Karmarkar [1984] an impressive number of papers have been written directed to the convergence study of different variants of interior point methods, as well as to the implementation of the corresponding algorithms for solving the linear programming problem.

All these methods could be classified into three main categories: *projective-scaling* methods, *affine-scaling* methods and *path-following* methods. Projective scaling methods have been proposed by Karmarkar [1984] and studied by numerous researchers along the last decade. Affine scaling was originaly considered by Dikin [1967] and further developed in two directions: *the range-space* and *the null-space* affine-scaling methods. Barnes [1986] and Vanderbei, Meketon and Freedman [1986] proposed a range-space version of the affine-scaling method in *primal* form. Adler, Karmarkar, Resende and Veiga [1986] proposed an affine-scaling method in *dual* form. Nazareth [1987] and Kim and Nazareth [1994] proposed and studied a null-space version of the affine-scaling method in primal form and tested its performance on a number of small relative dense problems from the NETLIB collection, [Gay, 1988]. A very important and significant advantage of the null-space version over the range-space version of the affine-scaling algorithms is that the descent direction can be computed by an iterative method, i.e. it does not have to be computed exactly as it is the case also for the projective-scaling methods.

The path-following algorithms start with the paper of Sonnevend [1985] who introduced the concept of *the analytic center of a polytope* as the unique point that minimizes the barrier function associated to that polytop. The analytic centers of all the constant-cost slices of the feasible set of the polytope determine the so called *the central path*. This path is a region with vey attractive primal-dual properties, and was a key concept in developing the path-following algorithms for linear programming. The Karmarkar's algorithm and its variants was the first algorithm which implements the idea to stay near the central path leading to a polynomial complexity of $O(n^{3.5}L)$ operations, lower than of Khachian which was of $O(n^4L)$. ($L$ is the total number of bits used in the description of the problem data.) Renegar [1988] followed this idea to stay near the central path and obtained the *first* real path-following algorithm with a complexity $O(\sqrt{n}L)$ lower than that of Karmarkar's method $O(nL)$ in terms of number of iterations. Later, Vaidya [1990] and Gonzaga [1991] following a penalty function approach, described algorithms with a complexity of $O(n^3L)$ operations, limit that is still standing. Simultaneously, Kojima, Mizuno and Yoshise [1989a, 1989b], Monteiro and Adler [1989], Ye [1991], Freund [1991] and Gonzaga [1990, 1991, 1992] developed primal-dual path-following algorithms.

Recently Lustig, Marsten and Shanno [1991,1992,1994], and Carpenter, Lustig, Mulvey and Shanno [1993] have introduced a remarcable higher-order primal-dual and predictor-corrector logarithmic barrier methods for linear programming. They developed the theory of these methods and implemented the corresponding algorithms in some professional packages for linear programming (OB1 and OBN) and tested their performances on the NETLIB collection.

Some other efficient LP codes based on interior point methods have been developed. Almost all of them are based on primal-dual or predictor-corrector algorithms, and they differ in many implementational details which are very important. We can mention here the public domain research codes developed by:

Vanderbei - LoQo, written in C, which is an implementation of the predictor-corrector and primal-dual algorithms for LP and QP, being available from:
http://www.sor.princeton.edu/˜rvdb/;

Zhang - LIPSOL, written in MATLAB and FORTRAN, also an implementation of the predictor-corrector method, available from:
http://pc5.math.umbc.edu/˜yzhang/;

Gondzio - HOPDM, written in FORTRN, which is an implementation of the higher order primal-dual algorithm, available from:
http://ecolu-info.unige.ch/˜logilab/software/;

Meszaros - BPMPD, written in FORTRAN, which is another implementation of the higher order primal-dual algorithm, available from:
ftp://ftp.sztaki.hu/pub/oplab/SOFTWARE/BPMPD.

Motivated by the performances of these interior point path-following algorithms some researchers, notably Kortanek, Potra and Ye [1991], Goldfarb, Liu and Wang [1991], Wright [1992], and El-Bakry, Tapia, Tsuchiya and Zhang [1996] have extended these methods and algorithms to the nonlinear problems. In this paper we shall present an extension of the predictor-corrector approach to the linear optimization problems with convex, nonlinear objective function.

## 2. Predictor-Corrector Interior-Point Algorithms

The problem we are concerned with is:

$$\begin{aligned} &min f(x) \\ &subject\ to: \\ &Ax = b, \\ &0 \leq x \leq u, \end{aligned} \tag{2.1}$$

where $f(x) : R^n \rightarrow R$ is a real, convex and at least continously, two-differential function, $A \in R^{mxn}$ and some or all of the upper bounds $u$ may be infinite.

In fact we can consider the problem in the following form:

$$\begin{aligned} &min f(x) \\ &subject\ to: \\ &Ax = b, \\ &x + s = u, \\ &x \geq 0, s \geq 0, \end{aligned} \tag{2.2}$$

where $s \in R^n$ is the vector of slack variables. The predictor corrector interior-point algorithm is motivated by applying the logarithmic barrier function to eliminate the inequality constraints in (2.2). This elimination can be done by incorporating them into the logarithmic barrier function, thus obtaining:

$$\begin{aligned} &min f(x) - \mu \textstyle\sum_{j=1}^{n} \ln x_j - \mu \sum_{j=1}^{n} \ln s_j \\ &subject\ to: \\ &Ax = b, \\ &x + s = u. \end{aligned} \tag{2.3}$$

The Lagrangean function of (2.3) is:

$$L(x, s, y, w, \mu) \;=\; f(x) - \mu \sum_{j=1}^{n} \ln x_j - \mu \sum_{j=1}^{n} \ln s_j - y^T(Ax - b) + w^T(x + s - u). \qquad (2.4)$$

The first-order necessary optimality conditions for a stationary point of (2.4) are:

$$\begin{aligned} Ax &= b, \\ x + s &= u, \\ \nabla f(x) - A^T y - z + w &= 0, \\ XZe &= \mu e, \\ SWe &= \mu e, \end{aligned} \qquad (2.5)$$

where $X, Z, S$ and $W$ are diagonal matrices with the elements $x_j, z_j, s_j$ and $w_j$ respectively, and $z \in R^n$ is a vector of the dual slack variables.

The above optimality conditions express the primal and dual admisibility as well as the complementarity conditions: $x_j z_j = \mu$ and $s_j w_j = \mu, j = 1, ..., n$.

The idea of all interior point algorithms is to solve the optimality conditions (2.5) by generating a sequence of strictly feasible solutions $\{[x\ s\ y\ z\ w]^T\}$ that converges to an optimal feasible pair $\{[x^*\ s^*\ y^*\ z^*\ w^*]^T\}$. The objective of such algorithms is to drive the duality gap to zero. This is implemented by solving the optimality conditions by means of the Newton method. According to the techniques used to solve the nonlinear system (2.5) the interior point methods can be classified in:

1) Primal-Dual interior point methods,
2) Predictor-Corrector interior point methods,
3) Higher-order Predictor-Corrector interior point methods.

For linear programming problems, the primal-dual interior point methods were first introduced by Megiddo [1989] using a logarithmic barrier function approach. Megiddo's idea was developed by Kojima, Mizuno and Yoshise [1989a, 1989b]. A different approach was proposed by Todd and Ye [1990]. This is based on reducing a primal-dual potential function that is simillar to the Karmarkar's primal potential function. Some other works on primal-dual interior point algorithms are described by Monteiro and Adler [1989], Lustig [1988], Gonzaga and Todd [1992], Choi, Monma and Shanno [1990], Lustig, Marsten and Shanno [1990,1991,1992,1994], Zhang, Tapia and Dennis [1992], Zhang and Tapia [1993].

The predictor-corrector approach has been introduced by Mizuno, Todd and Ye [1990] to describe a particular algorithm which considers a linear combination of a primal-dual affine step and a centered step. The predictor step is the primal-dual affine step studied by Monteiro, Adler and Resende [1990]. This step is than corrected by a centering step toward the central path, a concept introduced by Sonnevend [1985] and excellent presented by Gonzaga [1992]. This ideas have been extended and implemented in computing programs by Lustig, Marsten and Shanno [1992], Carpenter, Lustig, Mulvey and Shanno [1993], Vanderbei and Carpenter [1993], Potra [1996], Andersen, Gondzio, Meszaros and Xu [1996].

Mehrotra [1990,1992] introduced the higher-order predictor-corrector interior point methods approach into the arsenal of the interior point methods for linear programming. This follows from the composite Newton method, and has been studied and extended by Bayer and Lagarias [1989], Adler, Karmarkar, Resende and Veiga [1989a,1989b], Domich, Boggs, Rogers and Witzgall 1989], Carpenter, Lustig, Mulvey and Shanno [1993], Tapia, Zhang, Saltzman and Weiser [1996], Gondzio [1996], Andersen, Gondzio, Meszaros and Xu [1996], etc.

Thus, the predictor-corrector interior-point algorithm for (2.1) is derived from the first-order optimality conditions (2.5) by applying the Newton method (to find a solution to (2.5)) for a fixed, positive value of $\mu$, and continuing this until the complementarity gap is reduced to a predetermined value.

Supposing that we have an estimation $t = [x\ s\ y\ z\ w]^T$ of the solution of (2.5) with $x > 0$, $s > 0$, $z > 0$ and $w > 0$, than we can compute the new point $t + \Delta t$, where $\Delta t = [\Delta x\ \Delta s\ \Delta y\ \Delta z\ \Delta w]^T$, is a solution of the following nonlinear algebraic system:

$$\begin{gathered}
A(x + \Delta x) = b, \\
(x + \Delta x) + (s + \Delta s) = u, \\
A^T(y + \Delta y) + (z + \Delta z) - (w + \Delta w) = \nabla f(x + \Delta x), \\
(X + \Delta X)(Z + \Delta Z)e = \mu e, \\
(S + \Delta S)(W + \Delta W)e = \mu e,
\end{gathered} \tag{2.6}$$

where $\Delta X, \Delta S, \Delta Z, \Delta W$ are diagonal matrices with elements $\Delta x_j$, $\Delta s_j$, $\Delta z_j$, and respectively $\Delta w_j$ (j=1,...,n).

Considering the objective $f(x)$ a convex, continuously two-differential function, from (2.6) we obtain:

$$\begin{gathered}
A\Delta x = b - Ax, \\
\Delta x + \Delta s = u - x - s, \\
-\nabla^2 f(x)\Delta x + A^T\Delta y + \Delta z - \Delta w = \nabla f(x) - A^T y + w - z \\
X\Delta z + Z\Delta x = \mu e - XZe - \Delta X\Delta Ze, \\
S\Delta w + W\Delta s = \mu e - SWe - \Delta S\Delta We.
\end{gathered} \tag{2.7}$$

As we can immediately see, the great difference between (2.7) and the corresponding sistem for linear programming case is the presence of the Hessian matrix $\nabla^2 f(x)$ of the objective function.

As in the case of linear programming, the idea of the predictor-corrector interior-point method is to solve the system (2.7) in two stages, [Mehrotra, 1990].

The first stage, the so called *the prediction stage*, we consider $\mu = 0$ and ignore the quadratical terms $\Delta X \Delta Z e$, and $\Delta S \Delta W e$,thus solving the linear system:

$$\begin{aligned} A\Delta x &= b - Ax, \\ \Delta x + \Delta s &= u - x - s, \\ -\nabla^2 f(x)\Delta x + A^T\Delta y + \Delta z - \Delta w &= \nabla f(x) - A^T y + w - z \\ X\Delta z + Z\Delta x &= -XZe, \\ S\Delta w + W\Delta s &= -SWe, \end{aligned} \tag{2.8}$$

subject to the *primal-dual affine direction*: $\Delta t_a = [\Delta x_a \ \Delta s_a \ \Delta y_a \ \Delta z_a \ \Delta w_a]^T$ .

These directions are then used for two distinct purposes: to approximate the quadratical terms in the right-hand side terms of (2.7), and to dynamically estimate the barrier parameter $\mu$.

For linear programming problem, to estimate $\mu$, some authors (Mehrotra [1990, 1992], Lustig, Marsten and Shanno [1991,1992,1994], Carpenter, Lustig, Mulvey and Shanno [1993]) performs the standard ratio test on both the primal and the dual variables in order to determine the step that would actually be taken if the primal-dual affine direction given by (2.8) were used. For linear constrained optimization problems we consider the same strategy, thus defining

$$\begin{aligned} &\hat{\delta}_p = \min\{\min_{j=1,n}\{\tfrac{x_j}{-\Delta x_{aj}}, \Delta x_{aj} < 0\}, \min_{j=1,n}\{\tfrac{s_j}{-\Delta s_{aj}}, \Delta s_{aj} < 0\}\}, \\ \text{and} \\ &\hat{\delta}_d = \min\{\min_{j=1,n}\{\tfrac{z_j}{-\Delta z_{aj}}, \Delta z_{aj} < 0\}, \min_{j=1,n}\{\tfrac{w_j}{-\Delta w_{aj}}, \Delta w_{aj} < 0\}\}, \end{aligned} \tag{2.9}$$

and considering

$$\begin{aligned} \delta_p &= \tau \ \hat{\delta}_p, \\ \delta_d &= \tau \ \hat{\delta}_d \ . \end{aligned}$$

where $\tau \in (0, 1)$, (usually $\tau = 0.99995$).

**Proposition 1**. *The duality gap corresponding to the current solution is:*

$$g = x^T z + s^T w. \tag{2.10}$$

**Proof.** For the problem (2.1) and its dual the duality gap is $g = \nabla f(x)^T x - b^T y + u^T w$. From (2.5) we have: $g = (y^T A + z^T - w^T)x - b^T y + u^T w = z^T x + u^T w - x^T w = x^T z + s^T w$.

The duality gap which would result from a step in the affine direction is :

$$\hat{g} = (x + \delta_p \Delta x_a)^T (z + \delta_d \Delta z_a) + (s + \delta_p \Delta s_a)^T (w + \delta_d \Delta w_a). \qquad (2.11)$$

With these simple algebraic constructions, an estimation of the barrier parameter $\mu$ is then:

$$\mu = \begin{cases} \sigma \left( \frac{\hat{g}}{x^T z + s^T w} \right)^2 \left( \frac{\hat{g}}{n} \right), & if\ x^T z + s^T w \geq 1, \\ \sigma \frac{x^T z + s^T w}{n^2}, & if\ x^T z + s^T w < 1, \end{cases} \qquad (2.12)$$

where $\sigma \in (0, 1)$ is a parameter for modifying the value of $\mu$, (usually $\sigma = 0.1$).

With this value of $\mu$ we can initiate the second stage, the so called the *correction (centering) stage*, in which the following linear algebraic system is solved:

$$\begin{aligned} A\Delta x &= 0, \\ \Delta x + \Delta s &= 0, \\ -\nabla^2 f(x) \Delta x + A^T \Delta y + \Delta z - \Delta w &= 0, \\ X\Delta z + Z\Delta x &= \mu e - \Delta X_a \Delta Z_a e, \\ S\Delta w + W\Delta s &= \mu e - \Delta S_a \Delta W_a e, \end{aligned} \qquad (2.13)$$

subject to *the centering direction* $\Delta t_c = [\Delta x_c\ \Delta s_c\ \Delta y_c\ \Delta z_c\ \Delta w_c]^T$, with which we can compute the actual new direction (the so called predictor corrector direction) $\Delta t = \Delta t_a + \Delta t_c$.

Alternatively, we can compute the *full* direction $\Delta t$ *directly* by solving the linear system:

$$\begin{aligned} A\Delta x &= b - Ax, \\ \Delta x + \Delta s &= u - x - s, \\ -\nabla^2 f(x) \Delta x + A^T \Delta y + \Delta z - \Delta w &= \nabla f(x) - A^T y + w - z \\ X\Delta z + Z\Delta x &= \mu e - XZe - \Delta X_a \Delta Z_a e, \\ S\Delta w + W\Delta s &= \mu e - SWe - \Delta S_a \Delta W_a e, \end{aligned} \qquad (2.14)$$

for $\Delta x$, $\Delta s$, $\Delta y$, $\Delta z$ and $\Delta w$. We notice that the system (2.14) includes the correction terms $\Delta X_a \Delta Z_a e$, and $\Delta S_a \Delta W_a e$ in the right-hand side of the Newton system.

Clearly, the difference between the two algebraic systems (2.8) and (2.13) (or (2.14)) is the right-hand side term, so the matrix algebra for solving (2.13) mainly remains the same as in the solution of (2.8).

With this new direction $\Delta t$ we perform the ratio tests to determine the actual step sizes $\hat{\alpha}_p$ and $\hat{\alpha}_d$:

$$\begin{aligned} &\hat{\alpha}_p = \min\{\min_{j=1,n}\{\tfrac{x_j}{-\Delta x_j}, \Delta x_j < 0\}, \min_{j=1,n}\{\tfrac{s_j}{-\Delta s_j}, \Delta s_j < 0\}\}, \\ \text{and} & \\ &\hat{\alpha}_d = \min\{\min_{j=1,n}\{\tfrac{z_j}{-\Delta z_j}, \Delta z_j < 0\}, \min_{j=1,n}\{\tfrac{w_j}{-\Delta w_j}, \Delta w_j < 0\}\}, \end{aligned} \tag{2.15}$$

and than the actual new point is computed as:

$$\begin{aligned} \tilde{x} &= x + \hat{\alpha}_p \, \Delta x, \\ \tilde{s} &= s + \hat{\alpha}_p \, \Delta s, \\ \tilde{y} &= y + \hat{\alpha}_d \, \Delta y, \\ \tilde{z} &= z + \hat{\alpha}_d \, \Delta z, \\ \tilde{w} &= w + \hat{\alpha}_d \, \Delta w. \end{aligned} \tag{2.16}$$

Given a current estimate to the optimal point that is primal and dual feasible, the direction $\Delta t$ is a *feasible direction* if satisfy:

$$\begin{aligned} A\Delta x &= 0, \\ \Delta x + \Delta s &= 0, \\ -\nabla^2 f(x)\Delta x + A^t \Delta y + \Delta z - \Delta w &= 0. \end{aligned} \tag{2.17}$$

**Proposition 2.** *For any* $\Delta t = [\Delta x \ \Delta s \ \Delta y \ \Delta z \ \Delta w]^T$ *that satisfy (2.17),* $\Delta x^T \Delta z + \Delta s^T \Delta w = \Delta x^T \nabla^2 f(x) \Delta x$.

**Proof.** Using (2.17) we have: $\Delta z = \nabla^2 f(x)\Delta x + \Delta w - A^T \Delta y$, and $\Delta x = -\Delta s$. Hence, $\Delta x^T \Delta z + \Delta s^T \Delta w = \Delta x^T(\nabla^2 f(x)\Delta x + \Delta w - A^T \Delta y) + \Delta s^T \Delta w = \Delta x^T \nabla^2 f(x)\Delta x - \Delta s^T \Delta w - (A\Delta x)^T \Delta y + \Delta s^T \Delta w = \Delta x^T \nabla^2 f(x) \Delta x$.

For the predictor-corrector method, which we consider here, assuming the same steplengths are taken in the primal and the dual subspaces, the change in complementarity is the following:

$$\begin{gathered} (x + \alpha(\Delta x_a + \Delta x_c))^T (z + \alpha(\Delta z_a + \Delta z_c)) + \\ (s + \alpha(\Delta s_a + \Delta s_c))^T (w + \alpha(\Delta w_a + \Delta w_c)) = \end{gathered}$$

$$\begin{gathered}
x^T z + s^T w + \\
\alpha(x^T \Delta z_a + z^T \Delta x_a + x^T \Delta z_c + z^T \Delta x_c + \\
s^T \Delta w_a + w^T \Delta s_a + s^T \Delta w_c + w^T \Delta s_c) + \\
\alpha^2((\Delta x_a + \Delta x_c)^T(\Delta z_a + \Delta z_c) + (\Delta s_a + \Delta s_c)^T(\Delta w_a + \Delta w_c) = \\
x^T z + s^T w - \alpha(x^T z + s^T w) - \alpha(\Delta x_a^T \Delta z_a + \Delta s_a^T \Delta w_a - 2n\mu) + \\
\alpha^2((\Delta x_a + \Delta x_c)^T(\Delta z_a + \Delta z_c) + (\Delta s_a + \Delta s_c)^T(\Delta w_a + \Delta w_c) = \\
x^T z + s^T w - \alpha(x^T z + s^T w + \Delta x_a^T \nabla^2 f(x) \Delta x_a - 2n\mu) + \\
\alpha^2((\Delta x_a + \Delta x_c)^T(\Delta z_a + \Delta z_c) + (\Delta s_a + \Delta s_c)^T(\Delta w_a + \Delta w_c)
\end{gathered}$$

by proposition 2. Since $(\Delta t_a + \Delta t_c)$ is itself a feasible direction, by proposition 2 we obtain:

$$\begin{gathered}
(x + \alpha(\Delta x_a + \Delta x_c))^T(z + \alpha(\Delta z_a + \Delta z_c)) + \\
(s + \alpha(\Delta s_a + \Delta s_c))^T(w + \alpha(\Delta w_a + \Delta w_c)) = \\
x^T z + s^T w - \alpha(x^T z + s^T w + \Delta x_a^T \nabla^2 f(x) \Delta x_a - 2n\mu) + \\
\alpha^2((\Delta x_a + \Delta x_c)^T \nabla^2 f(x)(\Delta x_a + \Delta x_c)).
\end{gathered}$$

In contrast with the primal-dual approach, for the predictor corrector method, although there is a second-order term aparently increasing the complementarity, there is now a new first-order term, $\Delta x_a^T \nabla^2 f(x) \Delta x_a$,reducing the complementarity. Thus, for the case of small steplength, the predictor-corrector method has better complementarity reduction than the primal-dual method. Having in view that $\alpha \in (0,1)$ this proves the convergence of the method.

## 2.1. The Predictor-Corrector Algorithm

The predictor-corrector interior-point algorithm can now be described quite simple. It involves the solving at each iteration of two linear algebraic systems with the same coefficient matrix, but different right-hand side terms, as well as some other simple algebraic manipulations.

**Algorithm PCLC**

*Step1.* Choose an initial approximation $t_0 = [x_0 \; s_0 \; y_0 \; z_0 \; w_0]^T$ for which $x_0 > 0$, $s_0 > 0$, $z_0 > 0$, and $w_0 > 0$. Consider $\sigma = 0.1$ and $k = 0$.

*Step 2.* Test the relative duality gap:

$$\frac{\left|\nabla f(x_k)^T x_k - b^T y_k + u^T w_k\right|}{1 + |f(x_k) - \nabla f(x_k)^T x_k + b^T y_k - u^T w_k|} \leq \varepsilon.$$

*Step 3.* With this estimation of variables, solve the linear algebraic system (2.8) subject to the affine direction $\Delta t_a$.

*Step 4.* Consider $\sigma_k = \min(\sigma, x_k^T z_k + s_k^T w_k)$. Compute the barrier parameter $\mu$ as in (2.12), where $\sigma$ is replaced by the value of $\sigma_k$.

*Step 5.* Solve the linear algebraic system (2.14) for the *full* direction $\Delta t = [\Delta x \; \Delta s \; \Delta y \; \Delta z \; \Delta w]^T$ .

*Step 6.* Choose $\tau \in (0,1)$ and consider $\alpha_p = \tau \overset{\wedge}{\alpha}_p$ and $\alpha_d = \tau \overset{\wedge}{\alpha}_d$, where $\overset{\wedge}{\alpha}_p$ and $\overset{\wedge}{\alpha}_d$ are given by (2.15).

*Step 7.* Consider the new estimation $t_{k+1}$ of the solution:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_p \Delta x, \\ s_{k+1} &= s_k + \alpha_p \Delta s, \\ y_{k+1} &= y_k + \alpha_d \Delta y, \\ z_{k+1} &= z_k + \alpha_d \Delta z, \\ w_{k+1} &= w_k + \alpha_d \Delta w, \end{aligned}$$

set $k = k+1$, and continue with step 2.

Some remarks are in order:

1. The systems we solve to determine the predictor, the corrector, or the full predictor-corrector direction involve the same matrix. That is, each of these directions is obtained based on the same factorization of the matrix.

2. At each iteration of the algorithm we obtain an admisible solution which satisfy the primal and dual constraints. The only motivation to continue the iterations is to reduce the duality gap.

3. In contrast with linear programming problems, in this case the corresponding Newton systems (2.8) and (2.14) involve the Hessian $\nabla^2 f(x)$ of the objective function. This will introduce some complications which we consider later.

**Example 1**. Let us consider the following problem:

$$\min[x_1^2 + \tfrac{1}{2}x_2^2 + x_3^2 + \tfrac{1}{2}x_4^2 - x_1x_3 + x_3x_4 - x_1 - 3x_2 + x_3 - x_4 + e^{x_1} + e^{x_2} + e^{x_3} + e^{x_4}]$$

subject to:

$$\begin{gathered} x_1 + 2x_2 + x_3 + x_4 + x_5 = 5, \\ 3x_1 + x_2 + 2x_3 - x_4 + x_6 = 4, \\ x_2 + 4x_3 - x_7 = 1.5, \\ 0 \leq x_i \leq 10, (i = 1, ..., 7). \end{gathered}$$

The PCLC algorithm initialized into the point: $x_0 = [0.5...0.5]^T, s_0 = [1...1]^T, y_0 = [111]^T, z_0 = [1...1]^T, w_0 = [1...1]^T$ gives the following solution:

| | |
|---|---|
| $x_1$ | 0.04421 |
| $x_2$ | 0.9654 |
| $x_3$ | 0.1336 |
| $x_4$ | 0.0000 |
| $x_5$ | 2.8912 |
| $x_6$ | 2.6346 |
| $x_7$ | 0.0000 |

Considering: $\sigma = 0.1$, $\tau_k = 0.99995$, and $\varepsilon = 10^{-8}$, the characteristics of the optimization process are presented in the following table:

| Nr. | FobP | FobD | $\mu$ | g | $\alpha$ |
|---|---|---|---|---|---|
| 1 | 4.0167 | 562.9565 | 0.4098 | 10.5000 | 0.1149 |
| 2 | 8.1241 | 1955.436 | 12.3577 | 110.9070 | 0.1895 |
| 3 | 6.8792 | 2580.719 | 4.8041 | 645.8158 | 0.1458 |
| 4 | 6.6720 | 4182.258 | $0.3451e^{-2}$ | 1051.835 | 1.5102 |
| 5 | 5.3038 | 51.4250 | $0.1104e^{-1}$ | 4631.276 | 1.0370 |
| 6 | 3.6913 | 3.6800 | $0.6004e^{-2}$ | 42.4592 | 0.9972 |
| 7 | 3.5014 | 3.4820 | $0.9916e^{-3}$ | 0.4859 | 1.0137 |
| 8 | 3.4887 | 3.4870 | $0.1479e^{-4}$ | $0.2692e^{-1}$ | 1.0115 |
| 9 | 3.4872 | 3.4871 | $0.6625e^{-7}$ | $0.1801e^{-2}$ | 1.0035 |
| 10 | 3.4871 | 3.4871 | $0.2484e^{-10}$ | $0.3488e^{-4}$ | 1.0000 |

$g = x_k^T z_k + s_k^T w_k$.(duality gap)
$\alpha = \min\{\alpha_p, \alpha_d\}$.(step length)

As we can see, the algorithm needs a small number of iterations, and than of functions evaluations. At each iteration the current solution is primal and dual feasible. The algorithm continue to execute iterations in order to reduce the complementarity. The value of the barrier parameter $\mu$ has a nonmonotone evolution. This is typical for the interior point methods, being a heritage from the penalty methods, Fiacco and McCormick [1968]. The step length along the full direction of minimization is going to 1, exactly as in the Newton method.

## 2.2. The Multiple Predictor-Corrector Algorithm

As we have already seen, the process of solving the necessary optimality conditions given by the nonlinear system (2.5) is splitted into the solving of the system (2.8) for the affine direction and the solving of the system (2.13) for the correction direction. The affine direction $\Delta t_a$, defined in step 3 of the PCLC algorithm, can very likely point toward the boundary of the positive orthant, thus involving a very small steplength. The role of the centering step $\Delta t_c$,defined by the system (2.13), is to correct this situation. As in the linear programming case, we can consider the *composite Newton method* for the determination of the correction direction, thus obtaining a *higher-order predictor corrector algorithm.*

Considering the optimality conditions (2.5), let us define the following nonlinear transformation:

$$F(t) = \begin{bmatrix} Ax - b \\ x + s - u \\ A^T y + z - w - \nabla f(x) \\ XZe \\ SWe \end{bmatrix} \tag{2.18}$$

Then, the PCLC algorithm in step 3 solves the linear algebraic system:

$$F'(t_k)\Delta t_a = -F(t_k), \tag{2.19}$$

subject to the affine direction $\Delta t_a$. In step 5 the PCLC algorithm solves the following linear algebraic system:

$$F'(t_k)\Delta t = -F(t_k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu - \Delta X_a \Delta Z_a e \\ \mu - \Delta S_a \Delta W_a e \end{bmatrix} \tag{2.20}$$

subject to the full direction $\Delta t$.

The higher-order predictor method considers the same strategy to determine the affine direction, but the composite Newton method for the full direction. Thus, *the level $m_k$ composite Newton interior point method*, at iteration k, instead of solving one linear system (2.20) for the full direction will solve a number of $m_k$ linear systems with different right-hand side terms of the following form:

$$F'(t_k)\Delta t_i = -F(t_k + \sum_{j=0}^{i-1} \Delta t_j) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu - \Delta X_a \Delta Z_a e \\ \mu - \Delta S_a \Delta W_a e \end{bmatrix} \tag{2.21}$$

for i=1,...,$m_k$, where $\Delta t_0 = \Delta t_a$, and define $\Delta t = \sum_{j=0}^{m_k} \Delta t_j$.

Observe that at each iteration of the corresponding algorithm it is necessary to solve $m_k + 1$ linear algebraic systems with the same coefficient matrix. After the coefficient matrix has been factored, the additional work of the composite predictor-corrector method is in the extra backsolve to compute the full direction. What is gained from this extra work is a better approximation of the centering direction.

Thus we can derive a *level-m composite Newton interior point algorithm* which is the same as PCLC in which in step 5 a number of $m_k$ (2.21) linear systems are solved for the full direction.

**Theorem 1**. *The predictor-corrector interior point algorithm PCLC is equivalent to the level-1 composite Newton interior point algorithm.*

**Proof.** This follows immediately from the above discussions.

Instead of following this strategy we can use another one which could be considered as a generalization of the PCLC algorithm in a *multiple predictor-corrector* *procedure.* Instead of solving the correcting system (2.13) once at each iteration of the predictor-corrector interior-point method, it can be solved repetitively with each direction corrected based on the previous direction. The number of corrections $m_k$ in an iteration k is dynamically chosen according to the reduction of the duality gap.

**Algorithm MPCLC**

*Step1.* Choose an initial approximation $t_0 = [x_0\ s_0\ y_0\ z_0\ w_0]^T$ for which $x_0 > 0$, $s_0 > 0$, $z_0 > 0$, and $w_0 > 0$. Consider $\sigma = 0.1$ and $k = 0$.

*Step 2.* Test the relative duality gap:

$$\frac{\left|\nabla f(x_k)^T x_k - b^T y_k + u^T w_k\right|}{1+|f(x_k) - \nabla f(x_k)^T x_k + b^T y_k - u^T w_k|} \leq \varepsilon.$$

*Step 3.* With this estimation of variables, solve the linear algebraic system (2.8) subject to the affine direction $\Delta t_a$.

*Step 4.* Consider $\sigma_k = \min(\sigma, x_k^T z_k + s_k^T w_k)$. Compute the barrier parameter $\mu$ as in (2.12), where $\sigma$ is replaced by the value of $\sigma_k$.

*Step 5.* For $i = 1, ..., m_k$ solve the following linear system for $\Delta t^i = [\Delta x^i\ \Delta s^i\ \Delta y^i\ \Delta z^i\ \Delta w^i]^T$:

$$\begin{aligned} A\Delta x^i &= b - Ax \\ \Delta x^i + \Delta s^i &= u - x - s \\ -\nabla^2 f(x)\Delta x^i + A^T \Delta y^i + \Delta z^i - \Delta w^i &= \nabla f(x) - A^T y + w - z \\ X\Delta z^i + Z\Delta x^i &= \mu e - XZe - \Delta X^{i-1}\Delta Z^{i-1} e \\ S\Delta w^i + W\Delta s^i &= \mu e - SWe - \Delta S^{i-1}\Delta W^{i-1} e \end{aligned} \tag{2.22}$$

where $\Delta X^0 = \Delta X_a, \Delta Z^0 = \Delta Z_a$, $\Delta S^0 = \Delta S_a$ and $\Delta W^0 = \Delta W_a$.
Define $\Delta t = \Delta t^{m_k}$.

*Step 6.* Choose $\tau \in (0, 1)$ and consider $\alpha_p = \tau\ \hat{\alpha}_p$ and $\alpha_d = \tau\ \hat{\alpha}_d$, where $\hat{\alpha}_p$ and $\hat{\alpha}_d$ are given by (2.15).

*Step 7.* Consider the new estimation $t_{k+1}$ of the solution:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_p \Delta x, \\ s_{k+1} &= s_k + \alpha_p \Delta s, \\ y_{k+1} &= y_k + \alpha_d \Delta y, \\ z_{k+1} &= z_k + \alpha_d \Delta z, \\ w_{k+1} &= w_k + \alpha_d \Delta w, \end{aligned}$$

set $k = k + 1$, and continue with step 2.

Observe again that in the MPCLC algorithm, after the coefficient matrix of the system (2.8) has been factored, the additional algebra is in some extra backsolve to compute the full direction.

**Example 2.** Let us consider the same problem as in example 1 above. Considering the same initialization, the MPCLC algorithm gives the same solution, but the characteristics of the optimization process are illustrated in the below table:

| Nr | FobP | FobD | $\mu$ | g | $\alpha$ | $m_k$ |
|---|---|---|---|---|---|---|
| 1 | 5.5787 | -0.5044 | 0.4098 | 10.500 | 0.1005 | 1 |
| 2 | 5.3408 | 7.7489 | 0.0463 | 4.4531 | 0.1868 | 1 |
| 3 | 4.7246 | 7.3848 | $0.3847e^{-2}$ | 2.2778 | 0.3008 | 1 |
| 4 | 4.0469 | 5.8586 | $0.5971e^{-3}$ | 1.3947 | 0.4714 | 1 |
| 5 | 3.4910 | 3.4285 | $0.1307e^{-2}$ | 0.6406 | 1.0081 | 10 |
| 6 | 3.4871 | 3.4871 | $0.6993e^{-5}$ | $0.1851e^{-1}$ | 1.0018 | 7 |
| 7 | 3.4871 | 3.4871 | $0.9767e^{-10}$ | $0.6918e^{-4}$ | 0.9995 | 2 |

$g = x_k^T z_k + s_k^T w_k$.(duality gap)
$\alpha = \min\{\alpha_p, \alpha_d\}$.(step length)
$m_k$ = number of corrections at iteration k.

As we can see the number of centering corrections is relatively reduced. This is in agreement with the theory of composite Newton method, as well as with the numerical experiments from the linear programming case, Ortega and Rheinboldt [1970], Tapia, Zhang, Saltzman and Weiser [1996].

**Theorem 2**. *If $m_k = 1$ for all $k$ then the multiple predictor-corrector algorithm (MPCLC) is equivalent to the predictor-corrector algorithm (PCLC).*

**Proof**. Considering $m_k = 1$ in algorithm MPCLC (step 5), we must solve the system (2.22) which the same linear system as (2.14).

The following theorem is a specialization of the theorem 2.1 by Carpenter, Lustig, Mulvey and Shanno [1993].

**Theorem 3**. *The multiple predictor-corrector algorithm (MPCLC) is equivalent to the composite Newton interior point method.*

To prove the equivalence of these two algorithms, notice that they are identical with the exception of step 5, than the proof of the theorem requires only that the direction $\Delta t$ given by MPCLC is the same as that given by composite Newton interior point method. Before proceeding with the proof of the theorem we need to prove some propositions.

**Proposition 3**. *For any $i \geq 1$, of the composite Newton method, we have:*

$$b - A(x_k + \sum_{j=0}^{i} \Delta x^j) = 0,$$

$$u-(x_k+\sum_{j=0}^{i}\Delta x^j)-(s_k+\sum_{j=0}^{i}\Delta s^j)=0,$$
$$\nabla f(x)-A^T(y_k+\sum_{j=0}^{i}\Delta y^j)-(z_k+\sum_{j=0}^{i}\Delta z^j)+(w_k+\sum_{j=0}^{i}\Delta w^j)=0.$$

**Proof**. From the definition of $F(t)$ in (2.18) we have that

$$A\Delta x^i = b-A(x_k+\sum_{j=0}^{i-1}\Delta x^j),$$
$$\Delta x^i+\Delta s^i = u-(x_k+\sum_{j=0}^{i-1}\Delta x^j)-(s_k+\sum_{j=0}^{i-1}\Delta s^j),$$
$$A^T\Delta y^i+\Delta z^i-\Delta w^i = \nabla f(x)-A^T(y_k+\sum_{j=0}^{i-1}\Delta y^j)-(z_k+\sum_{j=0}^{i-1}\Delta z^j)+(w_k+\sum_{j=0}^{i-1}\Delta w^j).$$

which immediately implies the result of the proposition.

**Proposition 4**. *For any $i\geq 0$, of the composite Newton method, we have:*

$$(X^k+\sum_{j=0}^{i}\Delta X^j)(Z^k+\sum_{j=0}^{i}\Delta Z^j)e=\mu e+(\sum_{j=0}^{i}\Delta X^j)(\sum_{j=0}^{i}\Delta Z^j)e-$$
$$(\sum_{j=0}^{i-1}\Delta X^j)(\sum_{j=0}^{i-1}\Delta Z^j)e,$$
$$(S^k+\sum_{j=0}^{i}\Delta S^j)(W^k+\sum_{j=0}^{i}\Delta W^j)e=\mu e+(\sum_{j=0}^{i}\Delta S^j)(\sum_{j=0}^{i}\Delta W^j)e-$$
$$(\sum_{j=0}^{i-1}\Delta S^j)(\sum_{j=0}^{i-1}\Delta W^j)e.$$

**Proof**. Considering the first relation and multiplying term by term, we have:

$$(X^k+\sum_{j=0}^{i}\Delta X^j)(Z^k+\sum_{j=0}^{i}\Delta Z^j)e=$$
$$X^kZ^ke+\sum_{j=0}^{i}(X^k\Delta Z^j+Z^k\Delta X^j)e+(\sum_{j=0}^{i}\Delta X^j)(\sum_{j=0}^{i}\Delta Z^j)e. \qquad (2.23)$$

But, from the composite Newton interior point method we have that:

$$(X^k\Delta Z^i+Z^k\Delta X^i)=\mu e-(X^k+\sum_{j=0}^{i-1}\Delta X^j)(Z^k+\sum_{j=0}^{i-1}\Delta Z^j)e, \qquad (2.24)$$

which implies that:

$$X^kZ^ke+\sum_{j=0}^{i}(X^k\Delta Z^j+Z^k\Delta X^j)e+(\sum_{j=0}^{i-1}\Delta X^j)(\sum_{j=0}^{i-1}\Delta Z^j)e=\mu e. \qquad (2.25)$$

Now, adding and substracting in the right-hand side of equation (2.23) the term $(\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e$ we get:

$$(X^k + \sum_{j=0}^{i} \Delta X^j)(Z^k + \sum_{j=0}^{i} \Delta Z^j)e = X^k Z^k e + \sum_{j=0}^{i} (X^k \Delta Z^j + Z^k \Delta X^j)e +$$
$$(\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e + (\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e - (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e.$$

Using (2.25) to substitute $\mu e$ for the first three terms we get the result of the proposition. Applying the same approach to the second relation in S and W matrices we obtain the complete proof of the proposition.

Now combining the results of the propositions 3 and 4 we have that

$$F(t^k + \sum_{j=0}^{i} \Delta t^j) = \begin{bmatrix} b - A(x_k + \sum_{j=0}^{i} \Delta x^j) \\ u - (x_k + \sum_{j=0}^{i} \Delta x^j) - (s_k + \sum_{j=0}^{i} \Delta s^j) \\ \nabla f(x) - A^T(y_k + \sum_{j=0}^{i} \Delta y^j) - (z_k + \sum_{j=0}^{i} \Delta z^j) + (w_k + \sum_{j=0}^{i} \Delta w^j) \\ (X^k + \sum_{j=0}^{i} \Delta X^j)(Z^k + \sum_{j=0}^{i} \Delta Z^j)e \\ (S^k + \sum_{j=0}^{i} \Delta S^j)(W^k + \sum_{j=0}^{i} \Delta W^j)e \end{bmatrix} \tag{2.26}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu e + (\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e - (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e \\ \mu e + (\sum_{j=0}^{i} \Delta S^j)(\sum_{j=0}^{i} \Delta W^j)e - (\sum_{j=0}^{i-1} \Delta S^j)(\sum_{j=0}^{i-1} \Delta W^j)e \end{bmatrix} \tag{2.27}$$

To prove the theorem 3 we must prove that the direction $\Delta t = \sum_{i=0}^{m_k} \Delta t^i$ is the same as the direction $\Delta t^{m_k}$ given by the MPCLC algorithm.

To do this, we note that $\Delta t$ solves the following system:

$$\begin{gathered} A\Delta x = b - Ax, \\ \Delta x + \Delta s = u - x - s, \\ A^T \Delta y + \Delta z - \Delta w = \nabla f(x) - A^T y + w - z, \\ Z\Delta x + X\Delta z = \mu e - XZe - (\sum_{i=0}^{m_k - 1} \Delta X^i)(\sum_{i=0}^{m_k - 1} \Delta Z^i)e, \end{gathered}$$

$$W\Delta s + S\Delta w = \mu e - SWe - (\sum_{i=0}^{m_k-1} \Delta S^i)(\sum_{i=0}^{m_k-1} \Delta W^i)e.$$

Thus, $\Delta t^{m_k}$ and $\Delta t$ solve precisely the same system when $(\sum_{i=0}^{m_k-1} \Delta t^i) = \Delta t^{m_k-1}$.

**Proposition 5**. *For all $p \geq 0$ we have that $(\sum_{i=0}^{p} \Delta t^i) = \Delta t^p$.*

**Proof**. (by induction) For $p = 0$, the result of proposition is true; $\Delta t^0$ given by the two algorithms is the affine direction.

Assume that the proposition is true for $1 \leq p \leq m-1$. We must prove that it is true for $p = m$. By adding all the systems defining each $\Delta t^i$ we have that $\sum_{i=0}^{m} \Delta t^i$ solves the system:

$$-F(t) + \sum_{i=0}^{m-1} [\mu \hat{e} - F(t + \sum_{j=0}^{i} \Delta t^j)] = \begin{bmatrix} A \sum_{i=0}^{m} \Delta x^i \\ \sum_{i=0}^{m} \Delta x^i + \sum_{i=0}^{m} \Delta s^i \\ A^T(\sum_{i=0}^{m} \Delta y^i) + \sum_{i=0}^{m} \Delta z^i - \sum_{i=0}^{m} \Delta w^i \\ Z(\sum_{i=0}^{m} \Delta x^i) + X(\sum_{i=0}^{m} \Delta z^i) \\ W(\sum_{i=0}^{m} \Delta s^i) + S(\sum_{i=0}^{m} \Delta w^i) \end{bmatrix} \tag{2.28}$$

where $\hat{e}$ is a vector with 1 in the last 2n components and 0 otherwise.

Replacing $F(t + \sum_{j=0}^{i} \Delta t^j)$ from (2.26) in (2.28) we get:

$$-F(t) + \sum_{i=0}^{m-1} [\mu \hat{e} - F(t + \sum_{j=0}^{i} \Delta t^j)] =$$

$$\begin{bmatrix} b - Ax \\ u - x - s \\ \nabla f(x) - A^T y - z + w \\ \mu e - XZe - \Delta X^0 \Delta Z^0 e + \{\sum_{i=1}^{m-1} (\mu e - [\mu e + (\sum_{j=0}^{i} \Delta X^j)(\sum_{j=0}^{i} \Delta Z^j)e - \\ (\sum_{j=0}^{i-1} \Delta X^j)(\sum_{j=0}^{i-1} \Delta Z^j)e])\} \\ \mu e - SWe - \Delta S^0 \Delta W^0 e + \{\sum_{i=1}^{m-1} (\mu e - [\mu e + (\sum_{j=0}^{i} \Delta S^j)(\sum_{j=0}^{i} \Delta W^j)e - \\ (\sum_{j=0}^{i-1} \Delta S^j)(\sum_{j=0}^{i-1} \Delta W^j)e])\} \end{bmatrix}$$

Taking into consideration the above relations (2.25), within the braces the $\mu$ terms cancel as do alternating terms in the summation over i. This implies

$$
-F(t)+ \sum_{i=0}^{m-1} [\mu \hat{e} - F(t + \sum_{j=0}^{i} \Delta t^j)] =
$$

$$
\begin{bmatrix} b - Ax \\ u - x - s \\ \nabla f(x) - A^T y - z + w \\ \mu e - XZe - (\sum_{i=0}^{m-1} \Delta X^i)(\sum_{i=0}^{m-1} \Delta Z^i)e \\ \mu e - SWe - (\sum_{i=0}^{m-1} \Delta S^i)(\sum_{i=0}^{m-1} \Delta W^i)e \end{bmatrix} =
$$

$$
\begin{bmatrix} b - Ax \\ u - x - s \\ \nabla f(x) - A^T y - z + w \\ \mu e - XZe - \Delta X^{m-1} \Delta Z^{m-1} e \\ \mu e - SWe - \Delta S^{m-1} \Delta W^{m-1} e \end{bmatrix}
$$

the last equality being true by the induction hypothesis.

Thus, $\sum_{i=0}^{m} \Delta t^i$ and $\Delta t^m$ are the solution to the same system, which prove the proposition.

**Proof of the Theorem 3**. The proof follows directly from proposition 5 applied when $p = m_k$.

In order to complete the algorithm MPCLC we must specify the number of centering corrections in step 5. This number is dynamically selected, based on the complementarity that would result if a step is considered. Let $g^i$ be the complementarity that would result taking a step in the direction obtained after i corrections. We perform the $(i+1)$-th correction only if $g^i < g^{i-1}$ and i is less than the specified maximum number of corrections. If $g^i \geq g^{i-1}$ then the correcting process is stopped and the direction $\Delta t^{i-1}$ is used. This is the strategy considered into an experimental implementation of the algorithm MPCLC.

### 2.3. Cubic Convergence

The convergence analysis of the predictor-corrector algorithms follows directly from the books by Ortega and Rheinboldt [1970] and Dennis and Schnabel [1983], and the paper by Tapia, Zhang, Saltzman and Weiser [1996].

The pure Newton method applied to the system $F(t) = 0$, where $F(t)$ is given in (2.18) can be written as:

$$
N(t) = t - F'(t)^{-1} F(t).
$$

On the other hand, the predictor-corrector interior point method can be written as:

$$\hat{N}(t) = t - \alpha F'(t)^{-1}[F(t) + F(N(t)) - \mu \hat{e}].$$

Therefore

$$\begin{gathered}
\hat{N}(t) - t^* = t - t^* - F'(t)^{-1}[F(t) + F(N(t))] + \\
(1-\alpha)F'(t)^{-1}[F(t) + F(N(t))] + \alpha\mu F'(t)^{-1} \hat{e} = \\
F'(t)^{-1}[F(t)(N(t)) - F(N(t)) - F'(t)t^*] + \\
(1-\alpha)F'(t)^{-1}[F(t) + F(N(t))] + \alpha\mu F'(t)^{-1} \hat{e} = \\
F'(t)^{-1}\{F(N(t)) - F(t^*) - F'(t^*)(N(t) - t^*)] + \\
[F'(t^*) - F'(t)](N(t) - t^*)\} + \\
(1-\alpha)F'(t)^{-1}[F(t) + F(N(t))] + \alpha\mu F'(t)^{-1} \hat{e}.
\end{gathered}$$

In a neighborhood of the solution $t^*$ from the standard Newton method analysis we know that $\| N(t) - t^* \| = O(\| t - t^* \|^2)$.

But, $\| F(t) \| = O(\| t - t^* \|)$ and $\| F(N(t)) \| = O(\| t - t^* \|^2)$. Hence, from the above relation we obtain immediately:

$$\begin{gathered}
\| \hat{N}(t) - t^* \| = O(\| t - t^* \|^4) + O(\| t - t^* \|^3) + \\
| 1 - \alpha | O(\| t - t^* \|) + \mu O(1),
\end{gathered}$$

which can be simplified to:

$$\| \hat{N}(t) - t^* \| = O(\| t - t^* \|^3) + | 1 - \alpha | O(\| t - t^* \|) + \mu O(1). \quad (2.29)$$

Now, the term $\mu O(1)$ can be made $O(\| t - t^* \|^3)$ by the choice of $\mu$. The only term which must be very carefully analysed is $| 1 - \alpha | O(\| t - t^* \|)$. It is quite clear that for cubic convergence we need $| 1 - \alpha |$ to be $O(\| t - t^* \|^2)$.

In the case of linear programming problems, for primal-dual interior point method, Zhang, Tapia and Dennis [1992], assuming strict complementarity, $t_k$ is feasible, and $t^*$ is a nondegenerate vertex solution, obtained the following useful expresion:

$$| 1 - \alpha_k | = \tfrac{1 - \tau_k + \sigma_k \theta_k}{1 - \sigma_k \theta_k} + O(x_k^T y_k), \quad (2.30)$$

where $\tau_k$ and $\sigma_k$ are as in the PCLC algorithm, and $\theta_k \in (\frac{1}{n}, 1]$. This is also valid for the predictor-corrector interior point method. Since for feasible $t_k$ we have $x_k^T y_k = \| F(t_k) \|_1$, than $O(x_k^T y_k) = O(\| t - t^* \|)$.

The predictor-corrector methods could be viewed as perturbed Newton method. The perturbation to the right-hand side is $\mu \hat{e} - F(t - F'(t)^{-1}F(t))$. From the above estimation of $| 1 - \alpha_k |$ we see that independent of the choices for $\tau_k$ and $\sigma_k$, this term is at best $O(\| t - t^* \|)$ and the predictor-corrector interior point method, even for nondegenerate problems, cannot be shown to be cubically convergent. However, by choosing $\alpha_k = 1$ near solution and $\mu_k = O((x_k^T y_k)^3)$, from (2.29) we see that it is possible to obtain cubic convergence. All these observations could be assembled into the

**Theorem 4**. *Let $t_k = [x_k\ s_k\ y_k\ z_k\ w_k]^T$ be generated by the PCLC predictor corrector interior point algorithm with $t_0$ strictly feasible. Assume:*
*1) Strict complementarity,*
*2) $x^*$ is a nondegenerate vertex, and*
*3) The sequence $\{t_k\}$ converges to $t^*$.*
*If the choice of $\sigma_k$ and $\tau_k$ satisfy:*

$$0 \leq \sigma_k \leq \min(\sigma, c_1(x_k^T y_k)) \tag{2.31}$$

*and*

$$0 \leq \tau_k \leq \min(\tau, 1 - c_2(x_k^T y_k)) \tag{2.32}$$

*where $\sigma \in [0,1), \tau \in (0,1)$ and $c_1, c_2 > 0$, then the convergence is quadratic, i.e. there is the positive constant $\gamma_1 > 0$ such that for k large*

$$\| t_{k+1} - t^* \| \leq \gamma_1 \| t_k - t^* \|^2 .$$

*On the other hand, if instead of (2.31) we have*

$$0 \leq \sigma_k \leq \min(\sigma, c_1(x_k^T y_k)^2) \tag{2.33}$$

*and for large k $\alpha_k = 1$, then the convergence is cubic, i.e. there exist the positive constant $\gamma_2 > 0$ such that for k large*

$$\| t_{k+1} - t^* \| \leq \gamma_1 \| t_k - t^* \|^3 .$$

**Proof**. The proof follows directly from the above discussions combined with some details given in Zhang, Tapia and Dennis [1992] and in Tapia, Zhang, Saltzman and Weiser [1996].

The remarkable aspect of the theorem 4 is that if into a neighborhood of the solution the steplength one is selected (like in the Newton method), and the centering barrier parameter $\mu$ is of the order of the duality gap cubed (like Mehrotra suggests), then for nondegenerate problems, the predictor-corrector algorithms have a cubic convergence. Therefore, the predictor-corrector interior point method and its variants, for solving the linear constrained optimization problems, should be implemented in such a manner that near solution the centering barrier parameter is zero and the steplength is one, exactly as in the classical Newton method.

In order to consolidate the understanding of the behaviour of the predictor-corrector interior point algorithms we shall present a large-scale linear constrained optimization problem.

**Example 3**. Let us consider the problem:

$$\min \sum_{k=1}^{3m} x_k e^{-x_k}$$

subject to:

$$\begin{gathered} x_1 + 2x_m + x_{2m} = 5, \\ x_j + 2x_{m+j} + x_{2m+j} = 10 + \tfrac{1}{j}, \\ (j = 1, ..., m), \\ 0 \le x_j \le 10. \end{gathered}$$

The algorithms PCLC si MPCLC, initialized in the point: $x_0 = [0.1...0.1]^T$, for different values of m, give the results from the following table.

| m | n | vfo | PCLC | | | MPCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 500 | 1500 | 84.279940 | 10 | 11 | 0:0:15:77 | 8 | 9 | 0:0:10:44 |
| 1000 | 3000 | 168.469280 | 10 | 11 | 0:0:57:62 | 8 | 9 | 0:0:32:40 |
| 1200 | 3600 | 202.149807 | 10 | 11 | 0:1:19:91 | 8 | 9 | 0:0:43:50 |
| 1500 | 4500 | 252.673138 | 10 | 11 | 0:2:0:67 | 8 | 9 | 0:1:4:3 |
| 2000 | 6000 | 336.882943 | 10 | 11 | 0:3:40:30 | 8 | 9 | 0:1:55:02 |
| 3000 | 9000 | 505.311133 | 10 | 11 | 0:7:41:54 | 8 | 9 | 0:3:52:50 |

m = number of constraints.
n = number of variables.
vfo = value of the objective function.
it = number of iterations.
nf = number of objective function evaluations.

## 2.4. Alternative formulation

The systems of equations (2.8) and (2.14) with the coefficient matrix

$$\begin{bmatrix} A & & & & \\ I & I & & & \\ -\nabla^2 f(x) & & A^T & I & -I \\ Z & & & X & \\ & W & & & S \end{bmatrix} \tag{2.34}$$

are usually not solved directly, instead they are algebraically simplified.

Firstly, simple algebraic manipulations show that the solving of the above linear systems could be reduced to that of factorizing the matrix:

$$\begin{bmatrix} -[\nabla^2 f(x) + X^{-1}Z + S^{-1}W] & A^T \\ A & \end{bmatrix} \tag{2.35}$$

Note that the matrix (2.35) is symmetric whereas the original matrix (2.34) is not. On the other hand, the reduction process from (2.34) to (2.35) entails no off-diagonal fill-in in the system of equations, which is very appealing. However, the matrix (2.35) is indefinite.

The factorization of symmetric indefinite matrices, like (2.35) have been considered by Vanderbei and Carpentner [1993] (the packages LoQo and ALPO for linear programming, [Vanderbei, 1990a, 1990b, 1991]), Gill, Saunders and Shinnerl [1996], Gill, Murray, Ponceleón and Saunders, [1995], Fourer and Mehrotra, [1991]. It is shown by Gill, Murray, Ponceleón and Saunders, [1995], and Fourer and Mehrotra, [1991] that performing the Bunch-Parlett [1971] factorization of the indefinite matrix (2.35) will prevent fill-in.

Secondly, it is quite clear that the reduction process of the above linear systems (2.8) and (2.14) could be continued, finally the computation of the predictor-corrector direction involving the factorization of the symmetric matrix:

$$A(\nabla^2 f(x) + X^{-1}Z + S^{-1}W)^{-1}A^T. \tag{2.36}$$

Due to the convexity of the function $f(x)$, the advantage of factorizing the matrix (2.36) over (2.35) is that it is based on a symmetric positive definite matrix guaranted to yeld a factorization of the lorm $L\Lambda L^T$, where $L$ is a lower triangular matrix with unit diagonal elements, and $\Lambda$ is a diagonal matrix with strictly positive entries.

However, the great disadvantage of solving the system with matrix (2.36) is that it suffers a great deal of fill-in when (2.36) is formed. This is especially true due to the presence of the Hessian $\nabla^2 f(x)$ of the objective function, and if the A matrix has a number of dense columns.

For linear programming problems, most implementations of interior point methods, like that by Lustig, Marsten and Shanno [1991, 1992, 1994] (the package OB1), Carpenter, Lustig, Mulvey and Shanno [1993] (the package OBN), Gondzio [1996] (the package HOPDM), Andersen and Andersen [1997] (the package APOS), or Meszaros [1994] (the package BPMPD) are based on factorizing the matrix (2.36) where $\nabla^2 f(x) = 0$.

For linear constrained optimization problems, into an experimental code, we shall consider the solving of the linear system with the coefficient matrix (2.35).

## 3. Numerical Examples

In this section we shall present some numerical examples.

*Problem P1.* (Hock and Schittkowski [1981], Problem 112, page 121, Chemical equilibrium)

$$\min \sum_{j=1}^{10} x_j (c_j + \ln \frac{x_j}{x_1 + \ldots + x_{10}})$$

subject to:

$$\begin{gathered} x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2, \\ x_4 + 2x_5 + x_6 + x_7 = 1, \\ x_3 + x_7 + x_8 + 2x_9 + x_{10} = 1. \end{gathered}$$

Table P1 ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

| $c_j$ | *Lower bound* | $x_0$ | $x^*$ | *Upper bound* |
|---|---|---|---|---|
| -6.089 | 0.0001 | 0.1 | 0.040668 | 100.0 |
| -17.164 | 0.0001 | 0.1 | 0.147732 | 100.0 |
| -34.054 | 0.0001 | 0.1 | 0.783150 | 100.0 |
| -5.914 | 0.0001 | 0.1 | 0.001414 | 100.0 |
| -24.721 | 0.0001 | 0.1 | 0.485246 | 100.0 |
| -14.986 | 0.0001 | 0.1 | 0.000693 | 100.0 |
| -24.100 | 0.0001 | 0.1 | 0.27399 | 100.0 |
| -10.708 | 0.0001 | 0.1 | 0.017947 | 100.0 |
| -26.662 | 0.0001 | 0.1 | 0.037314 | 100.0 |
| -22.179 | 0.0001 | 0.1 | 0.096872 | 100.0 |

$f(x^*) = -47.76109$.

*Problem P2.*

$$\min \sum_{k=1}^{13} (\tfrac{1}{k} x_k^2 + k x_k + k^2)^2$$

subject to:

$$
\begin{gathered}
x_1 + x_3 + x_5 + x_7 + x_9 = 10, \\
x_2 + 2x_4 + 3x_6 + 4x_8 + 5x_{10} = 20, \\
2x_2 - 5x_5 + 8x_4 = 30, \\
-x_1 + 3x_4 - 5x_7 + x_{10} - x_{11} = 0, \\
x_1 + 2x_2 + 4x_4 + 8x_8 + x_{12} = 100, \\
x_1 + 3x_3 + 6x_6 - 9x_9 + x_{13} = 50.
\end{gathered}
$$

Table P2. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

| *Lower bound* | $x_0$ | $x^*$ | *Upper bound* |
|---|---|---|---|
| 0.0 | 0.1 | 5.666234 | 100.0 |
| 0.0 | 0.1 | 4.966837 | 100.0 |
| 0. | 0.1 | 4.333765 | 100.0 |
| 0.0 | 0.1 | 2.5 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 2.5 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 1.833765 | 100.0 |
| 0.0 | 0.1 | 54.33377 | 100.0 |
| 0.0 | 0.1 | 31.33247 | 100.0 |

$f(x^*) = 1562547.3$
*Problem P3.*

$$
\min \sum_{k=1}^{15} x_k^2 + \sum_{k=1}^{14} (x_k + x_{k+1})
$$

subject to:

$$
\begin{gathered}
0.1x_1 + 0.1x_7 + 0.3x_8 + 0.2x_9 + 0.2x_{11} = 1, \\
0.1x_2 + 0.2x_8 + 0.3x_9 + 0.4x_{10} + x_{11} = 2, \\
0.1x_3 + 0.2x_8 + 0.3x_9 + 0.3x_9 + 0.4x_{10} + 2x_{11} = 3, \\
x_4 + x_8 + 0.5x_9 + 0.5x_{10} + x_{11} - x_{12} = 3, \\
2x_5 + x_6 + 0.5x_7 + 0.5x_8 + 0.23x_9 + 0.25x_{10} + 0.5x_{11} - x_{13} = 4, \\
x_4 + x_6 + x_8 + x_9 + x_{10} + x_{11} - x_{14} = 5, \\
0.1x_1 + 1.2x_7 + 1.2x_8 + 1.4x_9 + 1.1x_{10} + 2x_{11} - x_{15} = 6.
\end{gathered}
$$

Table P3. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

| *Lower* *bound* | $x_0$ | $x^*$ | *Upper* *bound* |
|---|---|---|---|
| 0.0 | 0.1 | 0.452456 | 100.0 |
| 0.0 | 0.1 | 0.266273 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 1.096372 | 100.0 |
| 0.0 | 0.1 | 0.048186 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 1.492774 | 100.0 |
| 0.0 | 0.1 | 1.507982 | 100.0 |
| 0.0 | 0.1 | 0.489488 | 100.0 |
| 0.0 | 0.1 | 1.026627 | 100.0 |
| 0.0 | 0.1 | 0.518137 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 0.0 | 100.0 |
| 0.0 | 0.1 | 0.539443 | 100.0 |

$f(x^*) = 21.718976.$
*Problem P4.*

$$\min[3x_1 e^{-0.1x_1x_6} + 4x_2 + x_3^2 + 7x_4 + \tfrac{10}{x_5} + x_6]$$

subject to:

$$\begin{gathered}
-x_4 + x_5 - x_6 = 0.1, \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 20, \\
8x_1 + 3x_2 + 4x_3 + x_4 + x_5 + x_7 = 10, \\
2x_1 + 6x_2 + x_3 + 3x_4 + x_6 + x_8 = 13, \\
2x_1 + x_2 + x_3 + 3x_4 - x_9 = 2, \\
x_1 + 4x_2 + 5x_3 + 2x_4 + x_{10} = 18.
\end{gathered}$$

Table P4. ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

| *Lower* *bound* | $x_0$ | $x^*$ | *Upper* *bound* |
|---|---|---|---|
| 0.0 | 0.1 | 0.0 | 10.0 |
| 0.0 | 0.1 | 0.893288 | 10.0 |
| 0.0 | 0.1 | 1.146728 | 10.0 |
| 0.0 | 0.1 | 0.0 | 10.0 |
| 0.0 | 0.1 | 2.733221 | 10.0 |
| 0.0 | 0.1 | 2.633221 | 10.0 |
| 0.0 | 0.1 | 0.0 | 10.0 |
| 0.0 | 0.1 | 3.860319 | 10.0 |
| 0.0 | 0.1 | 0.040016 | 10.0 |
| 0.0 | 0.1 | 8.693204 | 10.0 |

$f(x^*) = 11.180048.$

*Problem P5.*

$$\min \sum_{k=1}^{5} (x_k - x_{k+1})^2 + \sum_{k=6}^{10} (x_k - x_{k+1})^4$$

subject to:

$$x_k + 2x_{k+1} + 3x_{k+2} = 6,$$
$$(k = 1, ..., 8)$$
$$x_9 + 2x_{10} + 3x_{11} = 9.$$

Table P5. $(\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8})$

| *Lower* *bound* | $x_0$ | $x^*$ | *Upper* *bound* |
|---|---|---|---|
| 0.0 | 0.1 | 1.016505 | 10.0 |
| 0.0 | 0.1 | 1.004545 | 10.0 |
| 0.0 | 0.1 | 0.991468 | 10.0 |
| 0.0 | 0.1 | 1.004173 | 10.0 |
| 0.0 | 0.1 | 1.000062 | 10.0 |
| 0.0 | 0.1 | 0.998567 | 10.0 |
| 0.0 | 0.1 | 1.000934 | 10.0 |
| 0.0 | 0.1 | 0.999854 | 10.0 |
| 0.0 | 0.1 | 0.999785 | 10.0 |
| 0.0 | 0.1 | 1.000191 | 10.0 |
| 0.0 | 0.1 | 1.999944 | 10.0 |

$f(x^*) = 0.9995046.$

*Problem P6.*

$$\min \sum_{k=1}^{m+2} x_k^2$$

subject to:

$$x_1 + 2x_2 + x_3 = 3,$$
$$x_1 + x_j + 2x_{j+1} + x_{j+2} = 4,$$
$$(j = 2, ..., m).$$

Table P6 (m=20) $(\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8})$

| $\mathit{Lower}$ $\mathit{bound}$ | $x_0$ | $x^*$ | $\mathit{Upper}$ $\mathit{bound}$ |
|---|---|---|---|
| 0.0 | 0.1 | 2.003768 | 10.0 |
| 0.0 | 0.1 | 0.031407 | 10.0 |
| 0.0 | 0.1 | 0.933417 | 10.0 |
| 0.0 | 0.1 | 0.097989 | 10.0 |
| 0.0 | 0.1 | 0.866834 | 10.0 |
| 0.0 | 0.1 | 0.164572 | 10.0 |
| 0.0 | 0.1 | 0.800251 | 10.0 |
| 0.0 | 0.1 | 0.231155 | 10.0 |
| 0.0 | 0.1 | 0.733668 | 10.0 |
| 0.0 | 0.1 | 0.297738 | 10.0 |
| 0.0 | 0.1 | 0.667085 | 10.0 |
| 0.0 | 0.1 | 0.364321 | 10.0 |
| 0.0 | 0.1 | 0.600502 | 10.0 |
| 0.0 | 0.1 | 0.430904 | 10.0 |
| 0.0 | 0.1 | 0.533919 | 10.0 |
| 0.0 | 0.1 | 0.497487 | 10.0 |
| 0.0 | 0.1 | 0.467336 | 10.0 |
| 0.0 | 0.1 | 0.564070 | 10.0 |
| 0.0 | 0.1 | 0.400753 | 10.0 |
| 0.0 | 0.1 | 0.630653 | 10.0 |
| 0.0 | 0.1 | 0.334170 | 10.0 |
| 0.0 | 0.1 | 0.697236 | 10.0 |

$f(x^*) = 10.345477.$

*Problem P7.*

$$\min \sum_{k=1}^{m+4} (x_k^2 + 2x_k + 6)e^{-x_k}$$

subject to:

$$\begin{gathered} x_1 + 2x_3 + x_5 = 4, \\ x_j + 2x_{j+2} + x_{j+4} = 8, \\ (j = 2, ..., m) \end{gathered}$$

Table P7 (m=10) ($\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8}$)

| *Lower bound* | $x_0$ | $x^*$ | *Upper bound* |
|---|---|---|---|
| 0.0 | 0.1 | 0.274447 | 10.0 |
| 0.0 | 0.1 | 2.286694 | 10.0 |
| 0.0 | 0.1 | 0.202727 | 10.0 |
| 0.0 | 0.1 | 1.713305 | 10.0 |
| 0.0 | 0.1 | 3.320098 | 10.0 |
| 0.0 | 0.1 | 2.286695 | 10.0 |
| 0.0 | 0.1 | 1.157077 | 10.0 |
| 0.0 | 0.1 | 1.713305 | 10.0 |
| 0.0 | 0.1 | 2.365749 | 10.0 |
| 0.0 | 0.1 | 2.286695 | 10.0 |
| 0.0 | 0.1 | 2.111426 | 10.0 |
| 0.0 | 0.1 | 1.713305 | 10.0 |
| 0.0 | 0.1 | 1.411399 | 10.0 |
| 0.0 | 0.1 | 2.286695 | 10.0 |

$f(x^*) = 33.242885.$

*Problem P8.*

$$\min \sum_{k=1}^{3m} \frac{x_k}{x_k+1} \ln x_k$$

subject to:

$$\begin{gathered} x_1 + 2x_{m+1} + x_{2m+1} = 4, \\ x_j + 2x_{m+j} + x_{2m+j} = 8 + \tfrac{1}{j}, \\ (j = 2, ..., m) \end{gathered}$$

Table P8 (m=5) $(\sigma = 0.1, \tau_k = 0.99995, \varepsilon = 10^{-8})$

| $Lower_{bound}$ | $x_0$ | $x^*$ | $Upper_{bound}$ |
|---|---|---|---|
| 0.0 | 0.1 | 0.386228 | 10.0 |
| 0.0 | 0.1 | 0.332300 | 10.0 |
| 0.0 | 0.1 | 0.333421 | 10.0 |
| 0.0 | 0.1 | 0.333997 | 10.0 |
| 0.0 | 0.1 | 0.334348 | 10.0 |
| 0.0 | 0.1 | 1.613771 | 10.0 |
| 0.0 | 0.1 | 3.917699 | 10.0 |
| 0.0 | 0.1 | 3.833245 | 10.0 |
| 0.0 | 0.1 | 3.791002 | 10.0 |
| 0.0 | 0.1 | 3.765651 | 10.0 |
| 0.0 | 0.1 | 0.386228 | 10.0 |
| 0.0 | 0.1 | 0.332300 | 10.0 |
| 0.0 | 0.1 | 0.333421 | 10.0 |
| 0.0 | 0.1 | 0.333997 | 10.0 |
| 0.0 | 0.1 | 0.334348 | 10.0 |

$f(x^*) = 1.824042.$

In oder to compare the interior point methods with the classical methods we shall solve these problems with the following packages for mathematical programming:

| Name Package | Author |
|---|---|
| TOLMIN | M.J.D. Powell<br>University of Cambridge |
| PCLC | N.Andrei<br>ICI - Bucharest |
| NLPQL | K.Schittkowski<br>Bayreuth University |
| MINOS | B.Murtagh, M.Saunders<br>Stanford University |
| SPENBAR | N.Andrei<br>ICI Bucharest |

The characteristics of the solving process corresponding to the P1-P8 linear optimization problems are presented into the following tables.

We emphasize that the TOLMIN and PCLC packages are specialized for linear optimization problems. The MINOS package is for large-scale nonlinear optimization and NLPQL and SPENBAR are dedicated to solve small or medium scale, strong nonlinear programming problems

| Prob | n | m | TOLMIN | | | PCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| P1 | 10 | 3 | 28 | 36 | 0:0:0:16 | 8 | 9 | 0:0:0:83 |
| P2 | 13 | 6 | 22 | 34 | 0:0:0:16 | 37 | 38 | 0:0:3:40 |
| P3 | 15 | 7 | 9 | 13 | 0:0:0:6 | 9 | 10 | 0:0:1:15 |
| P4 | 10 | 6 | 5 | 16 | 0:0:0:5 | 38 | 39 | 0:0:2:80 |
| P5 | 11 | 9 | 4 | 8 | 0:0:0:5 | 8 | 9 | 0:0:1:05 |
| P6 | 22 | 20 | 7 | 13 | 0:0:0:16 | 9 | 10 | 0:0:0:15 |
| P7 | 14 | 10 | 12 | 33 | 0:0:0:16 | 12 | 13 | 0:0:0:17 |
| P8 | 15 | 5 | 13 | 20 | 0:0:0:17 | 9 | 10 | 0:0:0:11 |

| Prob | NLPQL | | | MINOS | | | | SPENBAR | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | it | nf | Time | oit | init | nf | Time | oit | init | nf | Time |
| P1 | 21 | 30 | 0:0:0:44 | 1 | 19 | 46 | 0:0:0:27 | 6 | 228 | 1263 | 0:0:5:77 |
| P2 | 7 | 1045 | 0:0:0:50 | 1 | 18 | 15 | 0:0:0:17 | 9 | 395 | 2308 | 0:0:14:94 |
| P3 | 5 | 5 | 0:0:0:16 | 1 | 18 | 14 | 0:0:0:22 | 9 | 276 | 1898 | 0:0:14:44 |
| P4 | 10 | 10 | 0:0:0:17 | 1 | 12 | 8 | 0:0:0:11 | 9 | 401 | 2384 | 0:0:12:74 |
| P5 | 6 | 7 | 0:0:0:39 | 1 | 12 | 9 | 0:0:0:16 | 6 | 94 | 515 | 0:0:3:41 |
| P6 | 5 | 7 | 0:0:2:09 | 1 | 25 | 11 | 0:0:0:38 | 8 | 1021 | 10877 | 0:3:13:95 |
| P7 | 16 | 16 | 0:0:1:59 | 1 | 20 | 21 | 0:0:0:27 | 5 | 156 | 991 | 0:0:8:73 |
| P8 | 14 | 18 | 0:0:1:10 | 1 | 15 | 26 | 0:0:0:16 | 4 | 54 | 303 | 0:0:2:41 |

n = number of variables.
m = number of constraints.
it = number of iterations.
init = number of inner (minor) iterations.
oit = number of outer (major) iterations.
nf = number of objective function evaluations.

In the following we shall present a comparison among these optimization packages corresponding to the above P6-P8 problems in case of medium and large scale dimensions.

The following tables give the characteristics of solving the problem **P6,** for different values of m, with TOLMIN and PCLC packages and MINOS and NLPQL packages respectively.

| m | n | vfo | TOLMIN | | | PCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 20 | 22 | 10.345477 | 7 | 13 | 0:0:0:22 | 9 | 10 | 0:0:0:15 |
| 40 | 42 | 16.464030 | 3 | 7 | 0:0:0:72 | 10 | 11 | 0:0:0:59 |
| 60 | 62 | 22.664322 | 4 | 20 | 0:0:1:64 | 10 | 11 | 0:0:1:15 |
| 80 | 82 | 28.889140 | 3 | 8 | 0:0:3:18 | 9 | 10 | 0:0:1:92 |
| 100 | 102 | 35.123936 | 2 | 6 | 0:0:5:83 | 11 | 12 | 0:0:4:22 |
| 200 | 202 | 66.343333 | 6 | 30 | 0:0:46:47 | 6 | 7 | 0:0:5:76 |

| m | n | vfo | MINOS | | | NLPQL | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 20 | 22 | 10.345477 | 25 | 11 | 0:0:0:39 | 5 | 7 | 0:0:2:03 |
| 40 | 42 | 16.464030 | 43 | 7 | 0:0:0:77 | 3 | 4 | 0:0:6:92 |
| 60 | 62 | 22.664322 | 63 | 7 | 0:0:1:32 | 3 | 4 | 0:0:19:61 |
| 80 | 82 | 28.889140 | 85 | 7 | 0:0:2:15 | 3 | 4 | 0:0:41:03 |
| 100 | 102 | 35.123936 | 104 | 6 | 0:0:3:02 | 4 | 4 | 0:1:46:28 |
| 200 | 202 | 66.343333 | 204 | 6 | 0:0:9:94 | 2 | 2 | 0:5:55:53 |

For large scale dimensions the MINOS and PCLC packages have the following behaviour:

| m | n | vfo | MINOS | | | PCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 500 | 502 | 160.074858 | 503 | 6 | 0:0:50:20 | 11 | 12 | 0:1:12:19 |
| 1000 | 1002 | 316.318683 | 1003 | 6 | 0:2:42:36 | 11 | 12 | 0:4:38:36 |
| 1200 | 1202 | 378.817653 | 1203 | 6 | 0:3:51:29 | 11 | 12 | 0:5:37:47 |
| 1500 | 1502 | 472.566623 | a | | | 11 | 12 | 0:9:16:45 |
| 2000 | 2002 | 628.815592 | a | | | 11 | 12 | 0:16:13:12 |
| 3000 | 3002 | 941.314562 | a | | | 11 | 12 | 1:4:24:6 |

a = insufficient memory.

For the problem **P7** the following tables illustrate the behaviour of TOLMIN and PCLC packages and MINOS and NLPQL respectively, for different values of m.

| m | n | vfo | TOLMIN | | | PCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 20 | 24 | 52.688309 | 11 | 30 | 0:0:0:38 | 11 | 12 | 0:0:0:28 |
| 40 | 44 | 93.878536 | 11 | 25 | 0:0:1:26 | 10 | 11 | 0:0:0:55 |
| 60 | 64 | 134.915594 | 5 | 7 | 0:0:2:26 | 10 | 11 | 0:0:1:05 |
| 80 | 84 | 175.917332 | 7 | 25 | 0:0:4:51 | 10 | 11 | 0:0:1:82 |
| 100 | 104 | 216.893341 | 7 | 24 | 0:0:7:31 | 10 | 11 | 0:0:2:58 |
| 200 | 204 | 421.719855 | 11 | 43 | 0:0:49:10 | 10 | 11 | 0:0:9:51 |

| m | n | vfo | MINOS | | | NLPQL | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 20 | 24 | 52.688309 | 16 | 28 | 0:0:0:44 | 16 | 16 | 0:0:6:70 |
| 40 | 44 | 93.878536 | 46 | 13 | 0:0:0:83 | 10 | 10 | 0:0:22:41 |
| 60 | 64 | 134.915594 | 66 | 12 | 0:0:1:43 | 10 | 10 | 0:1:5:04 |
| 80 | 84 | 175.912732 | 86 | 12 | 0:0:2:19 | 10 | 10 | 0:2:22:09 |
| 100 | 104 | 216.893341 | 106 | 12 | 0:0:3:7 | 10 | 10 | 0:4:25:51 |
| 200 | 204 | 421.719855 | 206 | 13 | 0:0:10:5 | 12 | 12 | 0:39:40:42 |

For large scale dimensions the MINOS and PCLC give the results from the below table:

| m | n | vfo | MINOS | | | PCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 500 | 504 | 1036.073610 | 509 | 19 | 0:0:49:15 | 15 | 16 | 0:1:37:22 |
| 1000 | 1004 | 2059.953591 | 1010 | 20 | 0:2:40:05 | 11 | 12 | 0:5:3:62 |
| 1200 | 1204 | 2469.503050 | 1210 | 20 | 0:3:35:96 | 12 | 13 | 0:8:19:77 |
| 1500 | 1504 | 3083.826330 | a | | | 13 | 14 | 0:15:11:49 |
| 2000 | 2004 | 4107.697251 | a | | | 13 | 14 | 0:25:26:21 |
| 3000 | 3004 | 6155.437271 | a | | | 14 | 15 | 1:04:48:56 |

a = insufficient memory.

The tables which follows give the results of solving the problem **P8** with TOLMIN and PCLC and MINOS and NLPQL respectively for different values of m.

| m | n | vfo | TOLMIN | | | PCLC | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 20 | 60 | 9.076395 | 38 | 74 | 0:0:4:78 | 9 | 10 | 0:0:0:17 |
| 40 | 120 | 18.600455 | 41 | 81 | 0:0:22:85 | 9 | 10 | 0:0:0:39 |
| 60 | 180 | 28.086333 | 36 | 81 | 0:0:54:49 | 10 | 11 | 0:0:0:71 |
| 80 | 240 | 37.556382 | 37 | 95 | 0:1:52:44 | 10 | 11 | 0:0:0:99 |
| 100 | 300 | 47.017712 | 37 | 106 | 0:3:20:59 | 10 | 11 | 0:0:1:38 |
| 200 | 600 | 94.267018 | 36 | 106 | 0:21:49:59 | 11 | 12 | 0:0:4:45 |

| m | n | vfo | MINOS | | | NLPQL | | |
|---|---|---|---|---|---|---|---|---|
| | | | it | nf | Time | it | nf | Time |
| 20 | 60 | 9.076395 | 33 | 31 | 0:0:0:99 | 17 | 20 | 0:0:45:81 |
| 40 | 120 | 18.600455 | 58 | 42 | 0:0:3:42 | 20 | 21 | 0:6:5:86 |
| 60 | 180 | 28.086333 | 78 | 42 | 0:0:6:43 | 20 | 22 | 0:19:42:98 |
| 80 | 240 | 37.556382 | 98 | 42 | 0:0:10:55 | 18 | 19 | 0:41:55:54 |
| 100 | 300 | 47.017712 | 118 | 41 | 0:0:15:82 | 20 | 22 | 1:30:24:22 |
| 200 | 600 | 94.267018 | 415 | 631 | 0:4:58:8 | b | | |

b = Execution time greater that 4 hours.

For large scale dimensions the PCLC package gives the following results:

| m | n | vfo | PCLC | | |
|---|---|---|---|---|---|
| | | | it | nf | Time |
| 500 | 1500 | 235.856441 | 12 | 13 | 0:0:22:8 |
| 1000 | 3000 | 471.724819 | 12 | 13 | 0:1:21:2 |
| 1200 | 3600 | 566.059177 | 12 | 13 | 0:1:57:10 |
| 1500 | 4500 | 707.553823 | 12 | 13 | 0:3:0:54 |
| 2000 | 6000 | 943.366696 | 12 | 13 | 0:5:10:44 |
| 3000 | 9000 | 1414.969173 | 12 | 13 | 0:11:21:79 |

It is very instructive to notice the behaviour of the MPCLC algorithm in case of the problem P8 in which the maximum number of centering corrections at each iteration is limited to 5. The below table illustrate this behaviour for different values of m.

| m | n | vfo | MPCLC | | |
|---|---|---|---|---|---|
| | | | it | nf | Time |
| 500 | 1500 | 235.856441 | 8 | 9 | 0:0:15:38 |
| 1000 | 3000 | 471.724819 | 8 | 9 | 0:0:52:78 |
| 1200 | 3600 | 566.059177 | 8 | 9 | 0:1:15:47 |
| 1500 | 4500 | 707.553823 | 8 | 9 | 0:1:51:33 |
| 2000 | 6000 | 943.366696 | 8 | 9 | 0:3:12:30 |
| 3000 | 9000 | 1414.969173 | 8 | 9 | 0:6:58:42 |
| 4000 | 12000 | 1886.555515 | 8 | 9 | 0:12:21:22 |

The analysis of the results obtained on this limited set of problems illustrates that the predictor-corrector interior point algorithms are a powerfull alternative for solving linear constrained optimization problems.

Still, PCLC (and MPCLC) is a development code, but we notice that for small scale problems all the packages have the same behaviour. Significant differences appear in the case of medium and large scale problems.

For predictor-corrector interior point algorithms the number of iterations and function evaluations seems to be independent of the number of variables or constraints.

For the class of problems P6-P8, with the number of constraints between 20 and 200, from the above tables it follows that the total time (in milliseconds) needed by that 4 packages considered here is given in the below table:

| Problem | Packages | | | |
|---|---|---|---|---|
| | TOLMIN | PCLC | MINOS | NLPQL |
| P6 | 5806 | 1379 | 1759 | 53140 |
| P7 | 6482 | 1579 | 1801 | 291217 |
| P8 | 170474 | 809 | 33529 | |

It follows that for medium scale linear constrained optimization problems the PCLC and MINOS are the most suited packages. Anyway, MINOS remains the state-of-the-art and the most respectable package for large-scale optimization. Nowaday, TOLMIN does not implement sparse matrices techniques. Introducing these techniques in TOLMIN could give a very competitive package.

## 4. Conclusion

In this paper we have extended the predictor-corrector interior point approach to the linear constraint optimization problems with convex, nonlinear objective function. The idea used in linear programming case and implemented in some state-of-the-art predictor-corrector interior point packages is also considered in the case of linear constrained problems. Firstly, the algorithms allow iterates to be very close to the boundary of the positive orthant, and secondly they correct this step using a centering step. The theorems proved here provides a theoretical justification for such a practice for fast convergence.

The numerical results presented here are obtained with a crude experimental code which is still in progress of development. We expect improvements both in terms of efficiency and robustness. These could be obtained mainly using advanced strategies for solving augmented indefinite linear systems like (2.35). The very good stability properties of the augmented system approach motivated their incorporation into several interior point codes, [Duff, Gould, Reid, Scott and Turner, 1991], [Fourer and Mehrotra, 1993], [Maros and Meszaros, 1995], Turner [1991] and [Vanderbei and Carpenter, 1993]. Other advantages of this approach, namely an easy of handling dense columns in A and the ability of its easy extension to the quadratic programming problems emphasized the importance of the augmented system approach.

It is quite clear that the success of the augmented system factorization depends highly on the efficiency of the pivot selection strategy. For the class of problems considered here this strategy must take also into consideration the structure of the Hessian matrix of the objective function. This will introduce some complications. We must find a "good" partition both for A and $\nabla^2 f(x)$ matrices in order to determine which block can be inexpensively pivoted out and which one should be delayed as much as possible. These algebraic mechanisms are under current consideration.

## 5. Acknowledgements

I gratefully acknowledge the Alexander von Humbold Foundation for the constant moral, financial and material support generously given to me for more than 2 years spent in Duisburg and Bayreuth Universities - Germany.

Special thanks are due to Prof. Klaus Schittkowski (Bayreuth University) for his stimulating discussions and availability during my stay in Bayreuth, to Prof. Michael Saunders (Stanford University) for giving me the MINOS code, as well as some other practical hints for its usage, to Prof. M.J.D. Powell for TOLMIN package and some other papers, to Dr. Marc Breitfeld and Prof. David Shanno (Rutgers University) for PENBAR package and some papers on penalty-barrier algorithms for nonlinear programming.

This research has been supported by the Romanian Academy through grant No. 3649 / 16.07.1998. Special thanks are also extended.

## 6. References

Adler, I., Karmarkar, N., Resende, M.G.C., Veiga, G., (1989a) *Data structures and programming techniques for the implementation of Karmarkar's algorithm.* ORSA Journal on Computing, vol.1, 1989, pp.84-106.

Adler, I., Karmarkar, N., Resende, M.G.C., Veiga, G., (1989b) *An implementation of Karmarkar's algorithm for linear programming.* Mathematical Programming, vol.44, 1989, pp.297-335.

Andersen, E.D., Andresen, K.D., (1997) *The APOS linear programming solver: an implementation of the homogeneous algorithm.* Technical Report, April 28, 1997, Department of Management, Odense University DK-5230 Odense, and CORE, Universite Catholique De Louvain, Belgium.

Andersen, E.D., Gondzio, J., Meszaros, C., Xu, X., (1996) *Implementation of interior-point methods for large scale linear programs.* in Terlaky, T., (Ed.) Interior Point Methods of Mathematical Programming. Kluwer Academic Publishers, Dordrecht, 1996, pp.189-252.

Andrei, N., (1997a) PCIP - *Predictor Corrector Interior Point algorithm for large-scale linear programming.* Technical Paper No. AMOL-97-1, Research Institute for Informatics, Bucharest, March 1997.

Andrei, N., (1997b) *PCIPLC - Predictor Corrector Interior Point algorithm for large-scale Linear Constrained optimization problems.* Technical Paper No. AMOL-97-2, Research Institute for Informatics, Bucharest, May 1997.

Andrei, N., (1997c) *Programarea Matematica Avansata. Teorie, Metode computationale, Aplicatii.* Manuscript sent to "Editura Academiei Romane", 1997.

Anstreicher, K.M., (1990) *On long step path following and SUMT for linear and quadratic programming.* Manuscris. Department of Operations Research, Yale University, New Haven, 1990.

Barnes, E.R., *A variation on Karmarkar's algorithm for solving linear programming problems.* Mathematical Programming, vol.36,, 1986, pp.174-182.

Bayer, D.A., Lagarias, J.C., (1989) *The nonlinear geometry of linear programming, Part I: affine and projective scaling trajectories.* Trans. Amer. Math. Soc., vol.314, 1989, pp.499-526. *Part II: Legendre transform coordinates.* Trans.

Amer. Mathy. Soc., vol.314, 1989, pp.527-581.

Beigler, L.T., Nocedal, J., Schmid, C., Ternet, D., (1997) *Numerical experience with a reduced Hessian method for large scale constrained optimization.* Technical Report OTC 97/06, Argonne National Laboratory.

Bunch, J.R., Parlett, B.N., (1971) *Direct methods for solving symmetric indefinite systems of linear equations.* SIAM Journal on Numerical Analysis, vol.8, 1971, pp.639-655.

Byrd, R.H., Hribar, M.E., Nocedal, J., (1997) *An interior point algorithm for large scale nonlinear programming.* Technical Report OTC 97/05, Argonne National Laboratory.

Carpenter, T.J., Lustig, I.J., Mulvey, J.M., Shanno, D.F., (1993) *Higher-order predictor-corrector interior point methods with application to quadratic objectives.* SIAM Journal on Optimization, vol.3, no.4, November 1993, pp.696-725.

Choi, I.C., Monma, C.L., Shanno, D.F., (1990) *Further development of a primal dual interior point method.* ORSA Journal on Computing, vol.2, 1990, pp.304-311.

Dennis, J.E., Jr., Schnabel, R.B., (1983) *Numerical Methods for Unconstrained Optimization.* Prentice-Hall, Englewoog Cliffs, NJ, 1983.

Dikin, I.I., (1967) *Iterative solution of problems of linear and quadratic programming.* Soviet Mathematical Doklady, vol.8, 1967, pp.674-675.

Domich, P.D., Boggs, P.T., Donaldson, J.R., Witzgall, C., (1989) *Optimal 3-dimensional methods for linear programming.* NISTIR 89-4225. Center for Computing and Applied Mathematics, U.S. Department of Commercve, National Institute of Standards and Technology, 1989.

Duff, I.DS., Gould, N.I.M., Reid, J.K., Scott, J.A., Turner, K., (1991) *The factorization of sparse symmetric indefinite matrices.* IMA Journal on Numerical Analysis, vol.11, 1991, pp.181-204.

El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., Zhang, Y., (1996) *On the formulation and theory of the Newton interior point method for nonlinear programming.* Journal of Optimization Theory and Applications, vol.89, no.3, 1996, pp.507-541.

Fiacco, A.V., McCormick, G.P., (1968) *Nonlinear Programming. Sequential Unconstrained Minimization Techniques.* John Wiley

Fourer, R., Mehrotra, S., (1991) *Performance of an augmented system approach for solving least-squares problems in an interior point method for linear programming.* Mathematical Programming Society COAL Newsletter, vol.19, 1991, pp.26-31.

Fourer, R., Mehrotra, S., (1993) *Solving symmetric indefinite systems in an interior-point method for linear programming.* Mathematical Programming, vol.62, 1993, pp.15-39.

Freund, R.M., (1991) *Theoretical efficiency of a shifted barrier function algorithm for linear programming.* Linear Algebra and its Applications, vol.152, 1991, pp.19-41.

Frisch, K.R., (1955) *The logarithmic potential method for convex programming.* Manuscris. Institute of Economics, University of Oslo, Oslo, May 1955.

Gay, D.M., (1988) *Electronic mail distribution of linear programming test problems.* Mathematical Programming Society COAL Newsletter, 1988.

Gill, Ph.E., Murray, W., Ponceleon, D.B., Saunders, M.A., (1995) *Primal-dual methods for linear programming.* Mathematical Programming, vol.70, 1995, pp.251-277.

Gill, Ph.E., Saunders, M.A., Shinnerl, J.R., (1996) *On the stability of Cholesky factorization for symmetric quasidefinite systems.* SIAM Journal on Matrix Analysis and Applications, vol.17, no.1, January 1996, pp.35-46.

Goldfarb, D., Liu, S., Wang, S., (1991) *A logarithmic barrier function algorithm for quadratically constrained convex quadratic programming.* SIAM Journal on Optimization, vol.1, no.2, 1991, pp.252-267.

Gondzio, J., (1992) *Splitting dense columns of constraint matrix in interior point methods for large scale linear programming.* Optimization, vol.24, 1992, pp.285-297.

Gondzio, J., (1993) *Implementing Cholesky factorization for interior point methods of linear programming.* Optimization, vol.27, 1993, pp.121-140.

Gondzio, J., (1996) *Multiple centrality corrections in a primal-dual method for linear programming.* Computational Optimization and Applications, vol.6, no.2, September 1996, pp.137-156

Gonzaga, C.C., (1989) *An algorithm for solving linear programming problems in $O(n^3L)$ operations.* in: Megiddo, N., (Ed.) Progress in Mathematical Programming. Interior-Point and Related Methods. Springer-Verlag, New York, 1989, pp.1-28.

Gonzaga, C.C., (1990) *Polynomial affine algorithms for linear programming.* Mathematical Programming, vol.49, 1990, pp.7-21.

Gonzaga, C.C., (1991) *Interior point algorithms for linear programming problems with inequality constraints.* Mathematical Programming, vol.52, 1991, pp.209-226.

Gonzaga, C.C., (1992) *Path-following methods for linear programming.* SIAM Review, vol.34, 1992, pp.167-224.

Gonzaga, C.C., Todd, M., (1992) *An $O(\sqrt{n}L)$-iteration large-step primal-dual affine algorithm for linear programming.* SIAM Journal on Optimization, vol.2, 1992.

Hock, W., Schittkowski, K., (1981) *Test Examples for Nonlinear Programming Codes.* Springer Verlag, Berlin, Heidelberg, New York 1981.

Jansen, B., Ross, C., Terlaky, T., (1996) *Target-following methods for linear programming.* in: Terlaky, T., (Ed.) Interior Point Methods of Mathematical Programming. Kluwer Academic Publishers, Dordrecht, 1996, pp.83-124.

Karmarkar, N., (1984) *A new polynomial-time algorithm for linear programming.* Combinatorica vol.4, 1984, pp.373-395.

Karmarkar, N., (1985) *A new polynomial-time algorithm for linear programming.* Presented at the ORAS/TIMS meeting, Boston, November 1985.

Khachiyan, L.G., (1979) *A polynomial algorithm in linear programming.* Soviet Mathematics Doklady, vol.20, no.1, 1979, pp.191-194. [Doklady Akademiia Nauk SSSR, vol.244, 1979, pp.1093-1096.]

Khachiyan, L.G., (1980) *Polynomial algorithms in linear programming.* USSR Computational Mathematics and Mathematical Physics, vol.20, 1980, pp.53-72. [Zhurnal Vychisditel'noi Matematiki i Matematicheskoi Fiziki, vol.20, 1980, pp.51-68.]

Kim, K., Nazareth, J.L., (1994) *A primal null space affine scaling method.* ACM Transactions on Mathematical Software, vol.20, no.3, 1994, pp.373-392.

Kojima, M., Mizuno, S., Yoshise, A., (1989a) *A polynomial-time algorithm for a class of linear complementarity problems.* Mathematical Programming, vol.44, 1989, pp.1-26.

Kojima, M., Mizuno, S., Yoshise, A., (1989b) *A primal-dual interior point algorithm for linear programming.* in: Megiddo, N., (Ed.) Progress in Mathematical Programming. Interior-Point and Related Methods. Springer-Verlag, New York, 1989, pp.29-47.

Kortanek, K.O., Potra, F.A., Ye, Y., (1991) *On some efficient interior point methods for nonlinear convex pregramming.* Linear Algebra and its Applications, vol.152, 1991, pp.169-189.

Lustig, I.J., (1988) A generic primal-dual interior point algorithm. Technical Report SOR 88-3, Princeton University, 1988.

Lustig, I.J., Marsten, R.E., Shanno, D.F., (1990) *The primal-dual interior point method on the Cray supercomputer.* in: Coleman, T.F., Li, Y., (Eds.) Large-Scale Numerical Optimization. SIAM, Philadelphia, 1990, pp.70-80.

Lustig, I.J., Marsten, R.E., Shanno, D.F., (1991) *Computational experience with a primal-dual interior point method for linear programming.* Linear Algebra and its Applications, vol.152, 1991, pp.191-222.

Lustig, I.J., Marsten, R.E., Shanno, D.F., (1992) *On implementing Mehrotra's predictor-corrector interior-point method for linear programming.* SIAM Journal on Optimization, vol.2, no.3, August 1992, pp.435-449.

Lustig, I.J., Marsten, R.E., Shanno, D.F., (1994) *Computational experience with a globally convergent primal-dual predictor-corrector algorithm for linear programming.* Mathematical Programming, vol.66, 1994, pp.123-135.

Maros, I., Meszaros, C., (1995) *The role of the augmented system in interior point methods.* Technical Report TR/06/95, Brunel University, Department of Mathematics and Statistics, London, 1995.

Megiddo, N., (1989) *Pathways to the optimal set in linear programming.* in Megiddo, N., (Ed.) Progress in Mathematical Programming, Interior-point and realted methods. Springer Verlag, New York, 1989, pp.131-158.

Mehrotra, S., (1987) *Variants of Karmarkar's algorithm: Theoretical complexity and practical implementation.* Ph.D. Thesis, submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Science.

Mehrotra, S., (1990) *On finding a vertex solution using interior point methods.* Linear Algebra and its Applications, vol.152, 1990, pp.233-253.

Mehrotra, S., (1992) *On the implementation of a primal-dual interior point method.* SIAM J. Optimization, vol.2, no.4, November 1992, pp.575-601.

Meszaros, C., (1994) Fast Cholesky factorization for interior point methods of linear programming. Technical Report, Computer and Automation Institute, Budapest 1994.

Mizuno, S., Todd, M.J., Ye, Y., (1990) *On adaptive step primal dual interior point algorithms for linear programming.* Technical Report No. 944, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, 1990.

Monteiro, R.D.C., Adler, I., (1989a) I*nterior path-following primal-dual algorithms. Part I: Linear Programming.* Mathematical Programming, vol.44, 1989, pp.27-41.

Monteiro, R.D.C., Adler, I., (1989b) I*nterior path-following primal-dual algorithms. Part II: Convex quadratic programming.* Mathematical Programming, vol.44, 1989, pp.42-66.

Monteiro, R.C., Adler, I., Resende, M.G.C., (1990) *A polynomial time primal dual affine scaling algorithm for linear and convex quadratic programming and its power series extension.* Math. Oper. Res., vol.15, 1990, pp.191-214.

Nazareth, J.L., (1987) *Pricing criteria in linear programming.* in: Megiddo N., (Ed.) Progress in Mathematical Programming. Interior point and related methods. Springer Verlag, New York, 1989, pp.105-129.

Ortega, J.M., Rheinboldt, W.C., (1970) *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

Potra, F.A., (1996) *An infeasible interior point predictor corrector algorithm for linear programming.* SIAM Journal on Optimization, vol.6, no.1, 1996, pp.19-32.

Renegar, J.A., (1988) *A polynomial-time algorithm based on Newton's method for linear programming.* Mathematical Programming, vol.40, 1988, pp.59-93.

Ross, C., Terlaky, T., Vial, J.Ph., (1997) *Theory and Algorithms for Linear Optimization.* John Wiley

Shor, N.Z., (1970a) *Convergence rate of the gradient descent method with dilatation of the space.* Cybernetics, vol.6, no.2, 1970, pp.102-108.

Shor, N.Z., (1970b) *Utilization of the operation of space dilation in the minimization of covex functions.* Kibernetika (Kiev), vol.6, no.1, 1970, pp.6-12. [Traducere in limba engleza in: Cybernetics, vol.6, no.1, 1970, pp.7-15.]

Shor, N.Z., (1970c) *Convergence rate of the gradient descent method with dilatation of the space.* Kibernetika (Kiev), vol.6, no.2, 1970, pp.80-85.

Sonnevend, G., (1985) *An analytical centre for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming.* in Lecture Notes

Control Information Sci., vol.84, Springer Verlag, New York, 1985, pp.866-876.

Tapia, R., Zhang, Y., Saltzman, M., Weiser, A., (1996) *The Mehrotra predictor-corrector interior point method as a perturbed composite Newton method.* SIAM Journal on Optimization, vol.6, no.1, l996, pp.47-56.

Todd, M.J., Ye, Y., (1990) *A centered projective algorithm for linear programming.* Math. Oper. Res., vol.15, 1990, pp.508-529.

Traub, J.F., (1964) *Iterative Methods for the Solution of Equations.* Prentice-Hall, Inc., Englewoods Cliffs, New Jersey, 1964.

Tsuchiya T., (1996) *Affine Scaling Algorithm.* in: Terlaky, T., (Ed.) Interior Point Methods of Mathematical Programming. Kluwer Academic Publishers, Dordrecht, 1996, pp.35-82.

Turner, K., *Computing projections for Karmarkar algorithm.* Linear Algebra and its Applications, vol.152, 1991, pp.141-154.

Vaidya, P.M., (1990) *An algorithm for linear programming which requires* $O((m+n)n^2+(m+n)^{1.5}nL)$. Mathematical Programming, vol.47, 1990, pp.175-201.

Vanderbei, R.J., (1990a) ALPO: Another linear program optimizer. Technical Report, AT&T Bell Laboratories, 1990.

Vanderbei, R.J., (1990b) A brief description of ALPO. Technical Report, AT&T Bell Laboratories, 1990.

Vanderbei, R.J., (1991) *Splitting dense columns in sparse linear systems.* Linear Algebra and its Applications, vol. 152, 1991, pp.107-117.

Vanderbei, R.J., Carpenter, T.J., (1993) *Symmetric indefinite systems for interior point methods.* Mathematical Programming, vol.58, 1993, pp.1-32.

Vanderbei, R.J., Meketon, M.S., Freedman, B.F., (1986) *A modification of Karmarkar's linear programming algorithm.* Algorithmica, vol.1, 1986, pp.395-407.

Wright, S., (1992) *An interior point algorithm for linearly constrained optimization.* SIAM Journal on Optimization, vol.1, 1992.

Wright, S., (1996) *Modified Cholesky factorizations in interior point algorithms for linear programming.* Preprint ANL/MCS - P600-0596, May 1996.

Ye, Y., (1991) *An* $O(n^3L)$ *potential reduction algorithm for linear programming.* Mathematical Programming, vol.50, 1991, pp.239-258.

Zhang, Y., Tapia, R.A., (1993) *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming.* SIAM Journal on Optimization, vol.3, 1993, pp.118-133.

Zhang, Y., Tapia, R.A., Dennis, J.E., (1992) *On the superliniar and quadratic convergence of primal-dual interior point linear programming algorithms.* SIAM Journal on Optimization, vol.2, no.2, 1992, pp.304-324.