

USE OF CONTINUED ITERATION ON THE REDUCTION OF ITERATIONS OF THE INTERIOR POINT METHOD

Lilian F. Berti^{1*}, Aurelio R.L. Oliveira² and Carla T.L.S. Ghidini³

Received May 27, 2015 / Accepted November 7, 2016

ABSTRACT. Interior point methods have been widely used to determine the solution of large-scale linear programming problems. The predictor-corrector method stands out among all variations of interior point methods due to its efficiency and fast convergence. In each iteration it is necessary to solve two linear systems to determine the predictor-corrector direction. Solving such systems corresponds to the step which requires more processing time, and therefore, it should be done efficiently. The most common approach to solve them is the Cholesky factorization. However, Cholesky factorization demands a high computational effort in each iteration. Thus, searching for effort reduction, the continued iteration is proposed. This technique consists in determining a new direction through projection of the search direction and it was inserted into PCx code. The computational results regarding medium and large-scale problems, have indicated a good performance of the proposed approach in comparison with the predictor-corrector method.

Keywords: interior point methods, linear programming, continued iteration.

1 INTRODUCTION

Linear programming is an optimization technique widely used due to its robustness, the efficiency of its algorithms and the large number of problems that can be formulated through it. The study of interior point methods for linear programming has been one of the most active areas of research in optimization in recent decades because of great advances described by Gondzio [12]. Among the variations of the interior point methods for the solution of linear programming problems, the predictor-corrector method presented by Mehrotra [15] has great prominence due to its efficiency and fast convergence. This is a primal-dual method that needs to solve two linear systems with the same symmetric positive definite matrix in each iteration. To solve these systems the most used approach is the Cholesky factorization. Thus, interior point methods present a critical step in its computational performance, since this factorization in most cases demands a large computational effort. Vavasis & Ye [19] propose an algorithm that follows the central path, using

*Corresponding author.

¹Instituto de Matemática, Universidade Federal de Mato Grosso do Sul, 79070-900 Campo Grande, MS, Brasil.

²Departamento de Matemática Aplicada, Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, 13083-859 Campinas, SP, Brasil.

³Faculdade de Ciências Aplicadas, Universidade Estadual de Campinas, 13484-350 Limeira, SP, Brasil.

E-mails: lilian.berti@ufms.br / aurelio@ime.unicamp.br / carla.ghidini@fca.unicamp.br

short steps or a layered least squares step, which return an exact optimum after a finite number of steps. Besides Mehrotra's correction direction [15], other technique which determine correction directions have been developed over the last decades to improve interior point methods. Gondzio [5, 11] presented the multiple centrality corrections to interior point methods that consist in computing a direction so that $x_i z_i$ belongs to the neighborhood of the central path [20], thereby enhancing convergence.

Jarre & Wechs [13] propose to generate corrector directions making use of recursion in a modification of Mehrotra's corrector direction. The idea is to change the residual vector r_a for another vector r in the linear system which computes the predictor direction. They question what the best choice would be for r to get a good search direction. In general, it is not easy to find a good value for r . So, they propose to find a subspace spanned by k different directions, $(dx^1, dy^1, dz^1), \dots, (dx^k, dy^k, dz^k)$, generated for each r^1, r^2, \dots, r^k , where each one of these directions is a predictor-corrector type. They formulate a small linear subproblem to find the best combination of weights in the generated directions to increase the stepsize and the reduction of the complementarity gap.

Mehrotra & Li [16] follow the approach of Jarre & Wechs [13]. They found multiple corrector directions by using information generated by a suitable Krylov subspace. The affine-scaling direction $(\tilde{d}x, \tilde{d}y, \tilde{d}z)$, the Mehrotra's corrector direction $(\bar{d}x, \bar{d}y, \bar{d}z)$ and the first k directions $(dx^1, dy^1, dz^1), \dots, (dx^k, dy^k, dz^k)$ generated by Krylov subspace are combined with appropriate weights. Thus a linear programming problem is solved to determine the best set of weights in the combined search direction.

Numerical experiments have shown that direct methods provide sufficiently accurate solutions for interior point methods to achieve fast convergence regardless the ill-conditioning involved [1]. Such results are also supported by theory [20]. However, iterative methods have good accuracy when applied to well-conditioning systems or when preconditioners, as proposed by Oliveira & Sorensen [18], are used for an ill-conditioning system [12]. The direct methods take advantage of the factorization already computed to determine the solution of additional linear systems, but using iterative methods this advantage is lost.

We present in Section 3 an alternative technique to improve the predictor-corrector interior point method as [12, 13, 16], using direct methods. Inspired by the original continued iteration proposal presented on [17] for the dual affine-scaling method, we present the continued iteration for the predictor-corrector interior point method. The technique is incorporated in the predictor-corrector method in order to reduce the total number of iterations required to solve a linear programming problem.

The continued iteration consists in determining a new projection direction. Additionally, to take advantage of the Cholesky factorization already computed, the new direction is found by minimizing the deviation from the original rescaled direction. This technique is used when an iteration of the predictor-corrector method is completed.

This paper is organized as follows: In Section 2, we define the linear programming problem and describe, briefly, the primal dual interior point method. In Sections 3 and 4, the continued iteration is presented and incorporated into the predictor-corrector interior point method for linear programming problems, whether with bounded variables or not. Section 5 shows the results of computational experiments conducted with several problems selected from different collections. Finally, in Section 6, the conclusions are presented.

2 INTERIOR POINT METHODS

The linear programming problem in standard form is given by:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{1}$$

in which $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = m$, $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.

The problem (1) is called primal problem. The dual problem, associated with it, is given by:

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T y + z = c \\ & z \geq 0, \end{aligned} \tag{2}$$

in which $y \in \mathbb{R}^m$ represents the vector of free dual variables and $z \in \mathbb{R}^n$ represents the slack dual variables.

The first order optimality conditions (Karush-Kuhn-Tucker) of problems (1) and (2) are given by [20]:

$$\begin{aligned} Ax - b &= 0 \\ A^T y + z - c &= 0 \\ XZe &= 0 \\ (x, z) &\geq 0, \end{aligned} \tag{3}$$

in which $X = \text{diag}(x)$, $Z = \text{diag}(z)$ and $e \in \mathbb{R}^n$, such that $e = (1.1 \dots, 1)^T$.

2.1 Primal-Dual Affine-Scaling Interior Point Method

The primal-dual affine-scaling method entails applying Newton’s method to $F(x, y, z) = 0$, which is formed by the optimality conditions (3), ignoring $(x, z) \geq 0$, but starting from an interior point $(x, z) > 0$. Thus, the primal and dual problems are solved simultaneously. The affine-scaling direction is obtained by solving the linear system given below:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \tilde{d}x \\ \tilde{d}y \\ \tilde{d}z \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix}, \tag{4}$$

in which $r_p = b - Ax$, $r_d = c - A^T y - z$ and $r_a = -XZe$.

We use variables substitution to determine the solution of (4). By eliminating the variable $d\tilde{z} = X^{-1}(r_a - Z\tilde{d}x)$, we obtain the augmented system:

$$\begin{bmatrix} -D^{-1} & A^t \\ A & 0 \end{bmatrix} \begin{bmatrix} \tilde{d}x \\ \tilde{d}y \end{bmatrix} = \begin{bmatrix} r_1 \\ r_p \end{bmatrix},$$

in which $D = XZ^{-1}$ and $r_1 = r_d - X^{-1}r_a$. Now eliminating $\tilde{d}x = D(A^t \tilde{d}y - r_1)$, the following symmetric and positive definite normal equations system is obtained

$$ADA^T \tilde{d}y = r_p + AD r_1. \tag{5}$$

As ADA^T is symmetric positive definite, the most used approach for solving system (5) is the Cholesky factorization [9].

The primal α_P and dual α_D stepsize, to keep the interior point in each iteration are given by:

$$\alpha_P = \min \left\{ 1, \tau \min_t \left\{ -\frac{x_t}{\tilde{d}x_t} |x_t + \tilde{d}x_t \leq 0 \right\} \right\},$$

$$\alpha_D = \min \left\{ 1, \tau \min_t \left\{ -\frac{z_t}{\tilde{d}z_t} |z_t + \tilde{d}z_t \leq 0 \right\} \right\},$$

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

Remark 2.1. In the primal-dual affine-scaling method, the products $x_i z_i$ of r_a can converge to zero at different speeds. Thus, the method may fail or progress slowly. To avoid this, the parameter (μ) is added to the optimality conditions so that $x_i z_i = \mu$, which is updated every iteration and tends to zero as the method approximates to a solution.

2.2 Predictor-Corrector Interior Point Method

The predictor-corrector method presented by Mehrotra [15] consists in using a direction that contains three components: the affine-scaling direction, the centering direction and the non-linear correction direction. The affine-scaling direction is determined by (4). The centering direction is computed to prevent products $x_i z_i$ from r_a converging to zero at different speeds as mentioned in Remark (2.1). Mehrotra [15] presents a proposal for computing μ .

When $\alpha_P = \alpha_D = 1$, the primal and dual constraints are satisfied, but the complementary products have $x_i z_i = \tilde{d}x_i \tilde{d}z_i$, then a non-linear correction direction is computed. Consider, $\tilde{D}x = \text{diag}(\tilde{d}x)$ and $\tilde{D}z = \text{diag}(\tilde{d}z)$, the corrector direction \tilde{d} is the combination of the centering and the non-linear correction direction, which is obtained by solving:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \tilde{d}x \\ \tilde{d}y \\ \tilde{d}z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e - \tilde{D}x \tilde{D}ze \end{bmatrix}. \tag{6}$$

Thus, the (d) predictor-corrector direction, given by the sum of two directions found, i.e., $d = \tilde{d} + \bar{d}$, can be interpreted as a solution of the linear system:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_s \end{bmatrix}, \tag{7}$$

in which $r_s = r_a + \mu e - \tilde{D}x \tilde{D}ze$. Solving the system (7), the predictor-corrector direction is given by:

$$\begin{aligned} dy &= (ADA^T)^{-1} [r_p + AD(r_d - X^{-1}r_s)], \\ dx &= D(A^T dy - r_d + X^{-1}r_s), \\ dz &= X^{-1}(r_s - Zdx). \end{aligned} \tag{8}$$

Note that, the most expensive step is solving a linear system with the same matrix ADA^T . The Cholesky factorization is used and it has been already computed to obtain the affine-scaling direction.

Therefore the predictor-corrector method is an iterative method that starts from an interior point (x^0, y^0, z^0) by computing a new point:

$$x^{k+1} = x^k + \alpha_P dx^k,$$

$$\begin{aligned} y^{k+1} &= y^k + \alpha_D dy^k, \\ z^{k+1} &= z^k + \alpha_D dz^k, \end{aligned}$$

in which (dx^k, dy^k, dz^k) is given in (8) and k varies from 0 to a maximum number of iterations.

2.2.1 Stopping criteria

Consider ε_p , ε_d , ε_o primal feasibility, dual feasibility and optimality tolerances, respectively, then the stopping criteria is given by:

$$\frac{\|r_p^k\|_2}{1 + \|b\|_2} \leq \varepsilon_p,$$

$$\frac{\|r_d^k\|_2}{1 + \|c\|_2} \leq \varepsilon_d,$$

$$\frac{(x^k)^T z^k}{1 + |c^T x^k|} \leq \varepsilon_o.$$

For further details on the predictor-corrector interior point methods see [14, 15, 19].

3 CONTINUED ITERATION

The continued iteration is proposed in order to reduce the total number of predictor-corrector interior point method iterations, and consequently, reduce the total computational time. This iteration determines a new direction, after computing the predictor-corrector step, without being necessary to perform a new Cholesky factorization. The new direction is determined by setting some of its components to zero and thus, the deviation from the predictor-corrector direction is kept to a minimum. Furthermore, the Cholesky factorization of ADA^t , already computed in the predictor-corrector method iteration, is used again for solving the linear systems involved. Therefore, the computing of this new direction is not very expensive in comparison with a new predictor-corrector method iteration. This new direction is called continued predictor-corrector direction.

3.1 Computing the continued predictor-corrector direction

The continued predictor-corrector direction is computed in an analogous way to the computing of the predictor-corrector direction. Initially, the continued affine-scaling direction is computed, after that the centering and nonlinear correction direction is computed. More details on these directions will be seen later, because before that we need to define the blocking components.

The components that must be fixed at zero, in the new direction, are those responsible for blocking in the previous direction. The blocking components in the directions (dx, dz) are given by:

$$\begin{aligned} i &= \arg \min_t \left\{ -\frac{x_t^k}{dx_t^k} |x_t^k + dx_t^k \leq 0 \right\}, \\ j &= \arg \min_t \left\{ -\frac{z_t^k}{dz_t^k} |z_t^k + dz_t^k \leq 0 \right\}. \end{aligned} \tag{9}$$

As blocking component in any or both directions may not exist, some likely cases can be considered: two, one or none blocking component.

Remark 3.1. If there is more than one blocking component i or j , only the first blocking determined component is considered in this study. When there is no blocking component, the continued iteration is not performed.

In the original proposal of continued iteration to the predictor-corrector method only one of the blocking components was fixed at zero [3]. In this work, the two blocking components (i and j) are fixed at zero.

3.1.1 Two blocking components

In the continued affine-scaling direction, in which \tilde{d} is to be computed, the components that perform blocking in the previous direction are fixed at zero, thus, we get:

$$\tilde{d}x_i = 0 \text{ and } \tilde{d}z_j = 0, \tag{10}$$

which are additional conditions to the linear system (4). We compute \tilde{d} approximately, since linear system (4) admits a unique solution. In addition, we are imposing new conditions (10) to such system.

Similarly to the idea of Dikin [7] by developing the primal affine-scaling method, we find the solution of this system using an auxiliary problem to determine $\tilde{d}x$. The auxiliary problem is built using (10) and the first equation of (4). The component $\tilde{d}x_j$ corresponding to $\tilde{d}z_j$ is computed using the third equation of (4), in which the value must be $-x_j$. We formulate a problem that determines $\tilde{d}x$ rescaling the new direction and minimizing the deviation in relation to the original rescaled direction, so that the Cholesky factorization is already computed and used. The problem is given by:

$$\begin{aligned} \min \quad & \frac{1}{2} \| D^{-\frac{1}{2}} \tilde{d}x - D^{-\frac{1}{2}} dx \|^2 \\ \text{s.t.} \quad & A\tilde{d}x = r_p \\ & \tilde{d}x_i = \beta_a \\ & \tilde{d}x_j = \beta_b, \end{aligned} \tag{11}$$

in which $D = Z^{-1}X$, $\tilde{d}x, dx \in \mathbb{R}^n$, $\beta_a = 0$, and $\beta_b = -x_j$.

Consider: $\theta_i = A_i^T (ADA^T)^{-1} A_i$, $\theta_j = A_j^T (ADA^T)^{-1} A_j$ and $\theta_{ij} = A_{ij}^T (ADA^T)^{-1} A_{ij}$.

Since the problem (11) is quadratic, we find the solution to this problem using the Lagrangian function, which is given by:

$$\tilde{d}x = dx - DA^T v - \alpha d_{ii} e_i - \gamma d_{jj} e_j,$$

in which

$$\begin{aligned} v &= -\alpha d_{ii} (ADA^T)^{-1} A_i - \gamma d_{jj} (ADA^T)^{-1} A_j, \\ \alpha &= \frac{dx_i - \beta_a + \gamma d_{ii} d_{jj} \theta_{ij}}{d_{ii} (1 - d_{ii} \theta_i)}, \\ \gamma &= \frac{dx_j - \beta_b + \left[\frac{(dx_i - \beta_a) d_{jj} \theta_{ij}}{1 - d_{ii} \theta_i} \right]}{d_{jj} \xi}, \\ \xi &= 1 - \left[\frac{d_{ii} d_{jj} \theta_{ij}^2}{1 - d_{ii} \theta_i} \right] - d_{jj} \theta_j. \end{aligned} \tag{12}$$

From the last two equations of (4), and using least squares in the overdetermined system to compute $\tilde{d}y$, we find the remaining directions given by:

$$\begin{aligned} \tilde{d}z &= X^{-1}(r_a - Z\tilde{d}x), \\ \tilde{d}y &= (ADA^T)^{-1}AD(r_d - \tilde{d}z). \end{aligned} \tag{13}$$

Remark 3.2. The continued affine-scaling direction computation implies solving three linear systems with ADA^T . We solve two linear systems: one to obtain v in (12) and another to get $\tilde{d}y$ in (13).

Given the continued affine-scaling direction, when performing the centering and non-linear correction, we obtain continued predictor-corrector direction, similarly as done in the predictor-corrector method (2.2). Keeping the components that block at zero, in direction \hat{d} we should have:

$$\hat{d}x_i = 0 \text{ and } \hat{d}z_j = 0. \tag{14}$$

Furthermore, the direction \hat{d} should be an approximate solution of the linear system (7). We determine \hat{d} in a similarly way to determine the continued affine-scaling direction described above. However in auxiliary problem (11) there is an change in the value β_b due to the computing of centering and non-linear correction. If $\hat{d}z_j = 0$, we find the value of the component $\hat{d}x_j$ corresponding to the last equation of (7). Then, the auxiliary problem, in this case, is given by:

$$\begin{aligned} \min \quad & \frac{1}{2} \|D^{-\frac{1}{2}}\hat{d}x - D^{-\frac{1}{2}}dx\|^2 \\ \text{s.t.} \quad & A\hat{d}x = r_p \\ & \hat{d}x_i = \beta_a \\ & \hat{d}x_j = \beta_b, \end{aligned} \tag{15}$$

in which $D = Z^{-1}X$, $\hat{d}x, dx \in \mathbb{R}^n$, $\beta_a = 0$ and $\beta_b = -x_j + \frac{\mu}{z_j}$.

Solving the problem (15), it is obtained:

$$\hat{d}x = dx - DA^T v - \alpha d_{ii} e_i - \gamma d_{jj} e_j.$$

We find the remaining directions in (7) by:

$$\begin{aligned} \hat{d}z &= X^{-1}(r_s - Z\hat{d}x), \\ \hat{d}y &= (ADA^T)^{-1}AD(r_d - \hat{d}z). \end{aligned}$$

Remark 3.3. To determine the direction \hat{d} above it is not necessary to solve again the linear systems involving the matrix ADA^T to compute v . In fact, it is necessary to solve only a linear system with matrix ADA^T to determine $\hat{d}y$.

Remark 3.4. The computing of direction \hat{d} is analogous whether there is only one or two blocking components. That is, in (10) and (14) only one condition is considered. In the auxiliary problems (11) and (15), the restriction that has no blocking component is disregarded.

Remark 3.5. As the direction \hat{d} is determined in such a way that the change from the direction d is as low as possible, the stepsizes $\hat{\alpha}_P$ and $\hat{\alpha}_D$, adopted are

$$\hat{\alpha}_P = \min \left\{ \tau \min_t \left\{ -\frac{x_t}{\hat{d}x_t} |x_t + \hat{d}x_t \leq 0 \right\}, 1 - \alpha_P \right\}$$

and

$$\hat{\alpha}_D = \min \left\{ \tau \min_t \left\{ -\frac{z_t}{\hat{d}z_t} |z_t + \hat{d}z_t \leq 0 \right\}, 1 - \alpha_D \right\}.$$

Thus, $\hat{\alpha}_P + \alpha_P \in (0, 1]$ and $\hat{\alpha}_D + \alpha_D \in (0, 1]$. In [3], there is no limitation on the new stepsizes.

3.2 Criteria for using continued iteration

Consider $\hat{r} = (\hat{r}_p, \hat{r}_d, \hat{r}_a)^t$, formed by new residuals produced by the continued iteration e $r = (r_p, r_d, r_a)^t$. Given that the primal infeasibility, the dual infeasibility and/or the complementary products are reduced with continued iteration, then the continued direction only is used if $\|\hat{r}\|_2 < \omega_1 \|r\|_2$ in which $\omega_1 \in (0, 1)$. This is an improvement over the implementation described in [3].

4 CONTINUED ITERATION WITH BOUNDED VARIABLES

Consider the primal linear programming problem with bounded variables in its standard form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x + s = u \\ & x \geq 0, s \geq 0, \end{aligned} \tag{16}$$

in which $s \in \mathbb{R}^n$ represents the slack of bounded variables, and $u \in \mathbb{R}^n$ represents the upper bound of x .

The dual problem in its standard form associated with (16) is given by:

$$\begin{aligned} \max \quad & b^T y - u^T w \\ \text{s.t.} \quad & A^T y + z - w = c \\ & z \geq 0, w \geq 0, \end{aligned} \tag{17}$$

in which $w \in \mathbb{R}^n$ represents the dual variables associated with the slack variables s .

The first order optimality conditions (Karush-Kuhn-Tucker) of the problems (16) and (17) are as follows:

$$\begin{aligned} Ax - b &= 0 \\ x + s - u &= 0 \\ A^T y + z - w - c &= 0 \\ XZe &= 0 \\ SWe &= 0 \\ (x, s) &\geq 0 \\ (z, w) &\geq 0, \end{aligned}$$

in which $S = \text{diag}(s)$ and $W = \text{diag}(w)$.

4.1 Computing the continued predictor-corrector direction

In this case, it is conducted the blocking components and they are given by the directions (dx, ds, dz, dw) . The blocking components i and j in directions dx and dz are defined as in (9). Consequently, the blocking components in the directions ds, dw are defined by:

$$l_1 = \arg \min_i \left\{ -\frac{s_i^k}{ds_i^k} |s_i^k + ds_i^k \leq 0 \right\},$$

$$l_2 = \arg \min_i \left\{ -\frac{w_i^k}{dw_i^k} |w_i^k + dw_i^k \leq 0 \right\}.$$

The computing for determining the null components in the new direction is performed as follows.

Consider $h = \min\{-x_i^k/dx_i^k, -s_{l_1}^k/ds_{l_1}^k\}$ and $g = \min\{-z_j^k/dz_j^k, -w_{l_2}^k/dw_{l_2}^k\}$. If $h = -x_i^k/dx_i^k$, then in the new direction $\hat{d}x_i = 0$. Otherwise, we assign $i \leftarrow l_1$ and $\hat{d}s_i = 0$. If $g = -z_j^k/dz_j^k$, then in the new direction $\hat{d}z_j = 0$; otherwise, $j \leftarrow l_2$ and $\hat{d}w_j = 0$.

4.1.1 Two blocking components

With the given components, we have computed the continued affine-scaling direction \tilde{d} so that:

$$\begin{aligned} \tilde{d}x_i &= 0 \text{ or } \tilde{d}s_i = 0, \\ \tilde{d}z_j &= 0 \text{ or } \tilde{d}w_j = 0. \end{aligned} \tag{18}$$

Additionally, the continued affine-scaling direction should be an approximate solution of the linear system obtained when Newton's method is applied to the optimality conditions of the problem, being the corresponding linear system given by:

$$\begin{bmatrix} A & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 \\ 0 & 0 & A^T & I & -I \\ Z & 0 & 0 & X & 0 \\ 0 & W & 0 & 0 & S \end{bmatrix} \begin{bmatrix} \tilde{d}x \\ \tilde{d}s \\ \tilde{d}y \\ \tilde{d}z \\ \tilde{d}w \end{bmatrix} = \begin{bmatrix} r_p \\ r_u \\ r_d \\ r_a \\ r_b \end{bmatrix}, \tag{19}$$

in which $\tilde{d}x, \tilde{d}s, \tilde{d}z, \tilde{d}w \in \mathbb{R}^n$ and $\tilde{d}y \in \mathbb{R}^m$.

To determine the direction \tilde{d} , initially, we compute $\tilde{d}x$ analogously to (3.1.1), using an auxiliary problem, which in this case is given by:

$$\begin{aligned} \min \quad & \frac{1}{2} \|D^{-\frac{1}{2}} \tilde{d}x - D^{-\frac{1}{2}} dx\|^2 \\ \text{s.t.} \quad & A\tilde{d}x = r_p \\ & \tilde{d}x_i = \beta_a \\ & \tilde{d}x_j = \beta_b, \end{aligned} \tag{20}$$

in which $D = (XZ^{-1} + SW^{-1})^{-1}$,

$$\beta_a = \begin{cases} 0, & \text{if } \tilde{d}x_i = 0; \\ (r_u)_i, & \text{if } \tilde{d}s_i = 0, \end{cases}$$

and

$$\beta_b = \begin{cases} -x_j, & \text{if } \tilde{d}z_j = 0; \\ (r_u)_j + s_j, & \text{if } \tilde{d}w_j = 0. \end{cases}$$

Problem (20) is similar to the problem (11), in which only the values β_a e β_b may change. Thus,

$$\tilde{d}x = dx - DA^T v - \alpha d_{ii} e_i - \gamma d_{jj} e_j.$$

For the remaining equations (19), we obtain:

$$\begin{aligned} \tilde{d}s &= r_u - \tilde{d}x, \\ \tilde{d}z &= X^{-1}(r_a - Z\tilde{d}x), \\ \tilde{d}w &= S^{-1}(r_b - W\tilde{d}s), \\ \tilde{d}y &= (ADA^T)^{-1}AD(r_d - \tilde{d}z + \tilde{d}w). \end{aligned}$$

The procedure performing centering and non-linear correction is similar to the case without bounds, thus obtaining the continued predictor-corrector direction.

Remark 4.1. When there is one blocking component, the direction computation is similar to (4.1.1). Furthermore, the equations related to the missing blocking component are disregarded in (18) and (20).

Remark 4.2. The criterion for using the continued iteration with bounded variables is similar to that given in (3.2); in this case, $\hat{r} = (\hat{r}_p, \hat{r}_u, \hat{r}_d, \hat{r}_a, \hat{r}_b)^t$ and $r = (r_p, r_u, r_d, r_a, r_b)^t$.

5 COMPUTATIONAL EXPERIMENTS

The computational experiments were performed on an Intel Core i7, 8 GB RAM, 1TB HD and Linux Operating System.

The continued iteration was implemented in language C and incorporated into the PCx code [6] with multiple centrality corrections turned off [11]. To analyze the performance of PCx with the proposed approach, the computational experiments were performed with free access problems belonging to Netlib [10], Kennington [4] and Qaplib [2] collection.

The PCx has a good behavior near a solution, thus the continued iteration was not used in the last iterations. Based on criteria defined in (2.2.1), the continued iteration is to be applied if:

$$\frac{x^t z + s^t w}{(1 + |c^t x|)} > \omega_2,$$

in which $\omega_2 \in (0, 1)$.

For the computational experiments $\omega_1 = 10^{-7}$ in (3.2) and $\omega_2 = 0.99$.

Table 1 shows, in the first column, the name of the selected problems and in the second and third columns, are the dimension of the problems (number of rows and columns) after the preprocessing of the PCx. In the fourth column, there is the density of the Cholesky factor provided by PCx, which performs a reordering to obtain this sparse factor. The fifth column relates the number of variables with bounds (VB) for each problem. The last column indicates the collection to which the problem belongs.

Table 2 shows the results by incorporating the continued iteration into PCx code. The first column shows the tests problems. The second column presents the results obtained by PCx standard version without multiple centrality corrections. The third column has the results obtained by PCx with the continued iteration, which will be referred to as PCx-IC. The columns PCx and PCx-IC indicate the total number of iterations (k) performed by the interior point methods and running time in seconds (t). In addition, PCx-IC has the number of performed continued iterations (ic).

Table 1 – Test problems.

Problem	Dimension		Density	VB	Collection
	Rows	Columns			
DFL001	5984	12143	0.09132	13	Netlib
MAROS-R7	2152	7440	0.23023	0	Netlib
PILOT87	1971	6373	0.21863	1578	Netlib
STOCFOR3	15362	22228	0.00144	0	Netlib
CRE-B	5336	36382	0.01728	0	Kennington
CRE-D	4102	28601	0.02497	0	Kennington
KEN11	10085	16740	0.00192	16740	Kennington
KEN13	22534	36561	0.01130	36561	Kennington
KEN18	78862	128434	0.00061	128434	Kennington
OSA-07	1081	25030	0.04747	0	Kennington
OSA-14	2300	54760	0.02255	0	Kennington
OSA-30	4313	104337	0.01214	0	Kennington
OSA-60	10243	243209	0.00497	0	Kennington
PDS06	9156	28472	0.01395	8448	Kennington
PDS10	15648	48780	0.01372	15125	Kennington
PDS20	38722	106180	0.01357	33150	Kennington
PDS30	47968	156042	0.01271	51389	Kennington
PDS40	64276	214385	0.01363	72686	Kennington
PDS50	80339	272513	0.01293	95668	Kennington
PDS60	96514	332862	0.01247	119785	Kennington
PDS70	111896	386238	0.01257	143325	Kennington
PDS80	126120	430800	0.01185	165193	Kennington
PDS90	139752	471538	0.01136	186292	Kennington
PDS100	152300	498530	0.01036	206383	Kennington
CHR25A	8149	15325	0.07689	0	Qaplib
CHR22B	5587	10417	0.08850	0	Qaplib
ELS19	4350	13186	0.40664	0	Qaplib
KRA30A	18059	85725	0.51777	0	Qaplib
KRA30B	18059	85725	0.51777	0	Qaplib
ROU20	7359	37640	0.76870	0	Qaplib
SCR15	2234	6210	0.53284	0	Qaplib
SCR20	5079	15980	0.51710	0	Qaplib
STE36A	27683	131076	0.46092	0	Qaplib
STE36B	27683	131076	0.46092	0	Qaplib
STE36C	27683	131076	0.46092	0	Qaplib

Table 2 – Comparison between PCx and PCx-IC.

Problema	PCx		PCx-IC		
	k	t	k	t	ic
DFL001	56	18.34	54	18.17	6
MAROS-R7	16	0.71	19	0.91	11
PILOT87	33	1.80	35	1.91	17
STOCFOR3	30	0.23	29	0.31	5
CRE-B	41	0.88	45	1.10	6
CRE-D	41	0.73	47	0.97	8
KEN11	20	0.13	19	0.16	10
KEN13	23	0.41	23	0.53	11
KEN18	29	3.50	31	4.37	11
OSA-07	22	0.13	22	0.18	8
OSA-14	25	0.40	18	0.42	6
OSA-30	24	0.96	18	1.06	10
OSA-60	33	4.02	18	3.77	9
PDS06	34	2.62	33	2.60	10
PDS10	39	13.88	39	14.15	8
PDS20	55	149.65	52	144.23	10
PDS30	67	448.41	60	407.16	7
PDS40	66	1223.58	64	1195.83	4
PDS50	66	2115.27	71	2275.74	8
PDS60	69	3695.15	69	3710.79	5
PDS70	75	6298.36	70	5912.77	9
PDS80	74	8890.05	71	8580.14	11
PDS90	76	11275.48	76	11303.87	9
PDS100	78	12841.34	77	12711.74	9
CHR25A	28	11.49	26	11.07	1
CHR22B	27	4.59	28	4.96	1
ELS19	26	33.48	26	33.92	1
SCR15	21	5.45	20	5.31	1
SCR20	20	64.04	20	64.63	1
ROU20	16	290.70	16	293.19	1
KRA30A	26	3704.38	25	3585.43	2
KRA30B	27	3841.13	28	3998.78	2
STE36A	32	13191.43	31	12857.36	1
STE36B	31	12801.62	31	12860.21	0
STE36C	31	12808.24	30	12467.26	1

In Table 2, we find 18 problems in which the number of iterations is reduced by PCx-IC. The PDS problems present the largest reductions in both of iterations and time by the proposed method. The OSA-60 problem has the highest iteration reduction ratio at 45%. Particularly in terms of time, the largest time reduction ratio stands at 7% in the PDS70 problem among the 14 problems displaying time reduction. As for the

problems that require more processing time whenever PCx-IC reduces the number of iterations the time is also reduced. However, there is no time reduction in the problems solved by PCx in a few seconds and with low reduction. This occurs due to the increased effort exerted by the continued iteration. Thus, excellent performance is obtained by PCx-IC in most large-scale problems regarding iterations reduction as well as for total computational effort.

Figure 1 shows the performance profile [8] by PCx and PCx-IC regarding the total number of iterations. Note that the method PCx-IC offers better performance compared with PCx. It solves 78% of the test problems with a smaller number of iterations.

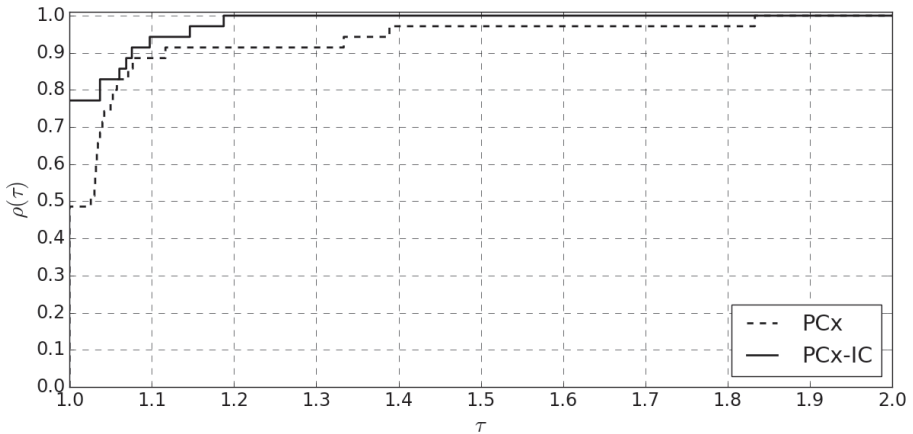


Figure 1 – Total number of iterations.

Figure 2 shows the performance profile of the methods considering the total running time as performance measure. In this case, PCx is the most efficient for approximately 60% of the test problems solved in less time. Moreover, both methods are robust for solving all tested problems.

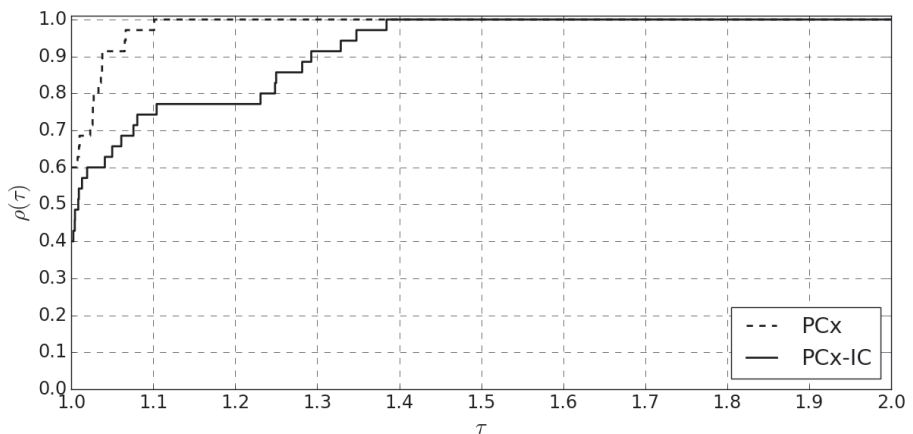


Figure 2 – Total time for solving problems.

6 CONCLUSIONS

In this study, the continued iteration has been proposed for the predictor-corrector interior point method with and without bounded variables. The main objective is to reduce the total number of predictor-corrector interior point method iterations and consequently, the total computational time. In the continued iteration, a new direction is determined, considering that some of its components are null according to the blocking component in the last computed direction. The continued predictor-corrector direction is determined by setting to zero some of its components and thus the deviation from the predictor-corrector direction is kept to a minimum. The Cholesky factorization, already computed in the predictor-corrector method iteration, is used again for solving the linear systems involved. Thus, it avoids computing a new Cholesky factorization. Then, the new method is applied on two levels. On the outer level, the Cholesky factorization and the traditional predictor-corrector direction are computed. On the inner level, a new direction employing the existing Cholesky factorization on the outer level is used by the continued iteration. The computational effort of each continued iteration is dominated by the solution of up to four linear systems with the same matrix already factored. The computational results show that the continued iteration reduces the number of iterations in most of the test problems. The problems that the PCx needs more time to solve also have their time reduced by PCx-IC. Although the continued iteration with the proposed direction has reduced the number of iterations in most problems, the extra effort per iteration to compute the new direction in smaller problems does not affect the reduction of total computational time. In future work other continued directions will be searched aiming to reduce the extra effort that such directions add in the predictor-corrector method.

ACKNOWLEDGEMENTS

This research was sponsored by the Brazilian Agencies CAPES, CNPq and FAPESP.

REFERENCES

- [1] ANDERSEN ED, GONDZIO J, MÉSZÁROS C & XU X. 1996. Implementation of interior point methods for large scale linear programming. In: Interior Point Methods in Mathematical Programming, TERLAKY T (Ed.), Kluwer Academic Publishers, pp. 189–252.
- [2] BURKARD RE, KARISCH S & RENDL F. 1991. QAPLIB – A quadratic assignment library problems. *European Journal of Operational Research*, **55**(1): 115–119.
- [3] BERTI LF & OLIVEIRA ARL. 2011. Iterações continuadas aplicadas ao método de pontos interiores preditor corretor. *Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional*, 2328–2338 (in Portuguese).
- [4] CAROLAN WJ, HILL JE, KENNINGTON JL, NIEMI S & WICHMAN SJ. 1990. An empirical evaluation of the KORBX algoritmos for military airlift applications. *Operations Research*, **38**(2): 240–248.
- [5] COLOMBO M & GONDZIO J. 2008. Further development of multiple centrality correctors for interior point methods. *Computational Optimization and Applications*, **41**: 277–305.
- [6] CZYZYK J, MEHROTRA S, WAGNER M & WRIGHT SJ. 1999. PCx an interior point code for linear programming. *Optimization Methods and Software*, **11-12**: 397–430.
- [7] DIKIN II. 1967. Iterative solution of problems of linear and quadratic programming. *Soviets Mathematics Doklady*, **8**: 674–675.
- [8] DOLAN ED & MORÉ JJ. 2003. Benchmarking optimization software with performance profiles. *Math. Program., Serie A*, **91**(2): 201–213.

- [9] DUFF IS, ERISMAN AM & REID JK. 1986. *Direct Methods for Sparse Matrices*. New York: Oxford University Press, Inc.
- [10] GAY DM. (1985). *Electronic mail distribution of linear programming test problems*, Mathematical Programming Society Committee on Algorithms COAL Newsletter, **13**: 10–12.
- [11] GONDZIO J. 1996. Multiple centrality corrections in a primal-dual methods for linear programming. *Computation Optimization and applications*, **6**: 137–156.
- [12] GONDZIO J. 2012. Interior point methods 25 years later. *European Journal of Operational Research*, **218**: 587–601.
- [13] JARRE F & WECHS M. 1999. Extending Mehrotra’s corrector for linear programs. *Advanced Modeling and Optimization*, **1**: 38–60.
- [14] LUSTIG IJ, MARSTEN RE & SHANNO DF. 1992. On Implementing Mehrotra’s Predictor-Corrector Interior Point Method for Linear Programming. *SIAM Journal on Optimization*, **2**: 435–449.
- [15] MEHROTRA S. 1992. On implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, **2**(4): 575–601.
- [16] MEHROTRA S & LI Z. 2005. Convergence conditions and Krylov subspace based corrections for primal-dual interior-point method. *SIAM Journal on Optimization*, **15**: 635–653.
- [17] OLIVEIRA ARL, LYRA C & CORREIRA PB. 1988. Implementação computacional de algoritmo polinomial de programação linear: aplicação ao planejamento de operação de sistemas hidrotérmicos. *Annals of the VII Congresso Brasileiro de Automática – CBA*, São José dos Campos, Brazil, 928–929 (in Portuguese).
- [18] OLIVEIRA ARL & SORENSEN DC. 2005. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and its Applications*, **394**: 1–24.
- [19] VAVASIS SA & YE Y. 1996. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming*, **74**: 79–120.
- [20] WRIGHT SJ. 1997. *Primal-dual interior-point methods*. Philadelphia: SIAM Publications, Inc.