Improving an interior-point approach for large block-angular problems by hybrid preconditioners

Silvana Bocanegra
Univ. Federal Rural
de Pernambuco
silvana@deinfo.ufrpe.br

Jordi Castro
Univ. Politècnica
de Catalunya
jordi.castro@upc.edu

Aurelio R.L. Oliveira
Univ. Estadual
de Campinas
aurelio@ime.unicamp.br

Research Report UPC-DEIO DR 2012-06
July 2012

Report available from http://www-eio.upc.es/~jcastro

Improving an interior-point approach for large block-angular problems by hybrid preconditioners

Silvana Bocanegra[a,b], Jordi Castro[b], Aurelio R. L. Oliveira[c],

[a]*Dept. de Estatística e Informática, Universidade Federal Rural de Pernambuco,*
R. Dom Manoel de Medeiros, s/n, Dois Irmãos - CEP: 52171-900, Recife, Brazil.

[b]*Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya,*
Campus Nord, Office C5 203, Jordi Girona 1-3, 08034 Barcelona, Catalonia.

[c]*Dept. de Matemática Aplicada - Universidade Estadual de Campinas,*
C.P. 6065, 13.083-970 Campinas - SP, Brazil.

Abstract

The computational time required by interior-point methods is often dominated by the solution of linear systems of equations. An efficient specialized interior-point algorithm for primal block-angular problems has been used to solve these systems by combining Cholesky factorizations for the block constraints and a conjugate gradient based on a power series preconditioner for the linking constraints. In some problems this power series preconditioner resulted to be inefficient on the last interior-point iterations, when the systems became ill-conditioned. In this work this approach is combined with a splitting preconditioner based on LU factorization, which is mainly appropriate for the last interior-point iterations. Computational results are provided for three classes of problems: multicommodity flows (oriented and nonoriented), minimum-distance controlled tabular adjustment for statistical data protection, and the minimum congestion problem. The results show that, in most cases, the hybrid preconditioner improves the performance and robustness of the interior-point solver. In particular, for some block-angular problems the solution time is reduced by a factor of 10.

Key words: Interior-point methods, Large-scale optimization,

Email addresses: silvana@deinfo.ufrpe.br (Silvana Bocanegra), jordi.castro@upc.edu (Jordi Castro), aurelio@ime.unicamp.br (Aurelio R. L. Oliveira)

Preconditioned conjugate gradient, Structured problems.

1. Introduction

Many important large-scale optimization problems exhibit a block-angular structure. Applications are found in fields such as control and planning, network flows, stochastic linear programming, and statistical data protection. Several interior-point methods have been devised to solve these structured problems [4, 6, 9, 13, 18]. These specialized algorithms exploit the particular structure of the constraints matrix, and some were implemented for parallel environments [4, 18]. The efficiency of interior-point methods critically depends of the linear system solver used at each iteration to compute the Newton direction. Such systems are often written in a symmetric indefinite form, known as the *augmented system*. They can also be reduced to a smaller positive definite form, the *normal equations*. Techniques based on direct and iterative solvers can be applied for their solution. For some classes of large scale problems the use of direct methods becomes prohibitive due to storage and time limitations, whereas iterative linear solvers with appropriate preconditioners may be more efficient.

The efficient interior-point algorithm for primal block-angular problems of [12] solved the normal equations in two stages: Cholesky factorizations for the block constraints and a Preconditioned Conjugate Gradient (PCG) for the linking constraints. The purpose of PCG is to avoid solving the system associated to the complicating linking constraints by Cholesky factorizations, in an attempt to make the problem block separable. The preconditioner is obtained by truncating an infinite power series that approximates the inverse of the system to be solved. For some difficult primal block-angular problems this approach outperformed state-of-the-art commercial solvers [13]. However, in some problems, systems become very ill-conditioned as the optimal solution is reached, and then PCG provides slow and inaccurate solutions. It was shown [13] that the efficiency of this approach depends on the spectral radius —in $[0, 1)$— of a certain matrix which appears in the definition of the preconditioner (which is itself related to the Schur complement of the normal equations). Spectral radius close to 1 degrade the performance of the preconditioner. When PCG gives inaccurate solutions, the code implemented in [12] switches to the solution of the normal equations by a Cholesky factorization, which may be prohibitive for large-scale problems.

In order to yield a reliable and efficient interior-point method based just on iterative solvers we introduce a hybrid and adaptive scheme for solving the normal equations. On the first interior-point iterations the normal equations are solved using the Cholesky-PCG approach of [12] outlined above. When the system associated to linking constraints becomes ill-conditioned, the normal equations are solved by a PCG using the splitting preconditioner of [22, 23], instead of switching to a direct solver. The splitting preconditioner is a generalization of the tree preconditioner of [25] for large-scale minimum cost network flow problems. Based on a LU factorization, the splitting preconditioner was specially tailored for the last interior-point iterations, when the systems are ill-conditioned. We developed a new and efficient criterion to identify when (i.e., at which interior-point iteration) to switch between iterative solvers. This criterion is based on both the Ritz values of the matrix that appears in the definition of the power series preconditioner, and the number of PCG iterations needed at each interior-point iteration. The Ritz values are approximations of the eigenvalues of a matrix; they will be used to estimate the spectral radius, which measures the efficiency of the power series preconditioner. An implementation of this new approach, combining the power series and the splitting preconditioners, was applied to three classes of primal block-angular instances [12]: multicommodity flows (oriented and nonoriented), minimum-distance controlled tabular adjustment for statistical data protection, and the minimum congestion problem. As it will be shown, the hybrid approach was more efficient than the power series preconditioner in many block-angular problems.

This paper is organized as follows. In Section 2 we recall the basic ideas of interior-point methods for primal block-angular problems using the power series preconditioner. The new hybrid approach is described in Section 3, together with an outline of the splitting preconditioner, and a description of the switching criterion between preconditioners. Numerical experiments are shown in Section 4. The effect of different regularization parameters for the splitting preconditioner are also discussed in Section 4. Finally, in Section 5 the conclusions are drawn and further developments are suggested.

2. The interior-point algorithm for primal block-angular problems

One of the most efficient interior-point methods for some classes of block-angular problems was initially developed for multicommodity flows [9] and later extended for general primal block-angular problems [12]. This method considers the following general formulation of a block-angular problem:

$$\begin{array}{ll} \min & \displaystyle\sum_{i=1}^{k} ({c^i}^T x^i + {x^i}^T Q_i x^i) \\ \text{subject to} & \begin{bmatrix} N_1 & & & & \\ & N_2 & & & \\ & & \ddots & & \\ & & & N_k & \\ L_1 & L_2 & \dots & L_k & I \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^k \\ x^0 \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 \\ \vdots \\ b^k \\ b^0 \end{bmatrix} \\ & 0 \le x^i \le u^i \quad i = 1, \dots, k. \end{array} \tag{1}$$

Matrices $N_i \in \mathbb{R}^{m_i \times n_i}$ and $L_i \in \mathbb{R}^{l \times n_i}$, $i = 1, \dots, k$, define, respectively, the block and linking constraints, k being the number of blocks. Vectors $x^i \in \mathbb{R}^{n_i}, i = 1, \dots, k$, are the variables for each block. $x^0 \in \mathbb{R}^l$ are the slacks of the linking constraints. $b^i \in \mathbb{R}^{m_i}, i = 1, \dots, k$ is the right-hand-side vector for each block of constraints, whereas $b^0 \in \mathbb{R}^l$ is for the linking constraints. The upper bounds for each group of variables are defined by $u^i, i = 1, \dots, k$. This formulation considers the general form of linking constraints $b^0 - u^0 \le \sum_{i=1}^k L_i x^i \le b^0$. $c^i \in \mathbb{R}^{n_i}$ and $Q_i \in \mathbb{R}^{n_i \times n_i}$, $i = 1, \dots, k$, are the linear and quadratic costs for each group of variables. We also consider linear and quadratic costs $c^0 \in \mathbb{R}^l$ and $Q_0 \in \mathbb{R}^{l \times l}$ for the slacks. We restrict our considerations to the separable case where $Q_i, i = 0, ..., k$, are diagonal positive semidefinite matrices.

Problem (1) can be written in standard form as

$$\begin{array}{rl} \min & c^T x + \frac{1}{2} x^T Q x \\ \text{subject to} & Ax = b \\ & 0 \le x \le u, \end{array} \tag{2}$$

where $A \in \mathbb{R}^{m \times n}$ ($m = l + \sum_{i=1}^k m_i$, $n = l + \sum_{i=1}^k n_i$ and $m \le n$), $Q \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^m$ and $c, x, u \in \mathbb{R}^n$. Replacing inequalities in (2) by a logarithmic barrier with parameter $\mu > 0$, we obtain the logarithmic barrier problem

$$\begin{array}{ll} \min & B(x, \mu) \triangleq c^T x + \frac{1}{2} x^T Q x - \mu \displaystyle\sum_{i=1}^n \ln(x_i) - \mu \sum_{i=1}^n \ln(u_i - x_i) \\ \text{subject to} & Ax = b. \end{array} \tag{3}$$

The first order KKT optimality conditions for the logarithmic barrier problem —or equivalently, the perturbed KKT-μ conditions for (2)— are

$$\begin{array}{rcl} Ax & = & b, \\ A^T y - Qx + z - w & = & c, \\ XZe & = & \mu e, \\ (U-X)We & = & \mu e, \\ (z,w) & > & 0, \quad u > x > 0, \end{array} \tag{4}$$

where $y \in \mathbb{R}^m$, $z \in \mathbb{R}^n$, $w \in \mathbb{R}^n$ are, respectively, the Lagrange multipliers of constraints $Ax = b$, $x \geq 0$ and $x \leq u$, $X, Z, U, W \in \mathbb{R}^{n \times n}$ are diagonal matrices made up of vectors x, z, u, w, and $e \in \mathbb{R}^n$ is a vector of 1's. The first two sets of equations of (4) impose, respectively, primal and dual feasibility, while the remaining two impose perturbed complementarity. The set of primal-dual solutions $\mathcal{C} = \{(x_\mu, y_\mu, z_\mu, w_\mu), \mu > 0\}$ of (4) is known as the *central path.* Primal-dual path-following interior-point algorithms approximately follow the central path by applying Newton's method to the nonlinear system of equations (4), reducing the barrier parameter μ at each iteration. When $\mu \rightarrow 0$ these solutions converge to the optimal solution of the original problem. Full details can be found in [28]. The Newton direction is obtained by solving a linear system in variables Δx, Δy, Δz and Δw. In practice, variables Δz and Δw are eliminated and the system reduces to the indefinite augmented system form

$$\left[\begin{array}{cc} -\Theta^{-1} & A^T \\ A & \end{array} \right] \left[\begin{array}{c} \Delta x \\ \Delta y \end{array} \right] = \left[\begin{array}{c} r \\ r_b \end{array} \right], \tag{5}$$

where Θ and r are defined as

$$\Theta = (Q + S^{-1}W + X^{-1}Z)^{-1} \qquad r = r_c + S^{-1} r_{sw} - X^{-1} r_{xz}, \tag{6}$$

and $S = U - X$. Eliminating Δx from the first group of equations system (5) is reduced to the smaller positive definite normal equations

$$(A\Theta A^T)\Delta y \quad = \quad r_b + A\Theta r = g. \tag{7}$$

For separable quadratic optimization problems Q and Θ are diagonal, and normal equations are usually the preferred choice for computing the Newton direction.

2.1. Normal equations for block-diagonal problems

The performance of interior-point methods relies on the efficient solution of either (5) or (7). For block-angular problems (1) matrices A and Θ have a special structure. The interior-point algorithm used in this work [9, 12] solves the normal equations (7) by exploiting the block decomposition of $A\Theta A^T$:

$$A\Theta A^T = \left[\begin{array}{ccc|c} N_1\Theta_1 N_1^T & & & N_1\Theta_1 L_1^T \\ & \ddots & & \vdots \\ & & N_k\Theta_k N_k^T & N_k\Theta_k L_k^T \\ \hline L_1\Theta_1 N_1^T & \dots & L_k\Theta_k N_k^T & \Theta_0 + \sum_{i=1}^k L_i\Theta_i L_i^T \end{array}\right]. \tag{8}$$

Considering the blocks of $A\Theta A$ and partitioning appropriately the dual variables direction Δy and the right-hand-side vector g, the normal equations system (7) can be written as

$$\left[\begin{array}{cc} B & C \\ C^T & D \end{array}\right]\left[\begin{array}{c} \Delta y_1 \\ \Delta y_2 \end{array}\right] = \left[\begin{array}{c} g_1 \\ g_2 \end{array}\right]. \tag{9}$$

where $B \in \mathbb{R}^{\tilde{m}\times\tilde{m}}$ ($\tilde{m} = \sum_{i=1}^k m_i$), $C \in \mathbb{R}^{\tilde{m}\times l}$ and $D \in \mathbb{R}^{l\times l}$ are the blocks of $A\Theta A^T$, and Θ_i, $i = 0, \dots, k$, are the submatrices of Θ, i.e., $\Theta_i = (Q_i + S_i^{-1}W_i + X_i^{-1}Z_i)^{-1}$. By eliminating Δy_1 from the first group of equations of (9), we obtain

$$\begin{aligned} (D - C^T B^{-1} C)\Delta y_2 &= (g_2 - C^T B^{-1} g_1) & (10) \\ B\Delta y_1 &= (g_1 - C\Delta y_2). & (11) \end{aligned}$$

System (10) is solved by PCG using the power series preconditioner described below. The symmetric and positive definite matrix $D - C^T B^{-1} C$ is the Schur complement of (9) and it has dimension l, which is the number of linking constraints. System (11) is solved by a Cholesky factorization for each diagonal block $N_i\Theta_i N_i^T, i = 1 \dots k$, of B.

2.2. Power series preconditioner

The power series preconditioner for the Schur complement matrix was initially applied to multicommodity flows [9] and later extended to any general primal block-angular problem [12]. This preconditioner uses the relation

[9]

$$(D - C^T B^{-1} C)^{-1} = \left(\sum_{i=0}^{\infty} (D^{-1}(C^T B^{-1} C))^i \right) D^{-1}$$

to compute an approximation of the inverse of the Schur complement. This approximation is obtained by truncating this series at some term h. Clearly, the more terms h, the better the preconditioner, at the expense of increasing the execution time of each PCG iteration. In practice, performances are best for $h = 0$ and, in some cases, for $h = 1$. The preconditioner M^{-1} for these choices are

$$\begin{array}{rcll} M^{-1} & = & D^{-1} & \text{if } h = 0, \\ M^{-1} & = & (I + D^{-1}(C^T B^{-1} C))D^{-1} & \text{if } h = 1. \end{array}$$

The value $h = 0$ was used for the computational results of Section 4. The effectiveness of this preconditioner depends on the spectral radius of $D^{-1}(C^T B^{-1} C)$, which is always in $[0, 1)$ [9, Theorem 1]. The farther away from 1, the closer M^{-1} is to $(D - C^T B^{-1} C)^{-1}$. In the last interior-point iterations matrix Θ is very ill-conditioned: some values of matrix Θ go to zero whereas others tend to infinity. This results in spectral radius of matrix $D^{-1}(C^T B^{-1} C)$ very close to 1, which hinders the efficient solution of (7).

2.3. Effect of regularizations on the power series preconditioner

Motivated by the observed better behavior of this specialized interior-point algorithm for separable quadratic than for linear instances, a quadratic regularization term was added to the logarithmic barrier to improve the quality of the power series preconditioner [13]. The logarithmic barrier problem for linear problems (3) (considering $Q = 0$) was replaced by the alternative regularized version

$$B_Q(x, \mu) \triangleq c^T x + \mu \left(\frac{1}{2} x^T Q x - \sum_{i=1}^{n} \ln x_i - \sum_{i=1}^{n} \ln(u_i - x_i) \right), \tag{12}$$

where Q is a diagonal positive semidefinite matrix and *dom* $B_Q = \{x : 0 < x < u\}$. Unlike other approaches [2], the reduction to zero of the regularization term is governed by the barrier parameter μ. This regularized barrier problem only changes the dual feasibility equations of (4), which become

$$A^T y + z - w = c + \mu Q x, \tag{13}$$

and the matrix Θ defined in (6) which is replaced by

$$\Theta = (\mu Q + S^{-1}W + X^{-1}Z)^{-1}. \tag{14}$$

When $\mu \rightarrow 0$, (13) and (14) converge to the expressions of the non-regularized algorithm. The quadratic regularization was shown to reduce the spectral radius of the power series preconditioner [13, Theorem 1, Proposition 1], improving the overall performance of the interior-point method in some classes of instances.

3. The hybrid approach for normal equations

The hybrid approach works as follows. Initially, the normal equations are solved by the procedure described in Section 2.1, i.e., solving (11) by Cholesky factorizations and (10) by PCG with the power series preconditioner. When the power series preconditioner becomes inefficient, then the method switches to the solution of the normal equations (7) by PCG with the splitting preconditioner [23]. The splitting preconditioner and the efficient criteria developed to identify the switch between preconditioners are described in next two subsections.

3.1. The Splitting preconditioner

The splitting preconditioner was introduced in [22, 23] for solving the augmented systems in general linear programming problems. This preconditioner is a generalization of the tree preconditioner introduced in [25], and extended in [15], for minimum cost network flow problems. An important feature of this preconditioner is the option to reduce the preconditioned indefinite system to the positive definite normal equations allowing the use of CG. The main appeal of this class of preconditioners is that they work better near a solution of the linear programming problem. This is a very welcome feature since the linear system is known to be very ill-conditioned close to the optimizer, making difficult its solution by iterative methods. However, since the preconditioner is specially tailored for the final iterations of the interior-point method, it may fail in the initial ones. Hybrid approaches with Cholesky controlled preconditioners on early iterations have been successfully used in large-scale linear programming [7, 17, 26]. A version of the splitting preconditioner for normal equations can be obtained as follows.

Let $AP = [B \;\; N]$ be a basic-nonbasic partition of the matrix A, i.e., $B \in \mathbb{R}^{m \times m}$ and $N \in \mathbb{R}^{(n-m)\times(n-m)}$, where P is a permutation matrix such that B is nonsingular. Considering the same partitioning for Θ, the normal equations can be rewritten as

$$A\Theta A^T = B\Theta_B B^T + N\Theta_N N^T. \tag{15}$$

The symmetric application of the preconditioner $M^{-1} = \Theta_B^{-\frac{1}{2}} B^{-1}$ to matrix (15) gives:

$$\begin{aligned} M^{-1}(A\Theta A^T)M^{-T} &= \Theta_B^{-\frac{1}{2}} B^{-1}(B\Theta_B B^T + N\Theta_N N^T)B^{-T}\Theta_B^{-\frac{1}{2}} \qquad (16)\\ &= I + (\Theta_B^{-\frac{1}{2}} B^{-1} N \Theta_N^{\frac{1}{2}})(\Theta_B^{-\frac{1}{2}} B^{-1} N \Theta_N^{\frac{1}{2}})^T \\ &= I + WW^T. \end{aligned}$$

Sufficiently close to an optimal solution at least $n - m$ entries of Θ are small. Thus, with a suitable choice of the columns of B, the diagonal entries of Θ_B^{-1} and Θ_N are very small close to a solution. In this situation, W approaches the zero matrix and the preconditioned matrix (16) approaches the identity. The preconditioner requires to find and to solve linear systems with B. Identifying a suitable basis matrix B is a nontrivial task. In [23], the first m linearly independent columns of $A\Theta$ with smallest 1-norm are selected for B. However, 1-norm has a tendency to diminish the effect of outliers, a feature not desirable in this context since the goal is to split the columns in two sets of size m and $n-m$, respectively. The use of the 2-norm instead has improved the performance of the splitting preconditioner for most problems [26]. In this work we are using the 2-norm criteria.

The techniques developed in [23] for determining the subset of columns B from A for the splitting preconditioner are rather costly and sophisticated. Fortunately, a nice property of the splitting preconditioner is that it can be reused for some iterations. It is important to notice that keeping the matrix B from previous iterations does not mean to keep the same preconditioner since Θ will be different. In this work the matrix B is recomputed if the PCG iterations exceeds 2% of the matrix size. This heuristic worked fine for the instances tested.

3.2. Switching between preconditioners

The criteria to identify the switch between preconditioners is instrumental for the performance of this approach. As shown in [9, Theorem 1], the effectiveness of the power series preconditioner depends on the spectral radius of matrix $D^{-1}C^TB^{-1}C$. We developed a heuristic based on Ritz Values to estimate the spectral radius of this matrix. It is based on the following relation between the eigenvalues of $D^{-1}C^TB^{-1}C$ and those of the preconditioned matrix of (10), the system solved by the power series preconditioner.

Proposition 1. *Let v be the eigenvector of matrix $I - D^{-1}(C^TB^{-1}C)$ associated with the eigenvalue λ. Then, v is eigenvector of $D^{-1}(C^TB^{-1}C)$ associated to eigenvalue $1 - \lambda$.*

Proof. Let v be the eigenvector of $I - D^{-1}(C^TB^{-1}C)$ for eigenvalue λ. Then

$$(I - D^{-1}(C^TB^{-1}C))v = \lambda v,$$

and, hence,

$$Iv - D^{-1}(C^TB^{-1}C)v = \lambda v \iff D^{-1}(C^TB^{-1}C)v = (1 - \lambda)v.$$

□

Note that $I - D^{-1}(C^TB^{-1}C)$ is the matrix of (10) preconditioned in asymmetric form (i.e., premultiplied) by D^{-1}. From Proposition 1 it is thus possible to obtain the spectral radius of $D^{-1}(C^TB^{-1}C)$ as follows:

Corollary 1. *Let $\lambda_{\min} \geq 0$ be the minimum eigenvalue of $I - D^{-1}(C^TB^{-1}C)$. Therefore, the spectral radius of $D^{-1}(C^TB^{-1}C)$ is $1 - \lambda_{\min}$.*

The conjugate gradient method is based implicitly on the Lanczos three-term recurrence relations (see for example [19]). The coefficients of these recurrences define a three-diagonal matrix T whose eigenvalues are the Ritz values. At iteration k, T_k is defined by

$$T_k = \begin{pmatrix} \gamma_1 & \eta_2 & & & \\ \eta_2 & \gamma_2 & \eta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \eta_{k-1} & \gamma_{k-1} & \eta_k \\ & & & \eta_k & \gamma_k \end{pmatrix}, \tag{17}$$

where the coefficients γ_i and η_j can be computed from the CG algorithm. Consider the following generic CG algorithm to solve the linear system $Mx = b$.

```
Given $x_0,\ r_0 = b - Mx_0,\ \rho_0 = r_0,\ k = 1$
**while** $r_k \neq 0$ **and** $k < k_{\max}$
    $\alpha_{k-1} = \left( \frac{||r_{k-1}||^2}{(\rho_{k-1}, M\rho_{k-1})} \right)$
    $x_k = x_{k-1} + \alpha_{k-1}\rho_{k-1}$
    $r_k = r_{k-1} - \alpha_{k-1}M\rho_{k-1}$
    $\beta_{k-1} = \left( \frac{||r_k||^2}{||r_{k-1}||^2} \right)$
    $\rho_k = r_k + \beta_k\rho_{k-1}$
**end_while.**
```

The coefficients in matrix T_k can be obtained from the following relations:

$$\gamma_k = \frac{1}{\alpha_{k-1}} + \frac{\beta_{k-1}}{\alpha_{k-2}}, \qquad \beta_0 = 0, \qquad \alpha_{-1} = 0, \qquad \eta_{k+1} = -\frac{\sqrt{\beta_k}}{\alpha_{k-1}}.$$

It is known from Lanczos theory that the eigenvalues of T_k (Ritz values) converge to those of the original matrix M. In general, the extreme eigenvalues of M are well approximated already during early GC iterations [20, 24]. We then used the extreme Ritz values to estimate the smallest eigenvalue $\lambda_{\min}$ of the preconditioned matrix $I - D^{-1}(C^T B^{-1} C)$. From Corollary 1 , as $\lambda_{\min}$ approaches zero, the spectral radius of $D^{-1}(C^T B^{-1} C)$ tends to one, and, consequently, the power series preconditioner becomes less efficient. Ritz values were computed using the SSTEQR LAPACK routine [3].

Supported by the previous results, the following heuristic criteria was developed. The switch between the power series and the splitting preconditioner is performed when all the following three criterion occur:

1. The minimum Ritz value of $I - D^{-1}(C^T B^{-1} C)$ is less than 0.001.
2. The number of PCG iterations with the power series preconditioner reaches $0.05l$, where l is the dimension of $I - D^{-1}(C^T B^{-1} C)$.
3. The centrality parameter μ is less than 0.5.

The two first criterion verify whether the power series preconditioner is becoming less efficient. The last one guarantees enough progress of the interior-point method. The satisfaction of the three criterion is a good indicator that the the splitting preconditioner will work better. In practice, this criteria provided good computational results.

4. Numerical experiments

The hybrid approach described in the previous section has been added to a MATLAB implementation of the specialized algorithm for general block-angular problems, named BlockIP [12]. BlockIP implements a standard infeasible primal-dual path-following algorithm, which solves the normal equations by either the specialized procedure described in Section 2.1 or a Cholesky factorization. The code uses the Ng-Peyton sparse Cholesky package [16, 21] for the solution of (11) and (7); the Ng-Peyton sparse Cholesky package was hooked to MATLAB for the LIPSOL interior-point solver [29]. System (10) associated to the linking constraints is solved by PCG, using the power series preconditioner. PCG may give inaccurate solutions when the matrix of this system becomes ill-conditioned. When this happened, the original version of BlockIP switched to the solution of the full normal equations (7) by a Cholesky factorization. The switch is performed when we are close enough to the optimal point ($gap^i < 0.5$) and gap^i increases from one iteration to the next ($gap^i > 1.05gap^{i-1}$), where $gap^i = |p^i - d^i|/(1+|p^i|)$, p^i and d^i being the primal and dual objective functions at iteration i, respectively.

The splitting preconditioner, initially coded in C, has been hooked to the MATLAB BlockIP implementation. The new version of the BlockIP package is thus able to solve the normal equations by either a Cholesky factorization or PCG using the power series or the splitting preconditioner. The default parameters of BlockIP described in [12] have been adopted, but for the stopping criteria of the PCG and the interior-point method. Since the splitting preconditioner may provide more accurate solutions than the power series preconditioner near the optimal solution, we allowed a more restrictive stopping criterion for the interior-point algorithm: it stops when, at a primal-dual feasible solution, $gap^i < 10^{-6}$ (instead of the original condition $gap^i < 10^{-4}$). In addition, when either $gap^i < 10^{-4}$ or the code switches from the power series to the splitting preconditioner, the stopping criterion of the PCG solver is tightened: it is stopped when at iteration k of PCG the relative residual norm satisfies $||r_k||/||r_0|| < 10^{-8}$. The precision 10^{-8} is also required for PCG if no switch of preconditioner is performed but we are close to the optimal solution ($\mu < 10^{-4}$) and the gap increases ($gap^i > 1.1gap^{i-1}$).

4.1. Test problems

Three classes of primal block-angular problems have been considered for testing the hybrid approach: multicommodity network flow problems (oriented and nonoriented), minimum congestion problems and minimum distance controlled tabular adjustment.

Multicommodity flow problems attempt to route a set of commodities at a minimum cost over a capacitated oriented or nonoriented network. We considered three kinds of oriented instances. The first two correspond to a subset of the PDS [8] and Mnetgen [1] instances. These instances are widely used in the literature, and they can be retrieved from `http://www.di.unipi.it/optimize/Data/MMCF.` The last set corresponds to instances obtained with the Tripartite generator, which can be retrieved from `http://www-eio.upc.edu/~jcastro/mmcnf-data.html`. We are not aware of any standard set of nonoriented multicommodity flow problems, thus nonoriented instances were generated from the oriented Mnetgen ones.

The minimum congestion problem (also known as the maximum concurrent flow problem) [5, 27] arises in practical applications on telecommunications networks. The purpose of this problem is to compute the maximum concurrent flow (or throughput) that can be transported through a network. Equivalently, it can be seen as the problem of finding, in an infeasible multicommodity flows network, the minimum of the maximum relative increments in arc capacities, for each arc of the network, that makes the problem feasible, i.e., all multicommodity flows can be sent from sources to destinations. The formulation used here is described in [12]. The infeasible multicommodity instances were generated from the oriented Mnetgen ones by increasing supplies and demands by a factor of two.

Minimum-distance controlled tabular adjustment (CTA for short) is a technique for the protection of statistical tabular data [10, 11, 14]. This is a major concern for National Statistical Institutes, which must guarantee that individual information cannot be disclosed from released data. For some classes of tables (three-dimensional tables) and using the Euclidean distance, this problem can be formulated as a quadratic multicommodity flow problem with saturated linking constraints [10, 13]. If the L_1 distance is used, the formulation no longer corresponds to a multicommodity flow problem, but it exhibits a block-angular structure. Several formulations for this problem are reported in [14]. In this work we considered some of the instances derived from the most efficient observed formulation of the L_1-CTA problem.

Table 1 summarizes the dimensions of these instances: number of blocks k; number of constraints and variables for each block, m' and n', respectively; number of linking constraints l; and overall number of constraints and variables of the resulting problem, m and n respectively. For minimum con-

Table 1: Dimensions of test instances.

Instances	k	m'	n'	l	m	n
oriented multicommodity flows						
M32-32-12	32	31	486	361	1353	15913
M64-64-12	64	63	511	371	4404	33075
M128-64-12	64	127	1171	860	8988	75804
M128-128-12	128	127	1204	932	17188	155044
PDS-10	11	1398	4792	814	16192	53526
PDS-30	11	4222	16148	2399	48841	180027
PDS-40	11	5651	22059	3199	65360	245848
PDS-50	11	7030	27668	3933	81263	308281
Tripart1	16	191	2096	238	3294	33774
Tripart2	16	767	8432	1029	13301	135941
Tripart3	20	1199	16380	1561	22541	329161
nonoriented multicommodity flows						
M32-32-12	32	31	972	361	1353	31465
M64-64-12	64	63	1022	387	4419	65795
M128-64-12	64	127	2342	896	9024	150784
M128-128-12	128	127	2408	959	17215	309183
controlled tabular adjustment						
CTAL1-10-10-5	5	219	400	100	1195	2100
CTAL1-15-15-10	10	479	900	225	5015	9225
CTAL1-20-20-20	20	839	1600	400	17180	32400
CTAL1-30-30-30	30	859	3600	900	56670	108900
CTAL1-40-40-20	20	3279	6400	1600	67180	129600
CTAL1-50-50-10	10	5099	10000	2500	257450	502500
CTAL1-500-500-50	50	500999	1000000	250000	25299950	50250000
minimum congestion						
M32-32-12	34	992	31591	972	1964	32563
M64-64-12	66	4032	66430	1532	5564	67962
M128-64-12	66	8128	152230	3512	11640	155742
M128-128-12	130	16256	310632	3611	19867	314243

gestion instances columns m' and n' show the overall number of constraints and variables for all the diagonal blocks, since not all the blocks have the same dimension, i.e., $m' = \sum_{i=1}^{k} m_i$ and $n' = \sum_{i=1}^{k} n_i$.

4.2. Effect of regularizations on the splitting preconditioner

As stated in Subsection 2.3, the power series preconditioner is improved by considering a quadratic regularization term in the barrier problem. However, this quadratic term may difficult the computation of the basic-nonbasic partition of the constraints matrix needed by the splitting preconditioner, which was developed for linear problems. The straightforward approach would be to inactivate the regularization term when switching to the splitting preconditioner. Instead, some numerical experiments were carried out to evaluate the impact of different regularization parameters in the performance of the splitting preconditioner. The regularization Q matrix of (12) is computed as

$$Q = t\frac{\delta}{\mu_0}I, \tag{18}$$

where $t \in \mathbb{Z}^+$ is the number of interior-point iterations, $\mu_0 \in \mathbb{R}$ is the value of the barrier parameter at the first interior-point iteration, and $\delta \in \mathbb{R}$ is a parameter to be provided by the user for initializing the regularization matrix Q. The value $\delta = 10^{-6}$ was always fixed for the power series preconditioner, and different values $\delta \in \{10^{-2}, 10^{-4}, 10^{-6}, 0\}$ were considered for the splitting preconditioner. When $\delta = 0$, the standard non-regularized algorithm is being used.

Results are illustrated in Table 2. For each δ, columns PCG and CPU report the number of PCG iterations and CPU time required by the splitting preconditioner. Information is only reported for the subset of the instances where the switch to the splitting preconditioner is performed (as it will be shown later in Table 3).

From Table 2 it is clear that the different regularization parameters affect the PCG and CPU time required by the splitting preconditioner. Note that in some cases (e.g., Tripart3) executions with a smaller number of PCG iterations provide larger CPU times; this may be due, first, to inaccuracies of the operating system timing routines, and second, to different sparsity patterns of the LU factorization of the splitting preconditioner. There is no regularization parameter providing the best performance for all problems. For multicommodity flow instances, the best results, in number of PCG iterations, were obtained with $\delta = 10^{-2}$ (oriented problems) and $\delta = 10^{-4}$

Table 2: Impact of different regularization parameters in the splitting preconditioner.

	$\delta = 10^{-2}$		$\delta = 10^{-4}$		$\delta = 10^{-6}$		$\delta = 0$	
Instances	PCG	CPU	PCG	CPU	PCG	CPU	PCG	CPU
oriented multicommodity flows								
M32-32-12	**477**	0.16	**477**	0.17	**477**	0.19	**477**	**0.15**
M64-64-12	**796**	**1.22**	799	1.25	798	1.25	801	1.28
M128-64-12	1755	9.03	1759	9.06	**1751**	**8.90**	1754	9.08
M128-128-12	**3354**	20.86	3385	20.95	3366	20.98	3362	**20.69**
PDS-10	1087	1.78	1135	1.85	1084	**1.77**	**1082**	1.78
PDS-30	**4717**	35.79	4826	35.75	4806	35.74	4801	**35.6**
PDS-40	4766	63.33	4849	65.41	**4756**	64.72	**4756**	**63.77**
PDS-50	5729	336.27	5937	315.25	5721	314,0	**5716**	**312.94**
Tripart1	**550**	1.69	771	1.09	755	**1.03**	757	1.04
Tripart2	1988	80.11	1461	**77.82**	**1451**	85.85	1456	83.79
Tripart3	**6011**	1069.61	6458	**884.78**	6423	902.04	6426	910.96
nonoriented multicommodity flows								
M32-32-12	369	0.22	369	0.24	373	**0.21**	**368**	0.23
M64-64-12	913	5.13	913	5.13	**900**	5.08	908	**5.06**
M128-64-12	2354	55.52	**2313**	53.76	2317	**51.07**	2346	51.10
M128-128-12	3267	142.01	**3242**	**133.83**	3244	142.48	3250	151.96
controlled tabular adjustment								
CTAL1-15-15-10	1825	1.78	**1366**	**1.24**	1472	1.52	1432	1.50
CTAL1-20-20-20	3426	15.44	3145	12.25	3118	**11.34**	**3096**	11.96
CTAL1-30-30-30	14681	223.18	12524	233.51	**12377**	**220.18**	12415	229.93
CTAL1-40-40-20	11157	303.07	**8458**	**230.34**	8618	234.66	8851	232.16
CTAL1-50-50-10	16255	270.10	**14476**	266.84	14617	264.65	14592	**253.38**
minimum congestion								
M128-64-12	†	†	†	†	**5261**	**28.53**	6429	35.63

† Numerical problems when $gap^i < 10^{-5}$

Number of instances with best results

for PCG	6		6		**7**		6	
for CPU		1		5		**8**		7

(nonoriented instances). The value $\delta = 10^{-4}$ also reported the best results for the CTA instances. However, the hybrid approach failed for the M128-64-12 minimum congestion instance when $\delta = 10^{-2}$ and $\delta = 10^{-4}$. As expected, in general, the small regularization parameters, i.e., $\delta = 10^{-6}$ and $\delta = 0$, performed well for most instances and reported the maximum number of best executions. The value $\delta = 10^{-6}$ will be used for the computational results of next subsection.

4.3. Computational results

Table 3 shows the computational results obtained. All the instances were ran with the original version of the BlockIP code (power series preconditioner followed by Cholesky factorization if ill-conditioning is found), and with the new version implementing the hybrid approach (power series followed by splitting preconditioner). Columns “original” and “hybrid” of Table 3 correspond to these two executions, respectively. For each test problem, columns “Iter”, “CPU” and “PCG” provide, respectively, the number of interior-point iterations, the CPU time in seconds and the number of PCG iterations. The numbers in brackets of columns “Iter” show the interior-point iteration at which the switch between linear solvers is performed (from power series preconditioner to Cholesky in the “original” columns, and from power series to splitting preconditioner in the “hybrid” columns). All runs were carried out on a Fujitsu Primergy RX300 server with 3.33 GHz Intel Xeon X5680 CPUs (24 cores) and 144 GB of RAM, under a GNU/Linux operating system (Suse 11.4), without exploitation of parallelism capabilities. The fastest executions are marked in boldface.

For the oriented multicommodity Mnetgen instances M32-32-12, M64-64-12 and M128-64-12 the hybrid approach significantly reduced the number of interior-point iterations performed. This is because the switch to the splitting preconditioner is performed very soon, and it provided better solutions than the power series preconditioner. For PDS there is no significant difference in the number of PCG iterations between the two approaches. For the Tripart instances, the “hybrid” approach is much better in number of PCG iterations. For all the instances the “hybrid” approach reduced the CPU time.

Similar results are obtained for nonoriented multicommodity flow problems. All the runs with the “original” approach switched from the power series preconditioner to the Cholesky factorization. We observed that normal equations are nearly singular when the switch is performed; directions computed by the Cholesky factorizations are inaccurate, thus requiring a

Table 3: Results for test instances.

	original			hybrid		
Instance	Iter	CPU	PCG	Iter	CPU	PCG
oriented multicommodity flows						
M32-32-12	$67^{(57)}$	15.79	1701	$37^{(25)}$	**2.46**	583
M64-64-12	$105^{(52)}$	114.35	948	$49^{(39)}$	**7.84**	1064
M128-64-12	83	504.23	31569	$60^{(46)}$	**20.66**	2053
M128-128-12	79	416.29	13344	$75^{(62)}$	**53.93**	3787
PDS-10	85	238.87	4555	$85^{(82)}$	**210.54**	4733
PDS-30	132	1123.87	11347	$133^{(127)}$	**998.96**	13660
PDS-40	139	1364.93	12205	$140^{(132)}$	**1064.43**	14264
PDS-50	144	2062.43	18203	$143^{(136)}$	**1877.33**	22062
Tripart1	58	12.21	2221	$53^{(49)}$	**8.64**	1760
Tripart2	79	1424.13	13258	$78^{(71)}$	**280.02**	3598
Tripart3	91	6057.77	34720	$84^{(70)}$	**1250.55**	8756
nonoriented multicommodity flows						
M32-32-12	$61^{(46)}$	15.66	1330	$38^{(28)}$	**3.12**	518
M64-64-12	$133^{(62)}$	164.81	1212	$58^{(47)}$	**17.09**	1410
M128-64-12	$95^{(71)}$	660.83	2774	$64^{(47)}$	**74.15**	2604
M128-128-12	$101^{(73)}$	3688.66	1248	$76^{(61)}$	**191.01**	3666
controlled tabular adjustment						
CTAL1-10-10-5	$70^{(65)}$	1.33	544	21	**0.48**	191
CTAL1-15-15-10	46	11.23	4499	$29^{(20)}$	**2.33**	1611
CTAL1-20-20-20	$47^{(40)}$	36.13	5514	$33^{(21)}$	**14.15**	3332
CTAL1-30-30-30	$200^{(63)*}$	2787.64	19539	$39^{(24)}$	**255.46**	12657
CTAL1-40-40-20	46	810.80	5921	$33^{(26)}$	**292.62**	8954
CTAL1-50-50-10	$183^{(57)}$	893.35	6572	$51^{(32)}$	**317.12**	15323
CTAL1-500-500-50	60	**10050.08**	237	60	10363.03	237
minimum congestion						
M32-32-12	27	3.11	211	27	**3.05**	211
M64-64-12	30	**5.45**	121	30	5.76	123
M128-64-12	$36^{(25)}$	218.79	49	$52^{(46)}$	**58.97**	5596
M128-128-12	$40^{(31)}$	362.89	90	39	**43.84**	251

* Maximum number of iterations reached without a solution.

$^{(\cdot)}$ Iteration at which the change of preconditioner is performed.

large number of interior-point iterations to converge. Since in the “hybrid” approach the switch to the splitting preconditioner is done earlier, better directions were obtained, reducing the number of interior-point iterations. The CPU time is also significantly smaller with the “hybrid” version.

For CTA problems, the “hybrid” version also reported, in general, better results. The “original” version reached the maximum number of iterations for instance CTAL1-30-30-30 without a solution with the required optimality tolerance. For this instance, when $gap^i < 10^{-5}$ system (10) became very ill-conditioned and PCG reported a large residual error. Bad directions were also obtained after switching to the Cholesky factorization. On the other hand, the “hybrid” approach converged to a solution in few iterations. Note that in some cases (e.g., CTAL1-40-40-20) although the “hybrid” approach needed more PCG iterations, it was much faster than the “original” one since it avoided a significant number of costly Cholesky factorizations for the block constraints.

For CTAL1-500-500-50 and the minimum congestion instances M32-32-12 and M64-64-12 the switching criteria was never satisfied, since the power series preconditioner behaved well for all the interior-point iterations. In some instances (e.g., CTAL1-10-10-5 and minimum congestion M128-128-12) the “original” approach switched to Cholesky, but the “hybrid” approach did no switch to the splitting preconditioner. This is because the tighter PCG termination criterion, which is used when there is no switch between preconditioners and *gap* gets worse, allowed the power series preconditioner to reach the optimal solution.

5. Conclusions

We have provided computational evidence that the hybrid approach significantly improved the performance of the specialized interior-point algorithm for some classes of primal block-angular problems. An estimate for the spectral radius of the matrix $D^{-1}(C^TB^{-1}C)$ was computed by using the Ritz values. This resulted in a criteria to switch between preconditioners that worked fine in the tested instances.

Improving the efficiency of the PCG by an adaptive selection of the number of terms h in the power series preconditioner is among the future tasks to be done. This could be done by using the Ritz values to measure the efficiency of the preconditioner at each interior-point iteration. The performance of the specialized algorithm would be improved if the optimal term

h^* could be identified at each iteration.

Acknowledgments

This work was developed when the first author was visiting the Department of Statistics and Operations Research of the Universitat Politècnica de Catalunya, funded by the CAPES/Fundação Carolina, Brazil. The second author has been supported by grants MTM2009-08747 of the Spanish Ministry of Science and Innovation, and SGR-2009-1122 of the Government of Catalonia. The authors thank F.F. Campos for suggesting the use of Ritz values.

References

[1] A. Ali and J.L. Kennington. Mnetgen program documentation. Technical Report 77003, Department of Industrial Engineering and Operations Research, Southern Methodist University, Dallas, TX, 1977.

[2] A. Altman and J. Gondzio. Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. *Optimization Methods & Software*, 11, 275–302, 1999.

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. LAPACK Users' Guide, Third Edition. SIAM, Philadelphia., 1999.

[4] P Beraldi, L. Grandinetti, R. Musmanno and C. Triki. Parallel algorithms to solve two-stage stochastic linear programs with robustness constraints. *Parallel Computing for Irregular Applications*, 26, 1889–1908, 2000.

[5] D. Bienstock and O. Raskina. Asymptotic analysis of the flow deviation method for the maximum concurrent flow problem. *Mathematical Programming*, 91, 479–492, 2002.

[6] J. R. Birge and L. Qi. Computing block angular Karmarkar projections with applications to stochastic programming. *Management Science*, 34, 1472–1479, 1988.

[7] S. Bocanegra, F. F. Campos and A. R. L. Oliveira. Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, 36, 149–164, 2007.

[8] W.J. Carolan, J.E. Hill, J.L. Kennington, S. Niemi, and S.J. Wichmann. An empirical evaluation of the KORBX algorithms for military airlift applications. *Operations Research*, 38, 240–248, 1990.

[9] J. Castro. A specialized interior-point algorithm for multicommodity network flows. *SIAM Journal on Optimization*, 10, 852–877, 2000.

[10] J. Castro. Quadratic interior-point methods in statistical disclosure control. *Computational Management Science*, 2, 107–121, 2005.

[11] J. Castro. Minimum-distance controlled perturbation methods for large-scale tabular data protection. *European Journal of Operational Research*, 171, 39-52, 2006.

[12] J. Castro. An interior-point approach for primal block-angular problems. *Computational Optimization and Applications*, 36, 195–219, 2007.

[13] J. Castro and J. Cuesta. Quadratic regularizations in an interior-point method for primal block-angular problems. *Mathematical Programming*, 130, 415–445, 2011.

[14] J. Castro and J. Cuesta. Solving $L1$-CTA in 3D tables by an interior-point method for primal block-angular problems. *TOP*, doi:10.1007/s11750-011-0247-z, 2012.

[15] A. Frangioni and C. Gentile. New preconditioners for KKT systems of network flow problems. *SIAM Journal on Optimization*, 14, 894–913, 2004.

[16] A. George and E. Ng. An implementation of Gaussian elimination with partial pivoting for sparse systems. *SIAM Journal on Scientific and Statistical Computing*, 6, 390–409, 1985.

[17] C. T.L.S. Ghidini, A.R.L. Oliveira, J. Silva and M.I. Velazco. Combining a hybrid preconditioner and a optimal adjustment algorithm to accelerate the convergence of interior point methods. *Linear Algebra and its Applications*, 436, 1267–1284, 2012.

[18] J. Gondzio and R. Sarkissian. Parallel Interior Point Solver for Structured Linear Programs. *Mathematical Programming*, 96, 561–584, 2003.

[19] G. Meurant. *The Lanczos and conjugate gradient algorithms: from theory to finite precision computations*. SIAM. 2006.

[20] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45, 225–280, 1950.

[21] E. Ng and B.W. Peyton. Block sparse Cholesky algorithms on advanced uniprocessor computers. *SIAM Journal on Scientific Computing*, 14, 1034–1056, 1993.

[22] A. R. L. Oliveira. *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1997.

[23] A. R. L. Oliveira and D. C. Sorensen. 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, 2005.

[24] B. N. Parlet. The Symmetric Eigenvalue Problem. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1998.

[25] M. G. C. Resende and G. Veiga. Implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks. *SIAM Journal on Optimization*, 3, 516–537, 2003.

[26] M. I. Velazco, A. R. L. Oliveira and F. F. Campos. A note on hybrid preconditioners for large-scale normal equations arising from interior-point methods. *Optimization Methods & Software*, 25:2, 321–332, 2010.

[27] F. Shahrokhi and D.W. Matula. The maximum concurrent flow problem. *Journal of the ACM*, 37, 318–334, 1990.

[28] S.J Wright. *Primal-Dual Interior-Point Methods*. SIAM. 1997.

[29] Y. Zhang. Solving large-scale linear programs by interior-point methods under the MATLAB environment. *Optimization Methods & Software*, 10, 1–31, 1998.