

Quadratic regularizations in an interior-point  
method for primal block-angular problems

Jordi Castro	Jordi Cuesta
Dept. of Stat. and Operations Research	Dept. of Chemical Engineering
Universitat Politècnica de Catalunya	Universitat Rovira i Virgili
jordi.castro@upc.edu	jordi.cuesta@urv.cat

Research Report UPC-DEIO DR 2008-07  
July 2008, updated June 2009

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



---

# Quadratic regularizations in an interior-point method for primal block-angular problems

Jordi Castro · Jordi Cuesta

the date of receipt and acceptance should be inserted later

**Abstract** One of the most efficient interior-point methods for some classes of primal block-angular problems solves the normal equations by a combination of Cholesky factorizations and preconditioned conjugate gradient for, respectively, the block and linking constraints. Its efficiency depends on the spectral radius—in  $[0, 1)$ —of a certain matrix in the definition of the preconditioner. Spectral radius close to 1 degrades the performance of the approach. The purpose of this work is twofold. First, to show that a separable quadratic regularization term in the objective reduces the spectral radius, significantly improving the overall performance in some classes of instances. Second, to consider a regularization term which decreases with the barrier function, thus with no need for an extra parameter. Computational experience with some primal block-angular problems confirms the efficiency of the regularized approach. In particular, for some difficult problems, the solution time is reduced by a factor of two to ten by the regularization term, outperforming state-of-the-art commercial solvers.

**Keywords** interior-point methods · primal block-angular problems · multicommodity network flows · preconditioned conjugate gradient · regularizations · large-scale computational optimization

**Mathematics Subject Classification (2000)** 90C06 · 90C08 · 90C51

## 1 Introduction

Many real-world problems exhibit a primal-block angular structure. Among them we find multicommodity network flow problems, extensively used in the linear program-

---

Jordi Castro\*

Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya, Jordi Girona 1–3, 08034 Barcelona, Catalonia, Spain. E-mail: [jordi.castro@upc.edu](mailto:jordi.castro@upc.edu), <http://www-eio.upc.es/~jcastro>

Jordi Cuesta

Statistics and Operations Research unit, Dept. of Chemical Engineering, Universitat Rovira i Virgili, Avda. Països Catalans 26, 43007 Tarragona, Catalonia, Spain. E-mail: [jordi.cuesta@urv.cat](mailto:jordi.cuesta@urv.cat)

**Table 1** CPU time ratio CPLEX/IPM for some linear and quadratic multicommodity instances (from [12])

Instance	$n$	$m$	LP ratio	QP ratio
PDS1	4167	1450	0.3	1.2
PDS10	53526	16192	1.5	2.9
M128-64-12	75804	8988	4.2	8.6
M128-128-12	155044	17188	9.7	7.5

**Table 2** Results for some quadratic multicommodity instances from a statistical data protection problem (from [13])

Instance	$n$	$m$	CPLEX		IPM		
			it.	CPU <sup>†</sup>	it.	PCG	CPU <sup>†</sup>
CTA-100-50-100	500000	20000	8	885	9	1.6	6
CTA-100-100-25	250000	15000	8	179	11	2.4	5
CTA-100-100-50	500000	20000	8	866	9	1.9	8
CTA-100-100-100	1000000	30000	*	*	9	1.6	16

\* Not enough memory

<sup>†</sup> CPU times on a laptop with a 1.8GHz Pentium Mobile and 512 MB RAM

ming literature. There have been many attempts to develop specialized implementations of linear or network optimization solvers for multicommodity problems [15, 27]. Over the years the approach of [10] based on an interior-point method has proved to be very efficient. This approach was recently applied to other primal block-angular problems [14]. It solved normal equations by a sensible combination of Cholesky factorizations for the block constraints and preconditioned conjugate gradient (PCG) iterations for the linking constraints (the procedure will be outlined in Section 2). This was recognized as the most efficient interior-point approach for some classes of multicommodity flows [8]. For some linear multicommodity flow problems it is known to outperform simplex implementations [11]. For separable convex quadratic multicommodity flows this approach is far more efficient than general interior-point solvers [12]. Although this specialized procedure makes use of the PCG, it is significantly different from other interior-point algorithms based on PCG which solve the full set of equations of either the augmented system [5] or normal equations [26], instead of only those associated to linking constraints. It is also remarkably different, and for some instances more efficient, than systems based on partitioned Cholesky factorizations [20]. The purpose of the specialized procedure is to eliminate the complicating linking constraints, making the problem block separable rather than solving the full (normal equations) system using an iterative solver.

This work was motivated by the much better behaviour of the specialized interior-point algorithm for separable quadratic than for linear instances. This is illustrated in Tables 1–2, and Figure 1. Table 1 reports the ratio between the barrier CPLEX algorithm and the specialized interior-point approach (named IPM) for some well-known multicommodity instances (i.e., some PDS [9] and Mnetgen [1] instances). Both the original linear instances and quadratic variants of them were considered in the study [12]. Columns  $n$  and  $m$  report the number of variables and constraints. From this ta-

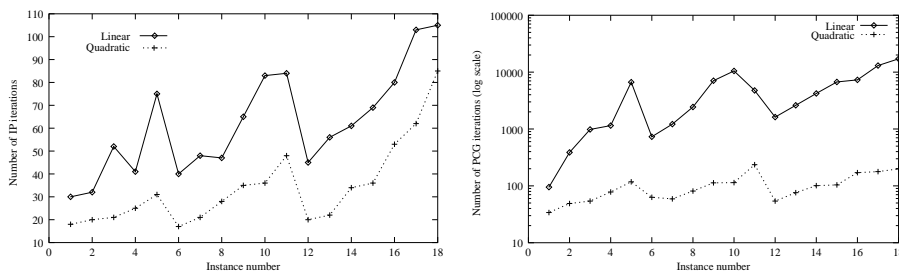


Fig. 1 Number of interior-point and PCG iterations for some Mnetgen problems

ble we conclude that in most cases (but for M128-128-12) the performance of IPM improves for quadratic problems. This is also confirmed by Table 2, which shows results in the solution of a statistical data protection problem in three-dimensional tables; this problem is modeled as quadratic multicommodity flows with equality linking constraints [13]. Columns  $n$  and  $m$  are as before, column “it.” provides the number of interior-point iterations, and column “PCG” provides the average number of PCG iterations required by the specialized approach at each interior-point iteration. Such good results (e.g., solving a one million variables and 30000 constraints quadratic multicommodity flows problem in 16 seconds —on a laptop with a 1.8GHz Pentium) have not been observed for linear problems. Although, in general, interior-point methods behave better for separable quadratic than for linear problems, the quadratic objective clearly improves the preconditioner too. This is observed in Figure 1 (from [12]), which plots for some linear and quadratic Mnetgen instances the number of interior-point iterations (left figure) and overall PCG iterations (right figure). The quadratic term decreases both, but the reduction is much more significant for PCG iterations: between one and two orders of magnitude.

The main purpose of this work is thus to reproduce the good behaviour of the specialized interior-point algorithm for quadratic problems in linear ones by adding a separable quadratic regularization term to the objective. Note that the focus is restricted to the separable convex quadratic case. It will be shown that the regularization term effectively decreases the spectral radius (in  $[0, 1)$ ) of a certain matrix in the definition of the preconditioner (up to now only empirical results were available). This is the most instrumental factor for the performance of the specialized approach. A general result for any primal block-angular problem will be presented. A second goal is to consider a regularization embedded in the barrier function, which decreases with the barrier parameter (unlike other approaches based, for instance, on proximal terms). This will allow us to show that, for bounded problems, the resulting barrier is self-concordant, preserving some of the good properties of interior-point algorithms. The computational results show the effectiveness of the approach. For some difficult problems in the literature (for which the specialized approach was already known to be more efficient than both simplex and interior-point variants of state-of-the art commercial solvers such as CPLEX), the regularized variant reduced the solution time by a factor of two to ten.

Regularization techniques for interior-point methods were already considered in the seminal book of Fiacco and McCormick [17]. They suggested to slack the constraints by a factor that depended on the barrier parameter, which vanished as optimality was approached. Other regularization approaches are based on proximal points. The proximal point algorithm, developed in a general context of minimizing a convex function, added a strong convex quadratic term to the objective function:

$$x_{k+1} = \min_x f(x) + \frac{\delta_k}{2} \|x - x_k\|^2.$$

In this form the proximal point algorithm has the same difficulty than the original problem. Rockafellar [28] showed that the requirement for exact minimization at each iteration of the proximal point algorithm can be relaxed. Setiono [30] used this proximal term in an interior-point approach for linear programming to get a better conditioned Newton system.

Several regularization approaches in interior-point algorithms considered the augmented system form of Newton equations. This indefinite system becomes quasidefinite and strongly factorizable with the regularization term [31]. Gondzio and Altman [2] considered the following regularization for the augmented system of an infeasible primal-dual path-following interior-point algorithm:

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} + \begin{bmatrix} -R_p & 0 \\ 0 & R_d \end{bmatrix},$$

where diagonal positive semidefinite matrices  $R_p$  and  $R_d$  were updated dynamically. They could be interpreted as adding proximal terms to the primal and dual objectives, respectively. This approach was in practice more flexible than the one of Saunders and Tomlin [29], which considered the regularized problem

$$\begin{aligned} \min \quad & c^T x + \frac{1}{2} \|\gamma x\|^2 + \frac{1}{2} \|p\|^2 \\ \text{s. to} \quad & Ax + \delta p = b, \quad x \geq 0 \end{aligned} \quad (1)$$

with fixed  $\delta$  and  $\gamma$  parameters, giving rise to the regularized augmented system matrix

$$\begin{bmatrix} -(D + \gamma^2 I_n) & A^T \\ A & \delta^2 I_m \end{bmatrix}.$$

The main differences of our approach with respect to the above ones are: (1) Although a proximal term could have been used, we considered a regularization term in the barrier function, which vanishes with the barrier parameter. (2) Our concern is not to improve the numerical stability of Cholesky factorizations, but the quality of the preconditioner of the combined Cholesky-PCG approach for primal block-angular problems. As it will be shown in the computational results, the number of PCG iterations is significantly reduced by the regularization term in most cases. (3) Our approach solves the normal equations instead of the augmented system. The recent interior-point approach of [4] for nonnegative least-squares problems also considers regularizations for improving the quality of a preconditioner, but, unlike ours, it solves the augmented system instead of normal equations.

Regularizations based on proximal terms have also been applied in the proximal analytic center cutting plane method (proximal-ACCPM) for multicommodity flows [3]. However, unlike our approach, the proximal term did not improve the performance of proximal-ACCPM, but just simplified the tuning of parameters with respect to ACCPM. In addition, our approach is not restricted to multicommodity flows, but it can deal with more general primal block-angular problems. Even for multicommodity flow problems, it can efficiently solve instances where (1) demands for commodities are not restricted to a single origin and a single destination (i.e., ACCPM subproblems are no longer shortest-path ones); and (2) the fraction of active linking constraints is large (e.g., as in instances of Table 2, where linking constraints were all equalities).

This paper is organized as follows. Section 2 outlines the specialized interior-point approach for primal block-angular problems. Section 3 analyzes the spectral properties of the preconditioner when a quadratic term is added to the objective function. Section 4 introduces and studies the main features of the quadratic regularization considered, providing computational results for general linear problems. Finally, Section 5 reports computational results for the solution of a set of primal block-angular problems, showing the benefits of the regularization term when combined with an iterative solver.

## 2 Outline of the interior-point algorithm for primal block-angular problems

The primal block-angular formulation dealt with by the algorithm is

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

Matrices  $N_i \in \mathbb{R}^{m_i \times n_i}$  and  $L_i \in \mathbb{R}^{l \times n_i}$ ,  $i = 1, \dots, k$ , respectively define the block-diagonal 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  $u^i$ ,  $i = 0, \dots, k$ . This formulation considers the general form of linking constraints  $b^0 - u^0 \leq \sum_{i=1}^k L_i x^i \leq b^0$ . If needed, equality constraints may be defined with this formulation by imposing (close to) zero upper bounds on the slacks. As it will be shown in Section 3, slacks improve the performance of the specialized

approach, as in the approach of [29], see (1).  $c^i \in \mathbb{R}^{n_i}$  and  $Q_i \in \mathbb{R}^{n_i \times n_i}$ ,  $i = 1, \dots, k$ , define 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  $Q_i$ ,  $i = 0, \dots, k$ , are diagonal positive semidefinite matrices.

Problem (2) can be written as

$$\begin{aligned} \min \quad & c^T x + \frac{1}{2} x^T Q x \\ \text{Ax} = & b \\ u \geq & x \geq 0 \end{aligned} \quad (3)$$

where  $c, x, u \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $Q \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^m$ . Replacing inequalities in (3) by a logarithmic barrier with parameter  $\mu > 0$ , the KKT conditions for the barrier problem become [32]:

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

$e \in \mathbb{R}^n$  is a vector of 1's, matrices  $X, Z, U, W \in \mathbb{R}^{n \times n}$  are diagonal matrices made up of vectors  $x, z, u, w$ , and  $S$  (to be used later) is defined as  $S = U - X$ . The first two sets of equations of (4) impose, respectively, primal and dual feasibility; the last two impose complementarity.

The normal equations for the Newton direction  $(\Delta x, \Delta y, \Delta z)$  of (4) reduce to (see [14] for details):

$$(A\Theta A^T)\Delta y = g \quad (5)$$

$$\Theta = (Q + (U - X)^{-1}W + X^{-1}Z)^{-1}, \quad (6)$$

for some right-hand-side  $g$ . For linear (i.e.,  $Q = 0$ ) or separable quadratic problems  $\Theta$  is a positive diagonal matrix and can be easily computed. Exploiting the structure of  $A$  and  $\Theta$  in (2) the matrix of (5) can be recast as

$$\begin{aligned} A\Theta A^T &= \left[ \begin{array}{ccc|ccc} N_1\Theta_1N_1^T & & & N_1\Theta_1L_1^T & & \\ & \ddots & & \vdots & & \\ & & N_k\Theta_kN_k^T & N_k\Theta_kL_k^T & & \\ \hline L_1\Theta_1N_1^T & \dots & L_k\Theta_kN_k^T & \Theta_0 + \sum_{i=1}^k L_i\Theta_iL_i^T & & \end{array} \right] \\ &= \begin{bmatrix} B & C \\ C^T & D \end{bmatrix}, \end{aligned} \quad (7)$$

$B \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$  ( $\tilde{m} = \sum_{i=1}^k m_i$ ),  $C \in \mathbb{R}^{\tilde{m} \times l}$  and  $D \in \mathbb{R}^{l \times l}$  being the blocks of  $A\Theta A^T$ , and  $\Theta_i$ ,  $i = 0, \dots, k$ , the submatrices of  $\Theta$  associated with the  $k + 1$  groups of variables in (2), i.e.,  $\Theta_i = (Q_i + S_i^{-1}W_i + X_i^{-1}Z_i)^{-1}$ .



Appropriately partitioning  $g$  and  $\Delta y$  in (5), the normal equations can be written as

$$\begin{bmatrix} B & C \\ C^T & D \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}. \quad (8)$$

By eliminating  $\Delta y_1$  from the first group of equations of (8), we obtain

$$(D - C^T B^{-1} C) \Delta y_2 = (g_2 - C^T B^{-1} g_1) \quad (9)$$

$$B \Delta y_1 = (g_1 - C \Delta y_2). \quad (10)$$

System (10) is solved a Cholesky factorization for each diagonal block  $N_i \Theta_i N_i^T$ ,  $i = 1 \dots k$ , of  $B$ . The system with matrix  $D - C^T B^{-1} C$ , the Schur complement of (8), is solved by a PCG. The dimension of this system is  $l$ , which is the number of linking constraints. In [10] it was proved that, under some conditions, which are guaranteed in our setting, the inverse of  $(D - C^T B^{-1} C)$  can be computed as

$$(D - C^T B^{-1} C)^{-1} = \left( \sum_{i=0}^{\infty} (D^{-1} (C^T B^{-1} C))^i \right) D^{-1}. \quad (11)$$

The preconditioner  $M^{-1}$ , an approximation of  $(D - C^T B^{-1} C)^{-1}$ , is thus obtained by truncating the infinite power series (11) at some term  $h$ . The more the terms included, the better the preconditioner will be, at the expense of increasing the execution time of each PCG iteration. However, in general,  $h = 0$  or  $h = 1$  are reasonable choices, which in practice yield

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

This preconditioner, initially developed for multicommodity flows [10] can be applied to any primal block-angular problem [14].

The effectiveness of the preconditioner depends on the spectral radius of matrix  $D^{-1} (C^T B^{-1} C)$ , which is always in  $[0, 1)$  [10, Theorem 1]. The farther away from 1 is the spectral radius of  $D^{-1} (C^T B^{-1} C)$  the better is the quality of the approximation of (11) obtained by truncation with  $h = 0$  or  $h = 1$ . Although the particular behaviour of the spectral radius value is problem dependent, in general, it comes closer to 1 as we approach the optimal solution, because of the ill-conditioning of the  $\Theta$  matrix. As stated in Section 1, in practice it was observed that when a quadratic term is present the spectral radius tends to be smaller than that obtained in the simplified linear formulation obtained by removing this quadratic objective term, and the preconditioner become more efficient. This behaviour is explained in next section.

### 3 Quadratic terms improve the preconditioner ...

The general case (2), i.e.,  $N_i \in \mathbb{R}^{m_i \times n_i}$  and  $L_i \in \mathbb{R}^{l \times n_i}$ ,  $i = 1, \dots, k$ , is first considered. Next result provides a bound for the spectral radius of  $D^{-1} (C^T B^{-1} C)$  for any primal block-angular problem, either with linear or separable quadratic objective function. This result will be specialized in Subsection 3.1 for a particularly important class of problems.

**Theorem 1** Let  $A$  be the constraint matrix of problem (2), with full row rank matrices  $N_i \in \mathbb{R}^{m_i \times n_i}$   $i = 1, \dots, k$ , and at least one full row rank matrix  $L_i \in \mathbb{R}^{l \times n_i}$ ,  $i = 1, \dots, k$ . Let  $\Theta$  be the symmetric diagonal matrix defined in (6), and  $B \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$  ( $\tilde{m} = \sum_{i=1}^k m_i$ ),  $C \in \mathbb{R}^{\tilde{m} \times l}$  and  $D \in \mathbb{R}^{l \times l}$  the submatrices of  $A\Theta A^T$  defined in (7). Then, the spectral radius  $\rho$  of  $D^{-1}(C^T B^{-1}C)$  is bounded by

$$0 \leq \rho \leq \max_{j \in \{1, \dots, l\}} \frac{\gamma_j}{\left(\frac{u_j}{v_j}\right)^2 \Theta_{0j} + \gamma_j} < 1, \quad (12)$$

where  $u$  is the eigenvector (or one of the eigenvectors) of  $D^{-1}(C^T B^{-1}C)$  for  $\rho$ ;  $\gamma_j, j = 1, \dots, l$ , and  $V = [V_1 \dots V_l]$ , are respectively the eigenvalues and matrix of columnwise eigenvectors of  $\sum_{i=1}^k L_i \Theta_i L_i^T$ ;  $v = V^T u$ ; and, abusing of notation, we assume that for  $v_j = 0$ ,  $(u_j/v_j)^2 = +\infty$ .

*Proof* Eigenvalue  $\lambda$  of  $D^{-1}(C^T B^{-1}C)$  satisfies  $(C^T B^{-1}C)u = \lambda Du$  for some eigenvector  $u$ . From the definition of  $B, C, D$  in (7) we have

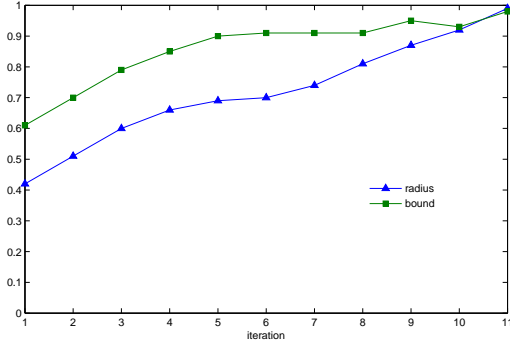
$$\begin{aligned} \left( \sum_{i=1}^k L_i \Theta_i N_i^T (N_i \Theta_i N_i^T)^{-1} N_i \Theta_i L_i^T \right) u &= \lambda \left( \Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T \right) u, \\ &\Downarrow \\ (1 - \lambda) \left( \Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T \right) u &= \left( \Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T \right) u - \left( \sum_{i=1}^k L_i \Theta_i N_i^T (N_i \Theta_i N_i^T)^{-1} N_i \Theta_i L_i^T \right) u, \\ &\Downarrow \\ (1 - \lambda) u^T \left( \Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T \right) u &= u^T \Theta_0 u + \sum_{i=1}^k u^T L_i \Theta_i^{1/2} \left( I - \Theta_i^{1/2} N_i^T (N_i \Theta_i N_i^T)^{-1} N_i \Theta_i^{1/2} \right) \Theta_i^{1/2} L_i^T u. \end{aligned} \quad (13)$$

$P_i = I - \Theta_i^{1/2} N_i^T (N_i \Theta_i N_i^T)^{-1} N_i \Theta_i^{1/2}$ ,  $i = 1, \dots, k$ , is a projection operator onto the null space of  $N_i \Theta_i^{1/2}$ , i.e.,  $P_i = P_i^2$  and  $w^T P_i w \geq 0$  for any vector  $w$ . Defining  $w^i = \Theta_i^{1/2} L_i^T u$  in (13), we have

$$1 - \lambda = \frac{u^T \Theta_0 u + \sum_{i=1}^k w^{iT} P_i w^i}{u^T \left( \Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T \right) u} \geq \frac{u^T \Theta_0 u}{u^T \Theta_0 u + u^T \left( \sum_{i=1}^k L_i \Theta_i L_i^T \right) u}$$

and therefore

$$\lambda \leq \frac{u^T \left( \sum_{i=1}^k L_i \Theta_i L_i^T \right) u}{u^T \Theta_0 u + u^T \left( \sum_{i=1}^k L_i \Theta_i L_i^T \right) u}. \quad (14)$$



**Fig. 2** Spectral radius and bound (12) for first iterations of problem PDS1

Since  $\sum_{i=1}^k L_i \Theta_i L_i^T \in \mathbb{R}^{l \times l}$  is symmetric and positive definite (because at least one  $L_i$  was assumed to be full row rank), their eigenvalues  $\gamma_j$ , with corresponding eigenvectors  $V_j$ ,  $j = 1, \dots, l$ , are real and positive. Defining  $\Pi = \text{diag}(\gamma_1, \dots, \gamma_l)$ ,  $V = [V_1 \dots V_l]$ ,  $v = V^T u$ ,  $\mathcal{L} = \{j \in \{1, \dots, l\} : v_j \neq 0\}$ ,  $\mathcal{L}^0 = \{1, \dots, l\} \setminus \mathcal{L}$ , and using  $\sum_{i=1}^k L_i \Theta_i L_i^T = V \Pi V^T$ , (14) can be recast as

$$\lambda \leq \frac{v^T \Pi v}{u^T \Theta_0 u + v^T \Pi v} = \frac{\sum_{j \in \mathcal{L}} \gamma_j v_j^2}{\sum_{j \in \mathcal{L}} (u_j^2 \Theta_{0j} + \gamma_j v_j^2) + \sum_{j \in \mathcal{L}^0} u_j^2 \Theta_{0j}} \leq \frac{\sum_{j \in \mathcal{L}} \gamma_j v_j^2}{\sum_{j \in \mathcal{L}} \left( \left( \frac{u_j}{v_j} \right)^2 \Theta_{0j} + \gamma_j \right) v_j^2}. \quad (15)$$

Denoting  $\alpha_j^2 = \left( \left( \frac{u_j}{v_j} \right)^2 \Theta_{0j} + \gamma_j \right) v_j^2$ , for  $j \in \mathcal{L}$ , in (15), we obtain

$$\lambda \leq \frac{\sum_{j \in \mathcal{L}} \frac{\gamma_j}{\left( \frac{u_j}{v_j} \right)^2 \Theta_{0j} + \gamma_j} \alpha_j^2}{\sum_{j \in \mathcal{L}} \alpha_j^2} \leq \frac{\sum_{j \in \mathcal{L}} \left( \max_{j \in \mathcal{L}} \frac{\gamma_j}{\left( \frac{u_j}{v_j} \right)^2 \Theta_{0j} + \gamma_j} \right) \alpha_j^2}{\sum_{j \in \mathcal{L}} \alpha_j^2} = \max_{j \in \mathcal{L}} \frac{\gamma_j}{\left( \frac{u_j}{v_j} \right)^2 \Theta_{0j} + \gamma_j}. \quad (16)$$

If, abusing of notation, we assume for  $v_j = 0$ ,  $(u_j/v_j)^2 = +\infty$  and thus  $\gamma_j / \left( \left( \frac{u_j}{v_j} \right)^2 \Theta_{0j} + \gamma_j \right) = 0$ , since  $\lambda \geq 0$  this particular  $j$  can not provide the maximum and then (16) is equivalent to (12).  $\square$

For instance, Figure 2 shows the evolution of the spectral radius of  $D^{-1}(C^T B^{-1} C)$  and the bound (12) for the first 11 interior-point iterations of a regularized version (see Section 4) of problem PDS1 of Table 1. After this iteration, both the spectral radius and the bound approach 1, significantly increasing the number of PCG iterations per interior-point iteration.

**Table 3** Results for a large (10 million variables, 210000 constraints) statistical data protection problem instance, with quadratic objective function  $\frac{1}{2}x^T Qx$ , for different  $Q = \beta I$

Instance	$\beta$	CPLEX-11		IPM			$f^*$
		it.	CPU	it.	PCG	CPU	
CTA-100-100-1000	0.01	7	29939	10	36	66	-2.6715e+08
CTA-100-100-1000	0.1	7	31328	9	40	61	-2.6715e+09
CTA-100-100-1000	1	8	33367	8	38	56	-2.6715e+10
CTA-100-100-1000	10	9	35220	7	37	51	-2.6715e+11

Using Theorem 1 we next show that the spectral radius tends to 0 when  $Q_i$ ,  $i = 1, \dots, k$ , (i.e., the quadratic costs of variables, excluding slacks) tends to infinity.

**Proposition 1** *Let assume the hypotheses of Theorem 1. Then*

$$\lim_{\substack{Q_i \rightarrow +\infty \\ i=1, \dots, k}} \rho = 0. \quad (17)$$

*Proof* At any interior-point iteration, there exists a constant  $\varepsilon$ , where  $0 < \varepsilon < 1$ , such that  $\varepsilon < s_i < 1/\varepsilon$ ,  $\varepsilon < w_i < 1/\varepsilon$ ,  $\varepsilon < x_i < 1/\varepsilon$  and  $\varepsilon < z_i < 1/\varepsilon$  [21]. Therefore, from (6), for some  $S, W, X$  and  $Z$ ,  $\lim_{Q_i \rightarrow +\infty} \Theta_i = 0$ . Then,  $\lim_{Q_i \rightarrow +\infty} \sum_{i=1, \dots, k}^k L_i \Theta_i L_i^T = 0$  and its eigenvalues satisfy  $\lim_{i=1, \dots, k} \gamma_j = 0$ ,  $j = 1, \dots, l$ . From (15), where  $\lambda$  is an eigenvalue of  $D^{-1}(C^T B^{-1} C)$ ,  $\Pi = \text{diag}(\gamma_1, \dots, \gamma_l)$ , and considering unit vectors  $u$  and  $V_j$ ,  $j = 1, \dots, l$ , (and thus  $v = V^T u$  is also a unit vector), we have

$$\lim_{\substack{Q_i \rightarrow +\infty \\ i=1, \dots, k}} \lambda \leq \lim_{\substack{Q_i \rightarrow +\infty \\ i=1, \dots, k}} \frac{v^T \Pi v}{u^T \Theta_0 u + v^T \Pi v} = 0.$$

Since  $\lambda \in [0, 1)$ , (17) holds.  $\square$

Proposition 1 means that adding large enough  $Q_i$ ,  $i = 1, \dots, k$ , to a linear problem, it is possible to reduce (actually to approach 0) the spectral radius of matrix  $D^{-1}(C^T B^{-1} C)$ , and thus to improve the quality of the preconditioner. It also explains the good behaviour of the specialized interior-point method in instances of Table 2: since that is a quadratic problem, without linear term,  $\arg \min \frac{1}{2}x^T Qx = \arg \min \frac{1}{2}x^T (\beta Q)x$  for any positive  $\beta \in \mathbb{R}$ ; therefore, the spectral radius is effectively reduced, and PCG is able to solve (9) in very few iterations, independently of the scaling factor  $\beta$ . This is shown in Table 3, which reports results for instance CTA-100-100-1000, using four different scaling factors  $\beta$  and  $Q = I$  (see [13, 14] for a description of the underlying statistical three-dimensional tabular data protection problem). The resulting primal block-angular problem has 10,000,000 variables, and 210,000 constraints. This instance has not been attempted before in the literature. Column “it.” provides the number of interior-point iterations, and column “PCG” provides the overall number of PCG iterations required by the specialized approach. Column “CPU” gives the CPU time in seconds, on a Dell PowerEdge 6950 server with four dual core AMD Opteron 8222 3.0 GHZ processors (without exploitation of parallelism capabilities) and 64 GB of RAM. Column “ $f^*$ ” provides the optimal

objectives found, which are consistent with the scaling factor. The tolerance for the PCG solution (i.e., solution of (9)) was set to  $10^{-8}$  for all the interior-point iterations. From Table 3 we see that the number of PCG iterations is independent of the  $\beta$  factor. The specialized interior-point approach was not only much more efficient than CPLEX-11 in terms of CPU time, but also in memory requirements: it needed 1.2 GB of RAM, while CPLEX-11 required 15 GB. Both codes successfully solved the problem, with relative differences in the objective function of about  $10^{-11}$ .

For linear problems, however, the addition of quadratic terms with large  $Q_i$ ,  $i = 1, \dots, k$ , is meaningless, and only small regularizations are used in practice [2, 29]. Proposition 2 below shows that, under some conditions, the bound (12) on the spectral radius for a linear problem is reduced by adding (even small) quadratic costs  $Q_i$ ,  $i = 1, \dots, k$ . Since both the bound and the spectral radius approach 1 in the last iterations of the interior-point method, a reduction in the bound also means a reduction in the spectral radius. We first prove the auxiliary Lemma 1. It states that the eigenvalues of  $\sum_{i=1}^k L_i \Theta_i L_i^T$  are reduced if we add a quadratic term with  $Q_i$  diagonal and positive definite to a linear cost function.

**Lemma 1** *Let  $\hat{\gamma}_j$  and  $\gamma_j$ ,  $j = 1, \dots, l$ , be the  $j$ -th largest eigenvalues of  $\sum_{i=1}^k L_i \hat{\Theta}_i L_i^T$  and  $\sum_{i=1}^k L_i \Theta_i L_i^T$  for, respectively, a linear and a quadratic problem (i.e.,  $Q = 0$  in  $\hat{\Theta}$ , and  $Q_i \succ 0$  and diagonal,  $i = 1, \dots, k$ , in  $\Theta$ ,  $\Theta$  defined as in (6)). Then,  $\hat{\gamma}_j > \gamma_j$ .*

*Proof* Since  $Q_i \succ 0$  is diagonal we have by (6) and (7) that  $\hat{\Theta}_i = \Theta_i + E_i$  for all  $i = 1, \dots, k$ , where  $E_i$  is diagonal and

$$E_i = \Theta_i \hat{\Theta}_i Q_i = (Q_i + S_i^{-1} W_i + X_i^{-1} Z_i)^{-1} (S_i^{-1} W_i + X_i^{-1} Z_i)^{-1} Q_i \succ 0.$$

Indeed, note that

$$E_i = \Theta_i \hat{\Theta}_i Q_i = \Theta_i \hat{\Theta}_i (Q_i + \hat{\Theta}_i^{-1} - \hat{\Theta}_i^{-1}) = \Theta_i \hat{\Theta}_i (\Theta^{-1} - \hat{\Theta}_i^{-1}) = \hat{\Theta}_i - \Theta_i.$$

Therefore

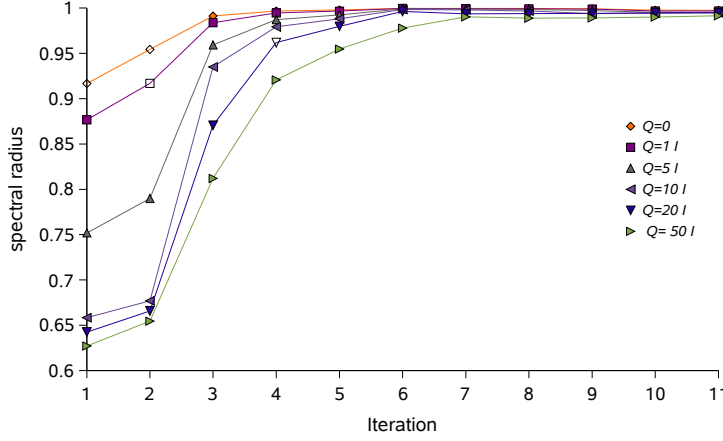
$$\sum_{i=1}^k L_i \hat{\Theta}_i L_i^T = \sum_{i=1}^k L_i \Theta_i L_i^T + \sum_{i=1}^k L_i E_i L_i^T. \quad (18)$$

The Courant-Fischer minimax theorem (see, for instance, [19, Theorem 8.1.2]) states that for a symmetric matrix  $M$  its  $j$ -th largest eigenvalue  $\lambda_j$  is equal to

$$\lambda_j = \max_{\dim(S)=j} \min_{0 \neq y \in S} \frac{y^T M y}{y^T y}. \quad (19)$$

Since the three matrices in (18) are positive definite, we have that, for all  $y$ ,  $\frac{y^T (\sum_{i=1}^k L_i \hat{\Theta}_i L_i^T) y}{y^T y} > \frac{y^T (\sum_{i=1}^k L_i \Theta_i L_i^T) y}{y^T y}$ . Therefore, by (19),  $\hat{\gamma}_j > \gamma_j$ .  $\square$

The strong assumption  $\hat{u}_j / \hat{v}_j \leq u_j / v_j$  of next proposition depends on the sensitivity of  $u$  (the eigenvector of  $D^{-1}(C^T B^{-1} C)$  for  $\rho$ ) and  $V_j$ ,  $j = 1, \dots, l$ , (eigenvectors of  $\sum_{i=1}^k L_i \Theta_i L_i^T$ ) against small changes in  $\Theta$  induced by  $Q_i \succ 0$ ,  $i = 1, \dots, k$ . We are aware that in the general case such an assumption cannot be verified in practice, but as we will show later in Subsection 3.1, for some important class of problems such an inequality does hold.



**Fig. 3** Evolution of spectral radius for first iterations of a quadratic version of problem PDS1 using different  $Q$  matrices

**Proposition 2** *Let assume the hypotheses of Theorem 1, and consider a linear problem and a quadratic one obtained by adding (likely small) quadratic costs  $Q_i > 0$ ,  $i = 1, \dots, k$ . Assume  $\hat{u}_j/\hat{v}_j \leq u_j/v_j$ ,  $j = 1, \dots, l$ , where “hatted” and “non-hatted” terms refer, respectively, to the linear and quadratic problems, and  $u$  and  $v$  are defined as in Theorem 1. Then bound (12) is smaller for the quadratic than for the linear problem.*

*Proof* By (12), the spectral radius  $\rho$  of  $D^{-1}(C^T B^{-1} C)$  after addition of quadratic costs  $Q_i > 0$  is bounded by

$$\rho \leq \max_{j \in \{1, \dots, l\}} \frac{\gamma_j}{\left(\frac{u_j}{v_j}\right)^2 \Theta_{0j} + \gamma_j} < 1,$$

$\gamma_j$  being the  $j$ -th eigenvalue of  $\sum_{i=1}^k L_i \Theta_i L_i^T$ . Denoting by  $\hat{\gamma}_j$  the  $j$ -th eigenvalue of  $\sum_{i=1}^k L_i \hat{\Theta}_i L_i^T$  corresponding to the linear problem, we have by Lemma 1  $\gamma_j = \hat{\gamma}_j - \varepsilon_j$ , for some  $\varepsilon_j > 0$ . Since  $\hat{u}_j/\hat{v}_j \leq u_j/v_j$ , then

$$\frac{\hat{\gamma}_j}{\left(\frac{\hat{u}_j}{\hat{v}_j}\right)^2 \Theta_{0j} + \hat{\gamma}_j} \geq \frac{\hat{\gamma}_j}{\left(\frac{u_j}{v_j}\right)^2 \Theta_{0j} + \hat{\gamma}_j} > \frac{\hat{\gamma}_j - \varepsilon_j}{\left(\frac{u_j}{v_j}\right)^2 \Theta_{0j} + \hat{\gamma}_j - \varepsilon_j}$$

for all  $j = 1, \dots, l$ , where last inequality holds because  $\left(\frac{u_j}{v_j}\right)^2 \Theta_{0j} \varepsilon_j > 0$ .  $\square$

It is noteworthy that Proposition 2 states that (i) the bound on the spectral radius is reduced, but not the spectral radius; (ii) and this is achieved by adding quadratic costs  $Q_i$ ,  $i = 1, \dots, k$ , to block variables, but not slacks costs  $Q_0$ . When the spectral radius is close to 1, a reduction in the bound means a reduction of the spectral radius (i.e., the preconditioner is improved). But when the spectral radius is far from 1 it may not be reduced for all diagonal  $Q \geq 0$ , and any  $A, x, z, s, w$ . For

instance, for a problem with no upper bounds (i.e., with no  $s$  and  $w$  variables),  $k = 1, L_1 = I_5, N_1 = \begin{bmatrix} 1 & 2 & 3 & 1 & 5 \\ 2 & 4 & 8 & 8 & 32 \end{bmatrix}, x_0 = [4.65 \ 0.704 \ 0.7206 \ 3.5957 \ 3.8952]^T, x_1 = [1.8289 \ 7.8537 \ 0.2577 \ 4.074 \ 9.0643]^T, z_1 = z_0 = [1 \ 1 \ 1 \ 1 \ 1]^T, Q = \text{diag}([1.108 \ 0.0111 \ 1.747 \ 0.0122 \ 0.0182])$  the spectral radius of  $D^{-1}(C^T B^{-1} C)$  is 0.8484 when  $Q_1 = Q_0 = 0$ , whereas for  $Q_1 = Q_0 = Q$  we obtain 0.8763, and for  $Q_1 = Q$  and  $Q_0 = 0$  we get 0.8632. However, the difficult and interesting case is when the spectral radius approaches 1, where just a small reduction can mean the efficient solution by PCG. In practice, even when we are far from 1, a quadratic term may reduce the spectral radius, as observed in Figure 3. It plots the evolution of the spectral radius for the first 11 iterations of a quadratic version of problem PDS1 of Table 1 using different  $Q = \delta I$  matrices,  $\delta \in \mathbb{R}$  and nonnegative.  $Q = 0$  corresponds to the standard linear PDS1 problem. It is shown that the spectral radius is reduced as  $\delta$  increases.

### 3.1 The case of weighted GUB constraints

Problems with weighted generalized upper bounds (GUB) are a particular case of the primal block-angular problem (2) for  $L_i = G_i, G_i \in \mathbb{R}^{n \times n}, i = 1, \dots, k$ , being a diagonal positive semidefinite matrix (note that  $n_i = n = l, i = 1, \dots, k$  because  $G_i$  are square matrices with the same number of rows). When  $G_i = I$  we have the standard non-weighted GUB constraints. If in addition  $N_i$ , for all  $i = 1, \dots, k$ , is the node-arc incidence matrix of a graph, we obtain a multicommodity flow problem. However, we consider in this Subsection that  $N_i$  is any  $\mathbb{R}^{m_i \times n}$  matrix.

If  $L_i = G_i$ , then  $\sum_{i=1}^k L_i \Theta_i L_i^T$  is diagonal, and its eigenvalues are  $\gamma_j = \sum_{i=1}^k \Theta_{ij} G_{ij}^2$ , with eigenvectors  $V_j = e_j, j = 1, \dots, n$ , i.e.,  $V = I$ . Therefore,  $v = V^T u = u$ , the ratio  $u_j/v_j$  in Theorem 1 is 1 for all  $j = 1, \dots, n$ , and the bound (12) has the simple and computable form:

$$\rho \leq \max_{j \in \{1, \dots, n\}} \frac{\sum_{i=1}^k \Theta_{ij} G_{ij}^2}{\Theta_{0j} + \sum_{i=1}^k \Theta_{ij} G_{ij}^2} < 1. \quad (20)$$

In addition, the strong assumption  $\hat{u}_j/\hat{v}_j \leq u_j/v_j$  of Proposition 2 is satisfied, since  $u_j/v_j = \hat{u}_j/\hat{v}_j = 1$ . Therefore bound (20) is effectively reduced by adding even a small quadratic term  $Q_i \succ 0, i = 1, \dots, k$ , to a linear problem. Note that if GUB constraints have to be imposed only for a subset  $\mathcal{J} \subseteq \{1, \dots, n\}$  of the variables, we just have to define  $G_{ij} = 0$  for all  $j \in \mathcal{J}$ , and (20) remains valid.

It is noteworthy that if, in addition to  $L_i = G_i$ , we have  $N_i, i = 1, \dots, k$ , are square nonsingular matrices, then (20) is not an upper bound, but it is actually the true spectral radius. This is because, under the above assumptions,

$$\begin{aligned} (D^{-1}(C^T B^{-1} C)) &= (\Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T)^{-1} \left( \sum_{i=1}^k L_i \Theta_i N_i^T (N_i \Theta_i N_i^T)^{-1} N_i \Theta_i L_i^T \right) \\ &= (\Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^T)^{-1} \left( \sum_{i=1}^k L_i \Theta_i L_i^T \right), \end{aligned}$$

which is equal to a diagonal matrix whose  $j$ th component is  $\left(\Theta_{0j} + \sum_{i=1}^k \Theta_{ij} G_{ij}^2\right)^{-1} \left(\sum_{i=1}^k \Theta_{ij} G_{ij}^2\right)$ . Although this result is not of practical interest (constraint matrix is square, with a unique feasible point), it shows how tight is the bound in a limit situation.

#### 4 ... then we add a quadratic regularization

Given the linear programming problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s. to} \quad & Ax = b \\ & 0 \leq x \leq u, \end{aligned} \quad (21)$$

the associated primal logarithmic barrier problem is

$$\begin{aligned} \min \quad & B(x, \mu) \triangleq c^T x + \mu \left( -\sum_{i=1}^n \ln x_i - \sum_{i=1}^n \ln(u_i - x_i) \right) \\ \text{s. to} \quad & Ax = b, \end{aligned} \quad (22)$$

$\mu$  being the barrier parameter,  $x, c, u \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$ , and  $A \in \mathbb{R}^{m \times n}$ . In the interior point context, regularizations based on proximal points have already been used in other approaches. They considered the alternative regularized function:

$$B_P(x, \mu) \triangleq c^T x + \frac{1}{2}(x - \bar{x})^T Q_P(x - \bar{x}) + \mu \left( -\sum_{i=1}^n \ln x_i - \sum_{i=1}^n \ln(u_i - x_i) \right), \quad (23)$$

$Q_P$  being the identity matrix in [30], and a diagonal positive definite matrix with small entries—dynamically updated at each interior-point iteration—in [2]; and  $\bar{x}$  the current point obtained by the interior-point algorithm. Therefore the definition of the barrier function changes at each iteration according to the current point, and it does not fit the general theory of structural optimization for interior-point methods of [22, 23]. However, it is worth noting that, in practice, the proximal point regularization of [2] has an excellent behaviour.

Instead, we suggest the alternative regularization

$$B_Q(x, \mu) \triangleq c^T x + \mu F_Q(x) \quad (24)$$

where

$$F_Q(x) \triangleq \frac{1}{2} x^T Q x - \sum_{i=1}^n \ln x_i - \sum_{i=1}^n \ln(u_i - x_i), \quad (25)$$

$Q$  being a diagonal positive semidefinite matrix, and  $\text{dom } F_Q = \{x : 0 < x < u\}$ . Some properties of  $B_Q(x, \mu)$  are:

1. Barrier function  $F_Q(x)$  is strictly convex in  $\{x : 0 < x < u\}$  ( $\nabla F_Q(x) = Qx - X^{-1}e + (U - X)^{-1}e$ ,  $\nabla^2 F_Q(x) = Q + X^{-2} + (U - X)^{-2} \succ 0$ ).
2.  $B_Q(x, \mu)$  does not depend on the current point, unlike  $B_P$  in (23).
3. The reduction to zero of the regularization  $Q$  matrix is governed by  $\mu$ , the standard barrier parameter of  $B(x, \mu)$  in (22).



4. This regularized barrier function fits the classical interior-point approach of Fiacco and McCormick [17] (see Subsection 4.1 below).
5.  $F_Q(x)$  is a self-concordant barrier in  $\{x : 0 < x < u\}$ , and then it fits the structural optimization framework of [22, 23]. As it will be shown in Subsection 4.1, for small enough regularizations, it is even possible to follow the central path at high speed.

Note that, using either  $B$ ,  $B_P$ , or  $B_Q$  only changes the dual feasibility of KKT conditions (4), and matrix  $\Theta$  defined in (6). Dual feasibility becomes

$$A^T y + z - w = c \quad \text{for } B, \quad (26)$$

$$A^T y + z - w = c + Q_P(x - \bar{x}) \quad \text{for } B_P, \text{ and} \quad (27)$$

$$A^T y + z - w = c + \mu Q x \quad \text{for } B_Q. \quad (28)$$

Evaluating at current point ( $x = \bar{x}$ ), (27) is equivalent to (26). (28) is also equivalent to (26) when  $\mu$  tends to zero. The  $\Theta$  matrices are

$$\Theta = ((U - X)^{-1}W + X^{-1}Z)^{-1} \quad \text{for } B, \quad (29)$$

$$\Theta = (Q_P + (U - X)^{-1}W + X^{-1}Z)^{-1} \quad \text{for } B_P, \text{ and} \quad (30)$$

$$\Theta = (\mu Q + (U - X)^{-1}W + X^{-1}Z)^{-1} \quad \text{for } B_Q. \quad (31)$$

The main difference between (30) and (31) is that  $\mu Q$  tends to zero with  $\mu$  and therefore (31) approximates (29) better than (30). The remaining linear algebra of interior-point methods is the same for the three variants.

#### 4.1 Self-concordancy of $F_Q(x)$

We first show that regularized barrier  $F_Q(x)$  (defined in (25)) fits the classical interior-point (or sequential unconstrained minimization) approach of Fiacco and McCormick. Rewriting (25) as

$$F_Q(x) = \sum_{i=1}^n B_{0_i}(x) + B_{u_i}(x), \quad B_{0_i}(x) = \frac{1}{2}q_i x_i^2 - \ln x_i, \quad B_{u_i}(x) = -\ln(u_i - x_i),$$

$B_{0_i}(x)$  and  $B_{u_i}(x)$  being the barrier for, respectively,  $g_{0_i}(x) = x_i \geq 0$  and  $g_{u_i}(x) = u_i - x_i \geq 0$ , and  $q_i \geq 0$  being the diagonal terms of  $Q$ , we have that: (1) both  $B_{0_i}(x)$  and  $B_{u_i}(x)$  are continuous in, respectively,  $\{x : g_{0_i}(x) > 0\}$  and  $\{x : g_{u_i}(x) > 0\}$ ; and (2),  $B_{0_i}(x) \rightarrow +\infty$  and  $B_{u_i}(x) \rightarrow +\infty$  when, respectively,  $g_{0_i}(x) \rightarrow 0$  and  $g_{u_i}(x) \rightarrow 0$ . Then:

- The sequence of minima of  $B_Q(x, \mu) = c^T x + \mu F_Q(x)$  converges to the compact set of minima of (21) [17, Theorem 8].
- Let assume barriers  $B_{0_i}(g_{0_i})$  and  $B_{u_i}(g_{u_i})$  satisfy these additional two conditions: (1) they are twice-differentiable functions of respectively  $g_{0_i}$  and  $g_{u_i}$ ; (2) if  $g_{0_i} > 0$  and  $g_{u_i} > 0$  then (i)  $\partial B_{0_i}(g_{0_i})/\partial g_{0_i} < 0$ ,  $\partial B_{u_i}(g_{u_i})/\partial g_{u_i} < 0$ ; (ii)  $\partial^2 B_{0_i}(g_{0_i})/\partial g_{0_i}^2 >$

0,  $\partial^2 B_{u_i}(g_{u_i})/\partial g_{u_i}^2 > 0$ ; and (iii)  $\partial^2 B_{0_i}(g_{0_i})/\partial g_{0_i}^2$  and  $\partial^2 B_{u_i}(g_{u_i})/\partial g_{u_i}^2$  are monotonically decreasing functions of respectively  $g_{0_i}$  and  $g_{u_i}$ . Then there exists an isolated once-differentiable trajectory of minima  $x(\mu)$  of  $B_Q(x, \mu) = c^T x + \mu F_Q(x)$  converging to  $x^*$ , solution of (21) [17, Theorem 12]. Conditions (i)-(iii) hold for  $B_{u_i}(g_{u_i})$ . Conditions (ii)-(iii) also hold for  $B_{0_i}(g_{0_i})$ :  $\partial^2 B_{0_i}(g_{0_i})/\partial g_{0_i}^2 = 1/g_{0_i}^2 + q_i > 0$  and it is a monotonically decreasing function of  $g_{0_i}$ . Condition (i) means  $\partial B_{0_i}(g_{0_i})/\partial g_{0_i} = -1/g_{0_i} + q_i g_{0_i} < 0$ , i.e.,  $q_i < 1/g_{0_i}^2 = 1/x_i^2$ . Since  $x_i < u_i$ , this condition is satisfied for  $q_i \leq 1/u_i^2$ . As seen below, such a small regularization term is also needed to guarantee  $F_Q(x)$  is a self-concordant barrier with parameter  $\nu = n$ .

Making use of the structural optimization theory, as described in [22], it can be seen that  $F_Q(x)$  is a self-concordant barrier, and for small enough regularizations, it has a small parameter. (We recall that self-concordant functions guarantee the efficiency of Newton's method, while self-concordant barriers—hopefully with small parameters—guarantee the efficiency of a path-following algorithm [22].) For this purpose, consider the linear problem (21), and rewrite its barrier  $F_Q(x)$  as

$$F_Q(x) = \sum_{i=1}^n F_{q_i}(x_i), \quad \text{where} \quad F_{q_i}(x_i) = \frac{1}{2} q_i x_i^2 - \ln x_i - \ln(u_i - x_i), \quad (32)$$

$q_i \geq 0$  being the diagonal terms of  $Q$ . Since the sum of self-concordant functions and self-concordant barriers is respectively a self-concordant function [22, Theorem 4.1.1] and a self-concordant barrier [22, Theorem 4.2.2] for the intersection of domains, we consider a particular  $F_{q_i}(x_i)$ . By definition,  $F_{q_i}(x_i) : (0, u_i) \rightarrow \mathbb{R}$  is a self-concordant barrier for its domain if: (1) it is a self-concordant function, i.e., there exists a constant  $M_{F_{q_i}}$  such that

$$|F_{q_i}'''(x_i)h^3| \leq M_{F_{q_i}} (F_{q_i}''(x_i)h^2)^{3/2}$$

for all  $x_i$  of domain of  $F_{q_i}$  and  $h \in \mathbb{R}$ ; and (2), there exists a value  $\nu_i$ , called the parameter of the barrier, such that

$$\sup_{h \in \mathbb{R}} 2F_{q_i}'(x_i)h - F_{q_i}''(x_i)h^2 \leq \nu_i, \quad (33)$$

or equivalently,

$$[F_{q_i}''(x_i)]^{-1} (F_{q_i}'(x_i))^2 \leq \nu_i \quad (34)$$

for all  $x_i$  of domain of  $F_{q_i}$ . We first show that  $F_{q_i}(x_i)$  is self-concordant function.

**Lemma 2**  $F_{q_i}(x_i)$ , defined in (32), is a self-concordant function in its domain  $\{x_i : 0 < x_i < u_i\}$ .

*Proof* Let us partition  $F_{q_i}$  in three terms

$$F_{q_i}(x_i) = F_1(x_i) + F_2(x_i) + F_3(x_i),$$

where

$$F_1(x_i) = \frac{1}{2} q_i x_i^2, \quad F_2(x_i) = -\ln x_i, \quad F_3(x_i) = -\ln(u_i - x_i).$$

The domains of  $F_1(x_i)$ ,  $F_2(x_i)$  and  $F_3(x_i)$  are, respectively,  $\mathbb{R}$ ,  $\{x_i : x_i > 0\}$  and  $\{x_i : x_i < u_i\}$ . It is known that the convex quadratic function  $F_1(x_i)$  and the logarithmic barriers  $F_2(x_i)$  and  $F_3(x_i)$  are self-concordant functions (see, e.g., [22]) with constants  $M_{F_1} = 0$ ,  $M_{F_2} = M_{F_3} = 2$ . By [22, Theorem 4.1.1] we have that the sum of self-concordant functions is self-concordant with constant equal to the maximum of the constants for all the functions. Therefore,  $F_{q_i}(x_i)$  is self-concordant function in  $\text{dom } F_1 \cap \text{dom } F_2 \cap \text{dom } F_3 = \{x_i : 0 < x_i < u_i\}$  and  $M_{F_{q_i}} = \max\{0, 2, 2\} = 2$ .  $\square$

Lemma 3 shows  $F_{q_i}(x_i)$  is a self-concordant barrier, and provides an upper bound for its parameter. It makes use of the fact that the logarithmic barrier

$$F_i(x_i) = -\ln x_i - \ln(u_i - x_i) \quad (35)$$

for the set  $0 < x_i < u_i$  is self-concordant with parameter 1. This is easily seen by noting that, from (34),

$$\begin{aligned} \frac{(F'_i(x_i))^2}{F''_i(x_i)} &= \frac{\left(-\frac{1}{x_i} + \frac{1}{u_i - x_i}\right)^2}{\frac{1}{x_i^2} + \frac{1}{(u_i - x_i)^2}} \cdot \frac{(u_i - x_i)^2 x_i^2}{(u_i - x_i)^2 x_i^2} = \frac{(2x_i - u_i)^2}{(u_i - x_i)^2 + x_i^2} \leq 1 \\ \iff (2x_i - u_i)^2 - ((u_i - x_i)^2 + x_i^2) &= 2x_i(x_i - u_i) \leq 0. \end{aligned} \quad (36)$$

**Lemma 3**  $F_{q_i}(x_i)$  is a self-concordant barrier in its domain  $\{x_i : 0 < x_i < u_i\}$ , with parameter  $v_i \leq q_i u_i^2 + 1$ .

*Proof* By (35), we have  $F_{q_i}(x_i) = \frac{1}{2} q_i x_i^2 + F_i(x_i)$ ,  $F'_{q_i}(x_i) = q_i x_i + F'_i(x_i)$  and  $F''_{q_i}(x_i) = q_i + F''_i(x_i)$ . Then, by (33) and (36),

$$\begin{aligned} \max_{h \in \mathbb{R}} 2F'_{q_i}(x_i)h - F''_{q_i}(x_i)h^2 &= \max_{h \in \mathbb{R}} 2(q_i x_i + F'_i(x_i))h - (q_i + F''_i(x_i))h^2 \\ &\leq \max_{h \in \mathbb{R}} (2q_i x_i h - q_i h^2) + \max_{h \in \mathbb{R}} (2F'_i h - F''_i(x_i)h^2) \leq q_i x_i^2 + 1 \leq q_i u_i^2 + 1. \end{aligned}$$

$\square$

Next Lemma shows that it is possible to obtain a parameter for  $F_{q_i}(x_i)$  smaller than the one provided by Lemma 3.

**Lemma 4** The parameter of the self-concordant barrier  $F_{q_i}(x_i)$  in its domain  $\{x_i : 0 < x_i < u_i\}$  is

$$\begin{aligned} v_i &= 1 & \text{if } & 0 \leq q_i \leq 1/u_i^2, \\ v_i &= q_i u_i^2 & \text{if } & q_i \geq 1/u_i^2. \end{aligned} \quad (37)$$

*Proof* From (34), it has to be proven that

$$T_i(x_i) = [F''_{q_i}(x_i)]^{-1} (F'_{q_i}(x_i))^2 = \frac{(q_i x_i - \frac{1}{x_i} + \frac{1}{u_i - x_i})^2}{q_i + \frac{1}{x_i^2} + \frac{1}{(u_i - x_i)^2}} \quad (38)$$

is upper bounded by  $v_i$  for  $0 < x_i < u_i$ . Consider first the case  $0 \leq q_i \leq 1/u_i^2$ . Since  $-1 \leq q_i x_i^2 - 1 \leq 0$ , then

$$T_i(x_i) = \frac{x_i^2 (q_i x_i - \frac{1}{x_i} + \frac{1}{u_i - x_i})^2}{x_i^2 (q_i + \frac{1}{x_i^2} + \frac{1}{(u_i - x_i)^2})} = \frac{(q_i x_i^2 - 1 + \frac{x_i}{u_i - x_i})^2}{q_i x_i^2 + 1 + \frac{x_i^2}{(u_i - x_i)^2}} \leq \frac{\max \left\{ (q_i x_i^2 - 1)^2, \left( \frac{x_i}{u_i - x_i} \right)^2 \right\}}{q_i x_i^2 + 1 + \frac{x_i^2}{(u_i - x_i)^2}} < 1.$$

For the second case,  $q_i \geq 1/u_i^2$ , defining

$$q_i = \alpha \frac{1}{u_i^2}, \quad \alpha \geq 1, \quad \text{and} \quad x_i = \beta u_i, \quad 0 < \beta < 1, \quad (39)$$

the inequality  $T_i(x_i) \leq q_i u_i^2$  is proven by noting that

$$\begin{aligned} \frac{(q_i x_i - \frac{1}{x_i} + \frac{1}{u_i - x_i})^2}{q_i + \frac{1}{x_i^2} + \frac{1}{(u_i - x_i)^2}} &= \frac{\left(\frac{\alpha\beta}{u_i} - \frac{1}{\beta u_i} + \frac{1}{u_i(1-\beta)}\right)^2}{\frac{\alpha}{u_i^2} + \frac{1}{u_i^2\beta^2} + \frac{1}{u_i^2(1-\beta)^2}} \\ &= \frac{\left(\alpha\beta - \frac{1}{\beta} + \frac{1}{(1-\beta)}\right)^2}{\alpha + \frac{1}{\beta^2} + \frac{1}{(1-\beta)^2}} \\ &\leq \frac{\left(\alpha - \frac{1}{\beta} + \frac{1}{(1-\beta)}\right)^2}{\alpha + \frac{1}{\beta^2} + \frac{1}{(1-\beta)^2}} \cdot \frac{(1-\beta)^2\beta^2}{(1-\beta)^2\beta^2} \\ &= \frac{(\alpha(1-\beta)\beta + 2\beta - 1)^2}{\alpha(1-\beta)^2\beta^2 + 2\beta^2 - 2\beta + 1} \leq q_i u_i^2 = \alpha, \end{aligned}$$

where for first inequality we used that  $\alpha \geq 1$  and  $0 < \beta < 1$ . We will show that the last inequality holds. Since the denominator  $\alpha(1-\beta)^2\beta^2 + 2\beta^2 - 2\beta + 1$  is positive (because  $\alpha(1-\beta)^2\beta^2 > 0$  and the minimum of  $2\beta^2 - 2\beta + 1$  is  $1/2$  at minimizer  $\beta = 1/2$ ),

$$\frac{(\alpha(1-\beta)\beta + 2\beta - 1)^2}{\alpha(1-\beta)^2\beta^2 + 2\beta^2 - 2\beta + 1} \leq \alpha \quad (40)$$

is equivalent to

$$\begin{aligned} (\alpha(1-\beta)\beta + 2\beta - 1)^2 &= \alpha^2(1-\beta)^2\beta^2 + 2\alpha(1-\beta)\beta(2\beta - 1) + (2\beta - 1)^2 \leq \\ \alpha(\alpha(1-\beta)^2\beta^2 + 2\beta^2 - 2\beta + 1) &= \alpha^2(1-\beta)^2\beta^2 + \alpha(2\beta^2 - 2\beta + 1). \end{aligned}$$

Simplifying the above inequality we obtain the equivalent one

$$\alpha(2\beta^2 - 2\beta + 1) - 2\alpha(1-\beta)\beta(2\beta - 1) - (2\beta - 1)^2 = \alpha(4\beta^3 - 4\beta^2 + 1) - (2\beta - 1)^2 \geq 0.$$

Since  $4\beta^3 - 4\beta^2 + 1$  is positive for  $0 < \beta < 1$ , and  $\alpha \geq 1$ ,

$$\alpha(4\beta^3 - 4\beta^2 + 1) - (2\beta - 1)^2 \geq 4\beta^3 - 4\beta^2 + 1 - (2\beta - 1)^2 = 4\beta(\beta - 1)^2 \geq 0,$$

and thus (40) holds.  $\square$

Figure 4 plots (38) for  $u_i = 1000$ , and  $q_i = 1/u_i^2 \cdot 0.9$  (left figure) and  $q_i = 1/u_i^2 \cdot 1000$  (right figure). It can be observed how tight are bounds (37).

By Lemma 4 and Theorem 4.2.2 of [22], barrier  $F_Q(x)$  is self-concordant in the domain  $\{x : 0 < x < u\}$ , with parameter  $\nu = \sum_{i=1}^n \nu_i$ . In view of Theorem 4.2.9 of [22] the path-following algorithm terminates in a number of iterations  $O(\sqrt{\nu} \ln 1/\varepsilon)$ ,  $\varepsilon$  being the accuracy of the solution. If  $q_i \leq 1/u_i^2$  for all  $i = 1, \dots, n$ , then  $\nu = n$ . This is the same parameter of the standard (non-regularized) logarithmic barrier, which is also the lowest possible value for any self-concordant barrier [22, Lemma 4.3.1].

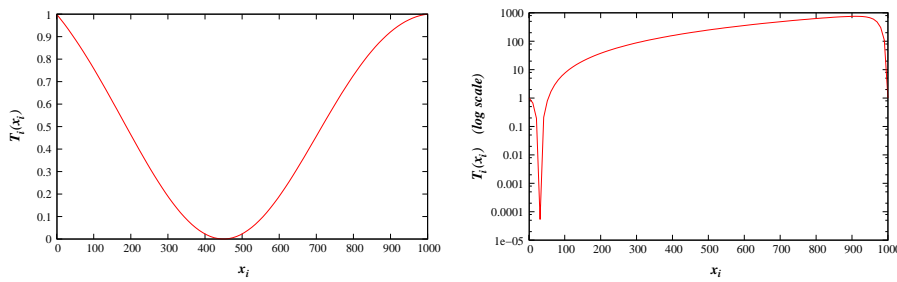


Fig. 4 Plot of (38) for  $u_i = 1000$ , and  $q_i = 1/u_i^2 \cdot 0.9$  (left figure) and  $q_i = 1/u_i^2 \cdot 1000$  (right figure)

Therefore, for small enough  $q_i, i = 1, \dots, n$ , the regularized interior-point approach is as efficient as the standard one in number of iterations. On the other hand, if  $q_i > 1/u_i^2$  the method is no longer strongly polynomial, and it depends on upper bounds  $u$ , according to  $v$ . However, in the tests performed, it has been observed that there is a significant wide range of values  $q_i$  that preserves a small number of interior-point iterations. This is also the usual behaviour of interior-point algorithms, which in practice need no more than 100 iterations, far from the number predicted by the theoretical complexity  $O(\sqrt{n} \ln 1/\varepsilon)$ . Next Subsection empirically shows that the number of interior-point iterations significantly increases for only large regularization terms.

A barrier similar to  $F_Q(x)$  was named *augmented barrier* in [24]. For any positive semidefinite  $Q$  matrix, it was shown to be a self-concordant function, but not a self-concordant barrier when its domain is a cone (i.e.,  $x \geq 0$ ). In our case  $Q$  is (positive semidefinite) diagonal, and its domain is defined by box constraints  $0 \leq x \leq u$ . This allowed us to prove that  $F_Q(x)$  is self-concordant barrier, though for very large regularizations it may increase the number of interior-point iterations. Note that, unless the problem is unbounded, there always exists a (possibly large)  $u$  such that  $0 \leq x \leq u$  are valid box constraints. According to (37), a small complexity bound is only guaranteed if  $q_i \leq 1/u_i^2$ , which for very large  $u$  can result in a negligible regularization term. However, as stated in next Subsection, it has been observed that in practice regularization terms significantly larger than  $1/u_i^2$  can be effectively used, without increasing the number of interior-point iterations. According to [24], it is even possible, at least theoretically, to trace the central path at linear speed by using augmented barriers for convex cones, though they don't fit the standard interior-point theory [23].

#### 4.2 Implementation and computational results for linear problems

Although our purpose is to regularize the specialized interior-point method for primal block-angular problems, we made a preliminary test with a general algorithm for linear problems. The primal regularized logarithmic barrier problem  $B_Q(x, \mu)$  of (24) has been included in a home-made primal-dual path-following code for linear programming. This code solves the normal equations using the Ng-Peyton sparse Cholesky package [25]. It also includes the second-order Merothra's heuristic direction [32]. It is similar to the well-known PCx code [16].

**Table 4** Results for Netlib problems, for both regularized and non-regularized variants

Instance	$n$	$m$	iter. IP	iter. R-IP	$q$	$f^*$
Adlittle	138	56	15	14	4.98e-03	2.25e+05
Afiro	51	27	10	10	2.3e-05	-4.65E+02
Agg	615	488	30	29	8.4e-11	-3.60e+07
Bandm	472	305	21	20	4.5e-04	-1.59E+02
Beaconfd	295	173	13	13	1.7e-05	3.36E+04
Blend	114	74	15	14	3.4e-02	-3.08E+01
Bore3d	333	233	16	16	3.4e-06	1.37E+03
Brandy	303	182	24	20	6.6e-05	1.52E+03
Degen3	2604	1503	22	22	6.1e-05	-9.87e+02
E226	472	223	24	22	4.5e-02	-1.88E+01
Etamacro	734	400	28	27	4.8e-08	-7.56E+02
Fit2d	10524	25	24	24	4.5e-05	-6.85E+04
Grow15	645	300	19	19	2.7e-11	-1.07E+08
Kb2	68	43	19	19	1.0e-06	-1.75E+03
Maros-r7	9408	3136	18	18	1.0e-07	1.50E+06
Recipe	178	87	11	11	1.0e-03	-2.67E+02
Sc105	163	104	12	11	6.1e-05	-5.22E+01
Wood1p	2595	244	24	20	0.3854	1.44E+00

**Table 5** Number of iterations for different regularizations and instances “Bore3d” and “Agg”

$\delta$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	0
Bore3d	53	19	17	16	16	16	16	16	16	16
Agg	> 100	> 100	> 100	> 100	> 100	55	31	30	29	30

Heuristically, the regularization  $Q$  matrix was computed as  $Q = qI$ , where  $q \in \mathbb{R}$  is

$$q = \delta \left( \frac{\sum_{i=1}^n |c_i|}{\sum_{i=1}^n x_i^0} \right). \quad (41)$$

$\delta > 0$  is a adjustable scalar parameter and  $x^0$  is the initial primal point. (41) guarantees that, for  $\delta = 1$ , both the linear and quadratic terms are of the same order of magnitude at  $x^0$ . We solved a subset of Netlib collection. Each problem was solved considering 13 different  $\delta$ , from  $10^0$  to  $10^{-12}$ . Table 4 shows the best execution of the regularized variant. For each problem, the table reports the number of variables ( $n$ ) and constraints ( $m$ ), the number of iterations performed by the standard algorithm (“iter. IP”) and the regularized variant (“iter. R-IP”), the scalar  $q$  for the definition of  $Q$  associated to the best  $\delta$ , and the optimal objective function ( $f^*$ ) found by both methods. Tuning  $q$ , the regularized variant was as fast (in number of iterations) as the standard algorithm. In general, the regularized approach is similar to the standard one, but it takes more iterations for significantly large  $q$ . This is observed in Table 5 which reports the number of iterations for two particular instances and different  $\delta$  values (case  $\delta = 0$  corresponds to the non-regularized method). A similar experiment was performed for the multicommodity instance PDS1 of Table 1 (also of Table 6). It was solved using the simplest regularization term  $Q = \delta I$ , for different  $\delta$  values. For  $\delta \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1\}$  the overall number of interior-point iterations for solving the problem was about 40 ( $\delta = 0$  corresponds to the standard non-regularized PDS1 linear problem). These values are greater than the theoretical

value  $1/u_i^2$ , which for many variables was of  $10^{-7}$ . For  $\delta = 10^2$  and  $\delta = 10^3$  the algorithm required about 50 and 60 interior-point iterations, respectively; and more than 120 iterations were needed for  $\delta = 10^3$ . Therefore, only large regularizations significantly increased the number of interior-point iterations.

The above results are consistent with Subsection 4.1: significantly large regularizations increase the number of interior-point iterations. On the other hand, the regularization provides a better conditioned  $\Theta$  matrix (31), and, most important, would improve the preconditioner of the specialized approach according to Section 3. As observed in next section, in spite of this tradeoff between number of iterations and time per iteration, in general, the regularization term improves the overall solution time for some primal block-angular problems.

## 5 Computational results for primal block-angular problems

Two primal block-angular problems have been considered: multicommodity network flows, and the minimum congestion (or maximum concurrent flow) problem [7]. They were solved with the specialized interior-point method for primal block-angular problems updated with a regularized function. Both regularizations (23) and (24) were implemented. For multicommodity flows we extended the code IPM of [10], mainly implemented in C, but for the Ng-Peyton Cholesky package [25] for factorizations, which is coded in Fortran. For the minimum congestion problem we extended the code PRBLOCK\_IP of [14] for general primal-block angular problems. PRBLOCK\_IP is implemented under the MATLAB environment, with Cholesky factorizations through external precompiled routines of the Ng-Peyton Cholesky package. All runs were carried out on a SUN Fire V20Z server with two AMD Opteron 2.46 GHZ processors (without exploiting parallelism capabilities) and 8 GB of RAM. Results are provided in next two subsections.

### 5.1 Multicommodity flow problems

Multicommodity flow problems match the primal block-angular formulation (2) for  $L_i = I$  and  $N_i$  being a node-arc incidence matrix. Blocks are related to commodities in this problem.

We considered three kind of instances. The first one corresponds to the PDS ones [9]. These problems arise from a logistic model for evacuating patients from a place of military conflict. Each instance depends on a parameter  $t$  that denotes the planning horizon under study (in number of days). The size of the network increases with  $t$ , whereas the number of commodities is always 11. Problems obtained with this generator are denoted as PDS $t$ . The PDS instances can be retrieved from <http://www.di.unipi.it/di/groups/optimize/Data/MMCF.html>.

The second type of instances was obtained with the Mnetgen generator [1]. It can be retrieved from the above URL. These instances will be denoted as  $Mm'-k-d$ , where  $m'$  is the number of nodes,  $k$  the number of commodities, and  $d$  is related to the density of the network; the larger  $d$  the denser is the network. In those instances,

**Table 6** Characteristics of multicommodity instances

Instance	$k$	$m'$	$n'$	$n$	$m$	$f^*$
PDS1	11	126	372	4464	1758	2.908e10
PDS5	11	686	2325	27900	9871	2.805e10
PDS10	11	1399	4792	57504	20181	2.673e10
PDS15	11	2125	7756	93072	31131	2.518e10
PDS20	11	2857	10858	130296	42285	2.832e10
PDS25	11	3554	13580	162960	52674	2.262e10
PDS30	11	4223	16148	193776	62601	2.139e10
PDS40	11	5652	22059	264708	84231	1.8886e10
PDS50	11	7031	27668	332036	105009	1.6603e10
PDS60	11	8423	33388	400656	126041	1.4265e10
PDS70	11	9750	38396	460752	145646	1.2241e10
PDS80	11	10989	42472	509664	163351	1.1469e10
PDS90	11	12186	46161	553932	180207	1.1087e10
M32-32-12	32	32	486	16038	1510	8.056e5
M64-64-12	64	64	511	33215	4607	4.624e6
M128-64-12	64	128	1171	76115	9363	1.927e7
M128-128-12	128	128	1204	155316	17588	4.014e7
M256-256-12	256	256	2204	566428	67740	3.979e8
M512-128-12	128	512	4786	617394	70322	8.280e8
M512-256-12	256	512	4810	1236170	135882	1.649e9
M512-512-12	512	512	4786	2455218	266930	3.488e9
M768-128-6	128	768	2317	298893	100621	2.118e9
M768-256-6	256	768	2370	609090	198978	4.249e9
tripart1	16	192	2096	35632	5168	6.348e7
tripart2	16	768	8432	143344	20720	3.870e8
tripart3	20	1200	16380	343980	40380	2.694e8
tripart4	35	1050	24815	893340	61565	1.775e7
gridgen1	320	1025	3072	986112	331072	622e12

80% of the arcs have mutual capacities (potential active linking constraints), 90% have individual capacities (upper bounds per commodity), and 30% of the arcs have a high cost [18].

The last set of instances was obtained with the Tripartite generator and with a variation for multicommodity flows of the Gridgen generator. They are known to be difficult linear programming instances, and interior-point algorithms outperformed simplex variants on them [6, 11]. Five such test examples are available. They can be obtained from [http://www-eio.upc.es/~jcastro/mmcnf\\_data.html](http://www-eio.upc.es/~jcastro/mmcnf_data.html).

Table 6 shows the main characteristics of these instances: number of commodities/blocks ( $k$ ), number of nodes ( $m'$ ) and arcs ( $n'$ ), number of variables ( $n$ ) and constraints ( $m$ ), and optimal objective value ( $f^*$ ). Table 7 shows the results for the specialized interior-point approach (columns “IPM”), the regularized version based on (24) (columns “RIPM”), and the proximal-point regularization based on (23) (columns “PIPM”). For each variant we show the number of interior-point iterations (columns “it.”), overall number of PCG iterations (columns “PCG”) and CPU time (columns “CPU”). CPU time of the fastest execution is marked in boldface. For all the instances (but “gridgen1” that was tuned to get a better result), the regularization  $Q$  matrix in (25) was heuristically computed as  $Q^t = t/\mu^0 \cdot \text{diag}(x_1^0/z_1^0, \dots, x_n^0/z_n^0)$ , where  $t$  is the iteration counter,  $\mu^0$  is the initial barrier parameter, and  $x^0$  and  $z^0$  are



**Table 7** Results with IPM, RIPM and PIPM for multicommodity instances

Instance	IPM			RIPM			PIPM		
	it.	PCG	CPU	it.	PCG	CPU	it.	PCG	CPU
PDS1	41	513	0.09	36	342	0.07	35	303	<b>0.06</b>
PDS5	62	1093	1.66	54	669	<b>1.29</b>	55	784	1.37
PDS10	78	1647	7.25	73	1373	<b>6.8</b>	77	1720	7.35
PDS15	90	2696	21.9	80	1674	<b>15.9</b>	86	2321	18.4
PDS20	107	4718	56.5	94	3273	<b>43.6</b>	112	6534	66.8
PDS25	114	3648	74.6	118	3348	67.0	103	2321	<b>48.3</b>
PDS30	119	4063	111	108	2839	<b>89.4</b>	111	3511	100
PDS40	136	7241	313	129	5031	<b>255</b>	128	5769	258
PDS50	140	10037	586	136	6433	446	128	6091	<b>395</b>
PDS60	137	6719	639	133	5278	<b>546</b>	134	7005	711
PDS70	143	5799	<b>864</b>	142	6044	894	142	7149	1030
PDS80	148	8239	1340	141	6192	<b>1140</b>	144	8718	1460
PDS90	158	12021	2270	144	6112	<b>1330</b>	149	8570	2220
M32-32-12	39	1300	0.44	30	940	<b>0.35</b>	33	938	0.36
M64-64-12	59	1387	1.71	50	722	<b>1.2</b>	55	909	1.43
M128-64-12	75	3543	<b>9.62</b>	66	2756	<b>9.62</b>	68	3760	13.2
M128-128-12	97	3839	26.6	82	2165	<b>17.7</b>	92	3722	25.8
M256-256-12	121	5182	203	112	3720	162	115	3621	<b>161</b>
M512-128-12	117	6164	377	124	5799	<b>366</b>	119	64150	394
M512-256-12	139	6945	872	139	6545	<b>842</b>	149	7241	925
M512-512-12	179	12074	<b>2780</b>	183	13167	2960	158	22180	4690
M768-128-6	130	26149	581	137	9248	<b>233</b>	134	116927	277
M768-256-6	128	10710	<b>561</b>	155	26446	1250	Too many iterations		
tripart1	58	1976	1.7	89	721	<b>1.57</b>	76	1546	1.78
tripart2	87	4092	17.3	97	1562	<b>10.2</b>	80	2576	12.8
tripart3	90	6978	62.4	106	3178	<b>36.4</b>	108	9238	74.0
tripart4	133	14660	265	136	4947	<b>128</b>	125	8321	171
gridgen1	242	96877	7400	219	5703	<b>618</b>	133	19977	1530

the initial primal and dual slack variables. No particular  $\delta$  term was needed, unlike (41). The effect of  $t$  is to avoid a too quick reduction of the regularization term as  $\mu$  approaches zero. The effect of  $1/\mu^0$  is to make the regularization term  $\mu Q$  relative to  $\mu^0$ . Several other regularization variants were also tested with similar results. Results might have been improved (specially in instances “PDS70”, “M512-512-12” and “M768-256-6”) by tuning  $Q$  for each particular instance. However, using the same problem independent choice confirms the robustness of the approach. As said above, tuning was only performed for instance “gridgen1”. Using the above default regularization, “RIPM” columns “it”, “PCG” and “CPU” of Table 7 for instance “gridgen1” would have been 212, 13381 and 1150, respectively.

It is observed that, in general, RIPM is more effective than both IPM and PIPM. Since the spectral radius of  $D^{-1}(C^T B^{-1} C)$  is lower in RIPM and PIPM than IPM, the number of PCG iterations is effectively decreased by the regularization term, as we see from columns “PCG”. This is observed in Figure 5, which shows the evolution of spectral radius of  $D^{-1}(C^T B^{-1} C)$  for instance M32-32-12 with IPM, RIPM and PIPM. For the difficult Tripartite and Gridgen instances, RIPM always outperformed IPM and PIPM. In instance “gridgen1”, RIPM significantly reduced the number of PCG iterations and was ten times faster than IPM.

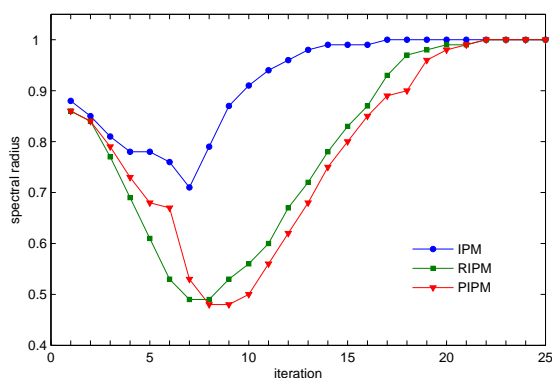


Fig. 5 Spectral radius for first iterations of problem M32-32-12, for IPM, RIPM and PIPM

Table 8 Results with CPLEX-11 for multicommodity instances

Instance	barrier		primal simplex		dual simplex	
	it.	CPU	it.	CPU	it.	CPU
PDS1	15	<b>0.02</b>	567	<b>0.02</b>	112	<b>0.02</b>
PDS5	36	1.35	3052	0.28	2170	<b>0.27</b>
PDS10	33	7.07	8409	0.94	5078	<b>0.75</b>
PDS15	39	30.76	32808	3.92	8963	<b>1.97</b>
PDS20	36	79.8	78508	15.77	16005	<b>4.75</b>
PDS25	38	123.76	147651	37.58	23209	<b>7.54</b>
PDS30	38	191.87	298932	105.38	27830	<b>10.26</b>
PDS40	38	297.66	451783	331.52	43295	<b>18.75</b>
PDS50	40	485.10	218455	339.26	56069	<b>27.72</b>
PDS60	38	645.24	700718	1138.37	63735	<b>35.59</b>
PDS70	40	719.63	764918	957.39	84593	<b>54.73</b>
PDS80	40	814.56	828482	1144.84	90133	<b>62.56</b>
PDS90	46	1050.53	864758	1378.26	93859	<b>65.50</b>
M32-32-12	14	1.75	17755	1.49	1682	<b>0.12</b>
M64-64-12	16	6.59	100575	26.72	8222	<b>0.88</b>
M128-64-12	20	59.7	275263	565.04	19857	<b>8.37</b>
M128-128-12	21	145.8	660643	2502.81	36483	<b>15.32</b>
M256-256-12	24	2149.5	2398575	14428.14	104108	<b>42.62</b>
M512-128-12	20	5594.1	3144031	21179.05	90977	<b>45.57</b>
M512-256-12	25	14330	not executed		165638	<b>67.27</b>
M512-512-12	failed		not executed		314765	<b>155.80</b>
M768-128-6	22	627.7	161141	139.44	88387	<b>18.52</b>
M768-256-6	29	1436.1	419979	688.32	177854	<b>61.72</b>
tripart1	21	3.99	14679	1.30	4197	<b>1.12</b>
tripart2	25	<b>36.01</b>	216854	111.26	58316	106.97
tripart3	28	<b>138.8</b>	681575	1024.05	104800	382.47
tripart4	29	<b>1323.2</b>	1847046	3037.08	197726	1638.12
gridgen1	64	<b>12288</b>	1469625	51289.02	416875	16467.79

**Table 9** Characteristics of minimum congestion instances

Instance	$k$	$\sum_{i=1}^k m_i$	$\sum_{i=1}^k n_i$	$l$	$n$	$m$	$f^*$
xPDS1	13	1357	8928	1115	10043	2472	1.999629
xM32-32	34	992	32076	1457	33533	2449	0.1157906
xM64-64	66	4032	66430	1532	67962	5564	1.651421
xM128-64	66	8128	152230	3512	155742	11640	1.802644
xM128-128	130	16256	310632	3611	314243	19867	1.9436

For the purpose of comparison, Table 8 reports the number of iterations (columns “it.”) and CPU time (columns “CPU”) with the three options of CPLEX-11: barrier, primal and dual simplex. As in previous table, CPU time of the fastest execution is marked in boldface. It is worth noting that IPM/RIPM are academic codes, whereas CPLEX-11 is a highly optimized package. In particular, RIPM uses the Ng-Peyton standard package for Cholesky factorization, and CPLEX-11 specialized linear algebra routines. In spite of this, RIPM was always much more efficient than the barrier method of CPLEX-11. The dual simplex algorithm is known to be very efficient for the PDS and Mnetgen instances [11], and it outperforms both RIPM and PIPM. However, for the difficult Tripart and Gridgen instances IPM was known to be the most efficient solver to date [11]. This still holds for CPLEX-11, RIPM being superior to IPM.

## 5.2 Minimum congestion problems

The minimum congestion problem is equivalent to the maximum concurrent flow problem. In the literature, both problems are usually seen as one, and denoted as the maximum concurrent flow problem [7]. These problems arise in practical applications on telecommunications networks. They have proved to be difficult for simplex algorithms [6]. The minimum congestion problem, which is defined on an infeasible nonoriented multicommodity network, finds 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. This min-max model can be transformed to a linear program by addition of auxiliary variables and constraints. The formulation considered in this work is described in [14]. The resulting model, which is no longer a nonoriented multicommodity flow problem, has primal block-angular structure.

Table 9 shows the dimensions and objective function of the instances considered. They were generated from some of the multicommodity instances of Subsection 5.1, increasing the supply and demand by a factor of two. For each instance, Table 9 reports number of blocks ( $k$ ), number of constraints and variables in diagonal blocks ( $\sum_{i=1}^k m_i$  and  $\sum_{i=1}^k n_i$ ), number of linking constraints ( $l$ ), overall number of variables and constraints of the linear problem ( $n$  and  $m$ ), and optimal objective function ( $f^*$ ).

Those instances were solved with the PRBLOCK\_IP code [14], which implements the specialized algorithm of Section 2 for general primal-block angular problems. Table 10 shows the results with that code, and two regularized versions based on (24) (columns “RPRBLOCK\_IP”) and (23) (columns “PPRBLOCK\_IP”). The

**Table 10** Results with PRBLOCK\_IP, RPRBLOCK\_IP and PPRBLOCK\_IP for minimum congestion instances

Instance	PRBLOCK_IP			RPRBLOCK_IP			PPRBLOCK_IP		
	it.	PCG	CPU*	it.	PCG	CPU*	it.	PCG	CPU*
xPDS1	27	65	1.75	26	72	<b>1.60</b>	25	69	1.63
xM32-32-12	24	156	4.88 <sup>(2)</sup>	24	170	<b>3.53</b>	26	202	5.23
xM64-64-12	27	113	14.43 <sup>(2)</sup>	29	118	4.99	23	98	<b>4.01</b>
xM128-64-12	33	211	50.22	37	179	54.92	27	165	<b>40.74</b>
xM128-128-12	31	176	52.82	31	188	58.32	30	140	<b>50.44</b>

\* CPU time spent on external precompiled Cholesky and incomplete Cholesky routines

<sup>(n)</sup>  $n$  is the number of interior-point iterations after switching to full Cholesky for normal equations

**Table 11** Results with CPLEX-11 for minimum congestion instances

Instance	barrier		primal simplex		dual simplex	
	it.	CPU	it.	CPU	it.	CPU
xPDS1	13	20.4	54	<b>0.24</b>	4	<b>0.24</b>
xM32-32-12	11	89.4	21542	3.82	11910	<b>1.71</b>
xM64-64-12	12	233.8	20591	<b>9.57</b>	11463	16.6
xM128-64-12	11	690.3	77528	<b>100.9</b>	29582	186.6
xM128-128-12	11	1421.5	274873	<b>673.5</b>	72217	768.5

meaning of columns “it.,” “PCG” and “CPU” is the same as in previous tables. For these instances the regularization  $Q$  matrix was heuristically computed as  $Q = \delta/\mu^0 \cdot \text{diag}(x_1^0/z_1^0, \dots, x_n^0/z_n^0)$ ,  $\delta \in \mathbb{R}$  being a small positive scalar with values in  $\{10^{-1}, 10^{-2}, 10^{-3}\}$ . Note that in this case, unlike for multicommodity problems, we adjusted  $\delta$  as in (41).

CPU time of fastest execution is marked in boldface in Table 10. Since MATLAB is an interpreted language, the overall execution time is meaningless. As it was done in [14], we only consider the execution time spent in the external precompiled Ng-Peyton Cholesky routines (including minimum degree ordering, symbolic factorization, numerical factorization, and numerical solution). This time would be a significant fraction (e.g., from 40% to 80%) of the overall execution time in a C implementation [14]. As we approach an optimal point, system (8) becomes more ill-conditioned, and PCG may provide inaccurate solutions. When this happens, PRBLOCK\_IP switches to the solution of normal equations by a Cholesky factorization, significantly increasing the solution time. We observed that, in instances “xM32-32-12” and “xM64-64-12”, regularization avoided this switching to full Cholesky, improving the performance when the factorization is expensive. It may also be observed that either RPRBLOCK\_IP or PPRBLOCK\_IP always provided slightly better executions. In general, the proximal point regularization outperformed RPRBLOCK\_IP, mainly for large instances. The improvement due to regularization was, in general, not as significant as for the multicommodity instances of Subsection 5.1. This behaviour, which is instance/problem dependent, can be explained by either: (1) matrices  $L_i$  of the minimum congestion problem are no longer diagonal, and thus conditions of Proposition 2 may not be satisfied; (2) even if these conditions are satisfied the reduction of bound (12) may be small or it may not significantly affect the real spec-

tral radius. Finally, Table 11 shows the results obtained with the three algorithms of CPLEX-11. As we see, the specialized algorithm outperformed CPLEX-11 in largest instances.

## 6 Conclusions

From both the theoretical and computational results of this work it can be stated that quadratic regularizations significantly improve the performance of the specialized interior-point algorithm for some classes of primal block-angular linear problems. Adding a regularization term, the specialized algorithm was able to outperform both simplex and interior-point variants of commercial state-of-the-art solvers in some significant difficult instances. However, this behaviour can not be generalized to any problem, and it depends on the reduction of the spectral radius of matrix  $D^{-1}(C^T B^{-1} C)$  due to the particular linking constraints structure. We also observed that for problems with box constraints the quadratic regularization fits the general framework of interior-point algorithms. Among the future tasks to be performed we find the automatic tuning of the regularization matrix  $Q$  for particular instances, and the application to alternative problems. Some work already started along these lines, applying the regularized specialized method to the solution of nonoriented convex multicommodity flow problems for routing in data telecommunications networks.

**Acknowledgements** This work has been supported by grant MTM2006-05550 of the Spanish Ministry of Science and Education. The authors thank Jacek Gondzio and Jean-Philippe Vial for fruitful discussions, and the two anonymous referees for their valuable suggestions.

## References

1. Ali, A., Kennington, J. L.: Mnetgen Program Documentation. Technical Report 77003. Department of Industrial Engineering and Operations Research, Southern Methodist University, Dallas (1977)
2. Altman, A., Gondzio, J.: Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. *Optim. Methods Software* 11, 275–302 (1999).
3. Babonneau, F., du Merle, O., Vial, J.-P.: Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-ACCPM. *Oper. Res.* 54, 184–197 (2006)
4. Bellavia, S., Gondzio, J., Morini, B.: Regularization and preconditioning of KKT systems arising in nonnegative least-squares problems, *Numer. Linear Algebra Appl.* 16, 39–61 (2009)
5. Bergamaschi, L., Gondzio, J., Zilli, G.: Preconditioning indefinite systems in interior point methods for optimization. *Comput. Optim. Appl.* 28, 149–171 (2004)
6. Bienstock, D.: *Potential Function Methods for Approximately Solving Linear Programming Problems. Theory and Practice.* Kluwer, Boston (2002)
7. Bienstock, D., Raskina, O.: Asymptotic analysis of the flow deviation method for the maximum concurrent flow problem, *Math. Prog.* 91, 479–492 (2002)
8. Bixby, R.E.: Solving real-world linear programs: a decade and more of progress. *Oper. Res.* 50, 3–15 (2002)
9. Carolan, W.J., Hill, J.E., Kennington, J.L., Niemi, S., Wichmann, S.J.: An empirical evaluation of the KORBX algorithms for military airlift applications. *Oper. Res.* 38, 240–248 (1990)
10. Castro, J.: A specialized interior-point algorithm for multicommodity network flows. *SIAM J. Optim.* 10, 852–877 (2000)
11. Castro, J.: Solving difficult multicommodity problems through a specialized interior-point algorithm. *Ann. Oper. Res.* 124, 35–48 (2003)

12. Castro, J.: Solving quadratic multicommodity problems through an interior-point algorithm. In: E.W. Sachs, R. Tichatschke (eds.) *System Modelling and Optimization XX*, pp. 199–212. Kluwer, Boston (2003)
13. Castro, J.: Quadratic interior-point methods in statistical disclosure control. *Comput. Manag. Sci.* 2, 107–121 (2005)
14. Castro, J.: An interior-point approach for primal block-angular problems. *Comput. Optim. Appl.*, 36, 195–219 (2007)
15. Chardaire P., Lisser A.: Simplex and interior point specialized algorithms for solving nonoriented multicommodity flow problems. *Oper. Res.* 50, 260–276 (2002)
16. Czyzyk, J., Mehrotra, S., Wagner, M., Wright, S.J.: PCx: An interior-point code for linear programming. *Optim. Methods Softw.*, 11/12, 97–430 (1999)
17. Fiacco, A.V., McCormick, G.P.: *Nonlinear programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York (1968). Reprinted in *Classics In Applied Mathematics* series, SIAM, Philadelphia (1990)
18. Frangioni, A., Gallo, G.: A bundle type dual-ascent approach to linear multicommodity min cost flow problems. *INFORMS J. Comput.* 11, 370–393 (1999)
19. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, Third Ed. Johns Hopkins Univ. Press, Baltimore (1996)
20. Gondzio, J., Sarkissian, R.: Parallel interior-point solver for structured linear programs. *Math. Prog.* 96, 561–584 (2003)
21. Güler O., Ye, Y.: Convergence behaviour of some interior-point algorithms. *Math. Prog.* 60, 215–228 (1993)
22. Nesterov, Y.: *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston (2004)
23. Nesterov, Y., Nemirovskii, A.: *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia (1994)
24. Nesterov, Y., Vial, J.-P.: Augmented self-concordant barriers and nonlinear optimization problems with finite complexity, *Math. Prog.* 99, 149–174 (2004)
25. Ng, E., Peyton, B.W.: Block sparse Cholesky algorithms on advanced uniprocessor computers. *SIAM J. Sci. Comput.* 14, 1034–1056 (1993)
26. Oliveira, A.R.L., Sorensen, D.C.: A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra Appl.* 394, 1–24 (2005)
27. Ouorou, A., Mahey M, Vial, J.-Ph.: A survey of algorithms for convex multicommodity flow problems. *Manag. Sci.* 46, 126–147 (2000)
28. Rockafellar, R.T.: Monotone Operators and the Proximal Point Algorithm. *SIAM J. Control Optim.* 14, 877–898 (1976)
29. Saunders, M.A., Tomlin, J.A.: Solving regularized linear programs using barrier methods and KKT systems. Technical Report SOL 96-4. Dept. of Operations Research, Stanford University (1996)
30. Setiono, R.: Interior proximal point algorithm for linear programs. *J. Optim. Theory Appl.* 74 425–444 (1992)
31. Vanderbei, R.J.: Symmetric quasidefinite matrices. *SIAM J. Optim.* 5, 100–113 (1995)
32. Wright, S.J.: *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia (1996)