A STUDY OF PRECONDITIONERS FOR NETWORK INTERIOR POINT METHODS

J.J. JÚDICE*, J. M. PATRÍCIO†, L.F. PORTUGAL‡, M.G.C. RESENDE§, AND G. VEIGA¶

Abstract. We study and compare preconditioners available for network interior point methods. We derive upper bounds for the condition number of the preconditioned matrices used in the solution of systems of linear equations defining the algorithm search directions. The preconditioners are tested using PDNET, a state-of-the-art interior point code for the minimum cost network flow problem. A computational comparison using a set of standard problems improves the understanding of the effectiveness of preconditioners in network interior point methods.

Key words. Interior point method, linear programming, network flows, conjugate gradient, preconditioners, experimental testing of algorithms.

1. Introduction. A number of implementations of interior point methods for the minimum cost network flow (MCNF) problem have been proposed [11, 14, 15]. These procedures show that interior point methods may be competitive with the classical algorithms in several classes of the MCNF problem. The efficiency of these codes is heavily dependent on the method used to compute the search directions within the interior point algorithm framework.

Finding a search direction in interior point methods for linear programming involves the solution of a system of linear equations. In the most common variants, this results in a system of normal equations:

$$A\Theta A^\top x = b, \tag{1.1}$$

where A is the $m \times n$ matrix of the coefficients and Θ is a $n \times n$ diagonal matrix of positive elements. Either a direct factorization or an iterative method can be applied to solve (1.1). If we assume that A has full row rank, then $A\Theta A^\top$ is a symmetric positive definite matrix. The most commonly used direct and iterative approaches to solve this system are, respectively, the *LU* factorization and the preconditioned conjugate gradient (PCG) algorithm, respectively.

Even for some small scale MCNF problems, direct methods suffer from excessive memory requirements. This deficiency is illustrated in Figures 1.1-1.2 and Table 1.1 for a class of transshipment problems on a grid. Let L and U denote the triangular factors of $A\Theta A^\top$. Figures 1.1 and 1.2, respectively, display the nonzero structures of the matrices $A\Theta A^\top$ and $L+U$ after minimum degree ordering, illustrating the large amount of fill-in that occurs during the factorization.

In the particular case of the MCNF problem, direct factorization methods are not competitive with the PCG algorithm. Efficient preconditioning strategies for $A\Theta A^\top$ have been devised, making the PCG method a powerful approach in this context. Table 1.1 compares CPLEX BAROPT (a state-of-the-art primal-dual interior point algorithm based on *LU* factorization) and PDNET (a network flow conjugate-gradient based primal-dual interior point algorithm). In this table, the label *nodes* indicates the number of nodes (the number of arcs is $8 \times nodes$), nzlw and nz denote the numbers of nonzero elements of the lower triangular part of $A\Theta A^\top$ and factor L, respectively, DSW represents the size of the dense window [1], ITRS and TIME represent the number of interior-point iterations and the CPU time in seconds taken by CPLEX and PDNET, respectively, and CGITRS represents the total number of the conjugate gradient iterations required by PDNET. The experiment was done on a Silicon Graphics Challenge (196 MHz MIPS R10000 processor).

Table 1.1 shows the rapidly increasing difficulty encountered by the direct factorization method as the size of the network increases. Even for the small 4096 node instance, the amount of fill-in is overwhelming. Standard techniques such the use of a dense window appear to be of little help. A conjugate based method solves this same instance 60 times faster, and is capable of solving much larger problems.

*Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal.
†Escola Superior de Tecnologia de Tomar, Instituto Politécnico de Tomar, 2300 Tomar, Portugal.
‡Luis F. Portugal passed away on July 21, 1998.
§AT&T Labs Research, Florham Park, NJ 07932 USA.
¶AT&T, Middletown, NJ 07748 USA.

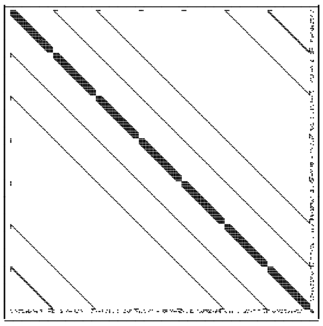

FIG. 1.1. *$A\Theta A^\top$ sparsity pattern*

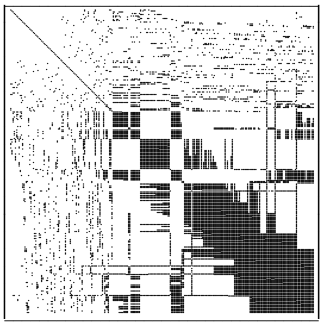

FIG. 1.2. *Sparsity patterns of LU factors of $A\Theta A^\top$*

The key ingredients to construct an efficient conjugate gradient based interior point code are the choice of stopping criterion and preconditioners for the PCG algorithm. The stopping criterion should minimize the computational effort involved in computing search directions that guarantee global convergence for the interior point algorithms. This is the idea behind the truncated optimization methods [6, 7]. In [14], a truncated Newton interior point method for linear programming is described. PDNET is an implementation of this algorithm for the MCNF problem.

A preconditioning matrix M is used to indirectly solve (1.1) by transformed system

$$M^{-1}A\Theta A^\top x = M^{-1}b.$$

When no preconditioner is used, the efficiency and the convergence of the conjugate gradient method depends on the distribution of the eigenvalues of $A\Theta A^\top$. An appropriate starting solution usually results in well conditioned system matrices for the initial iterations of the interior point method. However, as the algorithm progresses, the values of the scaling diagonal Θ spread and the conditioning of $A\Theta A^\top$ degrades rapidly. An effective preconditioner must improve on the spectral properties of the original system matrix without adding significant computational effort to the PCG algorithm.

Several families of preconditioners have been proposed for the PCG applied to network interior point methods [11, 13, 16, 20]. In this paper, we review the diagonal [20], the maximum spanning tree (MST) [16], and the diagonally compensated maximum spanning tree (DCMST) [11] preconditioning procedures. We derive upper bounds for the condition numbers of the preconditioned matrices obtained with these schemes. PDNET is used to compare the behavior of these preconditioning techniques in the initial, intermediate and final stages of the truncated Newton method.

TABLE 1.1
CPLEX & PDNET *on transshipment on a grid*

	CPLEX					PDNET		
nodes	nzlw	DSW	nz	ITRS	TIME	ITRS	CGITRS	TIME
256	1705	62	8750	27	0.54	26	145	0.26
512	3505	209	47915	20	2.84	33	164	0.71
1024	7004	82	48123	44	5.38	33	137	1.59
2048	14048	113	126190	54	19.57	35	211	4.51
4096	28652	1654	2173278	29	674.61	38	237	11.40
8192	57151	165	746707	61	210.21	43	246	26.62
16384	112645	220	1679410	67	514.08	41	254	61.38
32768	229872	15306	1455019303			67	1697	501.10
65536		did not run				47	370	593.26

Before concluding this introduction, we present some notation and outline the remainder of the paper. We denote a vector of ones by e. Let $x \in R^n$. We represent by X the $n \times n$ diagonal matrix having the elements of x in the diagonal. The number of elements of the set $\mathcal{H}$ is denoted by $|\mathcal{H}|$. Let B be a square matrix. $\underline{\lambda}(B)$, $\overline{\lambda}(B)$ and $\kappa(B)$, represent the smallest eigenvalue, the largest eigenvalue and the condition number of B, respectively. $\mathrm{diag}(B)$ denotes the diagonal matrix having the diagonal elements of B in the diagonal.

In Section 2 we describe the truncated Newton algorithm for the linear optimization problem. In Section 3 the conjugate gradient algorithm is presented. In Section 4 we study the conditioning of $A\Theta A^\top$, describing preconditioners and derive upper bounds for the condition number of their corresponding matrices. The computational experience and the concluding remarks are reported in sections 6 and 7, respectively.

2. The truncated Newton algorithm. The minimum cost network flow problem is a linear optimization problem of the form

$$\min \{c^\top x \mid Ax = b;\ x + s = u;\ x, s \geq 0\}, \tag{2.1}$$

where A is an $m \times n$ matrix obtained from the node-arc incidence matrix after the redundant constraints are removed, and b, u, x and s are the vectors of supply/demand at the nodes, arc capacities, flow variables and primal slacks, respectively. The dual of (2.1) can be written as:

$$\max \{b^\top y - u^\top w \mid A^\top y - w + z = c;\ z, w \geq 0\}, \tag{2.2}$$

where y is the vector of dual variables and w and z are the vectors of dual slacks. The conditions

$$\begin{aligned} Ax &= b, \\ x + s &= u, \\ A^\top y - w + z &= c, \\ Xz &= 0, \\ Sw &= 0, \\ x, s, w, z &\geq 0, \end{aligned} \tag{2.3}$$

are necessary and sufficient for (x, y, s, w, z) to be an optimal solution of the pair of problems (2.1)-(2.2).

In [14], a dual feasible version of a truncated Newton interior point algorithm for the solution of the nonlinear system of equations (2.3) was presented. This algorithm stems from the work of Dembo et al. [6], Dembo and Steihaug [7] and Kojima, Megiddo and Mizuno [10], using approximate Newton search directions computed with an iterative linear system solver. The iterative solver converges to the exact Newton direction from any initial solution. However, it is terminated as soon as a stopping criterion guaranteeing global convergence for the interior point method is verified.

Let β_0, β_1, β_2, β_3, γ and γ_p be such that $0 < \gamma < 1$, $\gamma_p > 0$ and $0 \le \beta_0 < \beta_1 < \beta_2 < \beta_3 < 1$. Furthermore, let ε_c and ε_p represent tolerances required for the complementarity gap and primal feasibility, respectively. The algorithm starts with any solution $(x^0, y^0, s^0, w^0, z^0) \in \mathcal{K}$, where

$$
\begin{aligned}
\mathcal{K} = \{(x,s,y,w,z) :\ & (x,s,y,w,z) > 0, x+s=u, A^\top y - w + z = c, \\
& x_i z_i \ge \gamma\mu(x,s,w,z)\ (i=1,2,\ldots,n), \\
& w_i s_i \ge \gamma\mu(x,s,w,z)\ (i=1,2,\ldots,n), \\
& \mu(x,s,w,z) \ge \gamma_p \|Ax-b\| \ \text{ or } \ \|Ax-b\| \le \varepsilon_p\},
\end{aligned}
$$

where the central trajectory parameter

$$\mu(x,s,w,z) = \beta_1 \frac{x^\top z + w^\top s}{2n}.$$

At iteration k, $\mu_k = \mu(x^k, s^k, w^k, z^k)$ is computed and the search direction

$$(\Delta x^k, \Delta y^k, \Delta s^k, \Delta w^k, \Delta z^k)$$

is obtained as a solution of the system of equations

$$A\Delta x^k = b - Ax^k + r^k, \tag{2.4}$$

$$\Delta x^k + \Delta s^k = 0, \tag{2.5}$$

$$A^\top \Delta y^k - \Delta w^k + \Delta z^k = 0, \tag{2.6}$$

$$Z^k \Delta x^k + X^k \Delta z^k = \mu_k e - X^k Z^k e, \tag{2.7}$$

$$W^k \Delta s^k + S^k \Delta w^k = \mu_k e - W^k S^k e. \tag{2.8}$$

where r^k is a vector satisfying

$$\|r^k\| \le \beta_0 \|Ax^k - b\| \tag{2.9}$$

As is explained in [14] and later in this paper, if

$$\Theta^k = \left(Z^k (X^k)^{-1} + W^k (S^k)^{-1}\right)^{-1} \tag{2.10}$$

and

$$\bar{b} = -A\Theta^k \left(\mu_k (X^k)^{-1} e - \mu_k (S^k)^{-1} e - c + A^\top y^k\right) + (b - Ax^k) \tag{2.11}$$

then the computation of this search direction simply amounts to solving inexactly the system of normal equations

$$A\Theta^k A^\top \Delta y^k = \bar{b} \tag{2.12}$$

by an iterative method under the following stopping criterion

$$\|A\Theta^k A^\top \Delta y^k - \bar{b}\| \le \beta_0 \|Ax^k - b\| \tag{2.13}$$

and then recovering Δx^k, Δs^k, Δw^k and Δz^k by simple affine expressions.

The new iterates are computed as

$$
\begin{aligned}
x^{k+1} &= x^k + \alpha_p \Delta x^k, \\
s^{k+1} &= s^k + \alpha_p \Delta s^k, \\
y^{k+1} &= y^k + \alpha_d \Delta y^k, \\
w^{k+1} &= w^k + \alpha_d \Delta w^k, \\
z^{k+1} &= z^k + \alpha_d \Delta z^k,
\end{aligned}
$$

where the stepsizes in the primal and dual spaces $\alpha_p, \alpha_d \in (0,1]$ are chosen according to

$$
\begin{aligned}
(x^{k+1}, s^{k+1}, y^{k+1}, w^{k+1}, z^{k+1}) &\in \mathcal{K}, \\
\mu_{k+1} &\leq (1 - \bar{\alpha}(1-\beta_3))\, \mu_k,
\end{aligned}
$$

where

$$
\begin{aligned}
\bar{\alpha} = \max\{0 < \alpha \leq 1 (x^k(\alpha), s^k(\alpha), y^k(\alpha), w^k(\alpha), z^k(\alpha)) \in \mathcal{K}, \\
\mu(x^k(\alpha), s^k(\alpha), y^k(\alpha), w^k(\alpha), z^k(\alpha)) \leq (1 - \alpha(1-\beta_2))\, \mu_k\}
\end{aligned}
$$

with

$$
\begin{aligned}
x^k(\alpha) &= x^k + \alpha \Delta x^k, \\
s^k(\alpha) &= s^k + \alpha \Delta s^k, \\
y^k(\alpha) &= y^k + \alpha \Delta y^k, \\
w^k(\alpha) &= w^k + \alpha \Delta w^k, \\
z^k(\alpha) &= z^k + \alpha \Delta z^k.
\end{aligned}
$$

If we set $\beta_0 = 0$, which is equivalent to replacing (2.4) by

$$A \Delta x^k = b - A x^k,$$

then this algorithm results in the exact Newton method of Kojima, Megiddo and Mizuno [10]. Despite some years ago it was believed not to be possible to prove convergence for infeasible exact Newton methods, it was already shown that these algorithms possess global, polynomial and even superlinear convergence. Kojima, Megiddo and Mizuno [10] have shown that if $\beta_0 = 0$, then the above algorithm produces in a finite number of iterations an iterate that either satisfies a prescribed tolerance or exceeds a prescribed upper bound in magnitude. The authors also introduced a modification of the algorithm having some better properties. In [21], Zhang presented an exact Newton interior point algorithm which guarantees to return an approximate solution by assuming that the set of optimal solutions is nonempty. This algorithm was also proved to have polynomial complexity. The convergence theory of infeasible exact Newton methods is based on the fact that the generated sequence of iterates verifies the equality

$$Ax^k - b = \nu_k (Ax^0 - b), \tag{2.14}$$

where ν_k is positive and converges to zero as $k \to \infty$.

The inexact Newton method presented in this Section was introduced by Portugal et al. [14]. In this algorithm, the norm $\|A(x^k + \Delta x^k) - b\|$ measures the closeness between the exact and the approximated Newton directions. The sequence generated by the algorithm does not verify equality (2.14). Instead, the weaker condition

$$\|Ax^k - b\| = \nu_k \|Ax^0 - b\| \tag{2.15}$$

holds. A very slight modification of the analysis presented in [10] establishes the following convergence result for the truncated Newton method.

THEOREM 2.1. *Either there is an iteration k where $(x^k, s^k, y^k, w^k, z^k)$ is an approximate solution satisfying*

$$\mu_k < \varepsilon_c \quad \textit{and} \quad \|Ax^k - b\| < \varepsilon_p,$$

or $\|(w^k, z^k)\| \to \infty$ as $k \to \infty$.

Proof. The proof only differs of that presented in [10] for the exact Newton algorithm in one detail. Instead of satisfying equality (11a) of [10], the sequence generated by this inexact Newton method satisfies

$$A(x^k + \alpha \Delta x^k) - b = (1-\alpha)(Ax^k - b) + \alpha r^k,$$

where r^k is such that

$$\|r^k\| \le \beta_0 \|Ax^k - b\|.$$

By taking this into consideration, we obtain for every $\alpha \in [0,1]$ that (see [10], page 271)

$$\begin{array}{lll} g_p(\alpha) \ge (\beta_1 - \beta_0)\varepsilon^* \alpha - \eta\alpha^2 & \text{if} & g_p(0) \ge 0 \\ (1 - \alpha(1-\beta_0))\|Ax^k - b\| \le \varepsilon_p & \text{if} & g_p(0) < 0. \end{array}$$

Thus, also in this case the primal and dual stepsizes are bounded away from zero, that is, there exists a $\alpha^* > 0$ such that $\bar{\alpha} \ge \alpha^*$ holds for every k. □

It follows from this theorem that the algorithm converges into an approximate optimal solution if the sequence of dual iterates is bounded. Based on extensive computational experience, partly described in [13, 14], we make the conjecture that this sequence is bounded provided the linear program is primal feasible. Since dual feasibility is assumed in our algorithm then the linear program has a finite optimal solution in this case. We further note that detecting primal feasibility in a linear network flow problem simply amounts to solving a maximal flow problem.

After the design of our algorithm, several attempts have been made to develop globally convergent truncated interior-point methods for linear programs and other related problems [8, 12, 5, 4]. We have no intention in this paper to comment on the benefits and drawbacks of these alternative techniques in practice. Instead, our main objective is to provide bounds for the condition number of the matrices $M^{-1}(A\Theta^k A^\top)$ associated with the system of normal equations (2.12), where Θ^k is given by (2.10), M is a preconditioning matrix to be discussed later and A is a full-row rank node-arc incidence matrix. We believe that the results included in this paper may also have an important impact on the solution of linear network flow problems by these alternative interior-point techniques.

An important property of the interior point algorithms is concerned with the limit points of the iteration sequence. Let $\mathcal{S}$ denote the set of optimal solutions. Furthermore, suppose that the sets of indices

$$\begin{aligned} \mathcal{B} &= \{i : z_i = 0 \text{ and } \; w_i = 0 \text{ for all } \; (x,y,s,w,z) \in \mathcal{S}\}, \\ \mathcal{L} &= \{i : x_i = 0 \text{ and } \; w_i = 0 \text{ for all } \; (x,y,s,w,z) \in \mathcal{S}\}, \\ \mathcal{U} &= \{i : s_i = 0 \text{ and } \; z_i = 0 \text{ for all } \; (x,y,s,w,z) \in \mathcal{S}\}, \end{aligned}$$

constitute the optimal partition of $\{1, 2, \ldots, n\}$. Equality (2.14) can be used to prove that there exists a constant τ such that the sequences of iterates generated by infeasible exact Newton algorithms progressing inside neighborhoods of the central path similar to set $\mathcal{K}$ satisfy the following conditions for all k sufficiently large [17], [19]:

$$\begin{aligned} &\tau \le s_i^k, x_i^k \le \frac{1}{\tau} \text{ and } \; \gamma\tau\mu_k \le w_i^k, z_i^k \le \frac{\mu_k}{\tau} \;\text{ if } \; i \in \mathcal{B}, \\ &\tau \le s_i^k, z_i^k \le \frac{1}{\tau} \text{ and } \; \gamma\tau\mu_k \le x_i^k, w_i^k \le \frac{\mu_k}{\tau} \;\text{ if } \; i \in \mathcal{L}, \\ &\tau \le x_i^k, w_i^k \le \frac{1}{\tau} \text{ and } \; \gamma\tau\mu_k \le s_i^k, z_i^k \le \frac{\mu_k}{\tau} \;\text{ if } \; i \in \mathcal{U}. \end{aligned} \tag{2.16}$$

Unfortunately we have not been able to establish a similar result for the truncated version. Nevertheless, we observed that the generated iteration sequence satisfies conditions (2.16) in practice. So we assume in the sequel that conditions (2.16) are satisfied by the truncated Newton method presented in this section.

3. The preconditioned conjugate gradient algorithm. As discussed in [14], the search direction is obtained in two steps. First, the preconditioned conjugate gradient (PCG) algorithm is used to compute the component Δy^k as an approximate solution of the linear system

$$A\Theta^k A^\top \Delta y^k = \bar{b}, \tag{3.1}$$

procedure pcg$(A, \Theta^k, \bar{b})$
1 Compute Δy_0 as a guess of Δy^k;
2 $r_0 := \bar{b} - A\Theta^k A^\top \Delta y_0$;
3 solve $M z_0 = r_0$;
4 $p_0 := z_0$;
5 $i := 0$;
6 **do** stopping criterion not satisfied $\rightarrow$
7   $q_i := A\Theta^k A^\top p_i$;
8   $\alpha_i := z_i^\top r_i / p_i^\top q_i$;
9   $\Delta y_{i+1} := \Delta y_i + \alpha_i p_i$;
10   $r_{i+1} := r_i - \alpha_i q_i$;
11   solve $M z_{i+1} = r_{i+1}$;
12   $\beta_i := z_{i+1}^\top r_{i+1} / z_i^\top r_i$;
13   $p_{i+1} := z_{i+1} + \beta_i p_i$;
14   $i := i + 1$
15 **od**;
16 $\Delta y^k := \Delta y_i$
end pcg;

FIG. 3.1. *The preconditioned conjugate gradient algorithm*

where $\bar{b}$ and Θ^k are given by (2.10) and (2.11) until the stopping criterion (2.13) is satisfied. Then, the remaining components of the direction are recovered by

$$\begin{aligned}
\Delta x^k &= \Theta^k A^\top \Delta y^k + \Theta^k(\mu_k (X^k)^{-1} e - \mu_k (S^k)^{-1} e - c + A^\top y^k), \\
\Delta s^k &= -\Delta x^k, \\
\Delta z^k &= -z^k + \mu_k (X^k)^{-1} e - Z^k (X^k)^{-1} \Delta x^k, \\
\Delta w^k &= -w^k + \mu_k (S^k)^{-1} e - W^k (S^k)^{-1} \Delta s^k.
\end{aligned}$$

The PCG algorithm is described in Figure 3.1. This iterative linear system solver is a variant of the conjugate gradient method designed to find the solution of the linear system

$$M^{-1} A\Theta^k A^\top \Delta y^k = M^{-1}\bar{b}$$

without explicitly computing $M^{-1}A\Theta^k A^\top$. The matrix M, is called preconditioner or preconditioning matrix. Symmetry and Positive definiteness of M are sufficient conditions for the PCG algorithm to converge. The aim of preconditioning is to reduce the number of iterations required to find a search direction by obtaining a matrix $M^{-1}A\Theta^k A^\top$ with more favorable spectral properties than those of $A\Theta^k A^\top$. In general, the preconditioning techniques either compute an approximation of the matrix of the system of linear equations or an approximation of the inverse of this matrix. In the second case, the linear systems in steps 3 and 11 of the PCG algorithm are replaced by the matrix-vector products $z_0 := M^{-1} r_0$ and $z_{i+1} := M^{-1} r_{i+1}$, respectively. Preconditioning reduces the iterations count of the conjugate gradient algorithm but incurs some additional costs, namely those of constructing M (constructing M^{-1}) and solving the linear system $M z_{i+1} = r_{i+1}$ (computing the product $z_{i+1} := M^{-1} r_{i+1}$) at each iteration of the PCG algorithm. The gain in the convergence speed achieved with a preconditioning scheme must superior to the extra costs.

4. The condition number of the normal equations matrix. In this section we exploit the special structure of the node-arc incidence matrix to derive an upper-bound for the condition number of the normal equations matrix $A\Theta^k A^\top$ that depends on the size of the diagonal elements of the diagonal matrix Θ^k. By assuming property (2.16), we further show that the condition numbers of the matrices $A\Theta^k A^\top$ used by the truncated interior-point methods are uniformly bounded in the last iterations provided this algorithm converges to a primal nondegenerate solution. However, the boundedness of the condition number is not guaranteed under a primal degenerate optimal solution.

We start by recalling some properties that will be useful to establish the main results of this and next sections. We also omit the superscripts k for easier explanation.

PROPOSITION 4.1. *Let B be a $m \times m$ symmetric matrix. Then*

$$\min_{1\le l\le m}\left\{ b_{ll} - \sum_{\substack{h=1\\ h\neq l}}^{m} |b_{hl}| \right\} \le \underline{\lambda}(B) \le \overline{\lambda}(B) \le \max_{1\le l\le m}\left\{ b_{ll} + \sum_{\substack{h=1\\ h\neq l}}^{m} |b_{hl}| \right\}.$$

Proof. This is the well known Gerschgorin Theorem (see [3]). □

PROPOSITION 4.2. *Let B and C be symmetric matrices, and let $\lambda_i(B)$ denote the ith eigenvalue, where we order the eigenvalues in an increasing order. Then*

$$\lambda_i(B) + \underline{\lambda}(C) \le \lambda_i(B+C) \le \lambda_i(B) + \overline{\lambda}(C).$$

Proof. This is a Corollary of the Courant-Fischer lemma (see [3]). □

PROPOSITION 4.3. *Let B and C be symmetric positive semi-definite and symmetric positive definite matrices, respectively.*

(i) If ρ_1 is a number such that $\overline{\lambda}(\rho_1 C - B) \le 0$, then $\underline{\lambda}(C^{-1}B) \ge \rho_1$.

(ii) If ρ_2 is a number such that $\underline{\lambda}(\rho_2 C - B) \ge 0$, then $\overline{\lambda}(C^{-1}B) \le \rho_2$.

Proof. See [3], page 404. □

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ denote the network associated to the MCNF problem, where $\mathcal{N}$ and $\mathcal{A}$ represent the set of nodes and the set of directed arcs, respectively. Each element (i,j) of $\mathcal{A}$ corresponds to one column of A and vice-versa. The same correspondence between elements of $\mathcal{A}$ and diagonal elements of Θ exists. Suppose that $\mathcal{H}$ is a subset of $\mathcal{A}$ and let $A_{\mathcal{H}}$ and $\Theta_{\mathcal{H}}$ be the corresponding submatrices of A and Θ, respectively. The diagonal element $[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{ll}$ and the off-diagonal element $[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{lh}$ of $A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ are given by

$$[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{ll} = \sum_{(i,j):(i,j)\in\mathcal{H}_l} \theta_{ij}$$

and

$$[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{lh} = \begin{cases} \displaystyle\sum_{(i,j):(i,j)\in\mathcal{H}_l\cap\mathcal{H}_h} -\theta_{ij} & \text{if } \mathcal{H}_l\cap\mathcal{H}_h \neq \emptyset \\ 0 & \text{if } \mathcal{H}_l\cap\mathcal{H}_h = \emptyset \end{cases}$$

respectively, where $\mathcal{H}_l$ represents the set of arcs in $\mathcal{H}$ incident to node l and θ_{ij} denotes the diagonal element of $\Theta_{\mathcal{H}}$ associated to arc (i,j).

THEOREM 4.4. $\overline{\lambda}(A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top) \le 2\overline{\lambda}(\Theta_{\mathcal{H}})|\mathcal{H}|$.

Proof. We note that

$$[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{ll} \ge \sum_{\substack{h=1\\ h\neq l}}^{m} |[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{hl}|,\ l = 1,\dots,m.$$

So, from Proposition 4.1 we obtain

$$\begin{aligned}\overline{\lambda}(A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top) &\le 2\max_{1\le l\le m}\left\{[A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top]_{ll}\right\} = 2\max_{1\le l\le m}\left\{\sum_{(i,j):(i,j)\in\mathcal{H}_l}\theta_{ij}\right\}\\ &\le 2\overline{\lambda}(\Theta_{\mathcal{H}})\max_{1\le l\le m}\{|\mathcal{H}_l|\} \le 2\overline{\lambda}(\Theta_{\mathcal{H}})|\mathcal{H}|\end{aligned}$$

as desired. □

Since we are assuming that A is a full row rank matrix, $A\Theta A^\top$ is positive definite, that is, there exists at least one basis of A. However, $A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ is not positive definite for every subset $\mathcal{H}$ of $\mathcal{A}$. A necessary and sufficient condition for the positive semi-definite matrix $A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ to be a nonsingular matrix is the existence of a set of arcs $\mathcal{T} \subseteq \mathcal{H}$ such that $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ is a spanning tree.

THEOREM 4.5. *$A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ is positive definite and*

$$\underline{\lambda}(A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top) \geq \frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m^2},$$

if and only if there exists $\mathcal{T} \subseteq \mathcal{H}$ such that $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ is a spanning tree.

Proof. The rank of $A_{\mathcal{H}}$ equals the cardinality of $\mathcal{T}$, where $\mathcal{T} \subseteq \mathcal{H}$ is the maximum cardinality set such that $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ does not contain any cycle [2]. If $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ is not a spanning tree, then $|\mathcal{T}| < m$, $A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ is singular and $\underline{\lambda}(A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top) = 0$. If $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ is a spanning tree, then $|\mathcal{T}| = m$ and $A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ is positive definite. In this case, let $\bar{\mathcal{T}} = \mathcal{H} - \mathcal{T}$. Since $A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top$ and $A_{\mathcal{T}}(\Theta_{\mathcal{T}} - \underline{\lambda}(\Theta_{\mathcal{T}})I)A_{\mathcal{T}}^\top$ are positive semi-definite matrices and the latter is singular, then from Proposition 4.2 we have

$$\begin{aligned}\underline{\lambda}(A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top) &\geq \underline{\lambda}(A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top) + \underline{\lambda}(A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top) \geq \underline{\lambda}(A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top)\\ &\geq \underline{\lambda}(A_{\mathcal{T}}(\Theta_{\mathcal{T}} - \underline{\lambda}(\Theta_{\mathcal{T}})I)A_{\mathcal{T}}^\top) + \underline{\lambda}(\Theta_{\mathcal{T}})\underline{\lambda}(A_{\mathcal{T}}A_{\mathcal{T}}^\top) = \frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{\overline{\lambda}(A_{\mathcal{T}}^{-\top}A_{\mathcal{T}}^{-1})}.\end{aligned}$$

Now, $A_{\mathcal{T}}^{-1}$ is a matrix having nonzero elements equal to 1 or -1. So, according to Proposition 4.1, $\overline{\lambda}(A_{\mathcal{T}}^{-\top}A_{\mathcal{T}}^{-1}) \leq m^2$. This gives the above lower bound for $\underline{\lambda}(A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top)$ when $A_{\mathcal{H}}\Theta_{\mathcal{H}}A_{\mathcal{H}}^\top$ is nonsingular. ☐

Since $A_{\mathcal{T}}$ is a nonsingular matrix

$$\Theta_{\mathcal{T}} = A_{\mathcal{T}}^{-1}A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top A_{\mathcal{T}}^{-\top}$$

Furthermore it is easy to see that $\overline{\lambda}(A_{\mathcal{T}}^\top A_{\mathcal{T}}) \leq m(m-1)$. By using these two results and following a proof similar to the last one, we are able to show that

$$\underline{\lambda}(\Theta_{\mathcal{T}}) \geq \frac{\underline{\lambda}(A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top)}{m(m-1)} \tag{4.1}$$

The next result establishes an upper bound for the condition number of $A\Theta A^\top$.

COROLLARY 4.6. *Suppose that the elements of Θ represent weights associated to the arcs of $\mathcal{A}$ and let $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ be a maximum spanning tree of $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ and $\bar{\mathcal{T}} = \mathcal{A} - \mathcal{T}$. Then*

$$\kappa(A\Theta A^\top) \leq 2m^3\frac{\overline{\lambda}(\Theta_{\mathcal{T}})}{\underline{\lambda}(\Theta_{\mathcal{T}})} + 2m^2(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}. \tag{4.2}$$

Proof. From Theorem 4.5 we have that

$$\underline{\lambda}(A\Theta A^\top) \geq \frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m^2},$$

since $A\Theta A^\top$ is positive definite. On the other hand, Proposition 4.2 and Theorem 4.4 give

$$\begin{aligned}\overline{\lambda}(A\Theta A^\top) &\leq \overline{\lambda}(A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top) + \overline{\lambda}(A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top)\\ &\leq 2m\overline{\lambda}(\Theta_{\mathcal{T}}) + 2(n-m)\overline{\lambda}(\Theta_{\bar{\mathcal{T}}}).\end{aligned}$$

Condition (4.2) is obtained by dividing the upper bound by the lower bound estimated for the eigenvalues of $A\Theta A^\top$. ☐

At the first iterations of the truncated Newton algorithm, we can manage to obtain matrices $A\Theta A^\top$ well conditioned by choosing an appropriate starting point for this method. As the algorithm progresses, the values of the diagonal elements of Θ spread and the conditioning of $A\Theta A^\top$

degrades. At the final stages, it may happen that $A\Theta A^\top$ is very ill conditioned. Let $\mathcal{S}$ denote the set of primal dual optimal solutions of the MCNF problem and suppose that the sets

$$\begin{aligned}
\mathcal{B} &= \{(i,j) : z_{ij} = 0 \text{ and } \ w_{ij} = 0 \text{ for all } \ (x,y,s,w,z) \in \mathcal{S}\}\,, \\
\mathcal{L} &= \{(i,j) : x_{ij} = 0 \text{ and } \ w_{ij} = 0 \text{ for all } \ (x,y,s,w,z) \in \mathcal{S}\}\,, \\
\mathcal{U} &= \{(i,j) : s_{ij} = 0 \text{ and } \ z_{ij} = 0 \text{ for all } \ (x,y,s,w,z) \in \mathcal{S}\}\,,
\end{aligned}$$

constitute the optimal partition of $\mathcal{A}$. As discussed in the previous section, we assume that there exists a constant τ such that the conditions

$$\begin{aligned}
&\tau \le s_{ij}, x_{ij} \le \frac{1}{\tau} \text{ and } \ \gamma\tau\mu \le w_{ij}, z_{ij} \le \frac{\mu}{\tau} \ \text{ if } \ (i,j) \in \mathcal{B}, \\
&\tau \le s_{ij}, z_{ij} \le \frac{1}{\tau} \text{ and } \ \gamma\tau\mu \le x_{ij}, w_{ij} \le \frac{\mu}{\tau} \ \text{ if } \ (i,j) \in \mathcal{L}, \\
&\tau \le x_{ij}, w_{ij} \le \frac{1}{\tau} \text{ and } \ \gamma\tau\mu \le s_{ij}, z_{ij} \le \frac{\mu}{\tau} \ \text{ if } \ (i,j) \in \mathcal{U}
\end{aligned} \tag{4.3}$$

hold when μ becomes small.

COROLLARY 4.7. *Suppose that conditions (4.3) hold with μ small. Then, at the later iterations of the truncated Newton method*

$$\kappa(A\Theta A^\top) \le \begin{cases} \tau_1 + \tau_2 \frac{1}{\frac{1}{2\gamma\mu^2}+\frac{1}{2}} & \textit{if } |\mathcal{B}| = m \\ \tau_1 + \tau_2 & \textit{if } |\mathcal{B}| > m \\ \tau_1(\frac{1}{2\gamma\mu^2}+\frac{1}{2}) + \tau_2 & \textit{if } |\mathcal{B}| < m, \end{cases} \tag{4.4}$$

where the constants τ_1 and τ_2 are given by

$$\tau_1 = 2m^3\frac{1}{\gamma\tau^4} \quad \textit{and} \quad \tau_2 = 2m^2(n-m)\frac{1}{\gamma\tau^4}.$$

Proof. Since

$$\Theta = \left(ZX^{-1} + WS^{-1}\right)^{-1}$$

where Z, X, W and S are diagonal matrices with diagonal elements z_{ij}, x_{ij}, w_{ij} and s_{ij} respectively, then it immediately follows from the conditions (4.3) that

$$\begin{aligned}
\frac{\tau^2}{2\mu} &\le \underline{\lambda}(\Theta_{\mathcal{B}}) \le \overline{\lambda}(\Theta_{\mathcal{B}}) \le \frac{1}{\gamma\tau^2 2\mu} \\
\frac{1}{\frac{1}{\gamma\tau^2\mu}+\frac{\mu}{\tau^2}} &\le \underline{\lambda}(\Theta_{\mathcal{L}\cup\mathcal{U}}) \le \overline{\lambda}(\Theta_{\mathcal{L}\cup\mathcal{U}}) \le \frac{1}{\frac{\tau^2}{\mu}+\gamma\tau^2\mu}.
\end{aligned}$$

Furthermore for small and positive μ we have

$$\frac{1}{\frac{1}{\gamma\tau^2\mu}+\frac{\mu}{\tau^2}} = \frac{\tau^2}{2\mu}\left[\frac{2}{1+\frac{1}{\mu^2\gamma}}\right] < \frac{\tau^2}{2\mu}$$

$$\frac{1}{\frac{\tau^2}{\mu}+\gamma\tau^2\mu} = \frac{1}{2\gamma\tau^2\mu}\left[\frac{2}{1+\frac{1}{\gamma^2}}\right] < \frac{1}{2\gamma\tau^2\mu}$$

Now the conditions (4.4) follow from (4.2) by setting $\mathcal{T} = \mathcal{B}$, $\mathcal{T} \subset \mathcal{B}$ and $\mathcal{T} \supset \mathcal{B}$ respectively. ∎

We note that $|\mathcal{B}| = m$, $|\mathcal{B}| < m$ and $|\mathcal{B}| > m$ correspond to the cases of primal and dual nondegeneracy, dual degeneracy and primal degeneracy respectively. It follows from this theorem that under the assumption (2.16) the condition number of $A\Theta A^\top$ is uniformly bounded at the last iterations of the algorithm provided the optimal solution is primal nondegenerate. The upper bound found for $\kappa(A\Theta A^\top)$ when the primal and dual optimal solution are unique, is smaller than that obtained when dual degeneracy is present, since in the first case the second term in (4.4) vanishes as $\mu \to 0$. If the problem is primal degenerate, then the right-hand side of (4.4) converges to infinity as $\mu \to 0$. In this case, $A\Theta A^\top$ is usually poorly conditioned when μ becomes small. In certain implementations of interior point methods for linear programming where LU factorization is used to compute the search directions several schemes are used to overcome this problem, namely, procedures to handle negative and zero pivots encountered during the factorization process [18]. In the conjugate gradient based interior point algorithm we propose for the MCNF problem, preconditioning schemes for the matrix $A\Theta A^\top$ are used. As we show in the next section, we can manage to reduce to 1 the first term in (4.4) by using spanning tree based preconditioning techniques.

5. Condition Number of the Preconditioned Matrix. As is discussed in [14] and in this paper, the system of normal equations (2.12) is solved by using the Preconditioned Conjugate Gradient Method, with a preconditioner M. The Diagonal, the Maximum Spanning Tree (MST) and the Diagonal Compensated Maximum Spanning Tree (DCMST) Preconditioners have been the most recommended in practice to be used in interior-point network flow problems [11, 14, 20]. The success of these techniques depends on the boundedness of the condition number of the preconditioned matrix $M^{-1}(A\Theta_k A^\top)$. In this section we derive some condition number upper bounds for each one of the preconditioners stated before. These theoretical results seem to support our strategy for choosing the preconditioner discussed in [14]. The diagonal preconditioner

$$M = \mathrm{diag}(A\Theta A^\top)$$

was first used by Yeh [20] in the context of interior point algorithms for network flows. The cost of constructing M^{-1} is $O(n)$ additions and $O(m)$ divisions. The cost of computing $z_{i+1} = M^{-1}r_{i+1}$ is $O(m)$ multiplications. The next result gives an upper bound for $\kappa(M^{-1}A\Theta A^\top)$.

THEOREM 5.1. *Let $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ be a maximum spanning tree of $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, $\bar{\mathcal{T}} = \mathcal{A} - \mathcal{T}$ and $M = diag(A\Theta A^\top)$. Then*

$$\kappa(M^{-1}A\Theta A^\top) \le 2m^3 \frac{\overline{\lambda}(\Theta_{\mathcal{T}})}{\underline{\lambda}(\Theta_{\mathcal{T}})} + 2m^2(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}. \tag{5.1}$$

Proof. Since $2M - A\Theta A^\top$ is positive semi-definite, then $\underline{\lambda}(2M - A\Theta A^\top) \ge 0$ and by Proposition 4.3

$$\overline{\lambda}(M^{-1}A\Theta A^\top) \le 2.$$

Now, let

$$\rho_1 = \frac{1}{m^3 \frac{\overline{\lambda}(\Theta_{\mathcal{T}})}{\underline{\lambda}(\Theta_{\mathcal{T}})} + m^2(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}}.$$

From Propositions 4.1, 4.2 and Theorem 4.5 we have

$$\begin{aligned}\overline{\lambda}\left(\rho_1 M - A\Theta A^\top\right) &\le \rho_1 \overline{\lambda}(M) - \underline{\lambda}(A\Theta A^\top)\\ &\le \rho_1 m \overline{\lambda}(\Theta_{\mathcal{T}}) + \rho_1(n-m)\overline{\lambda}(\Theta_{\bar{\mathcal{T}}}) - \frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m^2} = 0.\end{aligned}$$

Thus, $\frac{2}{\rho_1}$ is an upper bound for the condition number of $M^{-1}A\Theta A^\top$. ☐

It is interesting to note that the upper-bound achieved for the preconditioned matrix

$$M^{-1}(A\Theta A^\top)$$

coincides with that of the matrix $A\Theta A^\top$. This result is not surprising. Computational experience described in [15] has shown that the diagonal preconditioner is effective at the first stages of the interior point algorithms when $A\Theta A^\top$ is itself well conditioned. However, as the algorithms progress, it performs similarly to the identity matrix preconditioner.

The lack of effectiveness of the diagonal preconditioner at the intermediate and final iterations of the interior point methods motivated the search for other preconditioning schemes. The maximum spanning tree (MST) preconditioner was introduced by Resende and Veiga [16]. It has the form

$$M = A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top,$$

where $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ is a maximum spanning tree of $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. Contrary to the diagonal preconditioner, this preconditioning matrix takes advantage of the structure of the optimal solution of the MCNF problem. Constructing M involves computing $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ and inverting $\Theta_{\mathcal{T}}$ ($\Theta_{\mathcal{T}}^{-1}$ is explicitly computed in order to perform multiplications instead of divisions when the linear systems $Mz_{i+1} = r_{i+1}$ are solved in the PCG algorithm). In [16], the Kruskal algorithm is used to compute an approximate maximum spanning tree, while in [14] an exact maximum spanning tree is obtained by employing an implementation of Prim's algorithm with a Fibonacci heap data structure. The running time for this last method is $O\left(n + mlog(m)\right)$. The cost of solving $Mz_{i+1} = r_{i+1}$ is $O(m)$ additions and multiplications.

In [11], Mehrotra and Wang proposed the Diagonally Compensated Maximum Spanning Tree (DCMST) Preconditioner

$$M = A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top + D$$

where $\overline{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ is a maximum spanning tree of $\overline{\mathcal{G}} = (\mathcal{N}, \mathcal{A})$, $\overline{\mathcal{T}} = \mathcal{A} - \mathcal{T}$, and

$$D = \phi \mathrm{diag}(A_{\overline{\mathcal{T}}}\Theta_{\overline{\mathcal{T}}}A_{\overline{\mathcal{T}}}^\top)$$

with ϕ a nonnegative parameter. We further notice that the MST preconditioner is a special case of this last one when $\phi = 0$. Next we establish an upper-bound for the condition number of the preconditioned matrix when M is the DCMST preconditioner.

THEOREM 5.2. *Let $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ be a maximum spanning tree of $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, $\bar{\mathcal{T}} = \mathcal{A} - \mathcal{T}$ and $M = A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top + D$, where $D = \phi diag(A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top)$ and $0 \leq \phi \leq 2$. Then*

$$\kappa(M^{-1}A\Theta A^\top) \leq \min\left\{ \frac{1 + m(m-1)\frac{\overline{\lambda}(D)}{\underline{\lambda}(\Theta_{\mathcal{T}})}}{1 + m(m-1)\frac{\underline{\lambda}(D)}{\overline{\lambda}(\Theta_{\mathcal{T}})}} \left(1 + 2m(m-1)(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right), \right.$$
$$\left. \frac{1 + m(m-1)\frac{\overline{\lambda}(D)}{\underline{\lambda}(\Theta_{\mathcal{T}})}}{1 + \frac{1}{2m}\frac{\underline{\lambda}(D)}{\overline{\lambda}(\Theta_{\mathcal{T}})}} \left(1 + m(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right) \right\}. \tag{5.2}$$

Proof. Let

$$\rho_1 = \frac{1}{1 + m(m-1)\frac{\overline{\lambda}(D)}{\underline{\lambda}(\Theta_{\mathcal{T}})}}$$

and

$$\hat{\rho}_2 = \frac{1 + 2m(m-1)(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}}{1 + m(m-1)\frac{\underline{\lambda}(D)}{\overline{\lambda}(\Theta_{\mathcal{T}})}}, \qquad \bar{\rho}_2 = \frac{1 + m(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}}{1 + \frac{1}{2m}\frac{\underline{\lambda}(D)}{\overline{\lambda}(\Theta_{\mathcal{T}})}}.$$

We first note that $0 \leq \rho_1 \leq 1$. Furthermore, it follows from the definition of the matrix D and $0 \leq \phi \leq 2$ that $\overline{\lambda}(D) \leq 2(n-m)\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})$ and this guarantees that $\hat{\rho}_2, \bar{\rho}_2 \geq 1$. From Proposition

4.2, inequality (4.1) and the positive semi-definiteness of $A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top$ we have

$$\begin{aligned}\overline{\lambda}(\rho_1 M - A\Theta A^\top) &= \overline{\lambda}\left((\rho_1-1)A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top + \rho_1 D - A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top\right)\\ &\le (\rho_1-1)\underline{\lambda}(A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top) + \rho_1\overline{\lambda}(D)\\ &\le (\rho_1-1)\frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m(m-1)} + \rho_1\overline{\lambda}(D)\\ &= \left(\frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m(m-1)} + \overline{\lambda}(D)\right)\rho_1 - \frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m(m-1)} = 0.\end{aligned}$$

On the other hand, from Proposition 4.2, Theorem 4.4 and (4.1) we obtain

$$\begin{aligned}\underline{\lambda}(\hat{\rho_2} M - A\Theta A^\top) &= \underline{\lambda}\left((\hat{\rho_2}-1)A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top + \hat{\rho_2} D - A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top\right)\\ &\ge (\hat{\rho_2}-1)\underline{\lambda}(A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top) + \hat{\rho_2}\underline{\lambda}(D) - \overline{\lambda}(A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top)\\ &\ge (\hat{\rho_2}-1)\frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m(m-1)} + \hat{\rho_2}\underline{\lambda}(D) - 2(n-m)\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})\\ &= \left(\frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m(m-1)} + \underline{\lambda}(D)\right)\hat{\rho_2} - \left(\frac{\underline{\lambda}(\Theta_{\mathcal{T}})}{m(m-1)} + 2(n-m)\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})\right)\\ &= 0.\end{aligned}$$

Now, it is well known that each element of $A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}$ equals either 0, 1 or -1. This implies that

$$\begin{aligned}&(i)\, 0 \le [A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top A_{\mathcal{T}}^{-\top}]_{ll} \le n-m, \quad l = 1,\dots,m,\\ &(ii) {-(n-m)} \le [A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top A_{\mathcal{T}}^{-\top}]_{lh} \le n-m \quad l,h = 1,\dots,m \quad l \ne h.\end{aligned}$$

Consequently, $\overline{\lambda}\left(A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top A_{\mathcal{T}}^{-\top}\right) \le m(n-m)$ according to Proposition 4.1. Furthermore,

$$\begin{aligned}&\underline{\lambda}\left((\bar{\rho_2}-1)\Theta_{\mathcal{T}} + \bar{\rho_2}A_{\mathcal{T}}^{-1}DA_{\mathcal{T}}^{-\top} - A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top A_{\mathcal{T}}^{-\top}\right)\\ &\qquad \ge (\bar{\rho_2}-1)\underline{\lambda}(\Theta_{\mathcal{T}}) + \bar{\rho_2}\underline{\lambda}(A_{\mathcal{T}}^{-1}DA_{\mathcal{T}}^{-\top}) - \overline{\lambda}\left(A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top A_{\mathcal{T}}^{-\top}\right)\\ &\qquad \ge (\bar{\rho_2}-1)\underline{\lambda}(\Theta_{\mathcal{T}}) + \bar{\rho_2}\frac{1}{\overline{\lambda}(A_{\mathcal{T}}D^{-1}A_{\mathcal{T}}^\top)} - \overline{\lambda}(\Theta_{\bar{\mathcal{T}}})\overline{\lambda}\left(A_{\mathcal{T}}^{-1}A_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top A_{\mathcal{T}}^{-\top}\right)\\ &\qquad \ge (\bar{\rho_2}-1)\underline{\lambda}(\Theta_{\mathcal{T}}) + \bar{\rho_2}\frac{\underline{\lambda}(D)}{2m} - \overline{\lambda}(\Theta_{\bar{\mathcal{T}}}))m(n-m)\\ &\qquad \ge \left(\underline{\lambda}(\Theta_{\mathcal{T}}) + \frac{\underline{\lambda}(D)}{2m}\right)\bar{\rho_2} - \left(\underline{\lambda}(\Theta_{\mathcal{T}\mathcal{T}}) + \overline{\lambda}(\Theta_{\bar{\mathcal{T}}})m(n-m)\right) = 0.\end{aligned}$$

Thus, $\underline{\lambda}(\bar{\rho_2}M - A\Theta A^\top) \ge 0$. According to Proposition 4.3, $\frac{\hat{\rho_2}}{\rho_1}$ and $\frac{\bar{\rho_2}}{\rho_1}$ are upper bounds for the condition number of $M^{-1}A\Theta A^\top$. □

Consider now the MST preconditioner. Since $\phi = 0$ in this case, then $\underline{\lambda}(D) = \overline{\lambda}(D) = 0$. Furthermore $2m(m-1) > m$ for all $m > 1$ and this leads to

$$\kappa(M^{-1}A\Theta A^\top) \le 1 + m(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})} \tag{5.3}$$

It is interesting to further note that this upper bound is smaller than the one for the DCMST independently of the choice of $\phi > 0$. In fact, as $\underline{\lambda}(D) < \overline{\lambda}(D)$ for any $\phi > 0$ and $\frac{1}{2m} < m(m-1)$ for any $m > 1$, then

$$1 + m(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})} < up$$

where up is the upper-bound provided by (5.2).

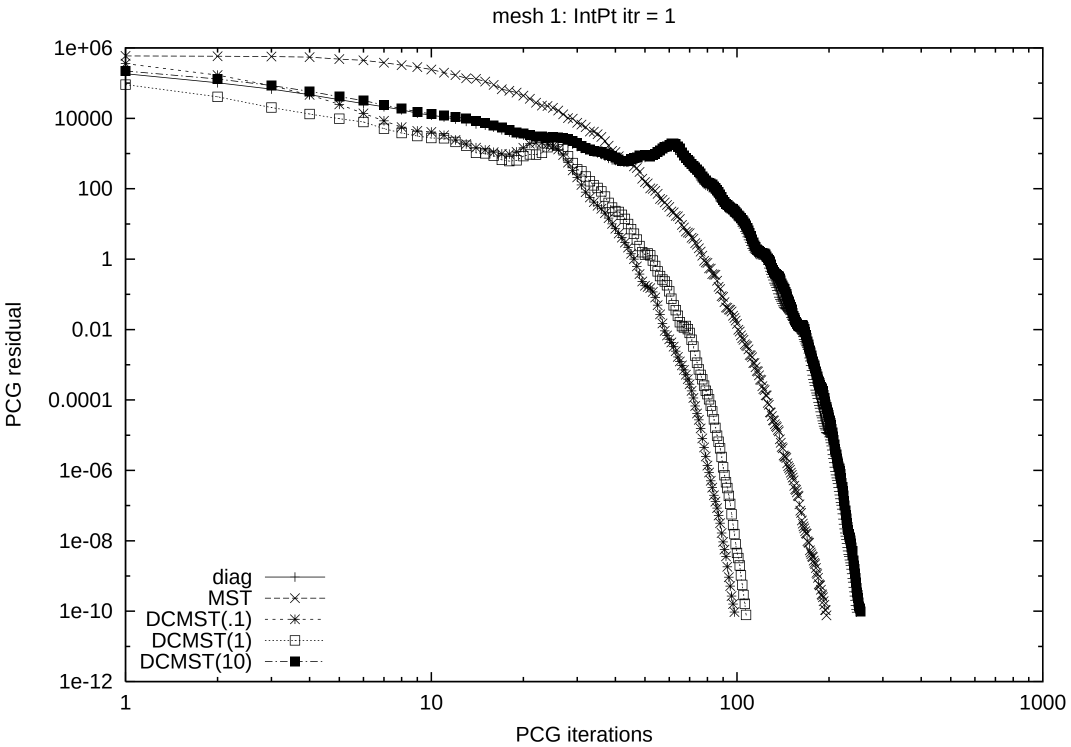

FIG. 5.1. *Convergence of PCG on 1024-node* `mesh-1` *instance on interior point iteration 1.*

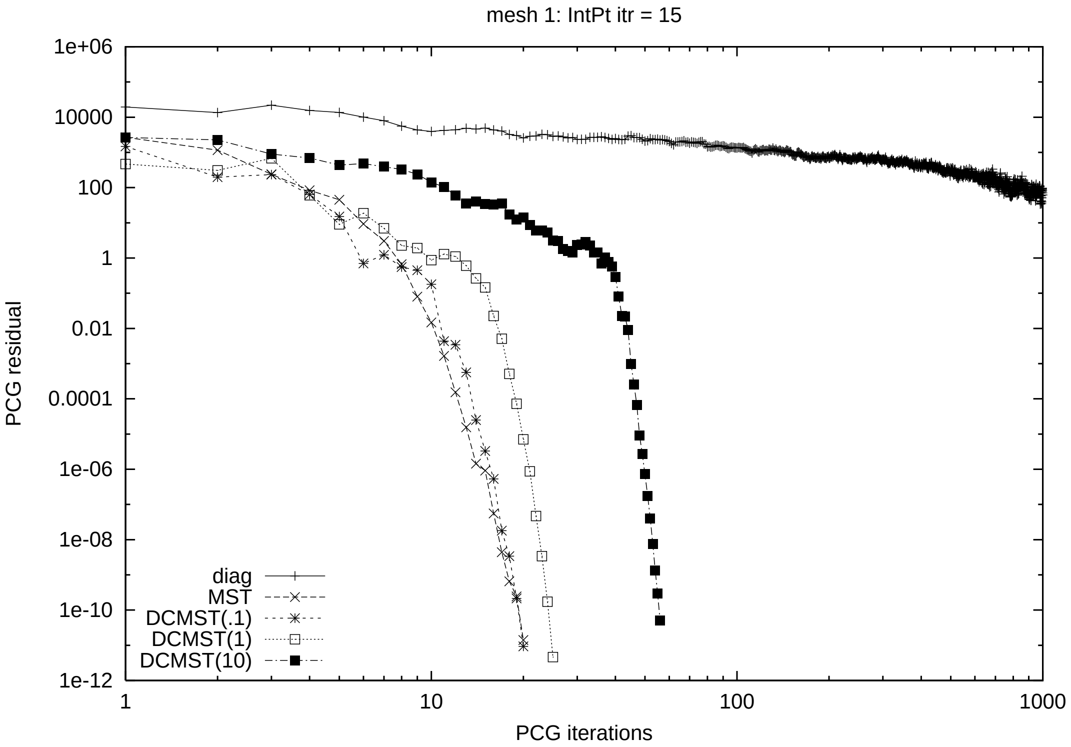

FIG. 5.2. *Convergence of PCG on 1024-node* `mesh-1` *instance on interior point iteration 15.*

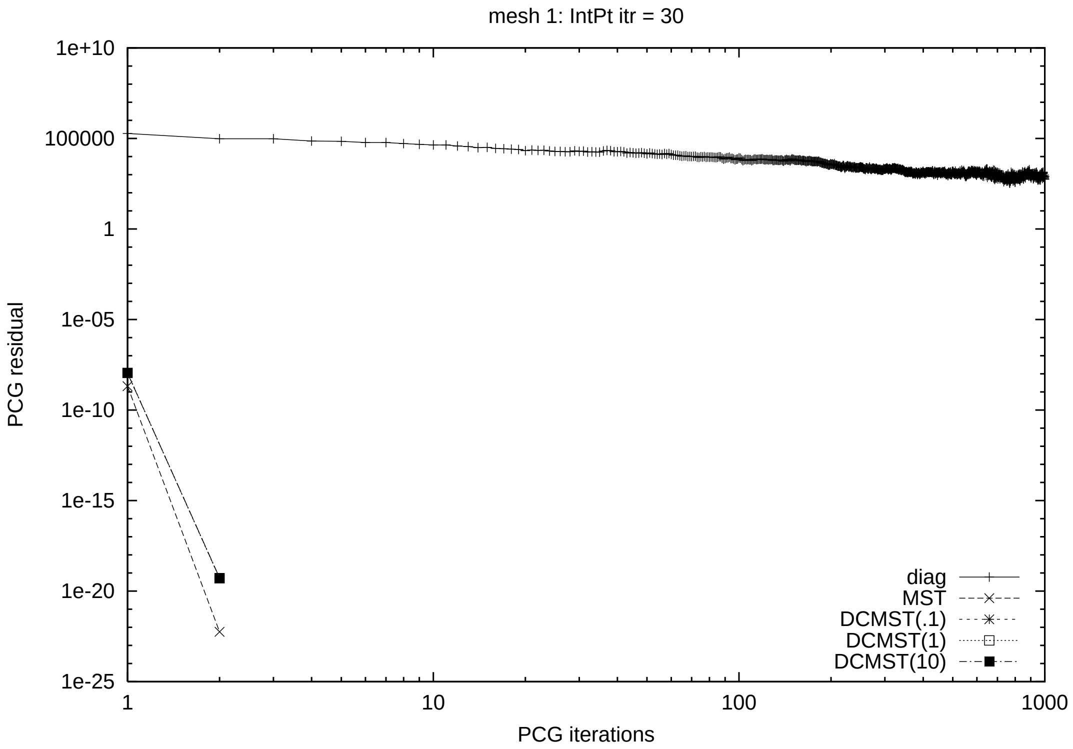

FIG. 5.3. *Convergence of PCG on 1024-node* `mesh-1` *instance on interior point iteration 30.*

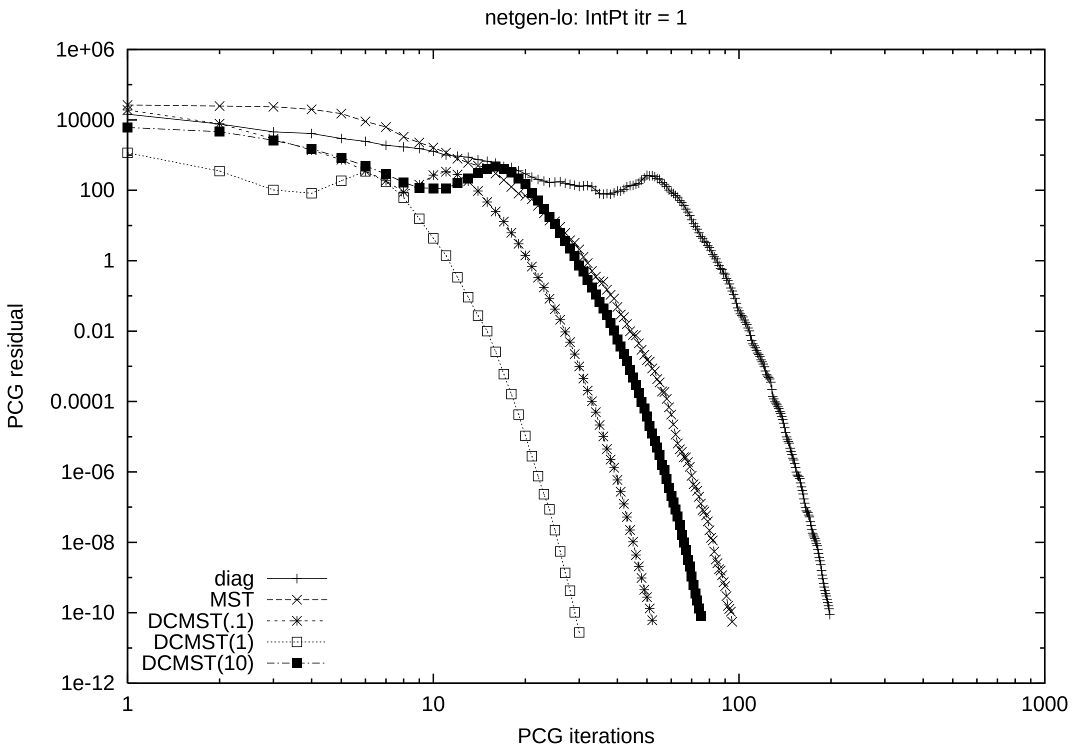

FIG. 5.4. *Convergence of PCG on 512-node* `netgen-lo` *instance on interior point iteration 1.*

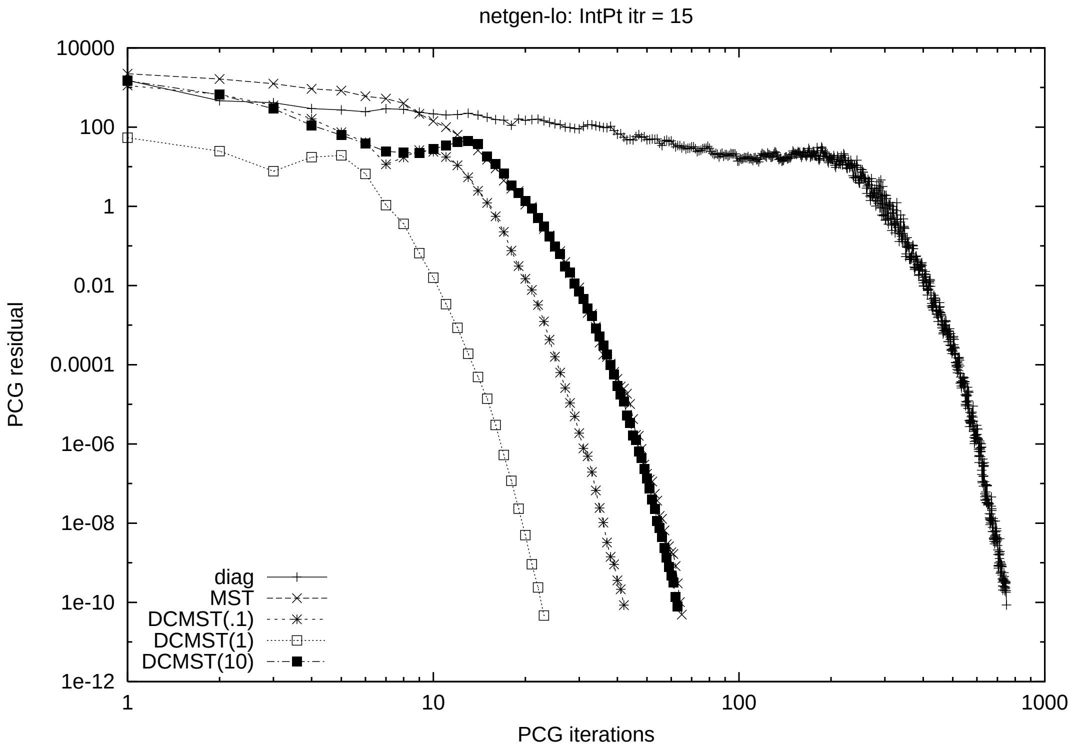

FIG. 5.5. *Convergence of PCG on 512-node* `netgen-lo` *instance on interior point iteration 15.*

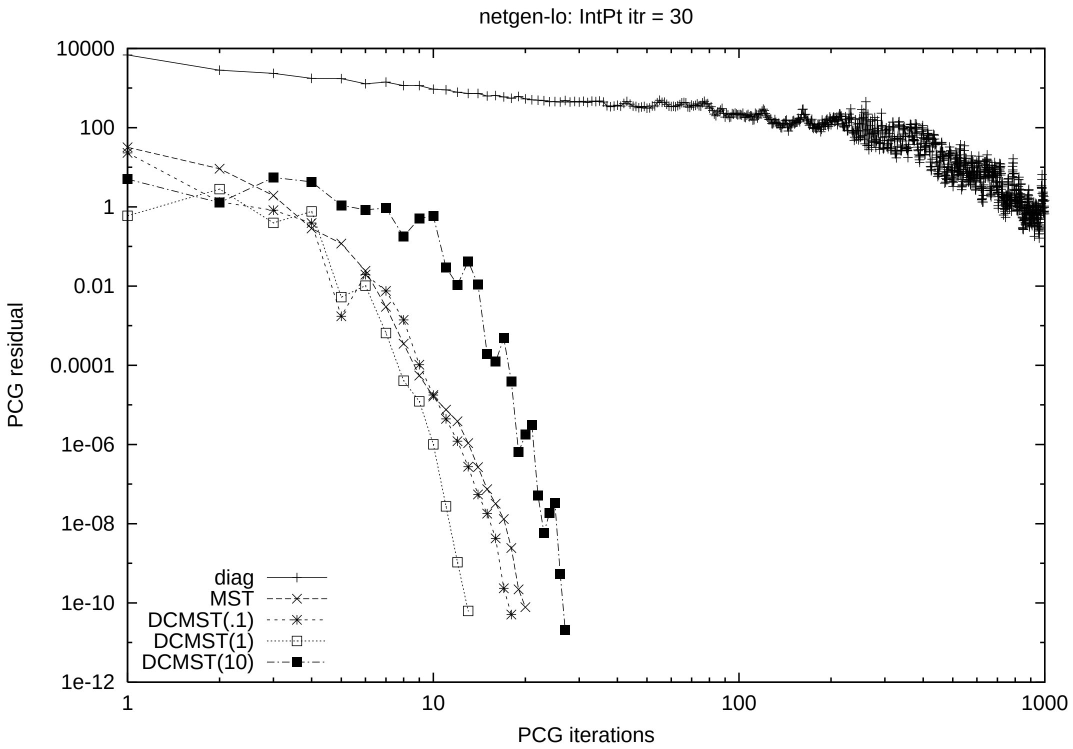

FIG. 5.6. *Convergence of PCG on 512-node* `netgen-lo` *instance on interior point iteration 30.*

As before, if we assume that the conditions (2.16) hold, it is possible to show that the condition number of $M^{-1}(A\Theta A^\top)$ is uniformly bounded in the last stages of the algorithm for both the MST and DCMST preconditioners. This is shown in the next theorem.

COROLLARY 5.3. *Suppose that conditions (4.3) hold with μ small. Let $\bar{\mathcal{G}} = (\mathcal{N}, \mathcal{T})$ be a maximum spanning tree of $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, $\bar{\mathcal{T}} = \mathcal{A} - \mathcal{T}$ and $M = A_{\mathcal{T}}\Theta_{\mathcal{T}}A_{\mathcal{T}}^\top + D$, where $D = \phi diag(A_{\bar{\mathcal{T}}}\Theta_{\bar{\mathcal{T}}}A_{\bar{\mathcal{T}}}^\top)$ with $0 \le \phi \le 2$. Then, at the later iterations of the truncated Newton method*

$$\kappa(M^{-1}A\Theta A^\top) \le \begin{cases} 1 + \tau_1 \frac{1}{\frac{1}{2\gamma\mu^2}+\frac{1}{2}} + \tau_2 \frac{1}{\left(\frac{1}{2\gamma\mu^2}+\frac{1}{2}\right)^2} & \text{if } |\mathcal{B}| = m \\ 1 + \tau_1 + \tau_2 & \text{if } |\mathcal{B}| > m \\ 1 + \tau_1 + \tau_2 & \text{if } |\mathcal{B}| < m, \end{cases}$$

where

$$\tau_1 = m^2(n-m)\frac{1}{\gamma\tau^4} \quad \text{and} \quad \tau_2 = \begin{cases} m^2(m-1)(n-m)^2(\frac{1}{\gamma\tau^4})^2 & \text{if} \quad \phi > 0 \\ 0 & \text{if} \quad \phi = 0 \end{cases}$$

Proof. From Theorem 5.2 we have

$$\begin{aligned}\kappa(M^{-1}A\Theta A^\top) &\le \left(1 + m(m-1)\frac{\overline{\lambda}(D)}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right)\left(1 + m(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right) \\ &\le \left(1 + m(m-1)(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right)\left(1 + m(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right) \\ &\le 1 + m^2(n-m)\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})} + m^2(m-1)(n-m)^2\left(\frac{\overline{\lambda}(\Theta_{\bar{\mathcal{T}}})}{\underline{\lambda}(\Theta_{\mathcal{T}})}\right)^2.\end{aligned}$$

The upper bounds are obtained by using this inequality an the same arguments of Corollary 4.7. □

This last result shows that, even in the presence of primal degeneracy, the condition number of $M^{-1}A\Theta A^\top$ is uniformly bounded after μ becomes small. Furthermore, when the problem is nondegenerate, $\kappa(M^{-1}A\Theta A^\top) \to 1$ as $\mu \to 0$. This nice property of the MST and DCMST preconditioners at the later stages of the interior point methods has been confirmed in practice.

6. Computational experience. In this section, we illustrate the theoretical results presented in this paper with examples solving linear systems with five different preconditioners: diagonal, maximum spanning tree (MST), and diagonally compensated maximum spanning tree (DCMST) using three values for the parameter ϕ (.1, 1, and 10).

Each preconditioner was tested on six instances taken from the First DIMACS Implementation Challenge [9] that were solved with PDNET in [14]: a 512-node instance from the Grid-Density-8 class, a 512-node instance from the Grid-Density-16 class, a 514-node instance from the Grid-Long class, a 514-node instance from the Grid-Wide class, a 1024-node instance from the Mesh-1 class, and a 512-node instance from the Netgen-Lo class.

For each instance, the preconditioners were used to solve three linear systems. The first system corresponds to an early interior point iteration (iteration 1). The second system is that of an intermediate iteration (iteration 15). The third system is of a late iteration (iteration 30). For each problem, a set of linear systems was generated by running PDNET with the default settings in [14]. In this way, identical systems were solved with each of the preconditioners. The conjugate gradient method stops when the norm of the residual satisfies a predetermined tolerance (we use the tolerance 10^{-10}). A limit of 1000 conjugate gradient iterations was imposed, with the purpose of detecting numerical difficulties.

Figures 6.1, 6.2, and 6.3 show the residual as a function of the conjugate gradient iteration for the 512-node instance from the Grid-Density-8 class and Figures 6.4, 6.5, and 6.6 show the residual as a function of the conjugate gradient iteration for the 512-node instance from the Grid-Density-16 class. On these six linear systems, the maximum spanning tree preconditioner is the only preconditioner to lower the residual monotonically. Diagonal compensation of the maximum

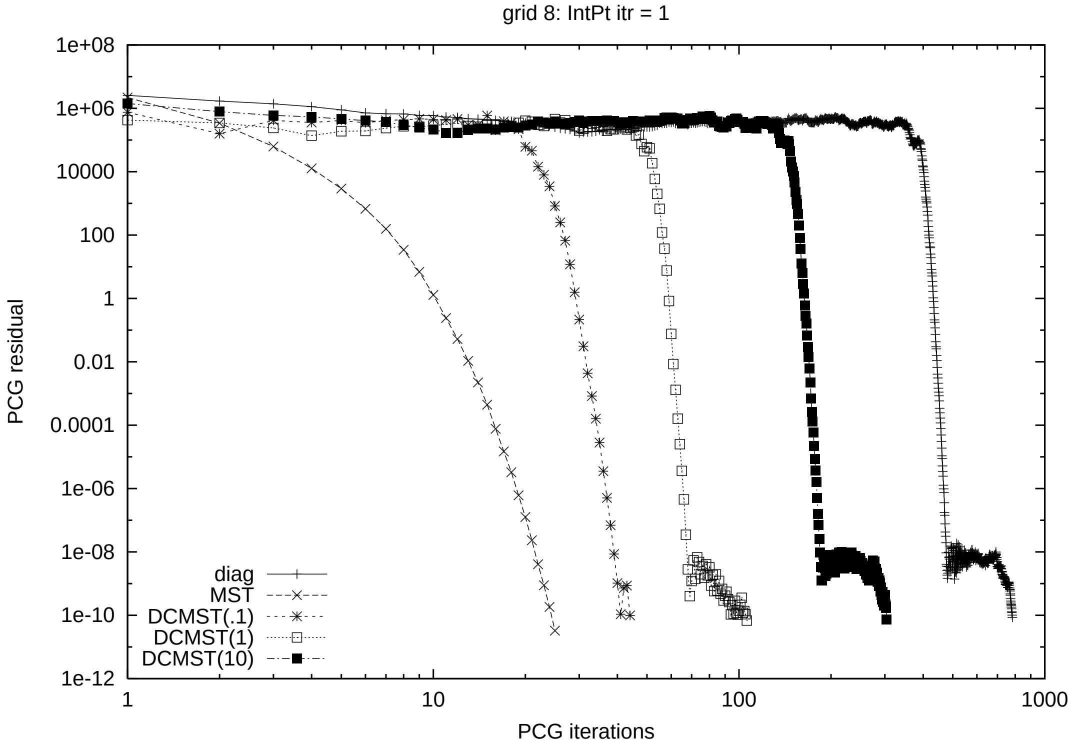

FIG. 6.1. *Convergence of PCG on 512-node* `grid-8` *instance on interior point iteration 1.*

spanning tree preconditioner degrades performance. Pure diagonal preconditioning can only solve the instances from iteration 1.

Figures 6.7, 6.8, and 6.9 show the residual as a function of the conjugate gradient iteration for the 514-node instance from the Grid-Long class. Figures 6.10, 6.11, and 6.12 show the residual as a function of the conjugate gradient iteration for the 514-node instance from the Grid-Wide class. Figures 5.1, 5.2, and 5.3 show the residual as a function of the conjugate gradient iteration for the 1024-node instance from the Mesh- 1 class. On these instances, diagonal compensation appears to help more on the cases for which the diagonal preconditioner is able to solve the system, i.e. the iteration 1 systems. When the diagonal preconditioner fails to solve the system, diagonal compensation degrades performance. Again, on eight of these nine instances, the only preconditioner able to decrease the residual monotonically is the maximum spanning tree. The only exception is on the iteration 30 Mesh-1 instance. The nonmonotone convergence patterns of the diagonally compensated preconditioners mimic those of the diagonal preconditioner.

Figures 5.4, 5.5, and 5.6 show the residual as a function of the conjugate gradient iteration for the 512-node instance from the Netflo-Lo class. The behavior of the diagonally compensated preconditioners when the diagonal preconditioner is able to solve the linear system is similar to the previous cases. Again, on these three instances, the only preconditioner able to monotonically decrease the residual is the maximum spanning tree. Oddly, even when the diagonal preconditioner fails to solve the system (iteration 30 instance), diagonal compensation appears to help.

7. Concluding remarks. As suggested by the theoretical bounds developed for the system matrix condition number, the MST preconditioner dominates the diagonal preconditioner in practice. However, countering the theoretical relationships of the condition number upper bounds, the DCMST preconditioner is sometimes slightly more effective than MST.

The computational experiments suggest that the effectiveness of the DCMST preconditioner depends strongly on the behavior of the diagonal preconditioner applied to the same linear system. In those cases where a PCG with a diagonal preconditioner is able to solve the linear system, a diagonal compensation appears to improve the performance of the maximum spanning tree

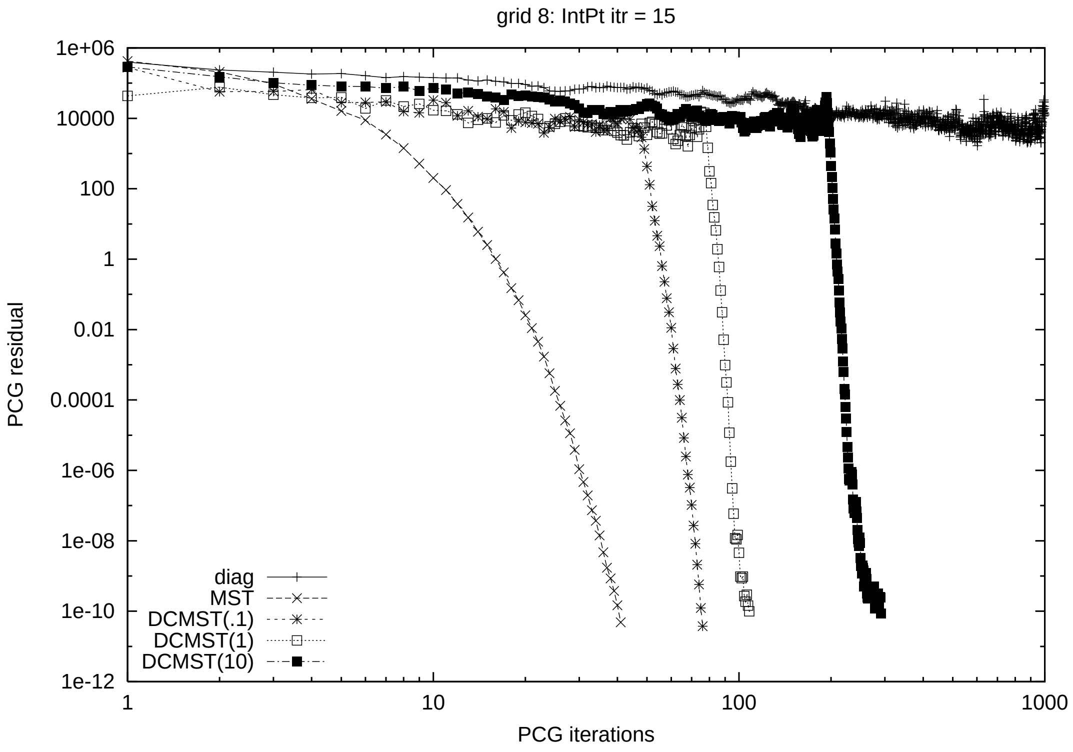

FIG. 6.2. *Convergence of PCG on 512-node* `grid-8` *instance on interior point iteration 15.*

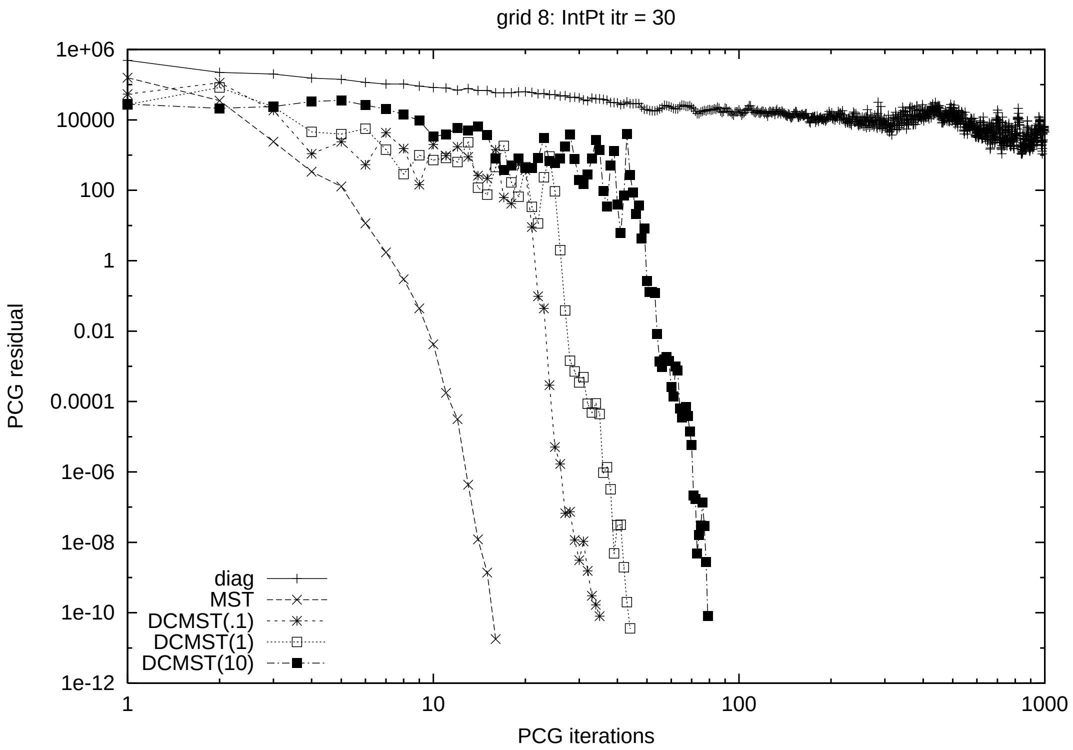

FIG. 6.3. *Convergence of PCG on 512-node* `grid-8` *instance on interior point iteration 30.*

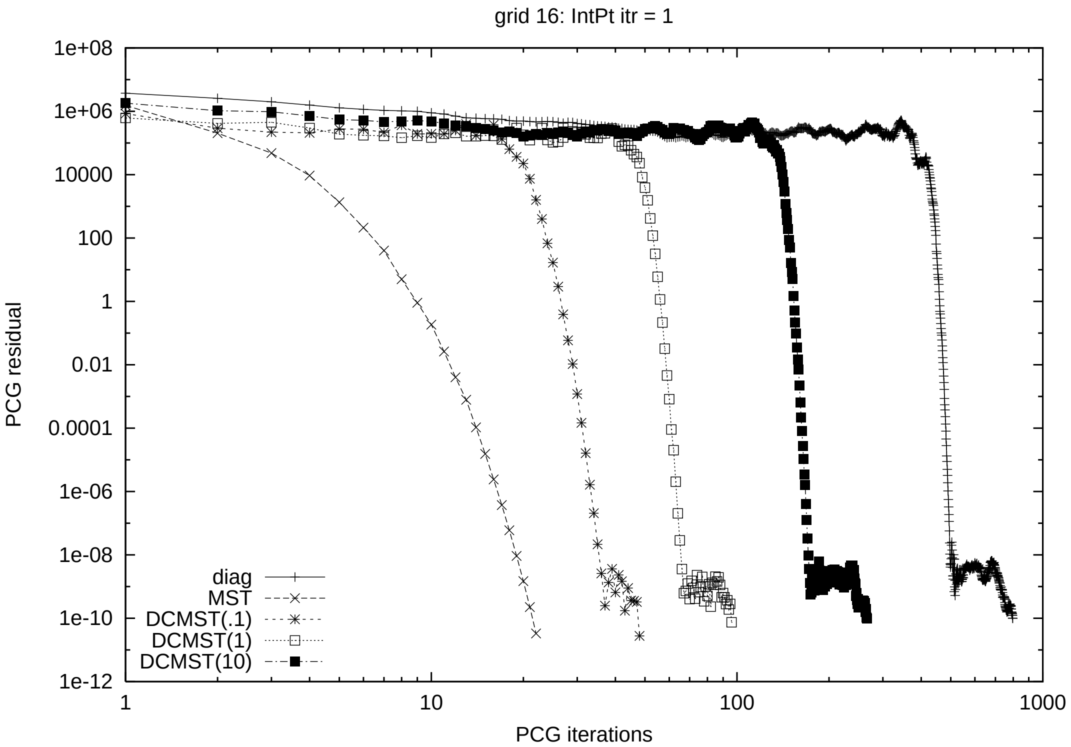

Fig. 6.4. *Convergence of PCG on 512-node* `grid-16` *instance on interior point iteration 1.*

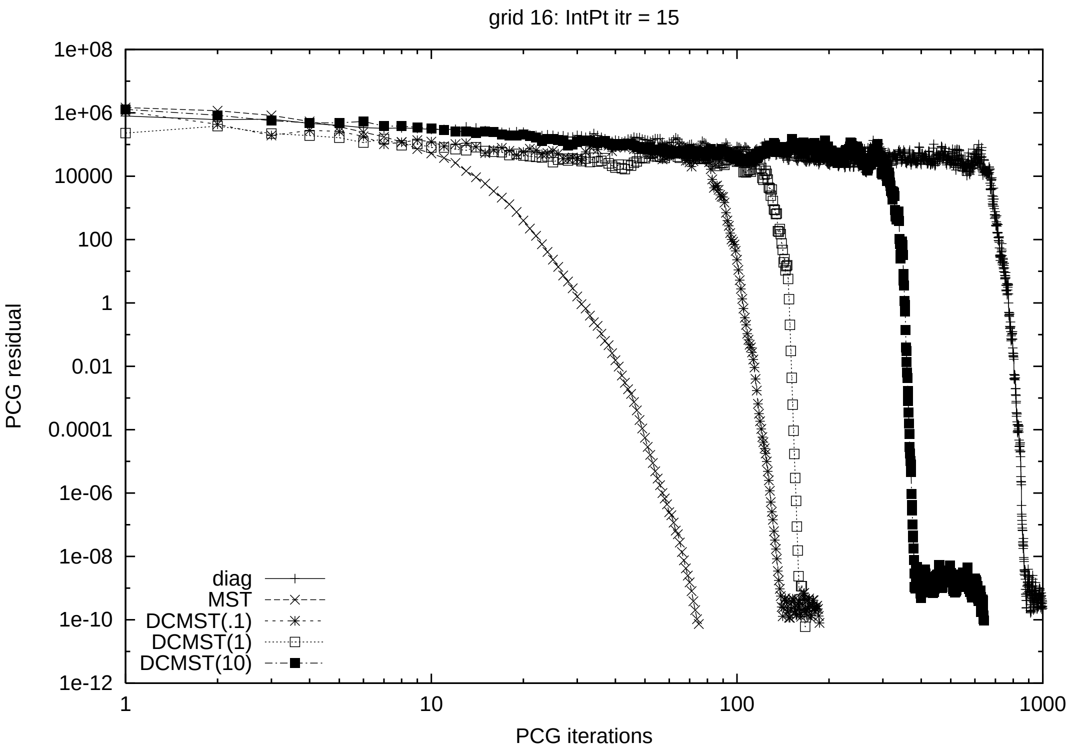

Fig. 6.5. *Convergence of PCG on 512-node* `grid-16` *instance on interior point iteration 15.*

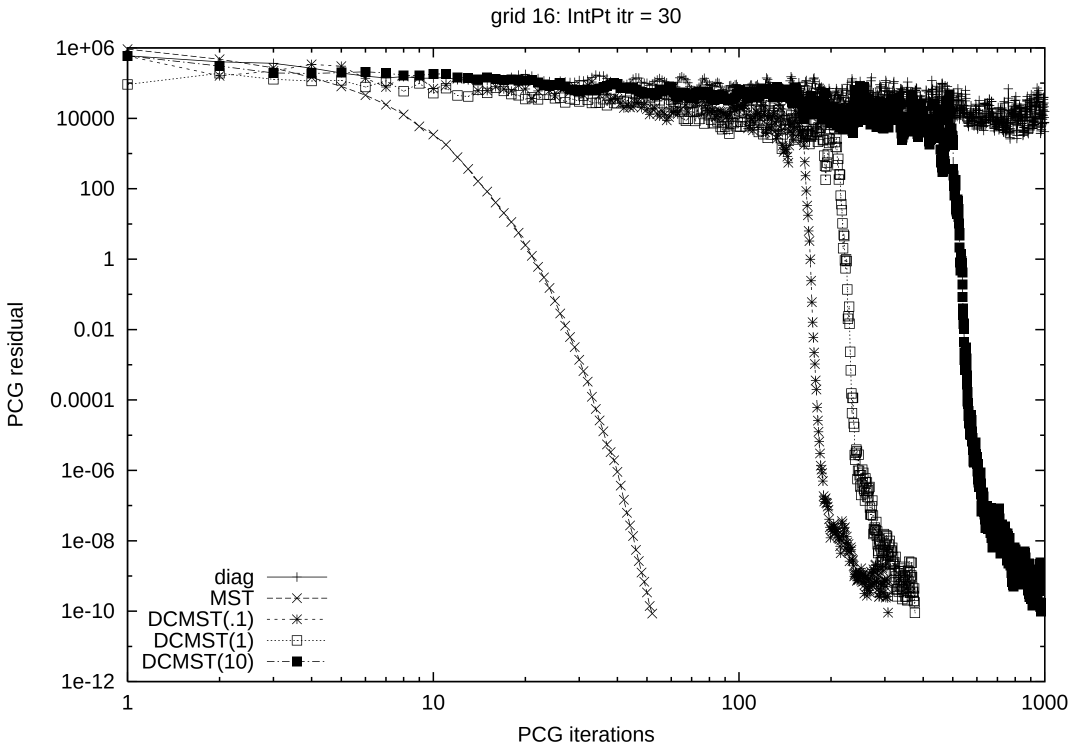

FIG. 6.6. *Convergence of PCG on 512-node* `grid-16` *instance on interior point iteration 30.*

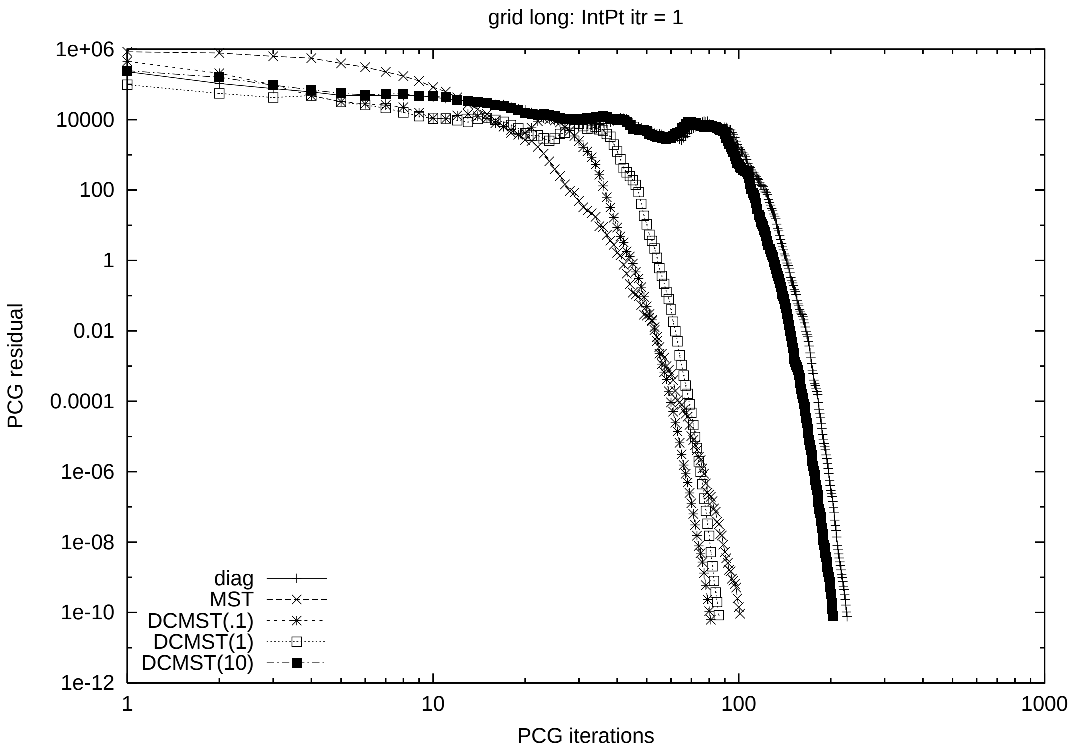

FIG. 6.7. *Convergence of PCG on 514-node* `grid-long` *instance on interior point iteration 1.*

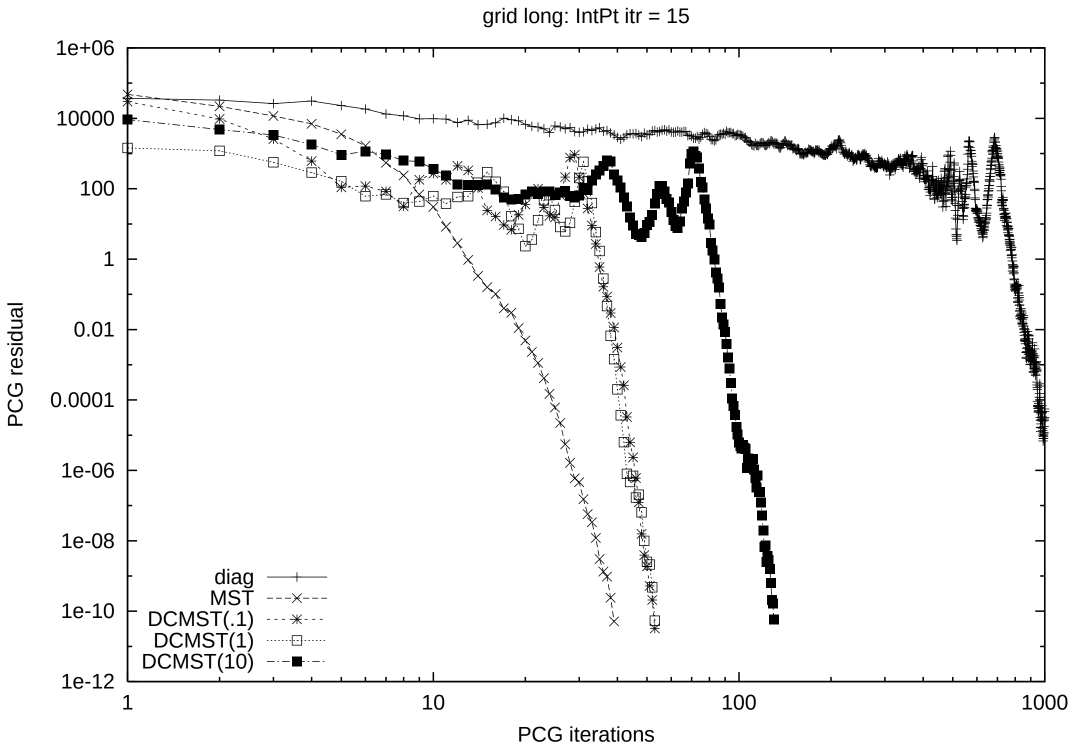

FIG. 6.8. *Convergence of PCG on 514-node* `grid-long` *instance on interior point iteration 15.*

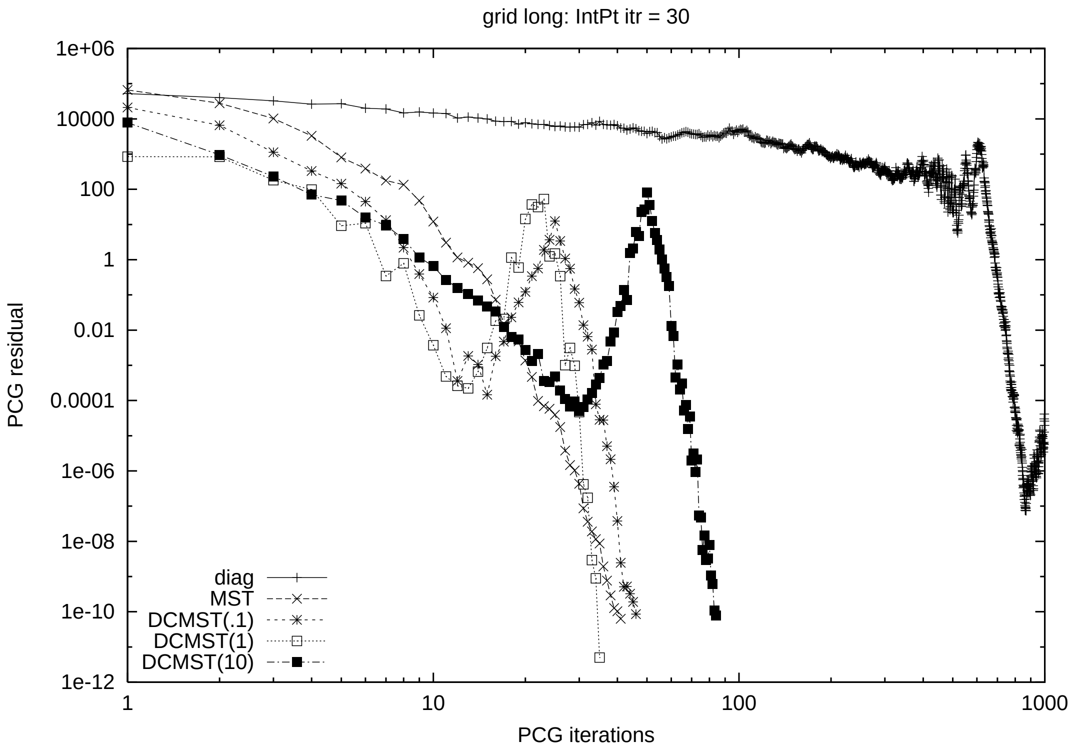

FIG. 6.9. *Convergence of PCG on 514-node* `grid-long` *instance on interior point iteration 30.*

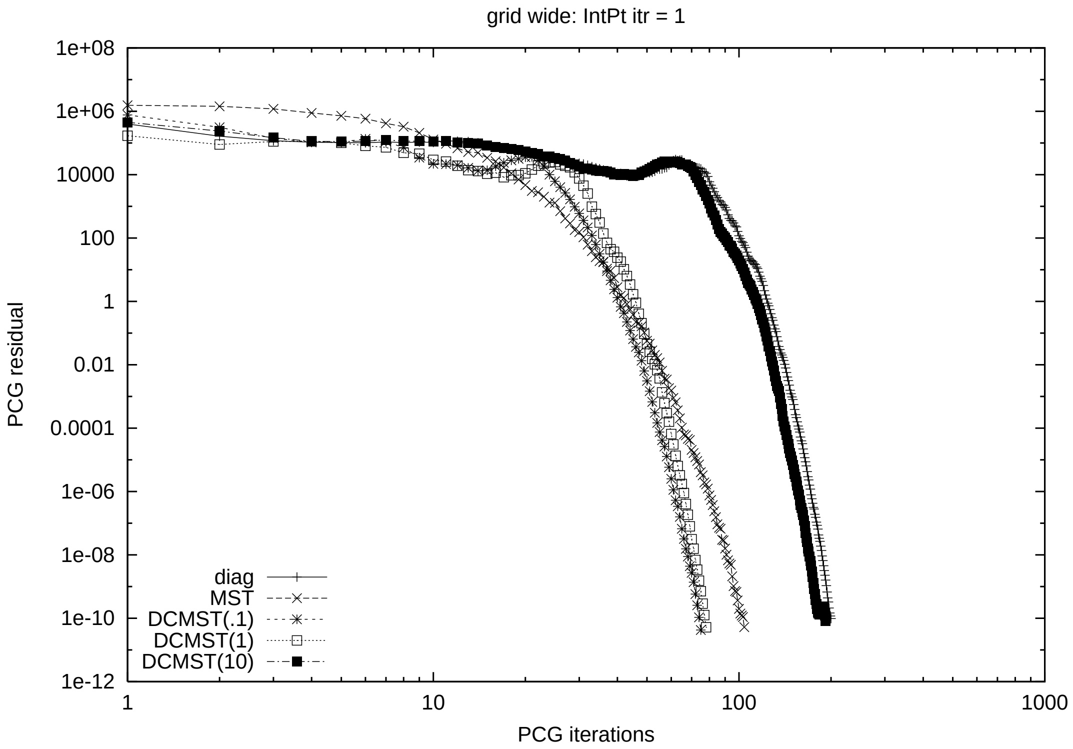

FIG. 6.10. *Convergence of PCG on 514-node* `grid-wide` *instance on interior point iteration 1.*

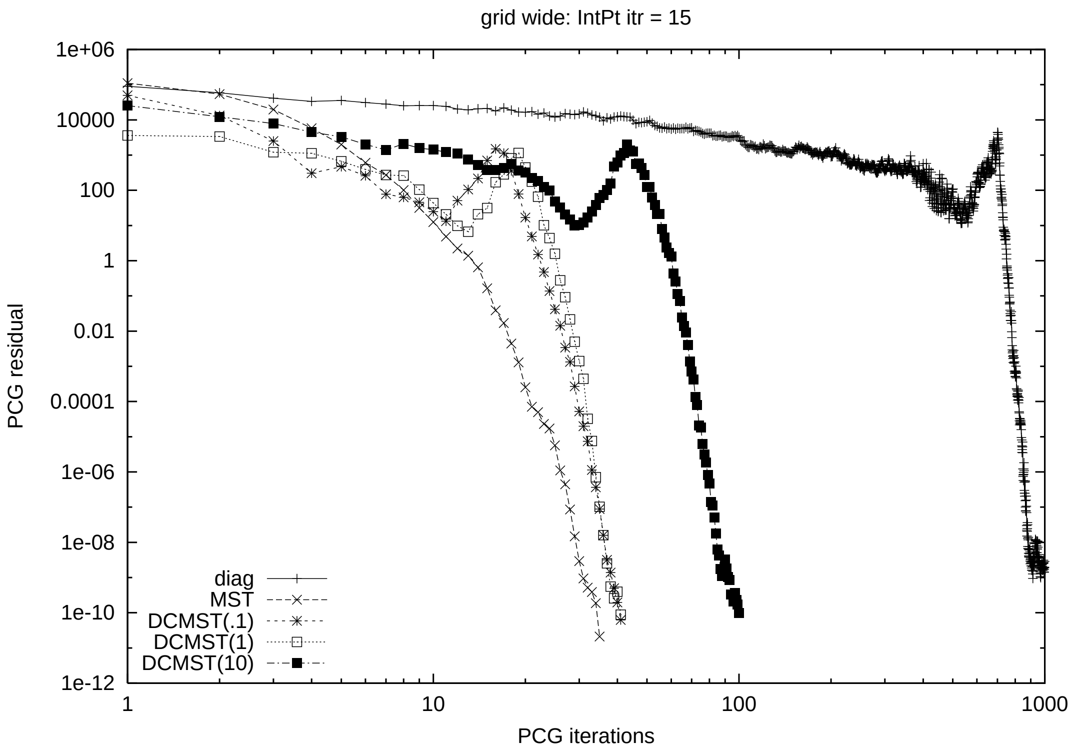

FIG. 6.11. *Convergence of PCG on 514-node* `grid-wide` *instance on interior point iteration 15.*

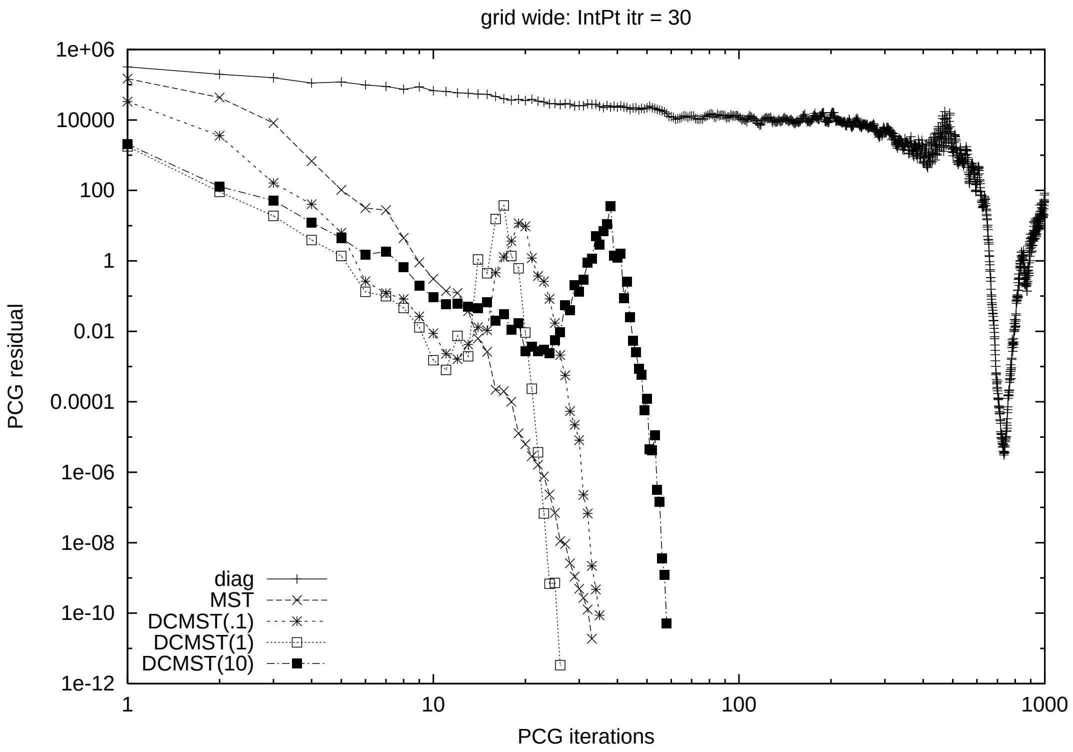

FIG. 6.12. *Convergence of PCG on 514-node* `grid-wide` *instance on interior point iteration 30.*

preconditioner. For the instances considered, the diagonal preconditioner was only effective during the early interior point iterations and only for some instances.

The MST preconditioner appears to be the most robust. It was the only preconditioner to monotonically reduced the residual on all linear systems considered. This fact, along with the strong dependence of the DCMST preconditioners on the suitability of the diagonal preconditioner, leads us to conclude that the loss of robustness of the DCMST preconditioners outweighs any gains that may be achieved with these preconditioners.

REFERENCES

[1] I. Adler, N. Karmarkar, M. Resende, and G. Veiga, *Data structures and programming techniques for the implementation of Karmarkar's algorithm*, ORSA Journal on Computing, 1 (1989), pp. 84–106.

[2] N. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[4] V. Baryamureeba and T. Steihaugh, *On the convergence of an inexact primal-dual interior point method for linear programming*, tech. report, Department of Informatics - University of Bergen, Bergen - Norway, March 2000.

[5] S. Bellavia and M. Maconni, *An inexact interior-point method for monotone NCP*, tech. report, University of Firenze, Italy, 1999.

[6] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.

[7] R. S. Dembo and T. Steihaug, *Truncated-Newton algorithms for large scale unconstrained optimization*, Mathematical Programming, 26 (1983), pp. 190–212.

[8] R. Freund, F. Jarre, and S. Mizuno, *Convergence of a class of inexact interior-point algorithms for linear programs*, tech. report, Lucent Technologies, 600 Mountain Avenue, Murray Hill, New Jersey 07974, USA, 1996.

[9] D. Johnson and C. McGeoch, eds., *Network flows and matching: First DIMACS implementation challenge*, vol. 12 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1993.

[10] M. Kojima, N. Megiddo, and S. Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Mathematical Programming, 61 (1993), pp. 263–280.

[11] S. Mehrotra and J. Wang, *Conjugate gradient based implementation of interior point methods for network*

flow problems, in Linear and Nonlinear Conjugate Gradient Related Methods, L. Adams and J. Nazareth, eds., SIAM, 1995.

[12] S. Mizuno and F. Jarre, *Global and polynomial time convergence of an infeasible interior point algorithm using inexact computation*, Research Memorandum 605, Institute of Statistical Mathematics, Tokyo, USA, 1996.

[13] L. Portugal, F. Bastos, J. Júdice, J. Paixão, and T. Terlaky, *An investigation of interior point algorithms for the linear transportation problem*, SIAM Journal on Scientific Computing, 17 (1996), pp. 1202–1223.

[14] L. Portugal, M. Resende, G. Veiga, and J. Júdice, *An efficient implementation of an infeasible primal-dual network flow method*, Networks, 35 (2000), pp. 91–108.

[15] M. Resende and G. Veiga, *An efficient implementation of a network interior point method*, in Network Flows and Matching: First DIMACS Implementation Challenge, D. S. Johnson and C. C. McGeoch, eds., vol. 12 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1993, pp. 299–348.

[16] ———, *An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks*, SIAM Journal on Optimization, 3 (1993), pp. 516–537.

[17] S. Wright, *Stability of linear algebra computations in interior point methods for linear programming*, Preprint ANL/MCS- P446-0694, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA, 1994.

[18] ———, *Modified cholesky factorizations in interior point algorithms for linear programming*, Preprint ANL/MCS- P600-0596, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA, 1996.

[19] ———, *Primal dual interior point methods*, SIAM, 1997.

[20] Q.-J. Yeh, *A reduced dual affine scaling algorithm for solving assignment and transportation problems*, PhD thesis, Columbia University, New York, NY, 1989.

[21] Y. Zhang, *On the convergenvce of a class of infeasible interior point methods for the horizontal linear complementarity problem*, SIAM Journal on Optimization, 4 (1994), pp. 208–227.