

HEURISTICS FOR IMPLEMENTATION OF A HYBRID PRECONDITIONER FOR INTERIOR-POINT METHODS

Marta Ines Velazco Fontova¹, Aurelio Ribeiro Leite de Oliveira^{2*}
and Frederico F. Campos³

Received January 7, 2008 / Accepted April 15, 2011

ABSTRACT. This article presents improvements to the hybrid preconditioner previously developed for the solution through the conjugate gradient method of the linear systems which arise from interior-point methods. The hybrid preconditioner consists of combining two preconditioners: controlled Cholesky factorization and the splitting preconditioner used in different phases of the optimization process. The first, with controlled fill-in, is more efficient at the initial iterations of the interior-point methods and it may be inefficient near a solution of the linear problem when the system is highly ill-conditioned; the second is specialized for such situation and has the opposite behavior. This approach works better than direct methods for some classes of large-scale problems. This work has proposed new heuristics for the integration of both preconditioners, identifying a new change of phases with computational results superior to the ones previously published. Moreover, the performance of the splitting preconditioner has been improved through new orderings of the constraint matrix columns allowing savings in the preconditioned conjugate gradient method iterations number. Experiments are performed with a set of large-scale problems and both approaches are compared with respect to the number of iterations and running time.

Keywords: Linear programming, Interior Point Methods, Preconditioning.

1 INTRODUCTION

Since the emergence of the interior-point method, sophisticated codes have been implemented in order to decrease the computational effort and improve its efficiency (Adler *et al.*, 1989; Lustig *et al.*, 1990; Mehrotra, 1992; Czyzyk *et al.*, 1999). The most expensive step to each

*Corresponding author

¹Faculty of Campo Limpo Paulista – FACCAMP, Rua Guatemala, 167, Bairro Jardim América, 13231-230 Campo Limpo Paulista, SP, Brazil. E-mail: marta.velazco@gmail.com

²Instituto de Matemática, Estatística e Computação Científica (IMECC), Universidade Estadual de Campinas (UNICAMP), Praça Sérgio Buarque de Holanda, 651, Cx. Postal 6065, 13083-859 Campinas, SP, Brazil. E-mail: aurelio@ime.unicamp.br

³Departamento de Ciência da Computação do ICEx – Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, Pampulha, 31270-010 Belo Horizonte, MG, Brazil. E-mail: fcampos@dcc.ufmg.br

iteration consists of the resolution of one or two linear systems. The most used approach for the solution of these systems is the Cholesky factorization (Golub & Van Loan, 1996), a procedure that may be expensive in large-scale problems.

An alternative for the solution of these systems is the use of iterative methods. The conjugate gradients method has shown to be the most efficient for the solution of the linear equations systems with large positive definite matrix.

To obtain the convergence of the iterative methods, it is fundamental to construct a preconditioner for the matrix of the linear system. These preconditioners should be easily built with relatively low computational cost and simultaneously, it should provide the convergence of the iterative method in a small number of iterations.

In this work, two specific preconditioners will be considered: controlled Cholesky factorization (Campos & Birkett, 1998) and the splitting preconditioner (Oliveira & Sorensen, 2005), presented in Section 3.

Based on tests performed by Bocanegra *et al.* (2007) it has been proved that these two preconditioners determine a different behavior for the conjugate gradients method in the solution of the linear systems which arise from interior-point methods. At the initial iterations of the interior-point method, the method of the conjugate gradients solves the linear systems involved more efficiently using the preconditioner obtained by the controlled Cholesky factorization (Campos & Birkett, 1998). At the final phase of the interior-point iterative process, where the linear systems matrices are highly ill-conditioned, it was verified that the splitting preconditioner (Oliveira & Sorensen, 2005), specially developed for these systems, allows the conjugate gradients method to solve the systems in a quite efficient way.

This opposite behavior was availed by Bocanegra *et al.* (2007) developing a hybrid approach where both preconditioners are used for the solution of the linear systems in the same problem of optimization by interior-point methods. In the first phase of the optimization, the preconditioner obtained by the controlled Cholesky factorization is used, and in the second phase (final phase) the splitting preconditioner is used.

This work presents new heuristics to identify the moment in which the preconditioners change. Moreover, improvements to the efficiency of the splitting preconditioner are presented, consequently decreasing the number of iterations of the conjugate gradients method. The results are presented using large-scale problems and comparing the results with the ones obtained by Bocanegra *et al.* (2007).

2 PRIMAL-DUAL INTERIOR POINTS METHODS

A linear optimization problem may be presented in the standard form, as following:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{1}$$

where $A \in \mathfrak{R}^{m \times n}$ is the matrix of restrictions with m rows and n columns and $\text{rank}(A) = m$, $c \in \mathfrak{R}^n$ is the vector of the cost of the problem, $b \in \mathfrak{R}^m$ is the vector of the restrictions and $x \in \mathfrak{R}^n$ is the vector of variables of the problem, restricted to values which are non-negative ($x \geq 0$; $x_i \geq 0, i = 1, 2, \dots, n$). Associated with the primal problem (1), the dual problem is represented by:

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

where $y \in \mathfrak{R}^m$ is the vector of dual variables, $z \in \mathfrak{R}^n$ is the vector of the *gap* variable z restricted to non-negative values.

The primal-dual methods (Wright, 1996) solve the primal and the dual problems simultaneously from an initial point, not necessarily feasible, but strictly positive (interior point). The method is obtained from the application of the Newton’s method to the non-linear system $F(x, y, z)$ (Equation 3) formed by the conditions of optimality, but not considering the non-negativity restrictions:

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

The Predictor-Corrector method (Monteiro *et al.*, 1990; Mehrotra, 1992) is considered the most efficient approach for the solution of generic problems of linear programming.

This method uses three components to calculate the direction. A predictor direction $(\Delta\tilde{x}^k, \Delta\tilde{y}^k, \Delta\tilde{z}^k)$ or the affine-scaling directions calculated from the system (4). This system of equations results from the Newton’s method applied to the system (3).

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta\tilde{x}^k \\ \Delta\tilde{y}^k \\ \Delta\tilde{z}^k \end{bmatrix} = \begin{bmatrix} b - Ax^k \\ c - A^T y^k - z^k \\ -X^k Z^k e \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_a^k \end{bmatrix} = r. \tag{4}$$

From the predictor direction, an auxiliary point is calculated $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)$,

$$\begin{aligned} \tilde{x}^k &= x^k + \tilde{\alpha}_p \Delta\tilde{x} \\ \tilde{y}^k &= y^k + \tilde{\alpha}_d \Delta\tilde{y} \\ \tilde{z}^k &= z^k + \tilde{\alpha}_d \Delta\tilde{z} \end{aligned} \tag{5}$$

where $\tilde{\alpha}_p$ and $\tilde{\alpha}_d$ are the steps calculated for the predictor direction, which guarantee the interi- ority of the auxiliary point. The steps are defined as follows (Wright, 1996).

$$\tilde{\alpha}_p^k = \min \left(1, \alpha \cdot \min_{\partial x_i^k < 0} \left(-\frac{x_i^k}{\partial x_i^k} \right) \right), \tilde{\alpha}_d^k = \min \left(1, \alpha \cdot \min_{\partial z_i^k < 0} \left(-\frac{z_i^k}{\partial z_i^k} \right) \right) \text{ and } \sigma \in [0, 1).$$

According to the predictor direction process, the disturbance μ^k is calculated (Wright, 1996)

$$\mu_k = \begin{cases} \frac{(\tilde{x}^k \tilde{z}^k)^3}{n(x^k z^k)^2} & \text{se } x^k z^k > 1 \\ \frac{(x^k z^k)^2}{n\sqrt{n}} & \text{otherwise} \end{cases},$$

and soon after that a direction of correction is computed. For this, the analogue system to the expression (4) (Mehrotra, 1992) known as Modified Newton’s method:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_d^k \\ r_c^k \end{bmatrix} = \begin{bmatrix} b - A\tilde{x}^k \\ c - A^T \tilde{y}^k - \tilde{z}^k \\ \mu^k e - X^k Z^k e - \Delta \tilde{X}^k \Delta \tilde{Z}^k e \end{bmatrix}.$$

From the new direction, we calculate the next point in an equivalent way to (5).

It may be observed that in the predictor-corrector method, two linear systems at each iteration with the same matrix are solved.

The implementations of interior points work with matrices of lower dimensions obtained from the elimination of some variables.

If we use the third equation to eliminate Δz we obtain the augmented system (Equation 6) defined by $D = X^{-1}Z$:

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_d - X^{-1}(\mu e - Ze - \Delta \tilde{X} \Delta \tilde{Z} e) \\ r_p \end{bmatrix}. \tag{6}$$

This system may still be reduced to a normal equations system by eliminating the Δx variable:

$$(AD^{-1}A^T)\Delta y = AD^{-1}(r_d - X^{-1}r_\mu) + r_p. \tag{7}$$

In the calculation of Newton’s directions, two linear systems involving the matrix $AD^{-1}A^T$ are solved. For the solution of this system, direct methods may be used through Cholesky factorization or iterative methods. The most used iterative method is the preconditioned conjugate gradient.

In the next section, the hybrid preconditioner used for the preconditioning of the matrix $AD^{-1}A^T$ of the normal equation system (7) is presented.

3 PRECONDITIONERS

The preconditioning of a matrix is used to facilitate the convergence of iterative methods in the solution of linear systems.

Most of the known preconditioners are obtained from an incomplete Cholesky factorization without promising results due to the ill condition of the matrix in the second phase of the process of optimization.

Mehrotra (1992b) and (Lustig *et al.*, 1990) define a preconditioner for the solution of linear systems by the conjugate gradient through the incomplete Cholesky factorization. The amount of zeros in the preconditioners is controlled and it highly influences the performance. The tests were carried out in the problems of *Netlib*.

Wang & O’Leary (2000) developed a preconditioner for the solution of the linear system by an “adaptive” method. The method identifies when we should use the direct method and when we should use the iterative method by the preconditioned conjugate gradient. The initial preconditioner would be Cholesky factorization of the matrix generated in one of the iterations of the interior-point method. When the *gap* varies in the course of the iterations of the method, the preconditioner is calculated through up to n updates of 1-rank according to the variation of the matrix D .

Bergamaschi *et al.* (2004) describe a preconditioner for the solution by iterative methods of the augmented system in the solution of problems of linear, non-linear and quadratic optimization. The preconditioner is calculated from Cholesky factorization of a new matrix $AE^{-1}A^T$ where the matrix E is obtained from modifications made about the problem of optimization to have a more sparse factorization. When we use the complete factorization of the matrix as a preconditioner, the computational cost when the problem is linear is the same as when the direct method is used.

3.1 Hybrid preconditioner

The hybrid preconditioner (Bocanegra *et al.*, 2007) is formed by the junction of two preconditioners: controlled Cholesky factorization and the splitting. The preconditioners are used in different phases of the process of optimization: controlled Cholesky factorization in the first phase and the splitting in the final phase where the system is very ill-conditioned.

3.2 Controlled Cholesky Factorization

The Controlled Cholesky Factorization (CCF) proposed by Campos and Birkett (1998) is a variation of the incomplete Cholesky factorization proposed by Jones and Plassmann (1995).

Being $Q = AD^{-1}A^T$ and $Qx = r$ the system of linear equations. Consider the complete Cholesky factorization of $Q = LL^T$ and the incomplete factorization of $Q = \tilde{L}\tilde{L}^T + R$ where \tilde{L} is an inferior triangular matrix obtained in the factorization and R , the remainder matrix. The matrix \tilde{L} is used as preconditioner matrix. It is constructed from the selection of a fixed number of elements per columns with the largest absolute values.

Defining $E = L - \tilde{L}$, the controlled Cholesky factorization is based on the minimization of the Frobenius norm of E given that when $\|E\| \rightarrow 0 \Rightarrow \|R\| \rightarrow 0$. Consider the following problem:

$$\min \|E\|_F^2 = \sum_{j=1}^n \sum_{i=1}^n |l_{ij} - \tilde{l}_{ij}|^2 = \sum_{j=1}^n \left(\sum_{i=1}^{m_j+\eta} |l_{ij} - \tilde{l}_{ij}|^2 + \sum_{i=m_j+\eta+1}^n |l_{ij}|^2 \right).$$

as:

- n = order of the matrix;
- m_j = number of nonzero elements below the diagonal in the j -th column of the matrix Q ;
- η = extra number of nonzero elements allowed per column.

The calculation of the norm is divided into two summations. In the first, we get the difference among the elements of the matrix L and the elements chosen which will form the matrix \tilde{L} . The selected elements are the elements of the largest absolute values per column. The quantity selected is equal to the number of nonzero elements below the diagonal (m_j) plus the extra number of nonzero elements allowed per columns (η). In the second summation the difference is not realized, because in these positions the matrix \tilde{L} will have elements equal to zeros. Note that the higher the filling of the matrix \tilde{L} the lower will be its difference with L obtained by the complete Cholesky factorization.

The value of η may vary from $-n$ to n . When η has a positive value we will have a column j with filling higher than the column j of the matrix Q . When the value is negative we will have a lower filling.

From tests carried out (Bocanegra *et al.*, 2007), it has been proved that this factorization presents good results in the first iterations of the interior-point method, however it may deteriorates itself in the last ones, as the matrix Q gets very ill-conditioned.

3.3 Splitting preconditioner

The splitting preconditioner was proposed in (Oliveira & Sorensen, 2005) for the solution of the linear systems that arise from interior-point methods by iterative approaches, specifically by the conjugate gradient method.

The splitting preconditioner was originally developed for the augmented matrix of form:

$$\begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix}.$$

The final matrix preconditioned takes the form:

$$M^{-1} \begin{pmatrix} -D & A^T \\ A & 0 \end{pmatrix} M^{-T} = \begin{pmatrix} -I + D^{-1/2} A^T G^T + G A D^{-1/2} & 0 \\ 0 & -D_B \end{pmatrix}$$

where M is the preconditioned matrix and

$$M^{-1} = \begin{pmatrix} D^{-1/2} & G \\ H & 0 \end{pmatrix}, \quad G = H^T D_B^{-1/2} B^{-1}, \quad H P^T = [I \ 0] \quad \text{and} \quad A P^T = [B \ N].$$

The permutation matrix P is such that the block B has m columns of A linearly independent and in block N as $n - m$ remaining columns.

The preconditioner M may also be defined for the normal equations system. The matrix G may be rewritten as:

$$G = AD^{-1}A^T = BD_B^{-1}B^T + ND_N^{-1}N^T. \quad (8)$$

Multiplying by $D_B^{-1/2}B^{-1}$ and post-multiplying by its transpose in the Equation (8) we obtain the new preconditioned matrix:

$$D_B^{-1/2}B^{-1}(AD^{-1}A^T)B^{-T}D_B^{-1/2} = I + D_B^{-1/2}B^{-1}ND_N^{-1}N^TB^{-T}D_B^{-1/2}.$$

This preconditioner is more efficient near an optimal solution of the problem (4) when the linear system is highly ill-conditioned. In (Oliveira & Sorensen, 2005), the use of a diagonal matrix in the first iterations is proposed and the splitting preconditioner in the final phase of the optimization. This approach does not converge to many problems, because the diagonal preconditioner fails in many of them.

The highest computational cost in the calculation of this preconditioner consists of the construction of the block B of the matrix A , where m columns linearly independent are chosen. The way in which these columns are chosen is fundamental for the good performance of the preconditioner. The authors (Oliveira & Sorensen, 2005) suggest the choice of the first m columns linearly independent of AD^{-1} with lower 1-norm. An advantageous property of this preconditioner is that the block B may be reused by several iterations making these iterations much cheaper.

3.4 Efficiency of the preconditioner

The splitting preconditioner builds a partition B of linearly independent columns and the way these columns are chosen influences its performance.

The authors (Oliveira & Sorensen, 2005) initially used the matrix D . New tests were carried out by the authors with the matrix AD^{-1} , selecting the first m linearly independent rows with lower 1-norm of the columns of this matrix. This new approach improved the performance of the preconditioner.

In this work, some problems have been tested using the ordering based on $AD^{-1/2}$, the results did not show better performance. Using the matrix $AD^{-3/2}$ the performance of the preconditioner was improved with some problems.

We also propose a new ordering of the columns of AD^{-1} from the 2-norm. The first m columns linearly independent of AD^{-1} will be chosen with lower 2-norm ($\|AD^{-1}\|_2$). This choice has shown better results in the performance of the splitting preconditioner. The number of iterations of the conjugate gradient method in the solution of the linear system was reduced.

3.5 Change of preconditioner

In the hybrid approach (Bocanegra *et al.*, 2007), several conditions were verified for the change of preconditioners as well as for the increase of the number of nonzero elements in the controlled Cholesky factorization.

The change of preconditioners is the fundamental point in the hybrid approach, because the efficiency of the preconditioners is very well defined in each phase of the optimization. However, each problem has a different behavior and consequently the phases are not easily identified. Bocanegra *et al.* (2007) realizes the change in the following conditions:

1. The initial GAP is reduced in 10^{-6} and the number of iterations of the conjugate gradient is higher than $m/4$ (m is the number of rows of the matrix A). It indicates that the process of optimization is very advanced and it may be the moment to change the preconditioner.
2. The number of iterations of the conjugate gradient in the solution of the linear system is near $m/2$ (m is the number of rows of the matrix A). This may indicate that the controlled Cholesky factorization is not having a good performance.
3. If more than 10 corrections are made in the diagonal (which means that the matrix \tilde{L} was recalculated more than 10 times). In the construction of CCF, there may be non-positive pivots and to avoid this problem, a value is added to the diagonal (correction in the diagonal) and the matrix \tilde{L} is recalculated.

If none of these three conditions is satisfied, then the η is increased by 10 and the method continues with the controlled Cholesky factorization.

In the study of this approach, it has been observed that these conditions are not always satisfactory. In all the tested problems, when a change of phase is realized very far from solution, the method does not reach the convergence.

In this way we verify that more coherent results are obtained when we use the controlled Cholesky factorization for the most number of iterations as possible. This new heuristic implies that the only condition that should be verified is the value of parameter η .

The new change of phase is realized when:

- If the iterations of the conjugate gradient are greater than $m/6$ then it is verified if η reached a maximum value. If so, the change of the phases is realized, otherwise, η is increased by 10 the process continues with the preconditioner obtained by the controlled Cholesky factorization. Note that η is only used in the first phase.

For reasonable values of η the preconditioner obtained by the controlled Cholesky factorization is cheaper in its construction and requires less memory than the splitting preconditioner and therefore the highest number of iterations possible should be maintained. For some problems this approach fails, because very close to a solution the controlled Cholesky factorization can not work well. In the numerical tests presented in the following section these results will be verified.

4 NUMERICAL EXPERIMENTS

The hybrid preconditioner was integrated to the PCx code (Czyzyk *et al.*, 1999). This code implements a variant of the predictor-corrector algorithm with multiple corrections. The procedures

to solve the linear systems are written in the language C, with the exception of the code of the controlled Cholesky factorization which was implemented in FORTRAN. The same parameters of PCx were used with the exception of the multiple corrections that are not allowed.

4.1 Test problems

All the tested problems are of public domain. The QAP problems are from the QAPLIB library (Burkard *et al.*, 1991) with the modifications described by Padberg and Rijal (1996). Some of the NUG problems were modified and the letter M, at the end of the name, was added as indicative. The Table 1 summarizes the problems tested showing the number of rows and columns after the preprocessing.

Table 1 – Test problems data.

Problems	Rows	Columns
CHR25A	8149	15325
STE36A	27683	131076
STE36B	27683	131076
QAP12	2794	8856
QAP15	5698	22275
NUG08M	742	1632
NUG12M	2794	8856
NUG15M	5698	22275
NUG05	210	225
NUG05-3RD	1410	1425
NUG06	372	486
NUG06-3RD	3972	4686
NUG07	602	931
NUG07-3RD	9742	12691
NUG08	912	1632
NUG08-3RD	19728	29856
NUG12	3192	8856
NUG15	6330	22275

4.2 Computational results

The performance of the hybrid approach proposed by Bocanegra *et al.* (2007) was compared with the performance of the new approach with the new condition of change of phases and the new ordering by the 2-norm. The results are summarized in Table 2.

In the column “Iterations”, the number of iterations of both approaches to reach the optimal is shown. In the column TIME, we have the computational times measured in an Intel Pentium IV 3.4GHz machine with 2Gb of memory using the Linux operating system.

Table 2 – Comparison between the hybrid approach and the new approach.

Problems	Iterations		Time	
	Hybrid	New	Hybrid	New
CHR25A	28	28	83.90	66.58
STE36A	37	38	31086.77	48532.79
STE36B	37	37	83347.97	58825.81
QAP12	20	20	475.59	293.0
QAP15	23	23	4866.9	3672.32
NUG08M	10	9	2.85	2.21
NUG12M	20	20	521.12	255.14
NUG15M	*	23	*	4597.23
NUG05	*	8	*	0.11
NUG05-3RD	*	6	*	2.32
NUG06	*	6	*	2.32
NUG06-3RD	*	7	*	27.2
NUG07	11	10	1.92	1.13
NUG07-3RD	*	8	*	219.29
NUG08	10	9	2.8	2.24
NUG08-3RD	*	9	*	1260.54
NUG12	20	20	473.0	245.91
NUG15	*	23	*	3526.91

*: The method has not converged.

The initial value of η in most of the problems was obtained from the average of the nonzero elements of the matrix $A(Mel)$. In problems of higher scale as in the NUG15 the initial η is obtained adding 100 to the value of Mel . In general, the best time in the construction of CCF is obtained when we use the initial value of $\eta = -Mel$, but this approach did not have good performance for the bigger problems.

In most of the tested problems it was possible to decrease the computational time and for all of them the optimal was reached. The hybrid approach of Bocanegra *et al.* (2007) realizes the change of preconditioners in a stage in the process of optimization where the splitting preconditioner still does not obtain a good computational performance. The tests have shown that the best results are obtained when the CCF is used until the end of the optimization process.

Table 3 shows the average of iterations realized using the new ordering by the 2-norm of the columns of AD^{-1} . The column “Change” shows the iteration where the change of preconditioners was realized and the “Average” column shows the average of iterations carried out after the change.

In the comparison between the two approaches considering the iteration of change of phase we can verify that with the new approach the change is realized in a more advanced stage of the process of optimization. Even when the splitting preconditioner manages to decrease the number

of iterations of the conjugate gradient compared with the CCF, the use of the CCF for some more iterations continues to be more advantageous, due to the computational effort to calculate B .

Table 3 – Comparison between the hybrid approach and the new approach in the change of preconditioners.

Problems	Change		Average	
	Hybrid	New	Hybrid	New
CHR25A	12	11	534	415
STE36A	26	28	3028	2553
STE36B	25	27	7343	4794
QAP12	2	5	777	405
QAP15	6	12	1676	1117
NUG08M	4	5	88	61
NUG12M	3	7	781	440
NUG15M	10	11	*	1473
NUG05	*	–	*	–
NUG05-3RD	2	–	*	–
NUG06	–	4	–	50
NUG06-3RD	2	–	*	–
NUG07	7	4	52	52
NUG07-3RD	2	–	*	–
NUG08	4	5	88	65
NUG08-3RD	5	–	*	–
NUG12	3	7	2406	440
NUG15	10	17	*	1818

*: The method has not converged; –: There was no change of preconditioner.

5 CONCLUSIONS

The problems presented in this work were tested by Bocanegra *et al.* (2007) using the PCx with the classic approach using direct methods for the solution of the linear systems. In most of the problems tested, it was possible to overcome the time of execution, the number of iterations and in some cases it was even possible to run new problems.

In this work, a new study of the hybrid approach applied to large-scale problems has been carried out. The study presents a new change of phase and a new reordering for the calculation of the matrix B of the splitting preconditioner.

The new proposal for the change of preconditioners is more robust. The preconditioner obtained with CCF is much cheaper and uses less memory than the splitting preconditioner. This suggests that it may be more advantageous to maintain it for the most number of iterations possible and with this new condition of change this is guaranteed. An initial proposal we are testing for the limit of η is the average of the nonzero values of the matrix (bearing in mind that the initial value of η is calculated from this average, but with negative sign).

The splitting preconditioner is very efficient in the last iterations of the interior-point method when the matrix D obtained from the values of x and z has a well defined separation. The main point which defines the good operation is the choice of the columns which form the partition B . The ideas developed in this work propose a new approach of ordering for the choice of this matrix that, as it has been shown, works better for all the problems tested. However, this ordering is not always respected, because from it the first m columns linearly independent are chosen and it may occur that the last columns are part of the matrix B . This makes the construction of B very expensive and still does not produce the expected results. Future works aim to improve this performance and eliminate these problems.

REFERENCES

- [1] ADLER I, KARMAKAR N, RESENDE MGC & VEIGA G. 1989. Data structures and programming techniques for the implementation of Karmarkar's algorithm. *ORSA Journal on Computing*, pp. 84–106.
- [2] BERGAMASCHI L, GONDZIO J & ZILLI G. 2004. Preconditioning Indefinite Systems in Interior Point Methods for Optimization. *Computational Optimization and Applications*, **28**: 149–171.
- [3] BOCANEGRA S, CAMPOS FF & OLIVEIRA ARL. 2007. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications* (**1-2**): 149–164. Special issue on “Linear Algebra Issues arising in Interior Point Methods”.
- [4] BURKARD RS, KARISCH S & RENDL F. 1991. QAPLIB – A quadratic assignment problem library. *European Journal of Operations Research*, **55**: 115–119.
- [5] CAMPOS FF & BIRKETT NRC. 1998. An efficient solver for multi-right-hand-side linear systems based on the CCCG(η) method with applications to implicit time-dependent partial differential equations. *SIAM J Sci Comput*, **19**(1): 126–138.
- [6] CZYZYK J, MEHROTRA S, WAGNER M & WRIGHT SJ. 1999. PCx an interior point code for linear programming. *Optimization Methods and Software*, **11**(2): 397–430.
- [7] GOLUB GH & VAN LOAN CF. 1996. *Matrix Computations*. Third Edition, The Johns Hopkins University Press, Baltimore.
- [8] JONES MT & PLASSMANN PE. 1995. An improved Incomplete Cholesky Factorization. *ACM Trans on Math Software*, **21**(1): 5–17.
- [9] LUSTIG IJ, MARSTEN RE & SHANNO DF. 1990. On Implementing Mehrotra's Predictor-Corrector Interior Point Method for Linear Programming. *TR SOR* 90-03.
- [10] MEHROTRA S. 1992. On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, **2**(4): 575–601.
- [11] MEHROTRA S. 1992b. Implementations of Affine Scaling Methods: Approximate Solutions of Systems of Linear Equations Using Preconditioned Conjugate Gradient Methods. *ORSA Journal on Computing*, **4**(2): 103–118.
- [12] MONTEIRO RDC, ILAN ADLER & RESENDE MGC. 1990. A Polynomial-time Primal-Dual Affine Scaling Algorithm for Linear and Convex Quadratic Programming and its Power Series Extension. *Mathematics of Operations Research*, **15**: 191–214.

- [13] 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.
- [14] PADBERG M & RIJAL MP. 1996. *Location, Scheduling, Design and Integer Programming*, Kluwer Academic, Boston.
- [15] WANG W & O'LEARY DP. 2000. Adaptive use of iterative methods in predictor-corrector interior point methods for linear programming. *Numerical Algorithms*, **25**: 387–406.
- [16] WRIGHT SJ. 1996. *Primal-Dual Interior-Point Methods*. SIAM Publications, SIAM, Philadelphia, PA, USA.