On geometrical properties of preconditioners in IPMs for classes of block-angular problems

Jordi Castro
Dept. of Stat. and Oper. Res.
Universitat Politècnica de Catalunya
jordi.castro@upc.edu

Stefano Nasini
Dept. of Prod., Tech. and Oper. Mngmt.
IESE Business School
snasini@iese.edu

Research Report UPC-DEIO DR 2016-03
February 2016

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

ON GEOMETRICAL PROPERTIES OF PRECONDITIONERS IN IPMS FOR CLASSES OF BLOCK-ANGULAR PROBLEMS*

JORDI CASTRO † AND STEFANO NASINI ‡

Abstract. One of the most efficient interior-point methods for some classes of block-angular structured problems solves the normal equations by a combination of Cholesky factorizations and preconditioned conjugate gradient for, respectively, the block and linking constraints. In this work we show that the choice of a good preconditioner depends on geometrical properties of the constraints structure. In particular, it is seen that the principal angles between the subspaces generated by the diagonal blocks and the linking constraints can be used to estimate ex-ante the efficiency of the preconditioner. Numerical validation is provided with some generated optimization problems. An application to the solution of multicommodity network flow problems with nodal capacities and equal flows of up to 127 million of variables and up to 7.5 million of constraints is also presented.

Key words. interior-point methods, structured problems, preconditioned conjugate gradient, principal angles, large-scale optimization

AMS subject classifications. 90C06, 90C08, 90C51

1. Introduction.

Many real-world optimization problems exhibit a primal block-angular structure, where decision variables and constraints can be grouped in different blocks which are dependent due to some linking constraints Applications of this class of problems can be found, for instance, in multicommodity telecommunications networks, statistical data protection, multistage control and planning, and complex networks.

Solution strategies to deal with this class of problems can be broadly classified into simplex-based methods [14, 22], decomposition methods [16, 1, 2, 26], approximation methods [5], and interior-point methods [10, 20] One of the most efficient interior-point methods (IPMs) for some classes of block-angular problems solves normal equations by a combination of Cholesky factorizations for the block constraints and preconditioned conjugate gradient (PCG) iterations for the linking constraints [10, 11]. The spectral radius of a certain matrix in the preconditioner, which is always in $[0, 1)$ plays an important role in the efficiency of this approach. It was observed that for separable convex problems with nonzero Hessians this spectral radius is reduced, and the PCG becomes more efficient [13]. When the spectral radius approaches 1, switching to a suitable preconditioner may be an efficient alternative [7]. It is worth noting that computing approximate Newton directions by PCG does not destroy the good convergence properties of IPMs, as shown in [19]. There is an extensive literature on the use of PCG within IPMs for other types of problems (e.g., [3, 4, 9, 17, 24, 27] to mention a few).

However, it is not yet clear why for some classes of block-angular problems the above approach may be very efficient (see, for instance, the results of [13, 12]), while it may need a large number of PCG iterations in others. The spectral radius may be used to monitor the good or bad behaviour, but an ex-ante explanation has to be found in the structural information of the block-angular constraints matrix. The purpose of this paper is to provide such a explanation by considering geometrical properties of the primal block-angular matrix structure which, in addition, may be used to improve the preconditioning in some instances. It will be shown that the principal angles between the subspaces generated by the diagonal blocks and the ones generated by the linking constraints play an important role, explaining the efficiency of the approach and providing a proper choice of the preconditioner.

This paper is organized as follows. Section 2 outlines the specialized IPM for primal block-angular problems. Section 3 analyses the quality of preconditioning techniques based on geometrical and spectral properties. This analysis is used to obtain new complementary preconditioners to deal with orthogonality and collinearity of the subspaces generated by the diagonal blocks and the ones generated by the linking constraints. Section 4 provides a numerical validation of the analysis in previous section. This is based on both a simulation analysis of the effect of geometrical relations on PCG iterations—Subsection 4.1—and a real world application to multicommodity network flow problems with nodal capacities—Subsection 4.2—, where the analyzed geometrical properties can be fully exploited.

*This work has been supported by grant MTM2012-31440 of the Spanish Scientific and Technical Research Program.

†Dept. of Statistics and Operations Research,Universitat Politècnica de Catalunya, Jordi Girona 1–3, 08034 Barcelona, Catalonia. jordi.castro@upc.edu

‡Dept. of Production, Technology and Operations Management, IESE Business School, University of Navarra, Av. Pearson 21, 08034 Barcelona, Catalonia, Spain. snasini@iese.edu

2. Outline of the IPM for primal block-angular problems. The primal block-angular formulation dealt with by the algorithm is

$$\min \quad \sum_{i=0}^{k} (c^i)^\top x^i \tag{2.1a}$$

$$\text{subject to} \quad \begin{bmatrix} N_1 & & & & \\ & N_2 & & & \\ & & \ddots & & \\ & & & N_k & \\ L_1 & L_2 & \dots & L_k & I \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^k \\ x^0 \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 \\ \vdots \\ b^k \\ b^0 \end{bmatrix} \tag{2.1b}$$

$$0 \le x^i \le u^i \quad i = 0, \dots, k. \tag{2.1c}$$

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 $u^i \in \mathbb{R}^{n_i}, i = 0, \dots, k$. Problems with equality linking constraints can be formulated by setting $u^0 = 0$. The total number of constraints and variables of (2.1) is thus, respectively, $\widetilde{m} = l + \sum_{i=1}^{k} m_i$ and $\widetilde{n} = l + \sum_{i=1}^{k} n_i$. Formulation (2.1) is a very general model which accommodates to several block-angular problems.

In this work we consider the specialized IPM of [10, 11]. Briefly, this approach requires the solution at each interior-point iteration of the normal equations system $A\Theta A^\top \Delta y = g$, where $A \in \mathbb{R}^{\widetilde{m} \times \widetilde{n}}$ is the constraints matrix of (2.1b); $\Theta = (W(U-X)^{-1} + ZX^{-1})^{-1} \in \mathbb{R}^{\widetilde{n}}$ is a diagonal matrix computed from the values $(x, w, z) \in \mathbb{R}^{3\widetilde{n}}$ of the current primal-dual point—w and z being the Lagrange multipliers associated to, respectively, upper and lower bounds; $\Delta y \in \mathbb{R}^{\widetilde{m}}$ is the direction of movement for the Lagrange multipliers of equality constraints y; and $g \in \mathbb{R}^{\widetilde{m}}$ is some right-hand-side. A derivation of the normal equations can be found in [28].

Exploiting the block structure of A and Θ we have

$$\begin{aligned} A\Theta A^\top \Delta y &= \left[\begin{array}{ccc|c} N_1\Theta_1 N_1^\top & & & N_1\Theta_1 L_1^\top \\ & \ddots & & \vdots \\ & & N_k\Theta_k N_k^\top & N_k\Theta_k L_k^\top \\ \hline L_1\Theta_1 N_1^\top & \dots & L_k\Theta_k N_k^\top & \Theta_0 + \sum_{i=1}^{k} L_i\Theta_i L_i^\top \end{array}\right] \Delta y \\ &= \begin{bmatrix} B & C \\ C^\top & D \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} = g, \end{aligned} \tag{2.2}$$

where Δy_1 and Δy_2 are the components of Δy associated to, respectively, block and linking constraints; and $\Theta_i = (W_i(U_i - X_i)^{-1} + Z_i X_i^{-1})^{-1}$, $i = 0, \dots, k$, are the blocks of Θ. By eliminating Δy_1 from the first group of equations of (2.2), we obtain

$$(D - C^\top B^{-1} C)\Delta y_2 = g_1 \tag{2.3a}$$

$$B\Delta y_1 = g_2, \tag{2.3b}$$

for a proper partition of the right-hand side into g_1 and g_2. The specialized IPM for this class of problems solves (2.3) by a combination of k Cholesky factorizations, for the systems involving B and PCG for (2.3a). Indeed, matrix $D - C^\top B^{-1} C \in \mathbb{R}^{l \times l}$ of (2.3a), whose dimension is the number of linking constraints, is symmetric and positive definite, since it is the Schur complement of the normal equations (2.2), which are symmetric and positive definite. System (2.3a) can thus be solved by PCG. A good preconditioner is instrumental. $D - C^\top B^{-1} C$ is a *P-regular splitting*, i.e., it is symmetric and positive definite, D is nonsingular and $D + C^\top B^{-1} C$ is positive definite. Therefore the *P-regular splitting theorem* [25] guarantees that

$$0 < \rho(D^{-1}(C^\top B^{-1} C)) < 1, \tag{2.4}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix (i.e., the maximum absolute eigenvalue). This allows us to compute the inverse of $D - C^\top B^{-1} C$ as the following infinite power series (see [10, Prop.

4] for a proof).

$$(D - C^\top B^{-1} C)^{-1} = \left(\sum_{i=0}^{\infty} (D^{-1}(C^\top B^{-1} C))^i \right) D^{-1}. \tag{2.5}$$

A preconditioner is thus obtained by truncating the infinite power series (2.5) at some term. The more the terms included, the better the preconditioner will be, at the expense of increasing the execution time of each PCG iteration. If we only include the first term, or the first and second terms, of the infinite power series (2.5) the resulting preconditioners will be, respectively, D^{-1} or $(I + D^{-1}(C^\top B^{-1} C))D^{-1}$. As shown in [12], in most test problems D^{-1} was the best option for the tradeoff between being a good and an efficient preconditioner. Thus, in this work we will focus on the first term preconditioner D^{-1}. Although the expected performance for a general primal block-angular matrix is problem dependent, the effectiveness of the preconditioner obtained by truncating the infinite power series (2.5) is governed by the spectral radius $\rho(D^{-1}(C^\top B^{-1} C))$—the farther from 1, the better [13]—and the structure of D, as each PCG iteration requires the solution of a system with this matrix.

The spectral radius tends to approach 1 when the IPM is close to the optimal solution. For this reason a hybrid preconditioner was introduced in [7]: the power series preconditioner (either D^{-1} or $(I + D^{-1}(C^\top B^{-1} C))D^{-1}$) is used in the initial IPM iterations, switching to the *splitting preconditioner* [24] when the former becomes inefficient. The above hybrid scheme is supported by the known good behaviour of the splitting preconditioner near the solution of the linear optimization problem. Briefly, given the normal equations matrix $A\Theta A^\top$, the splitting preconditioner is based on a partition of the columns of A into basic and nonbasic columns, forming the nonsingular basic matrix $A_B \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$ and the nonbasic one $A_N \in \mathbb{R}^{\tilde{m} \times (\tilde{n} - \tilde{m})}$, respectively. Applying the same partition to Θ, the normal equations matrix can be rewritten as

$$A\Theta A^\top = A_B \Theta_B A_B^\top + A_N \Theta_N A_N^\top. \tag{2.6}$$

The splitting preconditioner is thus $\Theta_B^{-1/2} A_B^{-1}$. The symmetric application of this preconditioner to $A\Theta A^\top$ gives:

$$\begin{aligned} (\Theta_B^{-1/2} A_B^{-1})(A\Theta A^\top)(\Theta_B^{-1/2} A_B^{-1})^\top &= \Theta_B^{-1/2} A_B^{-1}(A_B \Theta_B A_B^\top + A_N \Theta_N A_N^\top) A_B^{-T} \Theta_B^{-1/2} \\ &= I + (\Theta_B^{-1/2} A_B^{-1} A_N \Theta_N^{1/2})(\Theta_B^{-1/2} A_B^{-1} A_N \Theta_N^{1/2})^\top. \end{aligned} \tag{2.7}$$

Sufficiently close to an optimal solution, with a suitable choice of the columns of A_B, the diagonal entries of Θ_B^{-1} and Θ_N are very small and $(\Theta_B^{-1/2} A_B^{-1} A_N \Theta_N^{-1/2})$ approaches the zero matrix. Some strategies for identifying a suitable matrix A_B are discussed in [24, 7].

This specialized IPM has been recently implemented in a software package called BlockIP [12], that will be used for the computational results of this paper.

3. Analysis of preconditioning techniques for block-angular problems. The main source of difficulty in solving a primal block-angular problem of the form (2.1) is the presence of linearly independent linking constraints, $L_1 x_1 + L_2 x_2 + \ldots + L_k x_k \leq b_0$. Consider the lucky—and unlikely—case where A is such that for every $i = 1 \ldots k$ each row vector of L_i belongs to the column space of $N_i^\top$ (linearly dependent linking constraints). By the definition of B, C and D in (2.2), we may write

$$\begin{aligned} C^\top B^{-1} C &= \sum_{i=1}^{k} L_i \Theta_i N_i^\top \, (N_i \Theta_i N_i^\top)^{-1} \, N_i \Theta_i L_i^\top \\ &= \sum_{i=1}^{k} L_i \Theta_i^{1/2} P_i \Theta_i^{1/2} L_i^\top \end{aligned} \tag{3.1}$$

where

$$P_i = \Theta_i^{1/2} N_i^\top \, (N_i \Theta_i N_i^\top)^{-1} \, N_i \Theta_i^{1/2} \quad i = 1, \ldots, k \tag{3.2}$$

is the projection operator onto $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$, the range or column space of $\Theta_i^{1/2} N_i^\top$. If each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$ (which is equivalent to say that the row vectors of L_i might be written as a linear combination of the row vectors of N_i) then $P_i \Theta_i^{1/2} L_i^\top = \Theta_i^{1/2} L_i^\top$, as P_i is the

identity operator of the subspace generated by the columns of $\Theta_i^{1/2} N_i^\top$, and

$$
\begin{aligned}
D - C^\top B^{-1} C &= D - \sum_{i=1}^{k} L_i \Theta_i^{1/2} P_i \Theta_i^{1/2} L_i^\top \\
&= D - \sum_{i=1}^{k} L_i \Theta_i L_i^\top = \Theta_0.
\end{aligned} \tag{3.3}
$$

On the contrary, if each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{N}(N_i \Theta_i^{1/2})$, the null space of $N_i \Theta_i^{1/2}$, then $P_i \Theta_i^{1/2} L_i^\top = 0$ and

$$
D - C^\top B^{-1} C = D - \sum_{i=1}^{k} L_i \Theta_i^{1/2} P_i \Theta_i^{1/2} L_i^\top = D \tag{3.4}
$$

Thus Θ_0^{-1} and D^{-1} would be the inverse (i.e., the exact preconditioners) of $D - C^\top B^{-1} C$ when, for $i = 1 \dots k$, each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to either $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$ or $\mathcal{N}(N_i \Theta_i^{1/2})$, respectively. Needless to say, these two extreme cases will rarely appear in a real problem; for instance, when $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$ the linking constraints are redundant and can be removed, obtaining a block-separable problem. However, as seen in next subsection, they allow to decide whether D or Θ_0 will be a good preconditioner for (2.3a), and which of them will theoretically behave better.

3.1. Geometrical and spectral properties. According to previous discussion, the goodness of the approximation of Θ_0^{-1} and D^{-1} to $(D - C^\top B^{-1} C)^{-1}$ might be measured by the *principal angles* between the range spaces of $\Theta_i^{1/2} N_i^\top$ and $\Theta_i^{1/2} L_i^\top$, for $i = 1, \dots, k$. The principal angles provide information about the relative position of two subspaces of an inner product space. Consider two subspaces $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$ of $\mathbb{R}^n$, with $\dim \mathcal{L}_\Theta = l$, $\dim \mathcal{N}_\Theta = m$, and $q = \min\{l, m\}$. The principal angles between $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$, denoted as $0 \le \gamma_1 \le \dots \le \gamma_q \le \pi/2$, are obtained by solving for $j = 1, \dots, q$ this sequence of optimization problems [6][18, Chapter 12]:

$$
\cos(\gamma_j) = \max_{u,v} u^\top v \tag{3.5}
$$

$$
\text{subject to} \|u\| = 1, \ \|v\| = 1 \tag{3.6}
$$

$$
u \in \mathcal{L}_\Theta, \ v \in \mathcal{N}_\Theta \tag{3.7}
$$

$$
u^\top v_k = 0, \ u^\top u_k = 0, \ k = 1 \dots j-1, \tag{3.8}
$$

$u_j \in \mathbb{R}^n$ and $v_j \in \mathbb{R}^n$ being the optimal vectors. In other words, for $j = 1$ the procedure finds the unit vectors $u_1 \in \mathcal{L}_\Theta$ and $v_1 \in \mathcal{N}_\Theta$ which minimize the angle between them—named γ_1. For $j > 1$, the procedure considers the orthogonal complements of $\text{span}\{u_1, \dots, u_{j-1}\}$ in $\mathcal{L}_\Theta$ and of $\text{span}\{v_1, \dots, v_{j-1}\}$ in $\mathcal{N}_\Theta$, and computes a new pair of vectors u_j, v_j in the above subspaces minimizing the angle γ_j. In our case, we have that $\mathcal{L}_\Theta = \mathcal{R}(\Theta_i^{1/2} L_i^\top)$ and $\mathcal{N}_\Theta = \mathcal{R}(\Theta_i^{1/2} N_i^\top)$. The vectors $\{u_1, \dots, u_q\}$ and $\{v_1, \dots, v_q\}$ are called *principal vectors*, associated to principal angles $\{\gamma_1, \dots, \gamma_q\}$. The principal angles between subspaces can be graphically depicted as in Figure 1. If $q = m = l$, the distance between the equidimensional subspaces $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$ is defined as $\sin \gamma_q = \sqrt{1 - \cos^2 \gamma_q}$ [18, Chapter 2].

Principal angles between two subspaces can be computed by the singular value decomposition, as shown by next theorem from [6] (procedure also described in [18, Chapter 12]):

THEOREM 3.1. *Let the columns of matrices $Q_L \in \mathbb{R}^{n \times l}$ and $Q_N \in \mathbb{R}^{n \times m}$ form orthonormal bases for the subspaces $\mathcal{L}$ and $\mathcal{N}$, correspondingly. Principal vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$ must verify $u = Q_L \widetilde{u}$ and $v = Q_N \widetilde{v}$, where $\widetilde{u} \in \mathbb{R}^l$ and $\widetilde{v} \in \mathbb{R}^m$ are left and right singular vectors of $Q_L^\top Q_N$, associated to the singular value $\cos(\gamma(u, v))$, that is to say, $(Q_L^\top Q_N)\widetilde{v} = \cos(\gamma(u, v))\, \widetilde{u}$.*

The algorithm of Figure 2 outlines how to compute the principal angles of column spaces of matrices $L^\top \in \mathbb{R}^{n \times l}$ and $N^\top \in \mathbb{R}^{n \times m}$. The algorithm is just provided for completeness; as it will be shown later, it does not need to be used in practice. Indeed, its use would be prohibitive since the required singular value decomposition may be computationally very expensive for large optimization problems.

As $\Theta_i, i = 1, \dots, k$, are different at each interior-point iteration, the principal angles between the subspaces $\mathcal{L}_{\Theta_i}$ and $\mathcal{N}_{\Theta_i}$ also change. Accordingly, the goodness of the approximation of Θ_0^{-1} and D^{-1} to $(D - C^\top B^{-1} C)^{-1}$ dynamically changes along the interior-point iterations. Proposition 3.2

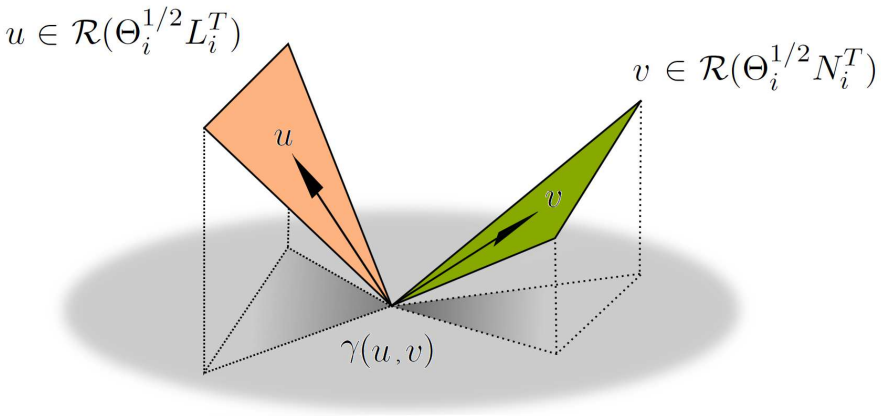

FIG. 1. *Angles between subspaces.*

Algorithm *Computation of principal angles*$(L^\top \in \mathbb{R}^{n\times l}, N^\top \in \mathbb{R}^{n\times m})$
$q = \min(m, l)$
//Compute orthonormal basis of $\mathcal{R}(L^\top)$ and $\mathcal{R}(N^\top)$ using QR factorizations
$Q_L \leftarrow QR(L^\top)$, $Q_L \in \mathbb{R}^{n\times l}$
$Q_N \leftarrow QR(N^\top)$, $Q_N \in \mathbb{R}^{n\times m}$
//Compute singular value decomposition of $Q_L^\top Q_N$: $U\Sigma V^\top = Q_L^\top Q_N$
$[U, \Sigma, V] \leftarrow SVD(Q_L^\top Q_N)$, $U \in \mathbb{R}^{l\times q}$, $V \in \mathbb{R}^{m\times q}$ orthonormal, $\Sigma \in \mathbb{R}^{q\times q}$
// Compute vector of principal angles γ
$\gamma \leftarrow \arccos(\mathrm{diag}(\Sigma))$, $\gamma \in \mathbb{R}^q$
// Compute principal vectors W_L and W_N
$W_L \leftarrow Q_L U$, $W_L \in \mathbb{R}^{n\times q}$
$W_N \leftarrow Q_N V$, $W_N \in \mathbb{R}^{n\times q}$
Return: γ, W_L, W_N
End_algorithm

FIG. 2. *Algorithm to compute principal angles between column spaces of* $L^\top$ *and* $N^\top$.

below provides a lower bound of the cosine of principal angles of $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$ as a function of the principal angles of $\mathcal{L}$ and $\mathcal{N}$.

PROPOSITION 3.2. *Consider the subspaces* $\mathcal{L} = \mathcal{R}(L^\top) \subseteq \mathbb{R}^n$ *and* $\mathcal{N} = \mathcal{R}(N^\top) \subseteq \mathbb{R}^n$ *and their image sets* $\mathcal{L}_\Theta = \mathcal{R}(\Theta^{1/2} L^\top) \subseteq \mathbb{R}^n$ *and* $\mathcal{N}_\Theta = \mathcal{R}(\Theta^{1/2} N^\top) \subseteq \mathbb{R}^n$ *under the linear transformation* $\Theta^{1/2}$*, where* Θ *is a positive diagonal matrix. Let* $\gamma(u, v)$ *be the first principal angle between* $\mathcal{L}$ *and* $\mathcal{N}$*, associated to principal vectors* u *and* v*; and* γ_Θ *the first principal angle between* $\mathcal{L}_\Theta$ *and* $\mathcal{N}_\Theta$*. If* $\mathcal{L}$ *and* $\mathcal{N}$ *are not orthogonal, then*

$$\cos(\gamma_\Theta) \geq \max\left\{0, \frac{\Theta_c}{\Theta_{max}} \cos\left(\gamma(u, v)\right)\right\}, \tag{3.9}$$

where $\Theta_{\min}$ *and* $\Theta_{\max}$ *are the smallest and largest diagonal components of* Θ*, and* $\Theta_c \in [-\Theta_{\max}, \Theta_{max}]$*. If in addition* $u_i v_i \geq 0$ *for all* $i = 1, \ldots, n$ *then we have*

$$\cos(\gamma_\Theta) \geq \frac{\Theta_{\min}}{\Theta_{\max}} \cos\left(\gamma(u, v)\right) > 0. \tag{3.10}$$

Proof. Let $\gamma\left(\frac{\Theta^{1/2}u}{\|\Theta^{1/2}u\|}, \frac{\Theta^{1/2}v}{\|\Theta^{1/2}v\|}\right)$ be the angle between the unitary transformed vectors $\frac{\Theta^{1/2}u}{\|\Theta^{1/2}u\|} \in \mathcal{L}_\Theta$ and $\frac{\Theta^{1/2}v}{\|\Theta^{1/2}v\|} \in \mathcal{N}_\Theta$. From the definition of principal angles as resulting from the maximization problem (3.5)–(3.8), and since vectors $\frac{\Theta^{1/2}u}{\|\Theta^{1/2}u\|}$ and $\frac{\Theta^{1/2}v}{\|\Theta^{1/2}v\|}$ are feasible for this problem, we have

$$\cos(\gamma_\Theta) \geq \cos\left(\gamma\left(\frac{\Theta^{1/2}u}{\|\Theta^{1/2}u\|}, \frac{\Theta^{1/2}v}{\|\Theta^{1/2}v\|}\right)\right). \tag{3.11}$$

Since u, v are principal vectors, by (3.6) $\|u\| = \|v\| = 1$. Moreover, since principal angles are always in $[0, \pi/2]$, and we are assuming $\mathcal{L}$ and $\mathcal{N}$ are not orthogonal, we have

$$0 < u^\top v = \|u\| \, \|v\| \cos(\gamma(u,v)) = \cos(\gamma(u,v)) \leq 1. \tag{3.12}$$

By definition of inner product, by (3.12), using that for any square matrix M and vector w $\|Mw\| \leq \|M\| \, \|w\|$, and that $\sqrt{\Theta_{\max}} \geq \|\Theta^{1/2}\|$,

$$\begin{aligned} \cos\left(\gamma\left(\frac{\Theta^{1/2}u}{\|\Theta^{1/2}u\|}, \frac{\Theta^{1/2}v}{\|\Theta^{1/2}v\|}\right)\right) &= \frac{u^\top \Theta v}{\|\Theta^{1/2}u\| \, \|\Theta^{1/2}v\|} \geq \frac{u^\top \Theta v}{\|\Theta^{1/2}\|^2 \, \|u\| \, \|v\|} \geq \\ &\geq \frac{1}{\Theta_{\max}} \frac{u^\top \Theta v}{u^\top v} \cos(\gamma(u,v)) \geq \frac{u^\top \Theta v}{\Theta_{\max}} \cos(\gamma(u,v)). \end{aligned} \tag{3.13}$$

Defining the set $\mathcal{P} = \{i \in \{1, \ldots, n\} : u_i v_i \geq 0\}$, since $\|u\| = \|v\| = 1$ and $u^\top v \leq 1$, we have that

$$0 \leq \sum_{i \in \mathcal{P}} u_i v_i \leq 1 \qquad \text{and} \qquad -1 \leq \sum_{i \notin \mathcal{P}} u_i v_i \leq 0. \tag{3.14}$$

Then, by (3.14),

$$u^\top \Theta v = \sum_{i \in \mathcal{P}} \Theta_i u_i v_i + \sum_{i \notin \mathcal{P}} \Theta_i u_i v_i \geq \Theta_{\max} \sum_{i \notin \mathcal{P}} u_i v_i \geq -\Theta_{\max},$$

and similarly,

$$u^\top \Theta v = \sum_{i \in \mathcal{P}} \Theta_i u_i v_i + \sum_{i \notin \mathcal{P}} \Theta_i u_i v_i \leq \Theta_{\max} \sum_{i \in \mathcal{P}} u_i v_i \leq \Theta_{\max}.$$

Then, there is a $\Theta_c \in [-\Theta_{\max}, \Theta_{\max}]$ such that

$$\Theta_c = u^\top \Theta v. \tag{3.15}$$

Applying (3.15) in (3.13), and since, by definition, $\cos(\gamma_\Theta) \geq 0$, inequality (3.9) is proved.

If $u_i v_i \geq 0$ for all $i = 1, \ldots, n$, we have

$$\frac{u^\top \Theta v}{u^\top v} \geq \frac{\Theta_{\min} u^\top v}{u^\top v} = \Theta_{\min}. \tag{3.16}$$

Applying (3.16) to the penultimate term of (3.13) we get (3.10). ∎

An important consequence of Proposition 3.2 is that we have no information about the principal angles in the final iterations, as $\Theta_{\min}/\Theta_{\max}$ can approach 0 and $\Theta_c/\Theta_{\max}$ can be any number in $[-1, 1]$ when the iterative process gets close to the optimal solution. In the worst case, it may even happen that almost-collinear subspaces $\mathcal{L}$ and $\mathcal{N}$ become almost-orthogonal for some Θ.

It is worth noting that the two extreme cases have opposite behaviour:

- If $\mathcal{L}$ and $\mathcal{N}$ are orthogonal (that is, their principal angles are $\pi/2$), then, for a diagonal Θ matrix such that some component $\Theta_{i,i}$ is very large, and for other components $j \neq i$ $\Theta_{jj} \approx 0$, the principal angles of $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$ will become close to 0. This is a limit case of (3.10).
- If $\mathcal{L} \subseteq \mathcal{N}$, then the principal angles are 0. This means that, for some matrix $Y \in \mathbb{R}^{m \times l}$, $L^\top = N^\top Y$, thus $(\Theta^{1/2} L^\top) = (\Theta^{1/2} N^\top) Y$ and the principal angles of $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$ will also be 0. However this case is not of practical interest, since if $\mathcal{L} \subseteq \mathcal{N}$ then the linking constraints are redundant and can be removed.

A downside of the use of principal angles between subspaces for this purpose appears when $\mathcal{R}(L_i^\top) = \mathbb{R}^{n_i}$, for $i = 1, \ldots, k$, such that $\mathcal{N}_i \subseteq \mathcal{L}_i$. In such case the principal angles are always zero, regardless of what $\mathcal{R}(N_i^\top)$ is. This happens, for instance, in multicommodity network flow problems [10] and edge-colored network problems [15], where $L_i = I$, for $i = 1, \ldots, k$. Since our interest is on whether linking constraints belong to $\mathcal{R}(N_i^\top)$, and not the opposite, a more sensible approach would be to provide the average of the principal angles between the l constraints of L_i and $\mathcal{R}(N_i^\top)$. This procedure has the added benefit that the computation of each single principal angle through (3.5)–(3.8) is highly simplified. In particular, denoting by $L_{i,j}$ the vector associated to the j-th constraint of L_i, (3.5)–(3.8) reduces to

$$\cos(\gamma) = \begin{array}{ll} \max\limits_{x \in \mathbb{R}^{m_i}} & c^\top x \\ \text{subject to} & x^\top N_i N_i^\top x = 1 \end{array} \qquad \text{where} \quad c = N_i \frac{L_{i,j}}{\|L_{i,j}\|},$$

whose solution can be directly computed as

$$x = (N_i N_i^\top)^{-1} \frac{c}{2\lambda} \qquad \text{where} \qquad \lambda = \frac{1}{2}\sqrt{c^\top (N_i N_i^\top)^{-1} c}.$$

As shown below, the goodness of the dynamical approximation of Θ_0^{-1} and D^{-1} to $(D - C^\top B^{-1} C)^{-1}$ along the interior-point iterations is related to the changes in the spectral radius of matrix $D^{-1}(C^\top B^{-1} C)$—which is always in $[0, 1)$ [10, Theorem 1]. Since D is the first term of the power series (2.5), the farther away from 1 is the spectral radius of $D^{-1}(C^\top B^{-1} C)$ the better is the quality of the approximation of the first few terms of (2.5). Although the particular behavior 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 Θ matrix. Next result provides a clear relationship between the spectral radius of $D^{-1}(C^\top B^{-1} C)$ and the projection operators in the subspaces $\mathcal{L}_\Theta$ and $\mathcal{N}_\Theta$.

PROPOSITION 3.3. *Let λ be an arbitrary eigenvalue of $D^{-1}(C^\top B^{-1} C)$. If each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$ then*

$$\lambda = \frac{r^\top \left(\sum_{i=1}^k L_i \Theta_i L_i^\top \right) r}{r^\top D r}, \tag{3.17}$$

where r is the corresponding eigenvector associated to λ. On the contrary, if each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{N}(N_i \Theta_i^{1/2})$, the null space of $N_i \Theta_i^{1/2}$, then

$$\lambda = 0. \tag{3.18}$$

Proof. Eigenvalue λ of $D^{-1}(C^\top B^{-1} C)$ satisfies $(C^\top B^{-1} C) r = \lambda D r$ for some eigenvector r. From the definition of C, B, D in (2.2) and projection matrices P_i in (3.2) we have

$$\left(\sum_{i=1}^k L_i \Theta_i N_i^\top \left(N_i \Theta_i N_i^\top \right)^{-1} N_i \Theta_i L_i^\top \right) r = \lambda D r$$

is equivalent to

$$(1 - \lambda) D r = \left(\Theta_0 + \sum_{i=1}^k L_i \Theta_i L_i^\top \right) r - \left(\sum_{i=1}^k L_i \Theta_i N_i^\top \left(N_i \Theta_i N_i^\top \right)^{-1} N_i \Theta_i L_i^\top \right) r,$$

which implies

$$(1 - \lambda) = \frac{r^\top \Theta_0 r + r^\top \left(\sum_{i=1}^k L_i \Theta_i^{1/2} (I - P_i) \Theta_i^{1/2} L_i^\top \right) r}{r^\top D r}. \tag{3.19}$$

From this expression and applying the definition of $W_\mathcal{R}$ and $W_\mathcal{N}$

$$W_\mathcal{N} = \sum_{i=1}^k L_i \Theta_i^{1/2} (I - P_i) \Theta_i^{1/2} L_i^\top \quad \text{and} \quad W_\mathcal{R} = \sum_{i=1}^k L_i \Theta_i^{1/2} P_i \Theta_i^{1/2} L_i^\top, \tag{3.20}$$

we find the following identities

$$\lambda = \frac{r^\top \left(\sum_{i=1}^k L_i \Theta_i L_i^\top \right) r - r^\top W_\mathcal{N} r}{r^\top D r} = \frac{r^\top W_\mathcal{R} r}{r^\top D r}. \tag{3.21}$$

It turns out that $r^\top W_\mathcal{N} r = 0$ when each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$; and $r^\top W_\mathcal{R} r = 0$ when each column vector of $\Theta_i^{1/2} L_i^\top$ belongs to $\mathcal{N}(N_i \Theta_i^{1/2})$. The proof is complete. ∎

Note that a clear lesson from Proposition 3.3 is that the spectral radius might be small even in the case of almost collinearity between the row vectors of $L_i \Theta_i^{1/2}$ and their projections into $\mathcal{R}(\Theta_i^{1/2} N_i^\top)$, when $r^\top \Theta_0 r \gg r^\top (\sum_{i=1}^k L_i \Theta_i L_i^\top) r$. This means that, although Θ_0 is in theory a better preconditioner in case of almost collinearity, in practice it can be even outperformed by D. This is consistent with

TABLE 1
CPU time of BlockIP using D^{-1} and Θ_0^{-1} preconditioners. The instances have $l = 100$ linking constraints and $k = 100$ equal diagonal block matrices $N \in \mathbb{R}^{10\times 50}$.

α	$\overline{r}$	CPU time		IPM iterations		PCG iterations	
		Θ_0^{-1}	D^{-1}	Θ_0^{-1}	D^{-1}	Θ_0^{-1}	D^{-1}
$\pi/14$	2	0.99	1.47	30	43	2.84	5.87
$\pi/10$	4	1.27	1.54	38	44	4.94	6.29
$\pi/6$	6	1.30	1.58	36	46	9.00	6.05
$\pi/2$	8	1.29	1.27	35	39	8.91	5.05

TABLE 2
CPU time of BlockIP using D^{-1} and Θ_0^{-1} preconditioners. The instances have $l = 200$ linking constraints and $k = 1000$ equal diagonal block matrices $N \in \mathbb{R}^{20\times 200}$.

α	$\overline{r}$	CPU time		IPM iterations		PCG iterations	
		Θ_0^{-1}	D^{-1}	Θ_0^{-1}	D^{-1}	Θ_0^{-1}	D^{-1}
$\pi/14$	2	191.52	547.21	28	77	1.85	7.08
$\pi/10$	4	512.61	479.12	80	61	6.40	12.05
$\pi/6$	6	496.39	574.33	64	76	17.50	11.75
$\pi/2$	8	565.52	497.19	67	70	26.29	11.85

some of the computational results of sections 4–4.2, and with Theorem 1 of [13], which stated that the spectral radius ρ of $D^{-1}(C^\top B^{-1}C)$ is bounded by

$$0 \leq \rho \leq \max_{j\in\{1,\ldots,l\}} \frac{\gamma_j}{\left(\frac{u_j}{v_j}\right)^2 \Theta_{0j} + \gamma_j} < 1, \tag{3.22}$$

where u is the eigenvector (or one of the eigenvectors) of $D^{-1}(C^\top B^{-1}C)$ for ρ; γ_j, $j = 1,\ldots,l$, and $V = [V_1 \ldots V_l]$, are respectively the eigenvalues and matrix of columnwise eigenvectors of $\sum_{i=1}^{k} L_i\Theta_i {L_i}^\top$; $v = V^\top u$ (and, abusing of notation, we assume that for $v_j = 0$, $(u_j/v_j)^2 = +\infty$). Clearly, from (3.22), the larger Θ_0, the closer ρ to 0, which it is also concluded from (3.17).

4. Numerical validation. The numerical validation of the proposed preconditioning technique is based on two types of computational experiments: i) assessing the effect of the described geometrical relations to the number of PCG iterations; ii) analyzing the global numerical performance of both preconditioning techniques, when applied to classes of multicommodity network flow problems. All the 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 multithreading capabilities.

4.1. Geometrical relations and PCG iterations. Consider two full rank matrices $N \in \mathbb{R}^{m\times n}$, $n > m$, and $Y \in \mathbb{R}^{l\times m}$, $n > l$, and let $L = YN$. The rows of $L \in \mathbb{R}^{l\times n}$ are linear combinations of the rows of N and the principal angles between the subspaces generated by $L^\top$ and $N^\top$ are zero. Each vector in $\mathcal{R}(L^\top)$ can be rotated an angle α around the i^{th} and j^{th} coordinate axes by pre-multiplying $L^\top$ by the $n \times n$ rotation matrix:

$$R_{ij}(\alpha) = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & 0 & \cdots & 0 & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos(\alpha) & 0 & \cdots & 0 & -\sin(\alpha) & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ldots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -\sin(\alpha) & 0 & \cdots & 0 & \cos(\alpha) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ldots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.1}$$

If we consider $\mathcal{H} = \{(i,j) : 1 \leq i \leq n-1, i < j \leq n\}$, the set of all possible pairs of coordinate axes, the $n(n-1)/2$ distinct $R_{ij}(\alpha)$ rotation matrices may be concatenated in some order to produce a new rotation matrix such as $\prod_{(i,j)\in\mathcal{S}} R_{ij}(\alpha)$, where $\mathcal{S} \subseteq \mathcal{H}$. (Rotations in three dimensions and higher do not commute, so that different orderings give different rotations.)

We are interested in showing the performance of the PCG method at each interior-point iteration when changing the geometrical relations between the diagonal block and the linking constraint

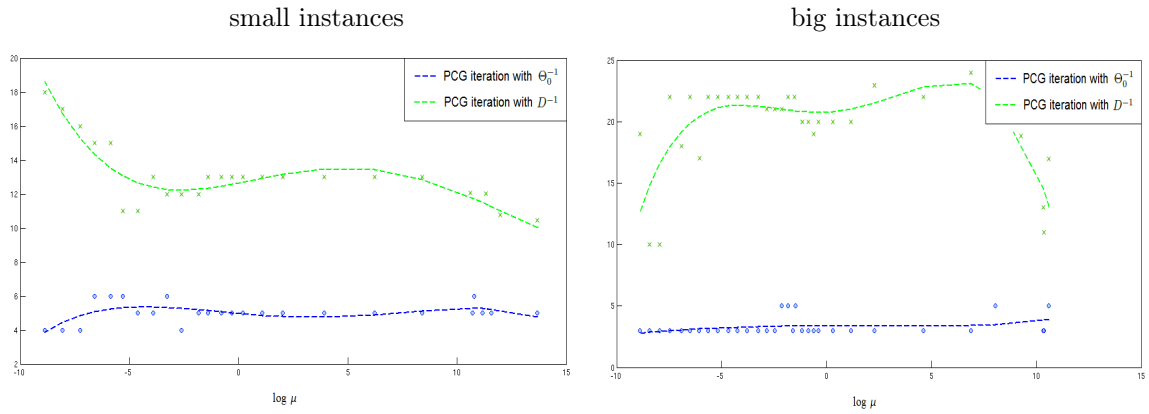

FIG. 3. *PCG iterations along the IPM iterations for* $\alpha = \pi/16$, $\overline{r} = 2$, *with preconditioners* Θ_0^{-1} *and* D^{-1}. *Plots have to be read from right to left.*

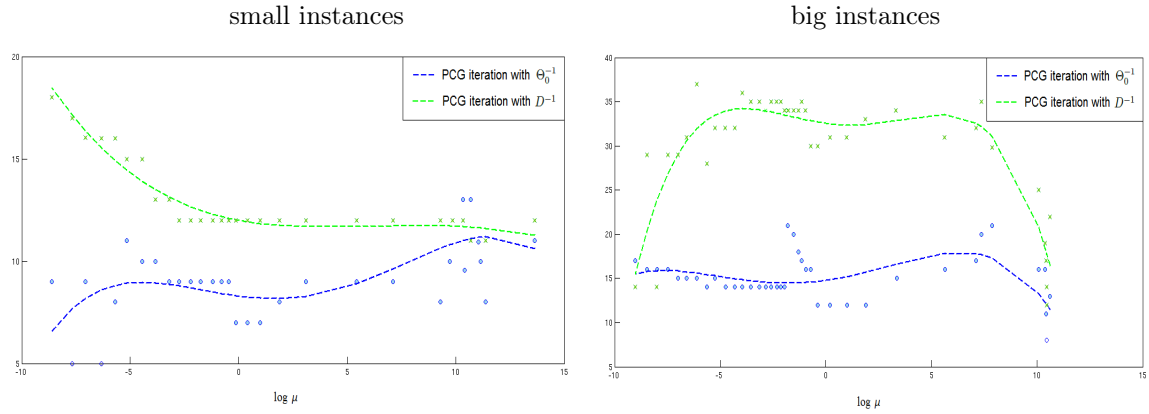

FIG. 4. *PCG iterations along the IPM iterations for* $\alpha = \pi/12$, $\overline{r} = 2$, *with preconditioners* Θ_0^{-1} *and* D^{-1}. *Plots have to be read from right to left.*

matrices, in accordance with specified rotations. To do so we randomly draw $N_i \in \mathbb{R}^{m\times n}$ and $Y_i \in \mathbb{R}^{l\times m}$, $i = 1,\ldots,k$, from a uniform probability distribution. Then $\mathcal{S} \subseteq \mathcal{H}$ is also randomly selected to compute the rotation matrix $\prod_{(i,j)\in\mathcal{S}} R_{ij}(\alpha)$. Finally, we obtain the linking constraint matrices $L_i = Y_i N_i \left(\prod_{(i,j)\in\mathcal{S}} R_{ij}(\alpha)\right)$, for $i = 1,\ldots,k$. This procedure relies on the number $\overline{r} = |\mathcal{S}|$ of concatenated rotation matrices and the angle of rotation α to evaluate the associated changes in the PCG iterations using Θ_0^{-1} and D^{-1} as preconditioners.

The BlockIP package [12] implementing the specialized interior-point method has been extended with the Θ_0^{-1} preconditioner introduced in this work. Tables 1 and 2 show the computational results obtained with BlockIP using D^{-1} and Θ_0^{-1} preconditioners. Instances of Table 1 are "small", with $l = 100$ linking constraints, and $k = 100$ equal diagonal block matrices $N \in \mathbb{R}^{10\times 50}$. For the "big" instances of Table 2 these dimensions are $l = 200$, $k = 1000$, and $N \in \mathbb{R}^{20\times 200}$. The first column of these tables show the angle α of each rotation, whereas the second column reports the number $\overline{r}$ of rotation matrices considered. The remaining columns give the CPU time, number of IPM iterations, and average number of PCG iterations per IPM iteration, for both preconditioners.

It is observed that the performance of the specialized IPM strongly relies on those angles for both the small and big instances. Indeed, the average number of PCG iterations appears to increase or decrease depending on the angles and the particular preconditioner. For Θ_0^{-1}, the smaller the angle, the better the preconditioner, whereas the opposite holds for the small instances and D (this is not so clearly observed for the big instances). This numerical evidence supports the previous discussion on the effect of the geometrical relations between the diagonal blocks and the linking constraint matrices on the quality of the preconditioner.

The plots of figures 3, 4, 5 and 6 illustrate the evolution of the number of PCG iterations (vertical axis) at each IPM iteration (horizontal axis) for both preconditioners. Small and large instances of

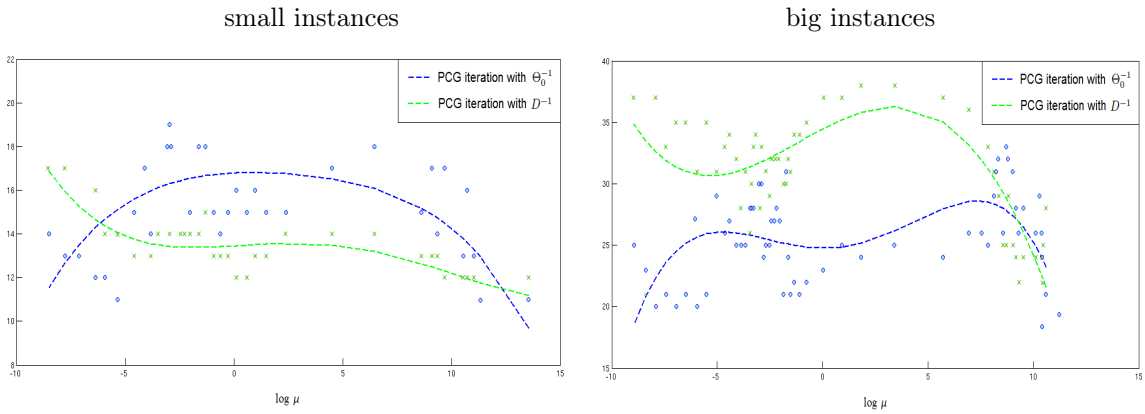

FIG. 5. *PCG iterations along the IPM iterations for $\alpha = \pi/8$, $\overline{r} = 2$, with preconditioners Θ_0^{-1} and D^{-1}. Plots have to be read from right to left.*

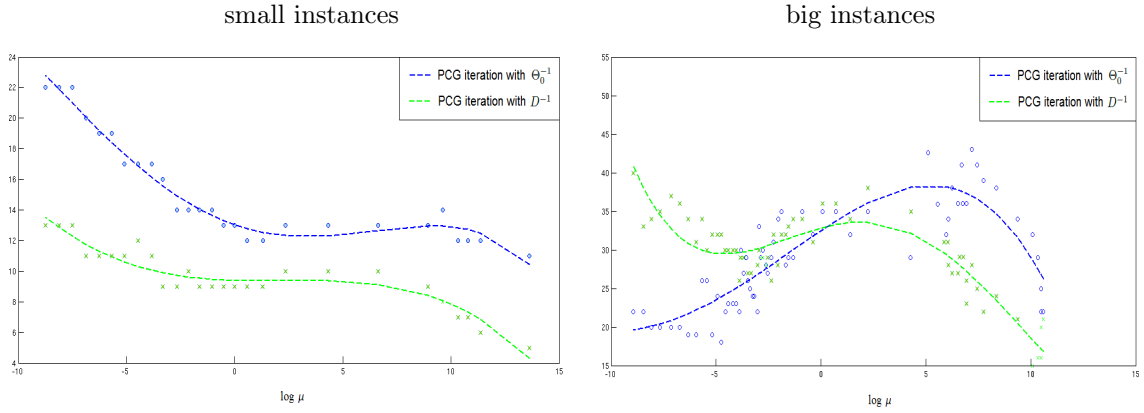

FIG. 6. *PCG iterations along the IPM iterations for $\alpha = \pi/4$, $\overline{r} = 2$, with preconditioners Θ_0^{-1} and D^{-1}. Plots have to be read from right to left.*

the same dimensions as in tables 1 and 2 have been used, but with different angles and always with $\overline{r} = 2$ rotations matrices. The horizontal axis shows the value of $\log \mu$ (the natural logarithm of the barrier parameter), such that the plots have to be read from right to left (the first and last IPM iterations correspond respectively to the rightmost and leftmost points). The blue and green dots are related to Θ_0^{-1} and D^{-1} respectively. The corresponding lines show polynomial curves of degree four which have been fitted to the observed number of PCG iterations. It is observed that for small angles (figures 3 and 4), as expected by the theory, the preconditioner D^{-1} usually requires more PCG iterations than Θ_0^{-1}; this difference increases as we approach the optimal solution, for the small instances. However, for the "larger" angles of figures 5 and 6, again in accordance with theory, preconditioner D^{-1} becomes more efficient than Θ_0^{-1} after some IPM iteration. The message should then be that, for problems with small angles, Θ_0^{-1} should be preferred in general, and D^{-1} otherwise. It is worth noting that a benefit of Θ_0^{-1} is that it is always a diagonal preconditioner, independently of the problem, unlike D^{-1}, which may require a Cholesky factorization.

4.2. Multicommodity network flow problem with nodal capacities and equal flows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a directed graph, where $\mathcal{V}$ is a set of n vertices or nodes and $\mathcal{A}$ is a set of n' arcs, and let $\mathcal{K}$ be a set of k commodities. The multicommodity network flow problem with nodal capacities (MNFPNC from now on) looks for the minimum cost routing of the flows for all the commodities from some source to some destination nodes, imposing capacities on the total outflow at some nodes $h \in \mathcal{C} \subseteq \mathcal{V}$. This problem differs from the standard multicommodity flow problem in that capacities are imposed at nodes, instead of at arcs. The node capacities constraints are

$$\sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{V}} x_{hj}^i \leq b_0^h, \qquad h \in \mathcal{C}, \tag{4.2}$$

TABLE 3
CPU time and iterations (within parenthesis) of BlockIP (using D^{-1} and Θ_0^{-1} preconditioners) and CPLEX available LP methods (Primal Simplex, Dual Simplex, Barrier Method) when solving MNFPNCs. Only 20% of nodes are constrained to have an outflow capacity, i.e., $l = 0.2n$.

n	k	n. var.	n. con.	CPLEX			BlockIP	
				Primal Simplex	Dual Simplex	Barrier	Θ_0^{-1}	D^{-1}
150	200	1788000	29830	57.5 (997332)	5.5 (85732)	5.8 (13)	5.7 (29)	5.6 (29)
150	400	3576000	59630	242.2 (428032)	14.0 (163391)	12.9 (14)	12.2 (30)	12.1 (30)
150	600	5364000	89430	362.2 (444258)	25.9 (249271)	19.7 (15)	19.6 (31)	18.8 (31)
150	800	7152000	119230	590.5 (575356)	38.0 (329632)	30.5 (16)	24.1 (29)	26.8 (32)
300	200	7176000	59860	459.3 (4676110)	27.5 (203873)	29.8 (15)	30.3 (30)	32.2 (32)
300	400	14352000	119660	> 3600 (8306650)	69.5 (415156)	57.2 (15)	57.4 (30)	59.1 (30)
300	600	21528000	179460	> 3600 (8453150)	112.7 (571964)	102.7 (17)	99.0 (31)	94.9 (30)
300	800	28704000	239260	> 3600 (9071240)	178.5 (784104)	140.3 (18)	123.1 (30)	133.8 (31)

TABLE 4
CPU time and iterations (within parenthesis) of BlockIP (using D^{-1} and Θ_0^{-1} preconditioners) and CPLEX available LP methods (Primal Simplex, Dual Simplex, Barrier Method) when solving MNFPNCs. All the nodes have an outflow capacity, i.e., $l = n$.

n	k	n. var.	n. con.	CPLEX			BlockIP	
				Primal Simplex	Dual Simplex	Barrier	Θ_0^{-1}	D^{-1}
150	200	1788000	29950	75.1 (967177)	6.7 (86446)	11.8 (17)	5.9 (29)	5.8 (29)
150	400	3576000	59750	257.0 (607837)	18.6 (163574)	23.8 (17)	14.0 (32)	12.9 (31)
150	600	5364000	89550	426.9 (729617)	32.3 (249227)	33.6 (17)	21.9 (34)	20.0 (33)
150	800	7152000	119350	585.4 (476154)	52.8 (330616)	52.0 (19)	29.9 (33)	27.9 (32)
300	200	7176000	60100	682.9 (4310028)	28.3 (203780)	85.4 (16)	54.9 (33)	30.4 (34)
300	400	14352000	119900	> 3600 (8226752)	72.2 (413460)	120.3 (13)	78.9 (37)	66.5 (33)
300	600	21528000	179700	> 3600 (8489927)	118.5 (567590)	205.6 (15)	140.5 (32)	140.1 (33)
300	800	28704000	239500	> 3600 (8408703)	181.6 (779265)	284.7 (16)	141.2 (32)	140.8 (40)

where b_0^h denotes the capacity of node $h \in \mathcal{C}$. MNFPNC matches the standard formulation (2.1) of primal block-angular problems, by considering $m_i = n$, $n_i = n'$, $N_i = N \in \mathbb{R}^{(n-1)\times n'}$ for all $i = 1, \ldots, k$, where N is the node-arc incidence matrix associated to $\mathcal{G}$; and $l = |\mathcal{C}|$, $L_i = L \in \mathbb{R}^{l\times n'}$ for all $i = 1, \ldots, k$, are derived from N by considering only positive coefficients to obtain (4.2).

Tables 3 and 4 report two computational experiments associated to MNFPNCs of different sizes. The eight instances of Table 3 correspond to problems where $l = 0.2n$ (i.e., only 20% of nodes are constrained to have an outflow capacity), whereas the eight instances of Table 4 correspond to problems where $l = n$ (all the nodes have capacities). For each instance, the tables provide the number of nodes n, number of commodities k, the total number of variables ("n.var"), total number of constraints ("n.con."), and the CPU time and (within parentheses) total number of iterations for all the solvers considered: primal simplex, dual simplex and barrier for CPLEX; and BlockIP with both Θ_0^{-1} and D^{-1} preconditioners. A time limit of 3600 seconds was considered.

It can be seen from tables 3 and 4 that the CPU time associated to the primal simplex is always far greater than the ones of the other solvers. The dual simplex is quite competitive and in some instances outperforms CPLEX barrier method and BlockIP. BlockIP with both preconditioners is in general more efficient than CPLEX barrier, although it requires more IPM iterations because—since it uses PCG—it computes Newton directions instead of second or higher order ones. The network size n does not seem to have a substantial effect on the comparative efficiency of the five solvers. Instead the increase in the number of linking constraints (either $l = 0.2n$, in Table 3 or $l = n$, in Table 4) almost double the CPU time of the CPLEX barrier method, whereas slightly affects the specialized IPM.

Let us consider a slight modification of the MNFPNCs, obtained by introducing to each commodity a set of equal flow constraints, i.e. constraints requiring that each arc in a specified set $\mathcal{R}_r$ must carry the same amount of flow, for every group of arcs $r \in R$; that is, $x^i_{a_j} = x^i_{a_h}$, for $i = 1 \ldots k$, $a_j, a_h \in \mathcal{R}_r$, $r = 1 \ldots R$ (where a_j, a_h denote two particular arcs in the network). These constraints arose while modeling some real-life problems, such as water resource system management [21]. We call this problem *multicommodity equal flow problem with nodal capacities* (MEFPNC from now on). Here we are considering MEFPNCs of different sizes with $R = \mathrm{EF} \cdot n$ groups of arcs having the same flows per each commodity and $|\mathcal{R}_r| = \mathrm{EF} \cdot 100$ number of arcs in each group $r = 1 \ldots R$, where EF is a given parameter to control the number of equal flow constraints.

Tables 5 and 6 report two computational experiments associated to these MEFPNCs. As before, the eight instances of Table 5 correspond to problems with $l = 0.2n$, whereas $l = n$ in Table 6. The parameter EF was set to 0.1 in these instances. The meaning of the columns is the same as in tables

TABLE 5

CPU time and iterations (within parenthesis) of BlockIP (using D^{-1} and Θ_0^{-1} preconditioner) and CPLEX available LP methods (Primal Simplex, Dual Simplex, Barrier Method) when solving MEFPNCs. Only 20% of nodes are constrained to have an outflow capacity, i.e., $l = 0.2n$.

n	k	n. var.	n. con.	CPLEX			BlockIP	
				Primal Simplex	Dual Simplex	Barrier	Θ_0^{-1}	D^{-1}
150	200	1788000	56830	59.3 (989973)	5.9 (90369)	7.7 (16)	6.5 (30)	6.8 (32)
150	400	3576000	113630	239.0 (516211)	19.5 (222725)	17.6 (18)	11.5 (32)	11.5 (33)
150	600	5364000	170430	514.1 (563073)	33.9 (280339)	42.2 (27)	25.4 (34)	24.0 (33)
150	800	7152000	227230	635.4 (650626)	52.0 (406268)	54.1 (26)	30.9 (32)	30.5 (32)
300	200	7176000	113860	1357.0 (958439)	37.9 (232976)	44.9 (20)	39.6 (37)	34.8 (34)
300	400	14352000	227660	> 3600.0 (1325388)	85.3 (456233)	118.2 (29)	70.7 (32)	70.1 (32)
300	600	21528000	341460	> 3600.0 (1213005)	142.5 (665822)	155.7 (25)	109.2 (33)	108.7 (34)
300	800	28704000	455260	> 3600.0 (1179502)	204.8 (860729)	239.8 (30)	152.8 (34)	150.8 (34)

TABLE 6

CPU time and iterations (within parenthesis) of BlockIP (using D^{-1} and Θ_0^{-1} preconditioner) and CPLEX available LP methods (Primal Simplex, Dual Simplex, Barrier Method) when solving MEFPNCs. All the nodes have an outflow capacity, i.e., $l = n$.

n	k	n. var.	n. con.	CPLEX			BlockIP	
				Primal Simplex	Dual Simplex	Barrier	Θ_0^{-1}	D^{-1}
150	200	1788000	56950	79.0 (2320414)	7.5 (86446)	11.3 (16)	5.9 (32)	5.0 (32)
150	400	3576000	113750	296.9 (964870)	26.7 (163574)	26.7 (18)	15.1 (33)	14.6 (33)
150	600	5364000	170550	619.9 (1839715)	43.0 (230553)	96.7 (27)	29.6 (34)	26.5 (33)
150	800	7152000	227350	891.7 (1061339)	74.7 (249227)	68.0 (22)	35.8 (33)	34.5 (32)
300	200	7176000	114100	820.8 (1184579)	36.9 (330616)	87.6 (18)	51.1 (36)	43.7 (34)
300	400	14352000	227900	> 3600.0 (4906436)	88.6 (203780)	232.8 (22)	98.8 (34)	80.2 (34)
300	600	21528000	341700	> 3600.0 (4701108)	177.3 (413460)	288.5 (21)	125.5 (35)	119.0 (36)
300	800	28704000	455500	> 3600.0 (4904433)	364.9 (1278588)	480.1 (25)	240.0 (35)	206.6 (34)

3 and 4. It is worth to mention that, for every set $\mathcal{R}_r = \{a_1, a_2, \ldots, a_{|\mathcal{R}_r|}\}$, and $i \in \mathcal{K}$, the equal flow constraints are formulated as $x^i_{a_j} = x^i_{a_{j+1}}, j = 1, \ldots, |\mathcal{R}_r| - 1$, instead of $x^i_{a_1} = x^i_{a_j}, j = 2, \ldots, |\mathcal{R}_r|$. With this formulation we avoid a "dense column" for variables $x^i_{a_1}$, which makes BlockIP more efficient. CPLEX barrier is not affected by this different reformulation due to its presolving capabilities. The principal angles between linking and block constraints are the same for both formulations, since they span the same subspace (it is easy to show that given the constraints matrices for the two formulations, one can obtain one from the other by simple linear manipulations).

The inclusion of equal flow constraints negatively affects the computational performance of all the considered solvers, though in different proportions. The dual simplex and barrier almost double its CPU times in the largest instances with respect to the MNFPNCs, whereas the ones associated to the specialized IPM slightly increase. In fact, the specialized IPM becomes the most efficient algorithm for this class of problems from $k = 400$. Also for the MEFPNCs, as it was for the MNFPNCs, the network size n does not seem to have a substantial effect on the ranking of the five solvers.

As for the two preconditioners of BlockIP, results with Θ_0^{-1} and D^{-1} are very similar in solving MEFPNCs with $l = 0.2n$, as shown in Table 5. Instead D^{-1} results to be slightly a better preconditioner when $l = n$. In both cases BlockIP outperforms the CPLEX available LP methods. It is worth noting that BlockIP uses an out-of-date sparse linear algebra package for the Cholesky factorization [23], while CPLEX implements highly tuned state-of-the-art factorization routines. Therefore, the performance of BlockIP could be significantly improved by the use of a more recent Cholesky solver.

The information of the average principal angles between the subspaces generated by the columns of $L^\top$ and $N^\top$ of the instances of tables 3–6 are reported in Table 7, differentiating for the 150 and 300 nodes instances. The first information we obtain from Table 7 is that the average principal angles are generally stable with respect to the number of nodes. We also see that average principal angles are in general far from 0, which may explain why preconditioner D^{-1} outperforms Θ_0^{-1} in general in tables 3–6.

We finally generated and solved a set of large instances, considering only the CPLEX barrier solver, and BlockIP with the two preconditioners, with a time limit of 3600 seconds. The corresponding results are reported in tables 8 and 9, for small dense and big sparse networks correspondingly. In these tables columns "density" provide the fraction "number of arcs/maximum number of arcs $(n(n-1))$"; columns "NC" show the fraction of nodal capacity constraints, i.e., "number of node capacities/n"; and columns "EF" provide the parameter considered for the equal flow constraints.

TABLE 7

Average principal angles between the subspaces generated by the columns of $L^\top$ and $N^\top$, for each instance of multi-commodity network flow problems with nodal capacities in tables 3, 4, 5 and 6.

n	average principal angles			
	Table 3	Table 4	Table 5	Table 6
150	0.8774	0.8319	0.7958	0.8225
300	0.8544	0.8297	0.7732	0.8247

TABLE 8

CPU time of BlockIP (using D^{-1} and Θ_0^{-1} preconditioner) and CPLEX barrier for MEFPNCs with small and dense networks.

Instance properties							CPU time		
n	k	density	NC	EF	n. var.	n. con.	Barrier	BlockIP Θ_0^{-1}	BlockIP D^{-1}
400	2000	0.20	0.10	0.00	63840000	798040	354.67	417.66	427.94
400	2000	0.20	0.10	0.05	63840000	958040	463.99	460.15	442.50
400	2000	0.20	0.10	0.10	63840000	1518040	699.63	550.11	531.12
400	2000	0.20	0.10	0.15	63840000	2478040	756.76	624.15	591.94
400	2000	0.20	0.10	0.20	63840000	3838040	671.20	685.70	694.74
400	2000	0.20	0.20	0.00	63840000	798080	401.50	432.82	427.62
400	2000	0.20	0.20	0.05	63840000	958080	504.83	471.38	455.42
400	2000	0.20	0.20	0.10	63840000	1518080	872.29	548.11	540.28
400	2000	0.20	0.20	0.15	63840000	2478080	704.73	621.72	617.27
400	2000	0.20	0.20	0.20	63840000	3838080	1008.74	750.51	736.59
400	2000	0.20	0.30	0.00	63840000	798120	477.00	442.55	429.09
400	2000	0.20	0.30	0.05	63840000	958120	529.90	498.52	455.59
400	2000	0.20	0.30	0.10	63840000	1518120	957.16	548.99	535.46
400	2000	0.20	0.30	0.15	63840000	2478120	865.097	640.99	611.09
400	2000	0.20	0.30	0.20	63840000	3838120	1291.96	749.23	718.98
400	2000	0.40	0.10	0.00	127680000	798040	918.03	636.26	629.18
400	2000	0.40	0.10	0.05	127680000	958040	1257.28	701.16	684.76
400	2000	0.40	0.10	0.10	127680000	1518040	1305.01	751.08	733.99
400	2000	0.40	0.10	0.15	127680000	2478040	1671.77	846.74	806.96
400	2000	0.40	0.10	0.20	127680000	3838040	1719.14	947.46	877.82
400	2000	0.40	0.20	0.00	127680000	798080	895.98	709.45	694.45
400	2000	0.40	0.20	0.05	127680000	958080	1078.44	714.52	698.59
400	2000	0.40	0.20	0.10	127680000	1518080	1842.88	764.32	768.55
400	2000	0.40	0.20	0.15	127680000	2478080	1480.11	880.15	850.83
400	2000	0.40	0.20	0.20	127680000	3838080	1665.74	925.35	892.44
400	2000	0.40	0.30	0.00	127680000	798120	1307.54	780.33	751.15
400	2000	0.40	0.30	0.05	127680000	998120	1333.14	759.27	717.97
400	2000	0.40	0.30	0.10	127680000	1598120	1842.15	760.14	730.76
400	2000	0.40	0.30	0.15	127680000	2598120	1651.68	899.92	840.19
400	2000	0.40	0.30	0.20	127680000	3838080	1765.22	929.24	879.18

The rest of columns have the same meaning as in previous tables.

Among the 60 instances of tables 8–9, CPLEX barrier resulted to be the most efficient strategy in 17 instances, whereas BlockIP outperformed CPLEX in the remaining 43. Within these latter instances, D^{-1} was a better preconditioner in 33, against the 10 corresponding to Θ_0^{-1}. However, it must be noted that in all the instances where D^{-1} behaved comparatively poorly, its total CPU time has been significantly high (> 3600). By contrast, Θ_0^{-1} has been outperformed by a more moderated comparative advantage in the aforementioned 33 instances. Unexpectedly, CPLEX barrier was the most efficient solver for some of the instances with a larger number of equal flow constraints of Table 9; this may be explained by its highly efficient factorization and presolving routines—which reduce the number of constraints and variable due to the equal flows. It is also worth to remark that BlockIP with Θ_0^{-1} was the only approach that solved all the instances within the time limit.

It must be noted that instances where CPLEX barrier resulted particularly efficient are associated to the big and sparse networks ($n = 800$, density≤ 0.04), of Table 9. Higher density matrices seem to favor BlockIP and to penalize CPLEX barrier, as observed in Table 8. However, the final behaviour is problem dependent, such that if the number of nodes is large enough, BlockIP can outperform CPLEX even for sparse networks. For instance, in a tough instance (not included in the above tables) of $n = 4000$, $k = 2000$, density$= 0.001$, NC$=0.2$, with no equal flow constraints, and without any time limit, CPLEX barrier spent about 37000 seconds (requiring 85 Gigabytes of memory), while BlockIP found a solution in 11000 seconds (using 15 Gigabytes of memory).

5. Conclusions. This work showed that the angles between the subspaces generated by the diagonal blocks and the linking constraints may explain the effectiveness of two complementary pre-

TABLE 9
CPU time of BlockIP (using D^{-1} and Θ_0^{-1} preconditioner) and CPLEX barrier for MEFPNCs with large and sparse networks.

Instance properties							CPU time		
n	k	density	NC	EF	n. var.	n. con.	Barrier	BlockIP Θ_0^{-1}	BlockIP D^{-1}
800	2000	0.02	0.10	0.00	25568000	1598080	487.21	860.17	916.84
800	2000	0.02	0.10	0.05	25568000	1918080	881.14	1001.97	984.47
800	2000	0.02	0.10	0.10	25568000	3038080	658.97	1211.42	1811.51
800	2000	0.02	0.10	0.15	25568000	4958080	1373.59	1901.41	2662.31
800	2000	0.02	0.10	0.20	25568000	7678080	2838.81	2203.28	> 3600
800	2000	0.02	0.20	0.00	25568000	1598160	565.29	987.79	> 3600
800	2000	0.02	0.20	0.05	25568000	1918160	1026.30	980.45	957.53
800	2000	0.02	0.20	0.10	25568000	3038160	1395.69	1296.92	1785.35
800	2000	0.02	0.20	0.15	25568000	4958160	1561.79	1870.65	> 3600
800	2000	0.02	0.20	0.20	25568000	7678160	3478.80	2136.35	> 3600
800	2000	0.02	0.30	0.00	25568000	1598240	1105.07	921.55	906.83
800	2000	0.02	0.30	0.05	25568000	1918240	1124.72	1018.78	988.78
800	2000	0.02	0.30	0.10	25568000	3038240	1040.20	1300.24	1888.08
800	2000	0.02	0.30	0.15	25568000	4958240	2021.58	2217.12	> 3600
800	2000	0.02	0.30	0.20	25568000	7678240	> 3600	3403.11	> 3600
800	2000	0.04	0.10	0.00	51136000	1598080	749.51	1290.82	1277.48
800	2000	0.04	0.10	0.05	51136000	1918080	1654.57	1340.82	1297.48
800	2000	0.04	0.10	0.10	51136000	3038080	1867.48	1826.23	1925.71
800	2000	0.04	0.10	0.15	51136000	4958080	1046.38	2236.19	2739.54
800	2000	0.04	0.10	0.20	51136000	7678080	1167.11	2695.33	> 3600
800	2000	0.04	0.20	0.00	51136000	1598160	846.26	1250.58	1247.37
800	2000	0.04	0.20	0.05	51136000	1918160	1814.96	1540.54	1525.10
800	2000	0.04	0.20	0.10	51136000	3038160	2146.20	1806.08	> 3600
800	2000	0.04	0.20	0.15	51136000	4958160	2709.08	2202.42	3201.02
800	2000	0.04	0.20	0.20	51136000	7678160	2267.28	2460.35	> 3600
800	2000	0.04	0.30	0.00	51136000	1598240	2473.51	1549.03	1506.47
800	2000	0.04	0.30	0.05	51136000	1918240	2579.88	1552.49	1521.36
800	2000	0.04	0.30	0.10	51136000	3038240	2765.23	1787.62	1874.88
800	2000	0.04	0.30	0.15	51136000	4958240	2712.73	2264.38	3083.99
800	2000	0.04	0.30	0.20	51136000	7678240	2378.00	2603.55	> 3600

conditioners, namely Θ_0^{-1} and D^{-1}, in the specialized IPM for block-angular problems. It was also shown that the evolution of principal angles along the IPM iterations rely on the diagonal Θ matrix. Algebraical properties of the two complementary preconditioners were supported by numerical results. We also analyzed the performance of the specialized IPM with the two preconditioners in the solution of the multicommodity network flow problem with nodal capacities and equal flows, observing that it outperforms all of the available CPLEX methods in some of the largest instances.

Θ_0^{-1} and D^{-1} have shown to be the exact preconditioners when a linking constraint belongs to respectively the range and null space of the matrix of block constraints. Since any vector defining a linking constraint can be decomposed as the sum of two orthogonal vectors belonging to the range and null space of the matrix of block constraints, it would be worth to exploit this fact to see whether a *multipreconditioner* based on the two above can be obtained, following for instance [8]. This is part of the further research to be done.

REFERENCES

[1] F. Babonneau, O. du Merle, and J.-P. Vial, *Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-ACCPM*, Oper. Res., 54 (2006), pp. 184–197.

[2] F. Babonneau and J.-P. Vial, *ACCPM with a nonlinear constraint and an active set strategy to solve nonlinear multicommodity flow problems*, Math. Prog., 120 (2009), pp. 179-210.

[3] S. Bellavia, J. Gondzio, and B. Morini, B., *A matrix-free preconditioner for sparse symmetric positive definite systems and least-squares problems*, SIAM J. Sci. Comp., 35 (2013), pp. A192-A211.

[4] L. Bergamaschi, J. Gondzio, J., and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Comput. Optim. Appl., 28 (2004), pp. 149–171.

[5] D. Bienstock, *Potential Function Methods for Approximately Solving Linear Programming Problems. Theory and Practice*, Kluwer, Boston, MA, 2002.

[6] A. Bjorck and G.H. Golub, *Numerical methods for computing angles between linear subspaces*, Math. Comp., 27 (1977), pp. 579–594.

[7] S. Bocanegra, J. Castro, and A.R.L. Oliveira, *Improving an interior-point approach for large block-angular problems by hybrid preconditioners*, Eur. J. Oper. Res., 231 (2013), pp. 263–273.

[8] R. Bridson and C. Greif, *A multipreconditioned conjugate gradient algorithm*, SIAM J Matrix Anal. Appl., 27 (2006), pp. 1056–1068.

[9] Y. Cao, C.D. Laird, and V.M. Zavala, *Clustering-based preconditioning for stochastic programs*, Preprint

ANL/MCS-P3050-1112, Argonne National Laboratory, Argonne, IL, 2014.
[10] J. Castro, *A specialized interior-point algorithm for multicommodity network flows*, SIAM J. Optim., 10 (2000), pp. 852–877.
[11] J. Castro, *An interior-point approach for primal block-angular problems*, Comput. Optim. Appl., 36 (2007), pp. 195–219.
[12] J. Castro, *Interior-point solver for convex separable block-angular problems*, Optim. Method. Softw., 31 (2016), pp. 88–109.
[13] J. Castro and J. Cuesta, *Quadratic regularizations in an interior-point method for primal block-angular problems*, Math. Prog., 130 (2011), pp. 415–445.
[14] J. Castro and N. Nabona, *An implementation of linear and nonlinear multicommodity network flows*, Eur. J. Oper. Res., 92 (1996), pp. 37–53.
[15] J. Castro and S. Nasini, *Mathematical programming approaches for classes of random network problems*, Eur. J. Oper. Res., 245 (2015), pp. 402–414.
[16] A. Frangioni, and G. Gallo, *A bundle type dual-ascent approach to linear multicommodity min cost flow problems*, INFORMS J. Comput., 11 (1999), pp. 370–393.
[17] A. Frangioni and C. Gentile, *New preconditioners for KKT systems of network flow problems*, SIAM J. Optim., 14 (2004), pp. 894–913.
[18] G.H. Golub and C.F. Van Loan, *Matrix Computations, Third Ed.*, Johns Hopkins Univ. Press, Baltimore, MD, 1996.
[19] J. Gondzio, *Convergence analysis of an inexact feasible interior point method for convex quadratic programming*, SIAM J. Optim., 23 (2013), pp. 1510–1527.
[20] J. Gondzio and R. Sarkissian, *Parallel interior-point solver for structured linear programs*, Math. Prog., 96 (2003), pp. 561–584.
[21] A. Manca, G. Sechi, and P. Zuddas, *Water supply network optimisation using equal flow algorithms*, Water Resour. Manag., 24 (2010), pp. 3665–3678.
[22] R.D. McBride, *Progress made in solving the multicommodity flow problem*, SIAM J. Optim., 8 (1998), pp. 947–955.
[23] E. Ng, and B.W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput., 14 (1993), pp. 1034–1056.
[24] 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 Appl., 394 (2005), pp. 1–24.
[25] J.M. Ortega, *Introduction to Parallel and Vector Solutions of Linear Systems*, Plenum Press, New York, NY, 1988.
[26] A. Ouorou, *A proximal cutting plane method using Chebychev center for nonsmooth convex optimization*, Math. Prog., 119 (2009), pp. 239–271.
[27] M.G.C. Resende and G. Veiga, *An implementation of the dual affine scaling algorithm for minimum-cost flow on bipartite uncapacitated networks*, SIAM J. Optim., 3 (1993), pp. 516–537.
[28] S.J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, 1996.