

A Newton-CG Augmented Lagrangian Method for Semidefinite Programming*

Xin-Yuan Zhao [†] Defeng Sun [‡] Kim-Chuan Toh [§]

March 12, 2008; Revised, February 03, 2009

Abstract. We consider a Newton-CG augmented Lagrangian method for solving semidefinite programming (SDP) problems from the perspective of approximate semismooth Newton methods. In order to analyze the rate of convergence of our proposed method, we characterize the Lipschitz continuity of the corresponding solution mapping at the origin. For the inner problems, we show that the positive definiteness of the generalized Hessian of the objective function in these inner problems, a key property for ensuring the efficiency of using an inexact semismooth Newton-CG method to solve the inner problems, is equivalent to the constraint nondegeneracy of the corresponding dual problems. Numerical experiments on a variety of large scale SDPs with the matrix dimension n up to 4,110 and the number of equality constraints m up to 2,156,544 show that the proposed method is very efficient. We are also able to solve the SDP problem **fap36** (with $n = 4,110$ and $m = 1,154,467$) in the Seventh DIMACS Implementation Challenge much more accurately than previous attempts.

Keywords: Semidefinite programming, Augmented Lagrangian, Semismoothness, Newton's method, Iterative solver.

1 Introduction

Let \mathcal{S}^n be the linear space of all $n \times n$ symmetric matrices and \mathcal{S}_+^n be the cone of all $n \times n$ symmetric positive semidefinite matrices. The notation $X \succeq \mathbf{0}$ means that X is a symmetric

*A preliminary version of this paper was presented at ICCOPT II, Hamilton, Canada, August 13, 2007.

[†]Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (zhaoxinyuan@nus.edu.sg).

[‡]Department of Mathematics and Risk Management Institute, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matsundf@nus.edu.sg). This author's research is supported in part by Academic Research Fund under Grant R-146-000-104-112.

[§]Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matttohkc@nus.edu.sg); and Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576. 117576. This author's research is supported in part by Academic Research Grant R-146-000-076-112.

positive semidefinite matrix. This paper is devoted to studying an augmented Lagrangian method for solving the following semidefinite programming (SDP) problem

$$(D) \quad \min \left\{ b^T y \mid \mathcal{A}^* y - C \succeq \mathbf{0} \right\},$$

where $C \in \mathcal{S}^n$, $b \in \mathbb{R}^m$, \mathcal{A} is a linear operator from \mathcal{S}^n to \mathbb{R}^m , and $\mathcal{A}^* : \mathbb{R}^m \rightarrow \mathcal{S}^n$ is the adjoint of \mathcal{A} . The dual of (D) takes the form

$$(P) \quad \max \left\{ \langle C, X \rangle \mid \mathcal{A}(X) = b, \quad X \succeq \mathbf{0} \right\}.$$

Given a penalty parameter $\sigma > 0$, the *augmented Lagrangian* function for problem (D) is defined as

$$L_\sigma(y, X) = b^T y + \frac{1}{2\sigma} (\|\Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^* y - C))\|^2 - \|X\|^2), \quad (y, X) \in \mathbb{R}^m \times \mathcal{S}^n, \quad (1)$$

where for any closed convex set \mathcal{D} in a finite dimensional real vector space \mathcal{X} equipped with a scalar inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$, $\Pi_{\mathcal{D}}(\cdot)$ is the metric projection operator over \mathcal{D} , i.e., for any $Y \in \mathcal{X}$, $\Pi_{\mathcal{D}}(Y)$ is the unique optimal solution to the following convex optimization problem

$$\min \left\{ \frac{1}{2} \langle Z - Y, Z - Y \rangle \mid Z \in \mathcal{D} \right\}.$$

Note that, since $\|\Pi_{\mathcal{D}}(\cdot)\|^2$ is continuously differentiable [44], the augmented Lagrangian function defined in (1) is continuously differentiable. In particular, for any given $X \in \mathcal{S}^n$, we have

$$\nabla_y L_\sigma(y, X) = b - \mathcal{A} \Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^* y - C)). \quad (2)$$

For given $X^0 \in \mathcal{S}^n$, $\sigma_0 > 0$, and $\rho > 1$, the augmented Lagrangian method for solving problem (D) and its dual (P) generates sequences $\{y^k\} \subset \mathbb{R}^m$ and $\{X^k\} \subset \mathcal{S}^n$ as follows

$$\begin{cases} y^{k+1} \approx \arg \min_{y \in \mathbb{R}^m} L_{\sigma_k}(y, X^k), \\ X^{k+1} = \Pi_{\mathcal{S}_+^n}(X^k - \sigma_k(\mathcal{A}^* y^{k+1} - C)), \quad k = 0, 1, 2, \dots \\ \sigma_{k+1} = \rho \sigma_k \text{ or } \sigma_{k+1} = \sigma_k, \end{cases} \quad (3)$$

For a general discussion on the augmented Lagrangian method for solving convex optimization problems and beyond, see [32, 33].

For small and medium sized SDP problems, it is widely accepted that interior-point methods (IPMs) with direct solvers are generally very efficient and robust. For large-scale SDP problems with m large and n moderate (say less than 5,000), the limitations of IPMs with direct solvers become very severe due to the need of computing, storing, and factorizing the $m \times m$ Schur complement matrix. In order to alleviate these difficulties, Toh and Kojima [39] and Toh [40] proposed inexact IPMs using an iterative solver to compute the search

direction at each iteration. The approach in [40] was demonstrated to be able to solve large sparse SDPs with m up to 125,000 in a few hours. Kočvara and Stingl [17] used a modified barrier method (a variant of the Lagrangian method) combined with iterative solvers for linear SDP problems having only inequality constraints and reported computational results in the code PENNON [16] with m up to 125,000. More recently, Malick, Povh, Rendl, and Wiegale [19] applied the Moreau-Yosida regularization approaches to solve SDP problems.

In this paper, we study an augmented Lagrangian dual approach to solve large scale SDPs with m large (say, up to a few millions) but n moderate (say, up to 5,000). Our approach is similar in spirit as those in [17] and [19], where the idea of augmented Lagrangian methods (or methods of multipliers in general) was heavily exploited. However, our points of view of employing the augmented Lagrangian methods are fundamentally different from them in solving both the outer and inner problems. It has long been known that the augmented Lagrangian method for convex problems is a gradient ascent method applied to the corresponding dual problems [30]. This inevitably leads to the impression that the augmented Lagrangian method for solving SDPs may converge slowly for the outer iteration sequence $\{X^k\}$. In spite of that, under mild conditions, a linear rate of convergence analysis is available (superlinear convergence is also possible when σ_k goes to infinity, which should be avoided in numerical implementations) [33]. However, recent studies conducted by Sun, Sun, and Zhang [37] and Chan and Sun [8] revealed that under the constraint nondegenerate conditions for (D) and (P) (i.e., the dual nondegeneracy and primal nondegeneracy in the IPMs literature, e.g., [1]), respectively, the augmented Lagrangian method can be locally regarded as an approximate generalized Newton method applied to a semismooth equation. It is this connection that inspired us to investigate the augmented Lagrangian method for SDPs.

The objective functions $L_{\sigma_k}(\cdot, X^k)$ in the inner problems of the augmented Lagrangian method (3) are convex and continuously differentiable but not twice continuously differentiable (cf. (2)) due to the fact that $\Pi_{S_+^n}(\cdot)$ is not continuously differentiable. It seems that Newton's method can not be applied to solve the inner problems. However, since $\Pi_{S_+^n}(\cdot)$ is strongly semismooth [36], the superlinear (quadratic) convergence analysis of generalized Newton's method established by Kummer [18], and Qi and Sun [26] for solving semismooth equations may be used to get fast convergence for solving the inner problems. In fact, the quadratic convergence and superb numerical results of the generalized Newton's method combined with the conjugate gradient (CG) method reported in [25] for solving a related problem strongly motivated us to study the semismooth Newton-CG method (see Section 3) to solve the inner problems.

In [32, 33], Rockafellar established a general theory on the global convergence and local linear rate of convergence of the sequence generated by the augmented Lagrangian method for solving convex optimization problems including (D) and (P) . In order to apply the general results in [32, 33], we characterize the Lipschitz continuity of the solution mapping for (P) defined in [33] at the origin in terms of the second order sufficient condition, and the extended strict primal-dual constraint qualification for (P) . In particular, under the uniqueness of Lagrange multipliers, we establish the equivalence among the Lipschitz continuity of the

solution mapping at the origin, the second order sufficient condition, and the strict primal-dual constraint qualification. As for the inner problems in (3), we show that the constraint nondegeneracy for the corresponding dual problems is equivalent to the positive definiteness of the generalized Hessian of the objective functions in the inner problems. This is important for the success of applying an iterative solver to the generalized Newton equations in solving these inner problems. The differential structure of the nonsmooth metric projection operator $\Pi_{\mathcal{S}_+^n}(\cdot)$ in the augmented Lagrangian function L_σ plays a key role in achieving this result.

Besides the theoretical results we establish for the Newton-CG augmented Lagrangian (in short, SDPNAL) method proposed in this paper, we also demonstrate convincingly that with efficient implementations, the SDPNAL method can solve some very large SDPs, with a moderate accuracy, much more efficiently than the best alternative methods such as the inexact interior-point methods in [40], the modified barrier method in [17], the boundary-point method in [19], as well as the dedicated augmented Lagrangian method for solving SDPs arising from the lift-and-project procedure of Lovász and Schrijver [5].

The remaining parts of this paper are as follows. In Section 2, we give some preliminaries including a brief introduction about concepts related to the method of multipliers and the characterizations of the Lipschitz continuity of the solution mapping for problem (P) at the origin. In Section 3, we introduce a semismooth Newton-CG method for solving the inner optimization problems and analyze its global and local superlinear (quadratic) convergence for solving these inner problems. Section 4 presents the Newton-CG augmented Lagrangian dual approach and its linear rate of convergence. Section 5 is on numerical issues of the semismooth Newton-CG algorithm. We report numerical results in Sections 6 and 7 for a variety of large scale linear SDP problems and make final conclusions in Section 8.

2 Preliminaries

From [32, 33], we know that the augmented Lagrangian method can be expressed in terms of the method of multipliers for (D). For the sake of subsequent discussions, we introduce related concepts to this.

Let $l(y, X) : \mathbb{R}^m \times \mathcal{S}^n \rightarrow \mathbb{R}$ be the ordinary Lagrangian function for (D) in extended form:

$$l(y, X) = \begin{cases} b^\top y - \langle X, \mathcal{A}^* y - C \rangle & \text{if } y \in \mathbb{R}^m \text{ and } X \in \mathcal{S}_+^n, \\ -\infty & \text{if } y \in \mathbb{R}^m \text{ and } X \notin \mathcal{S}_+^n. \end{cases} \quad (4)$$

The essential objective function in (D) is

$$f(y) = \sup_{X \in \mathcal{S}^n} l(y, X) = \begin{cases} b^\top y & \text{if } y \in \mathcal{F}_D, \\ +\infty & \text{otherwise,} \end{cases} \quad (5)$$

where $\mathcal{F}_D := \{y \in \mathbb{R}^m \mid \mathcal{A}^* y - C \succeq \mathbf{0}\}$ is the feasible set of (D), while the essential objective

function in (P) is

$$g(X) = \inf_{y \in \mathbb{R}^m} l(y, X) = \begin{cases} \langle C, X \rangle & \text{if } X \in \mathcal{F}_P, \\ -\infty & \text{otherwise,} \end{cases} \quad (6)$$

where $\mathcal{F}_P := \{X \in \mathcal{S}^n \mid \mathcal{A}(X) = b, X \succeq \mathbf{0}\}$ is the feasible set of (P) .

Assume that $\mathcal{F}_D \neq \emptyset$ and $\mathcal{F}_P \neq \emptyset$. As in Rockafellar [33], we define the following three maximal monotone operators

$$\begin{cases} T_g(X) &= \{U \in \mathcal{S}^n \mid -U \in \partial g(X)\}, & X \in \mathcal{S}^n, \\ T_f(y) &= \{v \in \mathbb{R}^m \mid v \in \partial f(y)\}, & y \in \mathbb{R}^m, \\ T_l(y, X) &= \{(v, U) \in \mathbb{R}^m \times \mathcal{S}^n \mid (v, -U) \in \partial l(y, X)\}, & (y, X) \in \mathbb{R}^m \times \mathcal{S}^n. \end{cases}$$

Throughout this paper, the following Slater condition for (P) is assumed to hold.

Assumption 1. Problem (P) satisfies the Slater condition

$$\begin{cases} \mathcal{A} : \mathcal{S}^n \rightarrow \mathbb{R}^m \text{ is onto,} \\ \exists X_0 \in \mathcal{S}_+^n \text{ such that } \mathcal{A}(X_0) = b, X_0 \succ \mathbf{0}, \end{cases} \quad (7)$$

where $X_0 \succ \mathbf{0}$ means that X_0 is a symmetric positive definite matrix.

For each $v \in \mathbb{R}^m$ and $U \in \mathcal{S}^n$, we consider the following parameterized problem:

$$(P(v, U)) \quad \max \left\{ \langle C, X \rangle + \langle U, X \rangle \mid \mathcal{A}(X) + v = b, \quad X \succeq \mathbf{0} \right\}.$$

By using the fact that g is concave, we know from Rockafellar [29, Theorem 23.5] that for each $U \in \mathcal{S}^n$,

$$T_g^{-1}(U) = \text{set of all optimal solutions to } (P(0, U)). \quad (8)$$

Similarly, we have that for each $v \in \mathbb{R}^m$,

$$T_f^{-1}(v) = \text{set of all optimal solutions to } (D(v, \mathbf{0})), \quad (9)$$

where for $(v, U) \in \mathbb{R}^m \times \mathcal{S}^n$, $(D(v, U))$ is the (ordinary) dual of $(P(v, U))$ in the sense that

$$(D(v, U)) \quad \min \left\{ b^T y - v^T y : \mathcal{A}^* y - U \succeq C \right\}.$$

Finally, for any $(v, U) \in \mathbb{R}^m \times \mathcal{S}^n$, under Assumption 1, we have that

$$\begin{aligned} T_l^{-1}(v, U) &= \arg \operatorname{minimax} \{ l(y, X) - v^T y + \langle U, X \rangle \mid y \in \mathbb{R}^m, X \in \mathcal{S}^n \}, \\ &= \text{set of all } (y, X) \text{ satisfying the Karush-Kuhn-Tucker} \\ &\quad \text{conditions for } (P(v, U)). \text{ (cf. (12))} \end{aligned} \quad (10)$$

Definition 1. [32] For a maximal monotone operator T from a finite dimensional linear vector space \mathcal{X} to itself, we say that its inverse T^{-1} is Lipschitz continuous at the origin (with modulus $a \geq 0$) if there is a unique solution \bar{z} to $z = T^{-1}(0)$, and for some $\tau > 0$ we have

$$\|z - \bar{z}\| \leq a\|w\| \quad \text{whenever} \quad z \in T^{-1}(w) \quad \text{and} \quad \|w\| \leq \tau. \quad (11)$$

The first order optimality conditions, namely the Karush-Kuhn-Tucker (KKT) conditions, of (D) and (P) are as follows:

$$\begin{cases} \mathcal{A}(X) = b, \\ \mathcal{S}_+^n \ni (\mathcal{A}^*y - C) \perp X \in \mathcal{S}_+^n, \end{cases} \quad (12)$$

where “ $(\mathcal{A}^*y - C) \perp X$ ” means that $(\mathcal{A}^*y - C)$ and X are orthogonal to each other, i.e., $\langle \mathcal{A}^*y - C, X \rangle = 0$. For any $X \in \mathcal{F}_P$, define the set

$$\mathcal{M}(X) := \{y \in \mathbb{R}^m \mid (y, X) \text{ satisfies the KKT conditions (12)}\}. \quad (13)$$

Let \bar{X} be an optimal solution to (P) . Since (P) satisfies the Slater condition (7), $\mathcal{M}(\bar{X})$ is nonempty and bounded [31, Theorems 17 & 18]. Let $y \in \mathcal{M}(\bar{X})$ be arbitrarily chosen. Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ be the eigenvalues of \bar{X} being arranged in the nonincreasing order and let $\mu_1 \leq \mu_2 \leq \dots \leq \mu_n$ be the eigenvalues of $(\mathcal{A}^*y - C)$ being arranged in the nondecreasing order. Denote $\alpha := \{i \mid \lambda_i > 0, i = 1, \dots, n\}$ and $\gamma := \{i \mid \mu_i > 0, i = 1, \dots, n\}$. Since $\bar{X}(\mathcal{A}^*y - C) = (\mathcal{A}^*y - C)\bar{X} = 0$, there exists an orthogonal matrix $P \in \mathbb{R}^{n \times n}$ such that

$$\bar{X} = P \begin{bmatrix} \Lambda_\alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} P^T \quad \text{and} \quad (\mathcal{A}^*y - C) = P \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Lambda_\gamma \end{bmatrix} P^T, \quad (14)$$

where Λ_α is the diagonal matrix whose diagonal entries are λ_i for $i \in \alpha$ and Λ_γ is the diagonal matrix whose diagonal entries μ_i for $i \in \gamma$.

Let $A := \bar{X} - (\mathcal{A}^*y - C) \in \mathcal{S}_n$. Then, A has the following spectral decomposition

$$A = P\Lambda P^T, \quad (15)$$

where

$$\Lambda = \begin{bmatrix} \Lambda_\alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\Lambda_\gamma \end{bmatrix}.$$

Denote $\beta := \{1, \dots, n\} \setminus (\alpha \cup \gamma)$. Write $P = [P_\alpha \ P_\beta \ P_\gamma]$ with $P_\alpha \in \mathbb{R}^{n \times |\alpha|}$, $P_\beta \in \mathbb{R}^{n \times |\beta|}$, and $P_\gamma \in \mathbb{R}^{n \times |\gamma|}$. From [2], we know that the tangent cone of \mathcal{S}_+^n at $\bar{X} \in \mathcal{S}_+^n$ can be characterized as follows

$$\mathcal{T}_{\mathcal{S}_+^n}(\bar{X}) = \{B \in \mathcal{S}^n \mid [P_\beta \ P_\gamma]^T B [P_\beta \ P_\gamma] \succeq 0\}. \quad (16)$$

Similarly, the tangent cone of \mathcal{S}_+^n at $(\mathcal{A}^*y - C)$ takes the form

$$\mathcal{T}_{\mathcal{S}_+^n}(\mathcal{A}^*y - C) = \{B \in \mathcal{S}^n \mid [P_\alpha \ P_\beta]^\top B [P_\alpha \ P_\beta] \succeq 0\}. \quad (17)$$

Recall that the *critical cone* of problem (P) at \bar{X} is defined by (cf. [4, p.151])

$$\mathcal{C}(\bar{X}) = \{B \in \mathcal{S}^n \mid \mathcal{A}(B) = 0, B \in \mathcal{T}_{\mathcal{S}_+^n}(\bar{X}), \langle C, B \rangle = 0\}. \quad (18)$$

Choose an arbitrary element $B \in \mathcal{C}(\bar{X})$. Denote $\tilde{B} := P^\top B P$. Since \bar{X} and $(\mathcal{A}^*y - C)$ have the spectral decompositions as in (14), we obtain that

$$\begin{aligned} 0 &= \langle C, B \rangle = \langle \mathcal{A}^*y - C, B \rangle = \langle P^\top (\mathcal{A}^*y - C) P, P^\top B P \rangle \\ &= \left\langle \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Lambda_\gamma \end{bmatrix}, \begin{bmatrix} \tilde{B}_{\alpha\alpha} & \tilde{B}_{\alpha\beta} & \tilde{B}_{\alpha\gamma} \\ \tilde{B}_{\alpha\beta}^\top & \tilde{B}_{\beta\beta} & \tilde{B}_{\beta\gamma} \\ \tilde{B}_{\alpha\gamma}^\top & \tilde{B}_{\beta\gamma}^\top & \tilde{B}_{\gamma\gamma} \end{bmatrix} \right\rangle, \end{aligned}$$

which, together with (16) and (18), implies that $\tilde{B}_{\gamma\gamma} = 0$. Thus

$$\tilde{B}_{\beta\gamma} = 0 \quad \text{and} \quad \tilde{B}_{\gamma\gamma} = 0.$$

Hence, $\mathcal{C}(\bar{X})$ can be rewritten as

$$\mathcal{C}(\bar{X}) = \{B \in \mathcal{S}^n \mid \mathcal{A}(B) = 0, P_\beta^\top B P_\beta \succeq 0, P_\beta^\top B P_\gamma = 0, P_\gamma^\top B P_\gamma = 0\}. \quad (19)$$

By using similar arguments as above, we can also obtain that

$$\mathcal{T}_{\mathcal{S}_+^n}(\mathcal{A}^*y - C) \cap \bar{X}^\perp = \{B \in \mathcal{S}^n \mid P_\alpha^\top B P_\alpha = 0, P_\alpha^\top B P_\beta = 0, P_\beta^\top B P_\beta \succeq 0\}, \quad (20)$$

where $\bar{X}^\perp := \{B \in \mathcal{S}^n \mid \langle B, \bar{X} \rangle = 0\}$.

In order to analyze the rate of convergence of the Newton-CG augmented Lagrangian method to be presented in Section 4, we need the following result which characterizes the Lipschitz continuity of T_g^{-1} at the origin. The result we establish here is stronger than that established in Proposition 15 of [8].

Proposition 2.1. *Suppose that (P) satisfies the generalized Slater condition (7). Let $\bar{X} \in \mathcal{S}_+^n$ be an optimal solution to (P). Then the following conditions are equivalent*

(i) $T_g^{-1}(\cdot)$ is Lipschitz continuous at the origin.

(ii) The second order sufficient condition

$$\sup_{y \in \mathcal{M}(\bar{X})} \Upsilon_{\bar{X}}(\mathcal{A}^*y - C, H) > 0 \quad \forall H \in \mathcal{C}(\bar{X}) \setminus \{0\} \quad (21)$$

holds at \bar{X} , where for any $B \in \mathcal{S}^n$, the linear-quadratic function $\Upsilon_B : \mathcal{S}^n \times \mathcal{S}^n \rightarrow \Re$ is defined by

$$\Upsilon_B(M, H) := 2 \langle M, H B^\dagger H \rangle, \quad (M, H) \in \mathcal{S}^n \times \mathcal{S}^n \quad (22)$$

and B^\dagger is the Moore-Penrose pseudo-inverse of B .

(iii) \bar{X} satisfies the extended strict primal-dual constraint qualification

$$\mathcal{A}^*\mathfrak{R}^m + \text{conv} \left(\bigcup_{y \in \mathcal{M}(\bar{X})} \left(\mathcal{T}_{\mathcal{S}^n_+}(\mathcal{A}^*y - C) \cap \bar{X}^\perp \right) \right) = \mathcal{S}^n, \quad (23)$$

where for any set $\mathcal{W} \subset \mathcal{S}^n$, $\text{conv}(\mathcal{W})$ denotes the convex hull of \mathcal{W} .

Proof. “(i) \Leftrightarrow (ii)”. From [4, Theorem 3.137], we know that (ii) holds if and only if the quadratic growth condition

$$\langle C, \bar{X} \rangle \geq \langle C, X \rangle + c\|X - \bar{X}\|^2 \quad \forall X \in \mathcal{N} \text{ such that } X \in \mathcal{F}_P \quad (24)$$

holds at \bar{X} for some positive constant c and an open neighborhood \mathcal{N} of \bar{X} in \mathcal{S}^n . On the other hand, from [33, Proposition 3], we know that $T_g^{-1}(\cdot)$ is Lipschitz continuous at the origin if and only if the quadratic growth condition (24) holds at \bar{X} . Hence, (i) \Leftrightarrow (ii).

Next we shall prove that (ii) \Leftrightarrow (iii). For notational convenience, let

$$\Gamma := \text{conv} \left(\bigcup_{y \in \mathcal{M}(\bar{X})} \left(\mathcal{T}_{\mathcal{S}^n_+}(\mathcal{A}^*y - C) \cap \bar{X}^\perp \right) \right). \quad (25)$$

“(ii) \Rightarrow (iii)”. Denote $\mathcal{D} := \mathcal{A}^*\mathfrak{R}^m + \Gamma$. For the purpose of contradiction, we assume that (iii) does not hold, i.e., $\mathcal{D} \neq \mathcal{S}^n$. Let $\text{cl}(\mathcal{D})$ and $\text{ri}(\mathcal{D})$ denote the closure of \mathcal{D} and the relative interior of \mathcal{D} , respectively. By [29, Theorem 6.3], since $\text{ri}(\mathcal{D}) = \text{ri}(\text{cl}(\mathcal{D}))$, the relative interior of $\text{cl}(\mathcal{D})$, we know that $\text{cl}(\mathcal{D}) \neq \mathcal{S}^n$. Thus, there exists $B \in \mathcal{S}^n$ such that $B \notin \text{cl}(\mathcal{D})$. Let \bar{B} be the metric projection of B onto $\text{cl}(\mathcal{D})$, i.e., $\bar{B} = \Pi_{\text{cl}(\mathcal{D})}(B)$. Let $H = \bar{B} - B \neq 0$. Since $\text{cl}(\mathcal{D})$ is a nonempty closed convex cone, from Zarantonello [44], we know that

$$\langle H, Z \rangle = \langle \bar{B} - B, Z \rangle \geq 0 \quad \forall Z \in \text{cl}(\mathcal{D}).$$

In particular, we have

$$\langle H, \mathcal{A}^*z + Q \rangle \geq 0 \quad \forall z \in \mathfrak{R}^m \text{ and } Q \in \Gamma,$$

which implies (by taking $Q = 0$)

$$\langle \mathcal{A}(H), z \rangle = \langle H, \mathcal{A}^*z \rangle \geq 0 \quad \forall z \in \mathfrak{R}^m.$$

Thus

$$\mathcal{A}(H) = 0 \quad \text{and} \quad \langle H, Q \rangle \geq 0 \quad \text{for any } Q \in \Gamma. \quad (26)$$

Since $0 \neq H \in \mathcal{C}(\bar{X})$ and (ii) is assumed to hold, there exists $y \in \mathcal{M}(\bar{X})$ such that

$$\Upsilon_{\bar{X}}(\mathcal{A}^*y - C, H) > 0. \quad (27)$$

By using the fact that (y, \overline{X}) satisfies (12), we can assume that \overline{X} and $(\mathcal{A}^*y - C)$ have the spectral decompositions as in (14). Then, we know from (20) that for any $Q \in \mathcal{T}_{S_+^n}(\mathcal{A}^*y - C) \cap \overline{X}^\perp$,

$$\begin{aligned} 0 &\leq \langle H, Q \rangle = \langle P\tilde{H}P^T, P\tilde{Q}P^T \rangle \\ &= \left\langle \begin{bmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \tilde{H}_{\beta\beta} & \tilde{H}_{\beta\gamma} \\ \tilde{H}_{\alpha\gamma}^T & \tilde{H}_{\beta\gamma}^T & \tilde{H}_{\gamma\gamma} \end{bmatrix}, \begin{bmatrix} 0 & 0 & \tilde{Q}_{\alpha\gamma} \\ 0 & \tilde{Q}_{\beta\beta} & \tilde{Q}_{\beta\gamma} \\ \tilde{Q}_{\alpha\gamma} & \tilde{Q}_{\beta\gamma} & \tilde{Q}_{\gamma\gamma} \end{bmatrix} \right\rangle, \end{aligned} \quad (28)$$

where $\tilde{H} = P^T H P$ and $\tilde{Q} = P^T Q P$. From (20) and (28), we have

$$\tilde{H}_{\alpha\gamma} = 0, \quad \tilde{H}_{\beta\gamma} = 0, \quad \tilde{H}_{\gamma\gamma} = 0, \quad \text{and} \quad \tilde{H}_{\beta\beta} \succeq 0. \quad (29)$$

By using (19), (26), and (29), we obtain that $H \in \mathcal{C}(\overline{X})$ and

$$P_\alpha^T H P_\gamma = 0. \quad (30)$$

Note that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ and $\mu_1 \leq \mu_2 \leq \dots \leq \mu_n$ are the eigenvalues of \overline{X} and $(\mathcal{A}^*y - C)$, respectively, and $\alpha = \{i \mid \lambda_i > 0, i = 1, \dots, n\}$ and $\gamma = \{j \mid \mu_j > 0, j = 1, \dots, n\}$. Therefore, from (22) and (14), we obtain that

$$\Upsilon_{\overline{X}}(\mathcal{A}^*y - C, H) = \sum_{i \in \alpha, j \in \gamma} \frac{\mu_j}{\lambda_i} (P_i^T H P_j)^2 = 0,$$

which contradicts (27). This contradiction shows $(ii) \Rightarrow (iii)$.

“(iii) \Rightarrow (ii)”. Assume that (ii) does not hold at \overline{X} . Then there exists $0 \neq H \in \mathcal{C}(\overline{X})$ such that

$$\sup_{y \in \mathcal{M}(\overline{X})} \Upsilon_{\overline{X}}(\mathcal{A}^*y - C, H) = 0. \quad (31)$$

Let y be an arbitrary element in $\mathcal{M}(\overline{X})$. Since (y, \overline{X}) satisfies (12), we can assume that there exists an orthogonal matrix $P \in \mathbb{R}^{n \times n}$ such that \overline{X} and $(\mathcal{A}^*y - C)$ have the spectral decompositions as in (14). From (14), (22), and (31), we have

$$0 \leq \sum_{i \in \alpha, j \in \gamma} \frac{\mu_j}{\lambda_i} (P_i^T H P_j)^2 = \Upsilon_{\overline{X}}(\mathcal{A}^*y - C, H) \leq \sup_{z \in \mathcal{M}(\overline{X})} \Upsilon_{\overline{X}}(\mathcal{A}^*z - C, H) = 0,$$

which implies

$$P_\alpha^T H P_\gamma = 0. \quad (32)$$

Then, by using (19), (20), and (32), we have that

$$\langle Q^y, H \rangle = \langle P^T Q^y P, P^T H P \rangle = \langle P_\beta^T Q^y P_\beta, P_\beta^T H P_\beta \rangle \geq 0 \quad \forall Q^y \in \mathcal{T}_{S_+^n}(\mathcal{A}^* y - C) \cap \overline{X}^\perp. \quad (33)$$

Since (iii) is assumed to hold, there exist $z \in \mathbb{R}^m$ and $Q \in \Gamma$ such that

$$-H = \mathcal{A}^* z + Q. \quad (34)$$

By Carathéodory's Theorem, there exist an integer $k \leq \frac{n(n+1)}{2} + 1$ and scalars $\alpha_i \geq 0$, $i = 1, 2, \dots, k$, with $\sum_{i=1}^k \alpha_i = 1$, and

$$Q_i \in \bigcup_{y \in \mathcal{M}(\overline{X})} \left(\mathcal{T}_{S_+^n}(\mathcal{A}^* y - C) \cap \overline{X}^\perp \right), \quad i = 1, 2, \dots, k$$

such that Q can be represented as

$$Q = \sum_{i=1}^k \alpha_i Q_i.$$

For each Q_i , there exists a $y^i \in \mathcal{M}(\overline{X})$ such that $Q_i \in \mathcal{T}_{S_+^n}(\mathcal{A}^* y^i - C) \cap \overline{X}^\perp$. Then by using the fact that $H \in \mathcal{C}(\overline{X})$ and (33), we obtain that

$$\langle H, H \rangle = \langle -\mathcal{A}^* z - Q, H \rangle = -\langle z, \mathcal{A}H \rangle - \langle Q, H \rangle = 0 - \sum_{i=1}^k \alpha_i \langle Q_i, H \rangle \leq 0,$$

which contradicts the fact that $H \neq 0$. This contradiction shows that (ii) holds. \square

Proposition 2.1 characterizes the Lipschitz continuity of T_g^{-1} at the origin by either the second sufficient condition (21) or the extended strict primal-dual constraint qualification (23). In particular, if $\mathcal{M}(\overline{X})$ is a singleton, we have the following simple equivalent conditions.

Corollary 2.2. *Suppose that (P) satisfies the generalized Slater condition (7). Let \overline{X} be an optimal solution to (P). If $\mathcal{M}(\overline{X}) = \{\bar{y}\}$, then the following are equivalent:*

(i) $T_g^{-1}(\cdot)$ is Lipschitz continuous at the origin.

(ii) The second order sufficient condition

$$\Upsilon_{\overline{X}}(\mathcal{A}^* \bar{y} - C, H) > 0 \quad \forall H \in \mathcal{C}(\overline{X}) \setminus \{0\} \quad (35)$$

holds at \overline{X} .

(iii) \overline{X} satisfies the strict primal-dual constraint qualification

$$\mathcal{A}^* \mathbb{R}^m + \mathcal{T}_{S_+^n}(\mathcal{A}^* \bar{y} - C) \cap \overline{X}^\perp = \mathcal{S}^n. \quad (36)$$

Remark 1. Note that in [8, Proposition 15], Chan and Sun proved that if $\mathcal{M}(\bar{X})$ is a singleton, then the *strong* second order sufficient condition (with the set $\mathcal{C}(\bar{X})$ in (35) being replaced by the superset $\{B \in \mathcal{S}^n \mid \mathcal{A}(B) = 0, P_\beta^T B P_\gamma = 0, P_\gamma^T B P_\gamma = 0\}$) is equivalent to the constraint nondegenerate condition, in the sense of Robinson [27, 28], at \bar{y} for (D) , i.e.,

$$\mathcal{A}^* \mathfrak{R}^m + \text{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\mathcal{A}^* \bar{y} - C)) = \mathcal{S}^n. \quad (37)$$

Corollary 2.2 further establishes the equivalence between the second order sufficient condition (35) and the strict constraint qualification (36) under the condition that $\mathcal{M}(\bar{X})$ is a singleton.

One may observe that the strict primal-dual constraint qualification condition (36) is weaker than the constraint nondegenerate condition (37). However, if strict complementarity holds, i.e., $\bar{X} + (\mathcal{A}^* \bar{y} - C) \succ 0$ and hence β is the empty set, then (36) and (37) coincide.

The constraint nondegenerate condition (37) is equivalent to the dual nondegeneracy stated in [1, Theorem 9]. Note that under such a condition, the optimal solution \bar{X} to (P) is unique.

Remark 2. In a similar way, we can establish parallel results for T_f^{-1} as for T_g^{-1} in Proposition 2.1 and Corollary 2.2. For brevity, we omit the details.

3 A Semismooth Newton-CG Method for Inner Problems

In this section we introduce a semismooth Newton-CG method for solving the inner problems involved in the augmented Lagrangian method (3). For this purpose, we need the practical CG method described in [12, Algorithm 10.2.1] for solving the symmetric positive definite linear system. Since our convergence analysis of the semismooth Newton-CG method heavily depends on this practical CG method and its convergence property (Lemma 3.1), we shall give it a brief description here.

3.1 A practical CG method

In this subsection, we consider a practical CG method to solve the following linear equation

$$Ax = b, \quad (38)$$

where $b \in \mathfrak{R}^m$ and $A \in \mathfrak{R}^{m \times m}$ is assumed to be a symmetric positive definite matrix. The practical conjugate gradient algorithm [12, Algorithm 10.2.1] depends on two parameters: a maximum number of CG iterations $i_{max} > 0$ and a tolerance $\eta \in (0, \|b\|)$.

Algorithm 1. A Practical CG Algorithm: $[CG(\eta, i_{max})]$

Step 0. Given $x^0 = 0$ and $r^0 = b - Ax^0 = b$.

Step 1. While $(\|r^i\| > \eta)$ or $(i < i_{\max})$

Step 1.1. $i = i + 1$

Step 1.2. If $i = 1$; $p^1 = r^0$; else; $\beta_i = \|r^{i-1}\|^2 / \|r^{i-2}\|^2$, $p^i = r^{i-1} + \beta_i p^{i-1}$; end

Step 1.3. $\alpha_i = \|r^{i-1}\|^2 / \langle p^i, Ap^i \rangle$

Step 1.4. $x^i = x^{i-1} + \alpha_i p^i$

Step 1.5. $r^i = r^{i-1} - \alpha_i Ap^i$

Lemma 3.1. Let $0 < \bar{i} \leq i_{\max}$ be the number of iterations when the practical CG Algorithm 1 terminates. For all $i = 1, 2, \dots, \bar{i}$, the iterates $\{x^i\}$ generated by Algorithm 1 satisfies

$$\frac{1}{\lambda_{\max}(A)} \leq \frac{\langle x^i, b \rangle}{\|b\|^2} \leq \frac{1}{\lambda_{\min}(A)}, \quad (39)$$

where $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the smallest and largest eigenvalue of A , respectively.

Proof. Let x^* be the exact solution to (38) and $e^i = x^* - x^i$ be the error in the i th iteration for $i \geq 0$. From [38, Theorem 38.1], we know that

$$\langle r^i, r^j \rangle = 0 \quad \text{for } j = 1, 2, \dots, i-1, \quad (40)$$

where $r^i = b - Ax^i$. By using (40), the fact that in Algorithm 1, $r^0 = b$, and the definition of β_i , we have that

$$\begin{aligned} \langle p^1, b \rangle &= \|r^0\|^2, \\ \langle p^i, b \rangle &= \langle r^{i-1}, b \rangle + \beta_i \langle p^{i-1}, b \rangle = 0 + \prod_{j=2}^i \beta_j \langle p^1, b \rangle = \|r^{i-1}\|^2 \quad \forall i > 1. \end{aligned} \quad (41)$$

From [38, Theorem 38.2], we know that for $i \geq 1$,

$$\|e^{i-1}\|_A^2 = \|e^i\|_A^2 + \langle \alpha_i p^i, A(\alpha_i p^i) \rangle, \quad (42)$$

which, together with $\alpha_i \|r^{i-1}\|^2 = \langle \alpha_i p^i, A(\alpha_i p^i) \rangle$ (see Step 1.3), implies that

$$\alpha_i \|r^{i-1}\|^2 = \|e^{i-1}\|_A^2 - \|e^i\|_A^2. \quad (43)$$

Here for any $x \in \mathfrak{R}^m$, $\|x\|_A := \sqrt{\langle x, Ax \rangle}$. For any $i \geq 1$, by using (41), (43), and the fact that $x^0 = 0$, we have that

$$\begin{aligned} \langle x^i, b \rangle &= \langle x^{i-1}, b \rangle + \alpha_i \langle p^i, b \rangle = \langle x^0, b \rangle + \sum_{j=1}^i \alpha_j \langle p^j, b \rangle = \sum_{j=1}^i \alpha_j \|r^{j-1}\|^2 \\ &= \sum_{j=1}^i [\|e^{j-1}\|_A^2 - \|e^j\|_A^2] = \|e^0\|_A^2 - \|e^i\|_A^2, \end{aligned} \quad (44)$$

which, together with (42), implies that

$$\langle x^i, b \rangle \geq \langle x^{i-1}, b \rangle, \quad i = 1, 2, \dots, \bar{i}.$$

Thus

$$\frac{1}{\lambda_{\max}(A)} \leq \alpha_1 = \frac{\langle x^1, b \rangle}{\|b\|^2} \leq \frac{\langle x^i, b \rangle}{\|b\|^2}. \quad (45)$$

Since $e^0 = x^* - x^0 = A^{-1}b$, by (44), we obtain that for $1 \leq i \leq \bar{i}$,

$$\frac{\langle x^i, b \rangle}{\|b\|^2} \leq \frac{\|e^0\|_A^2}{\|b\|^2} = \frac{\|A^{-1}b\|_A^2}{\|b\|^2} \leq \frac{1}{\lambda_{\min}(A)}. \quad (46)$$

By combining (45) and (46), we complete the proof. \square

3.2 A Semismooth Newton-CG method

For the augmented Lagrangian method (3), for some fixed $X \in \mathcal{S}^n$ and $\sigma > 0$, we need to consider the following form of inner problems

$$\min \{\varphi(y) := L_\sigma(y, X) \mid y \in \mathbb{R}^m\}. \quad (47)$$

As explained in the introduction, $\varphi(\cdot)$ is a continuously differentiable convex function, but fails to be twice continuously differentiable because the metric projector $\Pi_{\mathcal{S}_+^n}(\cdot)$ is not continuously differentiable. Fortunately, because $\Pi_{\mathcal{S}_+^n}(\cdot)$ is strongly semismooth [36], we can develop locally a semismooth Newton-CG method to solve the following nonlinear equation

$$\nabla \varphi(y) = b - \mathcal{A} \Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^*y - C)) = 0 \quad (48)$$

and expect a superlinear (quadratic) convergence for solving (48).

Since $\Pi_{\mathcal{S}_+^n}(\cdot)$ is Lipschitz continuous with modulus 1, the mapping $\nabla \varphi$ is Lipschitz continuous on \mathbb{R}^m . According to Rademacher's Theorem, $\nabla \varphi$ is almost everywhere Fréchet-differentiable in \mathbb{R}^m . Let $y \in \mathbb{R}^m$. The generalized Hessian of φ at y is defined as

$$\partial^2 \varphi(y) := \partial(\nabla \varphi)(y), \quad (49)$$

where $\partial(\nabla \varphi)(y)$ is the Clarke's generalized Jacobian of $\nabla \varphi$ at y [9]. Since it is difficult to express $\partial^2 \varphi(y)$ exactly, we define the following alternative for $\partial^2 \varphi(y)$

$$\hat{\partial}^2 \varphi(y) := \sigma \mathcal{A} \partial \Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^*y - C)) \mathcal{A}^*. \quad (50)$$

From [9, p.75], for $d \in \mathbb{R}^m$,

$$\partial^2 \varphi(y)d \subseteq \hat{\partial}^2 \varphi(y)d, \quad (51)$$

which means that if every element in $\hat{\partial}^2\varphi(y)$ is positive definite, so is every element in $\partial^2\varphi(y)$.

For the semismooth Newton-CG method to be presented later, we need to compute an element $V \in \hat{\partial}^2\varphi(y)$. Since $X - \sigma(\mathcal{A}^*y - C)$ is a symmetric matrix in $\Re^{n \times n}$, there exists an orthogonal matrix $Q \in \Re^{n \times n}$ such that

$$X - \sigma(\mathcal{A}^*y - C) = Q \Gamma_y Q^T, \quad (52)$$

where Γ_y is the diagonal matrix with diagonal entries consisting of the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ of $X - \sigma(\mathcal{A}^*y - C)$ being arranged in the nonincreasing order. Define three index sets

$$\alpha := \{i \mid \lambda_i > 0\}, \quad \beta := \{i \mid \lambda_i = 0\}, \quad \text{and} \quad \gamma := \{i \mid \lambda_i < 0\}.$$

Define the operator $W_y^0 : \mathcal{S}^n \rightarrow \mathcal{S}^n$ by

$$W_y^0(H) := Q(\Omega \circ (Q^T H Q))Q^T, \quad H \in \mathcal{S}^n, \quad (53)$$

where “ \circ ” denotes the Hadamard product of two matrices and

$$\Omega = \begin{bmatrix} E_{\bar{\gamma}\bar{\gamma}} & \nu_{\bar{\gamma}\gamma} \\ \nu_{\bar{\gamma}\gamma}^T & 0 \end{bmatrix}, \quad \nu_{ij} := \frac{\lambda_i}{\lambda_i - \lambda_j}, \quad i \in \bar{\gamma}, j \in \gamma, \quad (54)$$

$\bar{\gamma} = \{1, \dots, n\} \setminus \gamma$, and $E_{\bar{\gamma}\bar{\gamma}} \in \mathcal{S}^{|\bar{\gamma}|}$ is the matrix of ones. Define $V_y^0 : \Re^m \rightarrow \mathcal{S}^n$ by

$$V_y^0 d := \sigma \mathcal{A} [Q(\Omega \circ (Q^T (\mathcal{A}^* d) Q))Q^T], \quad d \in \Re^m. \quad (55)$$

Since, by Pang, Sun, and Sun [22, Lemma 11],

$$W_y^0 \in \partial \Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^*y - C)),$$

we know that

$$V_y^0 = \sigma \mathcal{A} W_y^0 \mathcal{A}^* \in \hat{\partial}^2\varphi(y).$$

Next we shall characterize the positive definiteness of any $V_y \in \hat{\partial}^2\varphi(y)$. From [33, p.107] and the definitions of $l(y, X)$ in (4), we know that for any $(y, X, \sigma) \in \Re^m \times \mathcal{S}^n \times (0, +\infty)$,

$$L_\sigma(y, X) = \max_{Z \in \mathcal{S}^n} \{l(y, Z) - \frac{1}{2\sigma} \|Z - X\|^2\}.$$

Since the Slater condition (7) is assumed to hold, by the definition of $g(\cdot)$ in (6), we can deduce from [31, Theorems 17 and 18] that

$$\begin{aligned} \min_{y \in \Re^m} \varphi(y) &= \min_{y \in \Re^m} \max_{Z \in \mathcal{S}^n} \left\{ l(y, Z) - \frac{1}{2\sigma} \|Z - X\|^2 \right\} = \max_{Z \in \mathcal{S}^n} \left\{ g(Z) - \frac{1}{2\sigma} \|Z - X\|^2 \right\} \\ &= \max_{\mathcal{A}(Z)=b, Z \succeq 0} \left\{ \langle C, Z \rangle - \frac{1}{2\sigma} \|Z - X\|^2 \right\}. \end{aligned} \quad (56)$$

Hence, (47) is the dual of

$$\max \left\{ \langle C, Z \rangle - \frac{1}{2\sigma} \|Z - X\|^2 \mid \mathcal{A}(Z) = b, \quad Z \succeq \mathbf{0} \right\}. \quad (57)$$

The KKT conditions of (57) are as follows

$$\begin{cases} \mathcal{A}(Z) = b, \\ \mathcal{S}_+^n \ni Z \perp [Z - (X - \sigma(\mathcal{A}^*y - C))] \in \mathcal{S}_+^n. \end{cases} \quad (58)$$

Proposition 3.2. *Suppose that the problem (57) satisfies the Slater condition (7). Let $(\hat{y}, \hat{Z}) \in \mathbb{R}^m \times \mathcal{S}^n$ be a pair that satisfies the KKT conditions (58) and let P be an orthogonal matrix such that \hat{Z} and $\hat{Z} - (X - \sigma(\mathcal{A}^*\hat{y} - C))$ have the spectral decomposition as (14). Then the following conditions are equivalent:*

(i) *The constraint nondegenerate condition*

$$\mathcal{A} \operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z})) = \mathbb{R}^m \quad (59)$$

holds at \hat{Z} , where $\operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z}))$ denotes the lineality space of $\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z})$, i.e.,

$$\operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z})) = \{B \in \mathcal{S}^n \mid [P_\beta \ P_\gamma]^T B [P_\beta \ P_\gamma] = 0\}. \quad (60)$$

(ii) *Every $V_{\hat{y}} \in \hat{\partial}^2 \varphi(\hat{y})$ is symmetric and positive definite.*

(iii) *$V_{\hat{y}}^0 \in \hat{\partial}^2 \varphi(\hat{y})$ is symmetric and positive definite.*

Proof. “(i) \Rightarrow (ii)”. This part is implied in [3, Proposition 2.8] by the Jacobian amicability of the metric projector $\Pi_{\mathcal{S}_+^n}(\cdot)$.

“(ii) \Rightarrow (iii)”. This is obvious true since $V_{\hat{y}}^0 \in \hat{\partial}^2 \varphi(\hat{y})$.

“(iii) \Rightarrow (i)”. Assume on the contrary that the constraint nondegenerate condition (59) does not hold at \hat{Z} . Then, we have

$$[\mathcal{A} \operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z}))]^\perp \neq \{0\}.$$

Let $0 \neq d \in [\mathcal{A} \operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z}))]^\perp$. Then

$$\langle d, \mathcal{A}(Q) \rangle = 0 \quad \forall Q \in \operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z})),$$

which can be written as

$$0 = \langle \mathcal{A}^* d, Q \rangle = \langle P^\top H P, P^\top Q P \rangle \quad \forall Q \in \operatorname{lin}(\mathcal{T}_{\mathcal{S}_+^n}(\hat{Z})), \quad (61)$$

where $H := \mathcal{A}^* d$. By using (60) and (61), we obtain that

$$P_\alpha^\top H P_\alpha = 0, \quad P_\alpha^\top H P_\beta = 0, \quad \text{and} \quad P_\alpha^\top H P_\gamma = 0.$$

By the definition of $W_{\hat{y}}^0$ in (53), it follows that $W_{\hat{y}}^0(H) = 0$. Therefore, for the corresponding $V_{\hat{y}}^0$ defined in (55), we have

$$\langle d, V_{\hat{y}}^0 d \rangle = \langle d, \sigma \mathcal{A} W_{\hat{y}}^0(\mathcal{A}^* d) \rangle = \sigma \langle H, W_{\hat{y}}^0(H) \rangle = 0,$$

which contradicts (iii) since $d \neq 0$. This contradiction shows that (i) holds. \square

Remark 3. The constraint nondegenerate condition (59) is equivalent to the primal nondegeneracy stated in [1, Theorem 6]. Under this condition, the solution \hat{y} for (58) is unique.

3.3 Convergence analysis

In this subsection, we shall introduce the promised semismooth Newton-CG algorithm to solve (47). Choose $y^0 \in \mathbb{R}^m$. Then the algorithm can be stated as follows.

Algorithm 2. A Semismooth Newton-CG Algorithm [$NCG(y^0, X, \sigma)$]

Step 0. Given $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, $\tau_1, \tau_2 \in (0, 1)$, and $\delta \in (0, 1)$.

Step 1. For $j = 0, 1, 2, \dots$

Step 1.1. Given a maximum number of CG iterations $n_j > 0$ and compute

$$\eta_j := \min(\bar{\eta}, \|\nabla \varphi(y^j)\|^{1+\tau}).$$

Apply the practical CG Algorithm 1 [$CG(\eta_j, n_j)$] to find an approximation solution d^j to

$$(V_j + \varepsilon_j I) d = -\nabla \varphi(y^j), \quad (62)$$

where $V_j \in \hat{\partial}^2 \varphi(y^j)$ is defined in (55) and $\varepsilon_j := \tau_1 \min\{\tau_2, \|\nabla \varphi(y^j)\|\}$.

Step 1.2. Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\varphi(y^j + \delta^m d^j) \leq \varphi(y^j) + \mu \delta^m \langle \nabla \varphi(y^j), d^j \rangle. \quad (63)$$

Step 1.3. Set $y^{j+1} = y^j + \alpha_j d^j$.

Remark 4. In Algorithm 2, since V_j is always positive semidefinite, the matrix $V_j + \varepsilon_j I$ is positive definite as long as $\nabla \varphi(y^j) \neq 0$. So we can always apply Algorithm 1 to equation (62).

Now we can analyze the global convergence of Algorithm 2 with the assumption that $\nabla \varphi(y^j) \neq 0$ for any $j \geq 0$. From Lemma 3.1, we know that the search direction d^j generated by Algorithm 2 is always a descent direction. This is stated in the following proposition.

Proposition 3.3. *For every $j \geq 0$, the search direction d^j generated in Step 1.2 of Algorithm 2 satisfies*

$$\frac{1}{\lambda_{\max}(\tilde{V}_j)} \leq \frac{\langle -\nabla\varphi(y^j), d^j \rangle}{\|\nabla\varphi(y^j)\|^2} \leq \frac{1}{\lambda_{\min}(\tilde{V}_j)}, \quad (64)$$

where $\tilde{V}_j := V_j + \varepsilon_j I$ and $\lambda_{\max}(\tilde{V}_j)$ and $\lambda_{\min}(\tilde{V}_j)$ are the largest and smallest eigenvalues of \tilde{V}_j respectively.

Theorem 3.4. *Suppose that problem (57) satisfies the Slater condition (7). Then Algorithm 2 is well defined and any accumulation point \hat{y} of $\{y^j\}$ generated by Algorithm 2 is an optimal solution to the inner problem (47).*

Proof. By Step 1.1 in Algorithm 2, for any $j \geq 0$, since, by (64), d^j is a descent direction, Algorithm 2 is well defined. Since problem (57) satisfies the Slater condition (7), from [31, Theorems 17 & 18], we know that the level set $\mathcal{L} := \{y \in \mathbb{R}^m \mid \varphi(y) \leq \varphi(y^0)\}$ is a closed and bounded convex set. Therefore, the sequence $\{y^j\}$ is bounded. Let \hat{y} be any accumulation point of $\{y^j\}$. Then, by making use of Proposition 3.3 and the Lipschitz continuity of $\Pi_{S_+^n}(\cdot)$, we can easily derive that $\nabla\varphi(\hat{y}) = 0$. By the convexity of $\varphi(\cdot)$, \hat{y} is an optimal solution of (47). \square

Next we shall discuss the rate of convergence of Algorithm 2.

Theorem 3.5. *Assume that problem (57) satisfies the Slater condition (7). Let \hat{y} be an accumulation point of the infinite sequence $\{y^j\}$ generated by Algorithm 2 for solving the inner problem (47). Suppose that at each step $j \geq 0$, when the practical CG Algorithm 1 terminates, the tolerance η_j is achieved (e.g., when $n_j = m + 1$), i.e.,*

$$\|\nabla\varphi(y^j) + (V_j + \varepsilon_j I) d^j\| \leq \eta_j. \quad (65)$$

Assume that the constraint nondegenerate condition (59) holds at $\hat{Z} := \Pi_{S_+^n}(X - \sigma(\mathcal{A}^\hat{y} - C))$. Then the whole sequence $\{y^j\}$ converges to \hat{y} and*

$$\|y^{j+1} - \hat{y}\| = O(\|y^j - \hat{y}\|^{1+\tau}). \quad (66)$$

Proof. By Theorem 3.4, we know that the infinite sequence $\{y^j\}$ is bounded and \hat{y} is an optimal solution to (47) with

$$\nabla\varphi(\hat{y}) = 0.$$

Since the constraint nondegenerate condition (59) is assumed to hold at \hat{Z} , \hat{y} is the unique optimal solution to (47). It then follows from Theorem 3.4 that $\{y^j\}$ converges to \hat{y} . From Proposition 3.2, we know that for any $V_{\hat{y}} \in \hat{\partial}^2\varphi(\hat{y})$ defined in (50), there exists a $W_{\hat{y}} \in \partial\Pi_{S_+^n}(X - \sigma(\mathcal{A}^*\hat{y} - C))$ such that

$$V_{\hat{y}} = \sigma\mathcal{A}W_{\hat{y}}\mathcal{A}^* \succ \mathbf{0}.$$

Then, for all j sufficiently large, $\{\|(V_j + \varepsilon_j I)^{-1}\|\}$ is uniformly bounded.

For any V_j , $j \geq 0$, there exists a $W_j \in \partial \Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^* y^j - C))$ such that

$$V_j = \sigma \mathcal{A} W_j \mathcal{A}^*. \quad (67)$$

Since $\Pi_{\mathcal{S}_+^n}(\cdot)$ is strongly semismooth [36], it holds that for all j sufficiently large,

$$\begin{aligned} \|y^j + d^j - \hat{y}\| &= \|y^j + (V_j + \varepsilon_j I)^{-1}((\nabla \varphi(y^j) + (V_j + \varepsilon_j I) d^j) - \nabla \varphi(y^j)) - \hat{y}\| \\ &\leq \|y^j - \hat{y} - (V_j + \varepsilon_j I)^{-1} \nabla \varphi(y^j)\| + \|(V_j + \varepsilon_j I)^{-1}\| \|\nabla \varphi(y^j) + (V_j + \varepsilon_j I) d^j\| \\ &\leq \|(V_j + \varepsilon_j I)^{-1}\| \|\nabla \varphi(y^j) - \nabla \varphi(\hat{y}) - V_j(y^j - \hat{y})\| \\ &\quad + \|(V_j + \varepsilon_j I)^{-1}\| (\varepsilon_j \|y^j - \hat{y}\| + \eta_j) \\ &\leq O(\|\mathcal{A}\| \|\Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^* y^j - C)) - \Pi_{\mathcal{S}_+^n}(X - \sigma(\mathcal{A}^* \hat{y} - C)) - W_j(\sigma \mathcal{A}^*(y^j - \hat{y}))\|) \\ &\quad + O(\tau_1 \|\nabla \varphi(y^j)\| \|y^j - \hat{y}\| + \|\nabla \varphi(y^j)\|^{1+\tau}) \\ &\leq O(\|\sigma \mathcal{A}^*(y^j - \hat{y})\|^2) + O(\tau_1 \|\nabla \varphi(y^j) - \nabla \varphi(\hat{y})\| \|y^j - \hat{y}\| + \|\nabla \varphi(y^j) - \nabla \varphi(\hat{y})\|^{1+\tau}) \\ &\leq O(\|y^j - \hat{y}\|^2) + O(\tau_1 \sigma \|\mathcal{A}\| \|\mathcal{A}^*\| \|y^j - \hat{y}\|^2 + (\sigma \|\mathcal{A}\| \|\mathcal{A}^*\| \|y^j - \hat{y}\|)^{1+\tau}) \\ &= O(\|y^j - \hat{y}\|^{1+\tau}), \end{aligned} \quad (68)$$

which implies that for all j sufficiently large,

$$y^j - \hat{y} = -d^j + O(\|d^j\|^{1+\tau}) \quad \text{and} \quad \|d^j\| \rightarrow 0. \quad (69)$$

For each $j \geq 0$, let $R^j := \nabla \varphi(y^j) + (V_j + \varepsilon_j I) d^j$. Then, for all j sufficiently large,

$$\begin{aligned} -\langle \nabla \varphi(y^j), d^j \rangle &= \langle d^j, (V_j + \varepsilon_j I) d^j \rangle - \langle R^j, d^j \rangle \\ &\geq \langle d^j, (V_j + \varepsilon_j I) d^j \rangle - \eta_j \|d^j\| \\ &\geq \langle d^j, (V_j + \varepsilon_j I) d^j \rangle - \|d^j\| \|\nabla \varphi(y^j)\|^{1+\tau} \\ &= \langle d^j, (V_j + \varepsilon_j I) d^j \rangle - \|\nabla \varphi(y^j) - \nabla \varphi(\hat{y})\|^{1+\tau} \|d^j\| \\ &\geq \langle d^j, (V_j + \varepsilon_j I) d^j \rangle - \sigma \|d^j\| \|\mathcal{A}\| \|\mathcal{A}^*\| \|y^j - \hat{y}\|^{1+\tau} \\ &\geq \langle d^j, (V_j + \varepsilon_j I) d^j \rangle - O(\|d^j\|^{2+\tau}), \end{aligned}$$

which, together with (69) and the fact that $\|(V_j + \varepsilon_j I)^{-1}\|$ is uniformly bounded, implies that there exists a constant $\hat{\delta} > 0$ such that

$$-\langle \nabla \varphi(y^j), d^j \rangle \geq \hat{\delta} \|d^j\|^2 \quad \text{for all } j \text{ sufficiently large.}$$

Since $\nabla \varphi(\cdot)$ is (strongly) semismooth at \hat{y} (because $\Pi_{\mathcal{S}_+^n}(\cdot)$ is strongly semismooth everywhere), from [11, Theorem 3.3 & Remark 3.4] or [21], we know that for $\mu \in (0, 1/2)$, there exists an integer j_0 such that for any $j \geq j_0$,

$$\varphi(y^j + d^j) \leq \varphi(y^j) + \mu \langle \nabla \varphi(y^j), d^j \rangle,$$

which means that for all $j \geq j_0$,

$$y^{j+1} = y^j + d^j.$$

This, together with (68), completes the proof. \square

Theorem 3.5 shows that the rate of convergence for Algorithm 2 is of order $(1 + \tau)$. If $\tau = 1$, this corresponds to quadratic convergence. However, this will need more CG iterations in Algorithm 1. To save computational time, in practice we choose $\tau = 0.1 \sim 0.2$, which still ensures that Algorithm 2 achieves superlinear convergence.

4 A Newton-CG Augmented Lagrangian Method

In this section, we shall introduce a Newton-CG augmented Lagrangian algorithm for solving problems (D) and (P). For any $k \geq 0$, denote $\varphi_k(\cdot) \equiv L_{\sigma_k}(\cdot, X^k)$. Since the inner problems can not be solved exactly, we will use the following stopping criteria considered by Rockafellar [32, 33] for terminating Algorithm 2:

$$(A) \quad \varphi_k(y^{k+1}) - \inf \varphi_k \leq \epsilon_k^2/2\sigma_k, \quad \epsilon_k \geq 0, \sum_{k=0}^{\infty} \epsilon_k < \infty.$$

$$(B) \quad \varphi_k(y^{k+1}) - \inf \varphi_k \leq (\delta_k^2/2\sigma_k)\|X^{k+1} - X^k\|^2, \quad \delta_k \geq 0, \sum_{k=0}^{\infty} \delta_k < \infty.$$

$$(B') \quad \|\nabla \varphi_k(y^{k+1})\| \leq (\delta'_k/\sigma_k)\|X^{k+1} - X^k\|, \quad 0 \leq \delta'_k \rightarrow 0.$$

Algorithm 3. A Newton-CG Augmented Lagrangian (SDPNAL) Algorithm

Step 0. Given $(y^0, X^0) \in \mathbb{R}^m \times \mathcal{S}_+^n$, $\sigma_0 > 0$, a threshold $\bar{\sigma} \geq \sigma_0 > 0$ and $\rho > 1$.

Step 1. For $k = 0, 1, 2, \dots$

Step 1.1. Starting with y^k as the initial point, apply Algorithm 2 to $\varphi_k(\cdot)$ to find $y^{k+1} = \text{NCG}(y^k, X^k, \sigma_k)$ and $X^{k+1} = \Pi_{\mathcal{S}_+^n}(X^k - \sigma_k(\mathcal{A}^*y^{k+1} - C))$ satisfying (A), (B) or (B').

Step 1.2. If $\sigma_k \leq \bar{\sigma}$, $\sigma_{k+1} = \rho\sigma_k$ or $\sigma_{k+1} = \sigma_k$.

The global convergence of Algorithm 3 follows from Rockafellar [32, Theorem 1] and [33, Theorem 4] without much difficulty.

Theorem 4.1. *Let Algorithm 3 be executed with stopping criterion (A). If (D) satisfies the Slater condition, i.e., if there exists $z^0 \in \mathbb{R}^m$ such that*

$$\mathcal{A}^*z^0 - C \succ \mathbf{0}, \tag{70}$$

then the sequence $\{X^k\} \subset \mathcal{S}_+^n$ generated by Algorithm 3 is bounded and $\{X^k\}$ converges to \bar{X} , where \bar{X} is some optimal solution to (P), and $\{y^k\}$ is asymptotically minimizing for (D) with $\max(P) = \inf(D)$.

If $\{X^k\}$ is bounded and (P) satisfies the generalized Slater condition (7), then the sequence $\{y^k\}$ is also bounded, and all of its accumulation points of the sequence $\{y^k\}$ are optimal solutions to (D).

Next we state the local linear convergence of the Newton-CG augmented Lagrangian algorithm.

Theorem 4.2. *Let Algorithm 3 be executed with stopping criteria (A) and (B). Assume that (D) satisfies the Slater condition (70) and (P) satisfies the Slater condition (7). If the extended strict primal-dual constraint qualification (23) holds at \bar{X} , where \bar{X} is an optimal solution to (P), then the generated sequence $\{X^k\} \subset \mathcal{S}_+^n$ is bounded and $\{X^k\}$ converges to the unique solution \bar{X} with $\max(P) = \min(D)$, and*

$$\|X^{k+1} - \bar{X}\| \leq \theta_k \|X^k - \bar{X}\| \quad \text{for all } k \text{ sufficiently large,}$$

where

$$\theta_k = [a_g(a_g^2 + \sigma_k^2)^{-1/2} + \delta_k] (1 - \delta_k)^{-1} \rightarrow \theta_\infty = a_g(a_g^2 + \sigma_\infty^2)^{-1/2} < 1, \quad \sigma_k \rightarrow \sigma_\infty,$$

and a_g is a Lipschitz constant of T_g^{-1} at the origin (cf. Proposition 2.1). The conclusions of Theorem 4.1 about $\{y^k\}$ are valid.

Moreover, if the stopping criterion (B') is also used and the constraint nondegenerate conditions (37) and (59) hold at \bar{y} and \bar{X} , respectively, then in addition to the above conclusions the sequence $\{y^k\} \rightarrow \bar{y}$, where \bar{y} is the unique optimal solution to (D), and one has

$$\|y^{k+1} - \bar{y}\| \leq \theta'_k \|X^{k+1} - X^k\| \quad \text{for all } k \text{ sufficiently large,}$$

where $\theta'_k = a_l(1 + \delta'_k)/\sigma_k \rightarrow \delta_\infty = a_l/\sigma_\infty$ and a_l is a Lipschitz constant of T_l^{-1} at the origin.

Proof. Conclusions of the first part of Theorem 4.2 follow from the results in [32, Theorem 2] and [33, Theorem 5] combining with Proposition 2.1. By using the fact that T_l^{-1} is Lipschitz continuous near the origin under the assumption that the constraint nondegenerate conditions (37) and (59) hold, respectively, at \bar{y} and \bar{X} [8, Theorem 18], we can directly obtain conclusions of the second part of this theorem from [32, Theorem 2] and [33, Theorem 5]. \square

Remark 5. Note that in (3) we can also add the term $\frac{1}{2\sigma_k}\|y - y^k\|^2$ to $L_{\sigma_k}(y, X^k)$ such that $L_{\sigma_k}(y, X^k) + \frac{1}{2\sigma_k}\|y - y^k\|^2$ is a strongly convex function. This actually corresponds to the proximal method of multipliers considered in [33, Section 5] for which the k -th iteration is given by

$$\begin{cases} y^{k+1} \approx \arg \min_{y \in \mathbb{R}^m} \left\{ L_{\sigma_k}(y, X^k) + \frac{1}{2\sigma_k}\|y - y^k\|^2 \right\}, \\ X^{k+1} = \Pi_{\mathcal{S}_+^n}(X^k - \sigma_k(\mathcal{A}^*y^{k+1} - C)), \\ \sigma_{k+1} = \rho\sigma_k \text{ or } \sigma_{k+1} = \sigma_k. \end{cases} \quad (71)$$

Convergence analysis for (71) can be conducted in a parallel way as for (3).

5 Numerical Issues in the Associated Semismooth Newton-CG Algorithm

In applying Algorithm 2 to solve the inner subproblem (47), the most expensive step is in computing the direction d at a given y from the linear system (62). Thus (62) must be solved as efficiently as possible. Let

$$M := \sigma A Q \otimes Q \text{diag}(\text{vec}(\Omega)) Q^T \otimes Q^T A^T,$$

where Q and Ω are given as in (52) and (54), respectively. Here A denotes the matrix representation of \mathcal{A} with respect to the standard bases of $\mathbb{R}^{n \times n}$ and \mathbb{R}^m . The direction d is computed from the following linear system:

$$(M + \varepsilon I) d = -\nabla \varphi(y). \quad (72)$$

To achieve faster convergence rate when applying the CG method to solve (72), one may apply a preconditioner to the system. By observing that the matrix Ω has elements all in the interval $[0, 1]$ and that the elements in the $(\bar{\gamma}, \bar{\gamma})$ block are all ones, one may simply approximate Ω by the matrix of ones, and hence a natural preconditioner for the coefficient matrix in (72) is simply the matrix $\widehat{M} := \sigma A A^T + \varepsilon I$. However, using \widehat{M} as the preconditioner may be costly since it requires the Cholesky factorization of $A A^T$ and each preconditioning step requires the solution of two triangular linear systems. The last statement holds in particular when the Cholesky factor has large number of fill-ins. Thus in our implementation, we simply use $\text{diag}(\widehat{M})$ as the preconditioner rather than \widehat{M} .

Next we discuss how to compute the matrix-vector multiplication Md for a given $d \in \mathbb{R}^m$ efficiently by exploiting the structure of Ω . Observe that $Md = \sigma \mathcal{A}(Y)$, where $Y = Q(\Omega \circ (Q^T D Q)) Q^T$ with $D = \mathcal{A}^* d$. Thus the efficient computation of Md relies on our ability to compute the matrix Y efficiently given D . By noting that

$$Y = [Q_{\bar{\gamma}} \ Q_{\gamma}] \begin{bmatrix} Q_{\bar{\gamma}}^T D Q_{\bar{\gamma}} & \nu_{\bar{\gamma}\gamma} \circ (Q_{\bar{\gamma}}^T D Q_{\gamma}) \\ \nu_{\bar{\gamma}\gamma}^T \circ (Q_{\gamma}^T D Q_{\bar{\gamma}}) & 0 \end{bmatrix} \begin{bmatrix} Q_{\bar{\gamma}}^T \\ Q_{\gamma}^T \end{bmatrix} = H + H^T, \quad (73)$$

where $H = Q_{\bar{\gamma}} \left[\frac{1}{2} (U Q_{\bar{\gamma}}) Q_{\bar{\gamma}}^T + (\nu_{\bar{\gamma}\gamma} \circ (U Q_{\gamma})) Q_{\gamma}^T \right]$ with $U = Q_{\bar{\gamma}}^T D$, it is easy to see that Y can be computed in at most $8|\bar{\gamma}|n^2$ flops. By considering $Y = D - Q((E - \Omega) \circ (Q^T D Q)) Q^T$, where E is the matrix of all ones, one can also compute Y in at most $8|\gamma|n^2$ flops. Thus Y can be computed in at most $8 \min\{|\bar{\gamma}|, |\gamma|\} n^2$ flops. The above computational complexity shows that the SDPNAL algorithm is able to take advantage of any low-rank or high-rank property of the optimal solution \bar{X} to reduce computational cost. In contrast, for inexact interior-point methods such as those proposed in [40], the matrix-vector multiplication in each CG iteration would require $\Theta(n^3)$ flops.

Finally, we should mention that the computational cost of the full eigenvalue decomposition in (52) can sometime dominate the cost of solving (72), especially when n is large. In our implementation, we use the LAPACK routine `dsyevd.f` (based on a divide-and-conquer strategy) to compute the full eigenvalue decomposition of a symmetric matrix. We have found it to be 7 to 10 times faster than MATLAB's `eig` routine when n is larger than 500.

5.1 Conditioning of M

Recall that under the conditions stated in Theorem 4.2 where the sequences $\{y^k\}$ and $\{X^k\}$ generated by Algorithm 3 converge to the solution \bar{y} and \bar{X} , respectively. Let

$$\bar{S} = \mathcal{A}^* \bar{y} - C.$$

For simplicity, we assume that strict complementarity holds for \bar{X}, \bar{S} , i.e., $\bar{X} + \bar{S} \succ 0$. We also assume that the constraint nondegenerate conditions (37) and (59) hold for \bar{y} and \bar{X} , respectively.

We shall now analyse the conditioning of the matrix M corresponding to the pair (\bar{y}, \bar{X}) . Proposition 3.2 assured that M is positive definite, but to estimate the convergence of the CG method for solving (72), we need to estimate the condition number of M .

From the fact that $\bar{X} \bar{S} = 0$, we have the following eigenvalue decomposition:

$$\bar{X} - \sigma \bar{S} = Q \begin{bmatrix} \Lambda^X & 0 \\ 0 & -\sigma \Lambda^S \end{bmatrix} Q^T, \quad (74)$$

where $\Lambda^X = \text{diag}(\lambda^X) \in \mathbb{R}^{r \times r}$ and $\Lambda^S = \text{diag}(\lambda^S) \in \mathbb{R}^{(n-r) \times (n-r)}$ are diagonal matrices of positive eigenvalues of \bar{X} and \bar{S} , respectively. Define the index sets $\bar{\gamma} := \{1, \dots, r\}$, $\gamma := \{r+1, \dots, n\}$. Let

$$\Omega = \begin{bmatrix} E_{\bar{\gamma}\bar{\gamma}} & \nu_{\bar{\gamma}\gamma} \\ \nu_{\bar{\gamma}\gamma}^T & 0 \end{bmatrix}, \quad \nu_{ij} := \frac{\lambda_i^X}{\lambda_i^X + \sigma \lambda_{j-r}^S}, \quad i \in \bar{\gamma}, j \in \gamma, \quad (75)$$

and

$$c_1 = \frac{\min(\lambda^X)}{\min(\lambda^X)/\sigma + \max(\lambda^S)}, \quad c_2 = \frac{\max(\lambda^X)}{\max(\lambda^X)/\sigma + \min(\lambda^S)} < \sigma.$$

Then $c_1 \leq \sigma \nu_{ij} \leq c_2$, $i \in \bar{\gamma}, j \in \gamma$.

Consider the decomposition in (74) for the pair (\bar{y}, \bar{X}) and let ν be defined as in (75). Then we have

$$M = \sigma \left(\tilde{A}_1 \tilde{A}_1^T + \tilde{A}_2 D_2 \tilde{A}_2^T + \tilde{A}_3 D_3 \tilde{A}_3^T \right) \quad (76)$$

where $\tilde{A}_1 = A Q_{\bar{\gamma}} \otimes Q_{\bar{\gamma}}$, $\tilde{A}_2 = A Q_{\gamma} \otimes Q_{\bar{\gamma}}$, $\tilde{A}_3 = A Q_{\bar{\gamma}} \otimes Q_{\gamma}$, $D_2 = \text{diag}(\text{vec}(\nu_{\bar{\gamma}\gamma}))$, and $D_3 = \text{diag}(\text{vec}(\nu_{\bar{\gamma}\gamma}^T))$. Since $c_1 I \preceq \sigma D_2$, $\sigma D_3 \preceq c_2 I \prec \sigma I$, it is rather easy to deduce from (76) that

$$c_1 \left(\tilde{A}_1 \tilde{A}_1^T + \tilde{A}_2 \tilde{A}_2^T + \tilde{A}_3 \tilde{A}_3^T \right) \preceq M \preceq \sigma \left(\tilde{A}_1 \tilde{A}_1^T + \tilde{A}_2 \tilde{A}_2^T + \tilde{A}_3 \tilde{A}_3^T \right).$$

Hence we obtain the following bound on the condition number of M :

$$\kappa(M) \leq \frac{\sigma}{c_1} \kappa \left(\tilde{A}_1 \tilde{A}_1^T + \tilde{A}_2 \tilde{A}_2^T + \tilde{A}_3 \tilde{A}_3^T \right) = \frac{\sigma}{c_1} \kappa \left([\tilde{A}_1, \tilde{A}_2, \tilde{A}_3] \right)^2. \quad (77)$$

The above upper bound suggests that $\kappa(M)$ can potentially be large if any of the following factors are large: (i) σ ; (ii) c_1 ; (iii) $\kappa([\tilde{A}_1, \tilde{A}_2, \tilde{A}_3])$. Observe that c_1 is approximately equal to $\min(\lambda^X)/\max(\lambda^S)$ if $\min(\lambda^X)/\sigma \ll \max(\lambda^S)$. Thus we see that a small ratio in $\min(\lambda^X)/\max(\lambda^S)$ can potentially lead to a large $\kappa(M)$. Similarly, even though the constraint nondegenerate condition (59) states that $\kappa([\tilde{A}_1, \tilde{A}_2, \tilde{A}_3])$ is finite (this is an equivalent condition), its actual value can affect the conditioning of M quite dramatically. In particular, if \bar{X} is nearly degenerate, i.e., $\kappa([\tilde{A}_1, \tilde{A}_2, \tilde{A}_3])$ is large, then $\kappa(M)$ can potentially be very large.

6 Numerical Experiments

We implemented the Newton-CG augmented Lagrangian (SDPNAL) algorithm in MATLAB to solve a variety of large SDP problems with m up to 2,156,544 and n up to 4,110 on a PC (Intel Xeon 3.2 GHz with 4G of RAM). We measure the infeasibilities and optimality for the primal and dual problems as follows:

$$R_D = \frac{\|C + S - \mathcal{A}^*y\|}{1 + \|C\|}, \quad R_P = \frac{\|b - \mathcal{A}(X)\|}{1 + \|b\|}, \quad \text{gap} = \frac{b^T y - \langle C, X \rangle}{1 + |b^T y| + |\langle C, X \rangle|}, \quad (78)$$

where $S = (\Pi_{\mathcal{S}_+^n}(W) - W)/\sigma$ with $W = X - \sigma(\mathcal{A}^*y - C)$. The above measures are the same as those adopted in the Seventh DIMACS Implementation Challenge [15], except that we used the Euclidean norms $\|b\|$ and $\|C\|$ in the denominators instead of ∞ -norms. We do not check the infeasibilities of the conditions $X \succeq 0$, $Z \succeq 0$, $XZ = 0$, since they are satisfied up to machine precision throughout the SDPNAL algorithm.

In our numerical experiments, we stop the SDPNAL algorithm when

$$\max\{R_D, R_P\} \leq 10^{-6}. \quad (79)$$

We choose the initial iterate $y^0 = 0$, $X^0 = 0$, and $\sigma_0 = 10$.

In solving the subproblem (47), we cap the number of Newton iterations to be 40, while in computing the inexact Newton direction from (62), we stop the CG solver when the maximum number of CG steps exceeds 500, or when the convergence is too slow in that the reduction in the residual norm is exceedingly small.

In this paper, we will mainly compare the performance of the SDPNAL algorithm with the boundary-point method, introduced in [24], that is coded in the MATLAB program `mprw.m` downloaded from F. Rendl's web page. It basically implements the following algorithm: given $\sigma_0 > 0$, $X^0 \in \mathcal{S}^n$, $y^0 \in \mathbb{R}^m$, accuracy level ε , perform the following loop:

$$\begin{aligned} W &= X^j - \sigma_j(\mathcal{A}^*y^j - C), \quad X^{j+1} = \Pi_{\mathcal{S}_+^n}(W), \quad S = (X^{j+1} - W)/\sigma_j \\ y^{j+1} &= y^j - (\sigma_j \mathcal{A} \mathcal{A}^*)^{-1}(b - \mathcal{A}(X^{j+1})) \\ R_P &= \|b - \mathcal{A}(X^{j+1})\|/(1 + \|b\|), \quad R_D = \|C + S - \mathcal{A}^*y^{j+1}\|/(1 + \|C\|) \\ \text{If } \max\{R_P, R_D\} &\leq \varepsilon, \text{ stop; else, update } \sigma_j, \text{ end} \end{aligned}$$

Note that in the second step of the above algorithm, it is actually applying one iteration of a modified gradient method to solve the subproblem (47). But as the iterate y^{j+1} in the above algorithm is not necessary a good approximate minimizer for (47), there is no convergence guarantee for the algorithm implemented. Next, a remark on the computational aspects of the above algorithm. Suppose that the Cholesky factorization of $\mathcal{A}\mathcal{A}^*$ is pre-computed. Then each iteration of the above algorithm requires the solution of two triangular linear systems and one full eigenvalue decomposition of an $n \times n$ symmetric matrix. Thus each iteration of the algorithm may become rather expensive when the Cholesky factor of $\mathcal{A}\mathcal{A}^*$ is fairly dense or when $n \geq 500$, and the whole algorithm may be very expensive if a large number of iterations is needed to reach the desired accuracy. In our experiments, we set the maximum number of iterations allowed in the boundary-point method to 2,000.

In the program `mprw.m`, the authors suggested choosing σ_0 in the interval $[0.1, 10]$ if the SDP data is normalized. But we should mention that the performance of the boundary-point method is quite sensitive to the choice of σ_0 . Another point mentioned in [24] is that when the rank of the optimal solution \bar{X} is much smaller than n , the boundary-point method typically would perform poorly.

6.1 Random sparse SDPs

We first consider the collection of random sparse SDPs tested in [19], which reported the performance of the boundary-point method introduced in [24].

In Table 1, we give the results obtained by the SDPNAL algorithm for the sparse SDPs considered in [19]. The first three columns give the problem name, the dimension of the variable y (m), the size of the matrix C (n_s), and the number of linear inequality constraints (n_l) in (D) , respectively. The middle five columns give the number of outer iterations, the total number of inner iterations, the average number of PCG steps taken to solve (72), the objective values $\langle C, X \rangle$ and $b^T y$, respectively. The relative infeasibilities and gap, as well as times (in the format hours:minutes:seconds) are listed in the last four columns.

Table 2 lists the results obtained by the boundary-point method implemented in the program `mprw.m`.

Comparing the results in Tables 1 and 2, we observe that the boundary-point method outperformed the SDPNAL algorithm. The former is about 2 to 5 times faster than the latter on most of the problems. It is rather surprising that the boundary-point method implemented in `mprw.m`, being a gradient based method and without convergence guarantee, can be so efficient in solving this class of sparse random SDPs, with all the SDPs solved within 250 iterations. For this collection of SDPs, the ratios $\text{rank}(\bar{X})/n$ for all the problems, except for Rn6m20p4, are greater than 0.25.

Table 1: Results for the SDPNAL algorithm on the random sparse SDPs considered in [19].

problem	$m \mid n_s; n_l$	it	itsub	pcg	$\langle C, X \rangle$	$b^T y$	$R_P \mid R_D$	gap	time
Rn3m20p3	20000 \mid 300;	7	47	77.6	7.61352657 2	7.61379700 2	6.1-7	9.2-7	-1.8-5 2:03
Rn3m25p3	25000 \mid 300;	5	55	81.3	7.38403202 1	7.38668475 1	9.5-7	9.1-7	-1.8-4 3:13
Rn3m10p4	10000 \mid 300;	11	50	16.9	1.65974701 2	1.65997274 2	1.4-7	7.0-7	-6.8-5 37

Table 1: Results for the SDPNAL algorithm on the random sparse SDPs considered in [19].

problem	$m \mid n_s; n_t$	it	itsub	pcg	$\langle C, X \rangle$	$b^T y$	$R_P \mid R_D \mid \text{gap}$	time
Rn4m30p3	30000 400;	8	44	64.3	1.07213940 3	1.07216947 3	1.0-7 7.0-7 -1.4-5	3:06
Rn4m40p3	40000 400;	7	77	86.6	8.05769056 2	8.05787521 2	3.7-7 4.6-7 -1.1-5	9:09
Rn4m15p4	15000 400;	12	54	16.4	-6.55000447 2	-6.54983628 2	2.2-7 4.0-7 -1.3-5	1:17
Rn5m30p3	30000 500;	11	51	43.8	1.10762661 3	1.10765984 3	1.9-7 7.4-7 -1.5-5	3:31
Rn5m40p3	40000 500;	10	52	46.6	8.16611193 2	8.16631468 2	1.4-7 4.1-7 -1.2-5	4:23
Rn5m50p3	50000 500;	8	45	62.8	3.64946178 2	3.64977699 2	8.1-7 5.6-7 -4.3-5	6:07
Rn5m20p4	20000 500;	12	50	15.1	3.28004397 2	3.28050465 2	3.1-7 7.7-7 -7.0-5	1:51
Rn6m40p3	40000 600;	12	55	39.0	3.06617262 2	3.06643231 2	9.9-8 4.7-7 -4.2-5	5:22
Rn6m50p3	50000 600;	10	50	58.2	-3.86413091 2	-3.86353173 2	2.8-7 8.5-7 -7.7-5	7:53
Rn6m60p3	60000 600;	9	47	48.3	6.41737682 2	6.41803361 2	5.0-7 8.7-7 -5.1-5	7:00
Rn6m20p4	20000 600;	13	56	13.8	1.04526971 3	1.04531605 3	1.6-7 7.7-7 -2.2-5	2:05
Rn7m50p3	50000 700;	12	52	31.6	3.13203609 2	3.13240876 2	7.4-7 5.4-7 -5.9-5	6:18
Rn7m70p3	70000 700;	10	48	41.6	-3.69557843 2	-3.69479811 2	2.4-7 8.7-7 -1.1-4	8:48
Rn8m70p3	70000 800;	11	51	33.3	2.33139641 3	2.33149302 3	1.8-7 9.9-7 -2.1-5	9:37
Rn8m100p3	100000 800;	10	52	55.8	2.25928848 3	2.25937157 3	1.3-7 7.3-7 -1.8-5	18:49

Table 2: Results obtained by the boundary-point method in [19] on the random sparse SDPs considered therein. The parameter σ_0 is set to 0.1, which gives better timings than the default initial value of 1.

problem	$m \mid n_s; n_t$	it	$\langle C, X \rangle$	$b^T y$	$R_P \mid R_D \mid \text{gap}$	time
Rn3m20p3	20000 300;	162	7.61352301 2	7.61351956 2	9.9-7 3.3-8 2.3-7	49
Rn3m25p3	25000 300;	244	7.38383593 1	7.38384809 1	9.3-7 4.8-8 -8.2-7	3:11
Rn3m10p4	10000 300;	148	1.65974687 2	1.65975074 2	9.8-7 7.3-8 -1.2-6	1:07
Rn4m30p3	30000 400;	143	1.07214130 3	1.07213935 3	9.6-7 2.7-8 9.1-7	57
Rn4m40p3	40000 400;	193	8.05769815 2	8.05768569 2	9.3-7 3.3-8 7.7-7	5:02
Rn4m15p4	15000 400;	168	-6.55000133 2	-6.54998597 2	9.9-7 1.1-7 -1.2-6	2:15
Rn5m30p3	30000 500;	151	1.10762655 3	1.10762734 3	9.9-7 8.4-8 -3.6-7	50
Rn5m40p3	40000 500;	136	8.16610180 2	8.16610683 2	9.6-7 3.6-8 -3.1-7	58
Rn5m50p3	50000 500;	149	3.64945604 2	3.64945078 2	9.7-7 2.3-8 7.2-7	2:56
Rn5m20p4	20000 500;	196	3.28004579 2	3.28010479 2	9.9-7 2.1-7 -9.0-6	3:36
Rn6m40p3	40000 600;	153	3.06617946 2	3.06618173 2	9.5-7 8.0-8 -3.7-7	1:21
Rn6m50p3	50000 600;	142	-3.86413897 2	-3.86413511 2	9.9-7 5.7-8 -5.0-7	1:21
Rn6m60p3	60000 600;	137	6.41736718 2	6.41736746 2	9.9-7 3.0-8 -2.2-8	2:09
Rn6m20p4	20000 600;	226	1.04526808 3	1.04528328 3	9.9-7 3.9-7 -7.3-6	2:52
Rn7m50p3	50000 700;	165	3.13202583 2	3.13205602 2	9.9-7 1.1-7 -4.8-6	2:07
Rn7m70p3	70000 700;	136	-3.69558765 2	-3.69558700 2	9.9-7 4.2-8 -8.9-8	2:10
Rn8m70p3	70000 800;	158	2.33139551 3	2.33139759 3	9.9-7 8.3-8 -4.5-7	2:54
Rn8m100p3	100000 800;	135	2.25928693 3	2.25928781 3	9.4-7 2.9-8 -1.9-7	4:16

6.2 SDPs arising from relaxation of frequency assignment problems

Next we consider SDPs arising from semidefinite relaxation of frequency assignment problems (FAP) [10]. The explicit description of the SDP in the form (P) is given in [6, equation (5)].

Observe that for the FAP problems, the SDPs contain non-negative vector variables in addition to positive semidefinite matrix variables. However, it is easy to extend the SDPNAL algorithm and the boundary-point method in `mprw.m` to accommodate the non-negative variables.

Tables 3 and 4 list the results obtained by the SDPNAL algorithm and the boundary-

point method for the SDP relaxation of frequency assignment problems tested in [6], respectively. For this collection of SDPs, the SDPNAL algorithm outperformed the boundary-point method. While the SDPNAL algorithm can achieve rather high accuracy in $\max\{R_P, R_D, \text{gap}\}$ for all the SDPs, the boundary-point method fails to achieve satisfactory accuracy after 2000 iterations in that the primal and dual objective values obtained have yet to converge close to the optimal values. The results in Table 4 demonstrate a phenomenon that is typical of a purely gradient based method, i.e., it may stagnate or converge very slowly well before the required accuracy is achieved.

It is interesting to note that for this collection, the SDP problems (D) and (P) are likely to be both degenerate at the optimal solution \bar{y} and \bar{X} , respectively. For example, the problem **fap01** is both primal and dual degenerate in that $\kappa(\tilde{A}_1) \approx 3.9 \times 10^{12}$ and $\kappa([\tilde{A}_1, \tilde{A}_2, \tilde{A}_3]) \approx 1.4 \times 10^{12}$, where $\tilde{A}_1, \tilde{A}_2, \tilde{A}_3$ are defined as in (76). Similarly, for **fap02**, we have $\kappa(\tilde{A}_1) \approx 2.3 \times 10^{12}$ and $\kappa([\tilde{A}_1, \tilde{A}_2, \tilde{A}_3]) \approx 1.7 \times 10^{12}$. It is surprising that the SDPNAL algorithm can attain the required accuracy within moderate CPU time despite the fact that the problems do not satisfy the constraint nondegeneracy conditions (37) and (59) at the optimal solution \bar{y} and \bar{X} .

The SDPs arising from FAP problems form a particularly difficult class of problems. Previous methods such as the spectral bundle (SB) method [13], the BMZ method (a log-barrier method applied to a nonlinear programming reformulation of (D)) [6], and inexact interior-point method [40] largely fail to solve these SDPs to satisfactory accuracy within moderate computer time. For example, the SB and BMZ methods took more than 50 and 3.3 hours, respectively, to solve **fap09** on an SGI Origin2000 computer using a single 300MHz R1200 processor. The inexact interior-point method [40] took more than 2.5 hours to solve the same problem on a 700MHz HP c3700 workstation. Comparatively, our SDPNAL algorithm took only 41 seconds to solve **fap09** to the same accuracy or better. In [20], the largest problem **fap36** was tested on the SB and BMZ methods using a 450MHz Sun Ultra 60 workstation. The SB and BMZ methods obtained the lower bounds of 63.77 and 63.78 for the optimal objective value after running for 4250 and 2036 hours, respectively. In contrast, our SDPNAL algorithm was able to solve **fap36** to a rather good accuracy in about 65 hours, and obtained the approximate optimal objective value of 69.85.

Table 3: Results for the SDPNAL algorithm on the frequency assignment problems.

problem	m	$n_s; n_l$	it itsub pcg	$\langle C, X \rangle$	$b^T y$	R_P R_D gap	time
fap01	1378	52; 1160	20 109 33.2	3.28834503-2	3.28832952-2	8.4-7 1.0-7 1.5-7	06
fap02	1866	61; 1601	20 81 21.4	6.90524269-4	7.02036467-4	8.4-7 3.5-7 -1.1-5	04
fap03	2145	65; 1837	20 102 38.6	4.93726225-2	4.93703591-2	1.2-7 2.5-7 2.1-6	07
fap04	3321	81; 3046	21 173 43.5	1.74829592-1	1.74844758-1	2.0-7 6.4-7 -1.1-5	17
fap05	3570	84; 3263	21 244 56.6	3.08361964-1	3.08294715-1	7.6-6 6.2-7 4.2-5	32
fap06	4371	93; 3997	21 187 55.3	4.59325368-1	4.59344513-1	7.6-7 6.8-7 -10.0-6	27
fap07	4851	98; 4139	22 179 61.4	2.11762487 0	2.11763204 0	9.9-7 4.9-7 -1.4-6	30
fap08	7260	120; 6668	21 113 45.0	2.43627884 0	2.43629328 0	2.8-7 9.9-7 -2.5-6	21
fap09	15225	174; 14025	22 120 38.4	1.07978114 1	1.07978423 1	8.9-7 9.6-7 -1.4-6	41
fap10	14479	183; 13754	23 140 57.4	9.67044948-3	9.74974306-3	1.5-7 9.3-7 -7.8-5	1:18
fap11	24292	252; 23275	25 148 69.0	2.97000004-2	2.98373492-2	7.7-7 6.0-7 -1.3-4	3:21
fap12	26462	369; 24410	25 169 81.3	2.73251961-1	2.73410714-1	6.0-7 7.8-7 -1.0-4	9:07
fap25	322924	2118; 311044	24 211 84.8	1.28761356 1	1.28789892 1	3.2-6 5.0-7 -1.1-4	10:53:22
fap36	1154467	4110; 1112293	17 197 87.4	6.98561787 1	6.98596286 1	7.7-7 6.7-7 -2.5-5	65:25:07

Table 4: Results obtained by the boundary-point method in [19] on the frequency assignment problems. The parameter σ_0 is set to 1 (better than 0.1).

problem	m	$n_s; n_l$	it	$\langle C, X \rangle$	$b^T y$	R_P	R_D	gap	time
fap01	1378	52; 1160	2000	3.49239684-2	3.87066984-2	5.4-6	1.7-4	-3.5-3	15
fap02	1866	61; 1601	2000	4.06570342-4	1.07844848-3	1.6-5	7.5-5	-6.7-4	16
fap03	2145	65; 1837	2000	5.02426246-2	5.47858318-2	1.5-5	1.5-4	-4.1-3	17
fap04	3321	81; 3046	2000	1.77516830-1	1.84285835-1	4.5-6	1.7-4	-5.0-3	24
fap05	3570	84; 3263	2000	3.11422846-1	3.18992969-1	1.1-5	1.6-4	-4.6-3	25
fap06	4371	93; 3997	2000	4.60368585-1	4.64270062-1	7.5-6	9.8-5	-2.0-3	27
fap07	4851	98; 4139	2000	2.11768050 0	2.11802220 0	2.5-6	1.5-5	-6.5-5	25
fap08	7260	120; 6668	2000	2.43638729 0	2.43773801 0	2.6-6	3.5-5	-2.3-4	34
fap09	15225	174; 14025	2000	1.07978251 1	1.07982902 1	9.2-7	9.8-6	-2.1-5	59
fap10	14479	183; 13754	2000	1.70252739-2	2.38972400-2	1.1-5	1.1-4	-6.6-3	1:25
fap11	24292	252; 23275	2000	4.22711513-2	5.94650102-2	8.8-6	1.4-4	-1.6-2	2:31
fap12	26462	369; 24410	2000	2.93446247-1	3.26163363-1	6.0-6	1.5-4	-2.0-2	4:37
fap25	322924	2118; 311044	2000	1.31895665 1	1.35910952 1	4.8-6	2.0-4	-1.4-2	8:04:00
fap36	1154467	4110; 1112293	2000	7.03339309 1	7.09606078 1	3.9-6	1.4-4	-4.4-3	46:59:28

6.3 SDPs arising from relaxation of maximum stable set problems

For a graph G with edge set \mathcal{E} , the stability number $\alpha(G)$ is the cardinality of a maximal stable set of G , and $\alpha(G) := \{e^T x : x_i x_j = 0, (i, j) \in \mathcal{E}, x \in \{0, 1\}^n\}$. It is known that $\alpha(G) \leq \theta(G) \leq \theta_+(G)$, where

$$\theta(G) = \max\{\langle e e^T, X \rangle : \langle E_{ij}, X \rangle = 0, (i, j) \in \mathcal{E}, \langle I, X \rangle = 1, X \succeq 0\}, \quad (80)$$

$$\theta_+(G) = \max\{\langle e e^T, X \rangle : \langle E_{ij}, X \rangle = 0, (i, j) \in \mathcal{E}, \langle I, X \rangle = 1, X \succeq 0, X \geq 0\}, \quad (81)$$

where $E_{ij} = e_i e_j^T + e_j e_i^T$ and e_i denotes column i of the identity matrix I . Note that for (81), the problem is reformulated as a standard SDP by replacing the constraint $X \geq 0$ by constraints $X - Y = 0$ and $Y \geq 0$. Thus such a reformulation introduces an additional $n(n+1)/2$ linear equality constraints to the SDP.

Table 5 lists the results obtained by the SDPNAL algorithm for the SDPs (80) arising from computing $\theta(G)$ for the maximum stable set problems. The first collection of graph instances in Table 5 are the randomly generated instances considered in [40] whereas the second collection is from the Second DIMACS Challenge on Maximum Clique Problems [42]. The last collection are graphs arising from coding theory, available from N. Sloane's web page [35].

Observe that the SDPNAL algorithm is not able to achieve the required accuracy level for some of the SDPs from Sloane's collection. It is not surprising that this may happen because many of these SDPs are degenerate at the optimal solution. For example, the problems 1dc.128 and 2dc.128 are degenerate at the optimal solutions \bar{y} even though they are nondegenerate at the optimal solutions \bar{X} .

In [19], the performance of the boundary-point method was compared with that of the iterative solver based primal-dual interior-point method in [40], as well as the iterative solver

based modified barrier method in [17], on a subset of the large SDPs arising from the first collection of random graphs. The conclusion was that the boundary-point method was between 5-10 times faster than the methods in [40] and [17]. Since the SDPNAL algorithm is at least as efficient as the boundary-point method on the theta problems for random graphs (not reported here in the interest of saving space), it is safe to assume that the SDPNAL algorithm would be at least 5-10 times faster than the methods in [40] and [17]. Note that the SDPNAL algorithm is more efficient than the boundary-point method on the collection of graphs from DIMACS. For example, the SDPNAL algorithm takes less than 100 seconds to solve the problem G43 to an accuracy of less than 10^{-6} , while the boundary-point method (with $\sigma_0 = 0.1$) takes more than 3,900 seconds to achieve an accuracy of 1.5×10^{-5} . Such a result for G43 is not surprising because the rank of the optimal X (equals to 58) is much smaller than n , and as already mentioned in [24], the boundary-point method typically would perform poorly under such a situation.

Table 5: Results for the SDPNAL algorithm on computing $\theta(G)$ in (80) for the maximum stable set problems.

problem	m $n_s; n_l$	it itsub pcg	$\langle C, X \rangle$	$b^T y$	R_P R_D gap	time
theta4	1949 200;	22 25 12.7	5.03212191 1	5.03212148 1	4.9-8 5.2-7 4.2-8	05
theta42	5986 200;	20 24 11.6	2.39317091 1	2.39317059 1	2.2-7 8.5-7 6.6-8	06
theta6	4375 300;	22 29 11.0	6.34770834 1	6.34770793 1	4.5-8 4.8-7 3.2-8	12
theta62	13390 300;	20 25 11.2	2.96412472 1	2.96412461 1	5.8-7 9.2-7 1.7-8	14
theta8	7905 400;	22 28 10.6	7.39535679 1	7.39535555 1	6.5-8 6.9-7 8.3-8	23
theta82	23872 400;	21 26 10.3	3.43668917 1	3.43668881 1	1.4-7 8.8-7 5.2-8	27
theta83	39862 400;	20 27 10.8	2.03018910 1	2.03018886 1	1.2-7 4.8-7 5.6-8	35
theta10	12470 500;	21 25 10.6	8.38059689 1	8.38059566 1	6.9-8 6.6-7 7.3-8	36
theta102	37467 500;	23 28 10.2	3.83905451 1	3.83905438 1	6.9-8 4.8-7 1.6-8	50
theta103	62516 500;	18 27 10.7	2.25285688 1	2.25285667 1	4.4-8 5.8-7 4.6-8	1:00
theta104	87245 500;	17 28 11.2	1.33361400 1	1.33361379 1	6.1-8 6.5-7 7.6-8	58
theta12	17979 600;	21 26 10.3	9.28016795 1	9.28016679 1	9.6-8 8.1-7 6.2-8	57
theta123	90020 600;	18 26 10.9	2.46686513 1	2.46686492 1	3.3-8 5.2-7 4.1-8	1:34
theta162	127600 800;	17 26 10.2	3.70097353 1	3.70097324 1	3.6-8 5.4-7 3.8-8	2:53
MANN-a27	703 378;	9 13 6.2	1.32762891 2	1.32762869 2	9.4-11 7.0-7 8.3-8	07
johnson8-4	561 70;	3 4 3.0	1.39999996 1	1.399999983 1	4.5-9 1.6-7 4.4-8	00
johnson16-	1681 120;	3 4 4.0	7.99998670 0	7.99999480 0	8.1-8 7.5-7 -4.8-7	01
san200-0.7	5971 200;	13 22 8.9	3.00000066 1	2.99999980 1	2.3-7 3.1-7 1.4-7	04
c-fat200-1	18367 200;	8 36 20.3	1.19999983 1	1.19999962 1	1.5-7 8.3-7 8.5-8	09
hamming-6-	1313 64;	3 4 4.2	5.33333334 0	5.33333330 0	4.4-11 5.8-9 2.7-9	00
hamming-8-	11777 256;	5 5 4.0	1.59999983 1	1.59999855 1	7.2-9 8.0-7 3.9-7	02
hamming-9-	2305 512;	6 6 5.2	2.24000000 2	2.24000049 2	1.2-10 2.4-7 -1.1-7	10
hamming-10	23041 1024;	7 9 5.6	1.02399780 2	1.02400070 2	7.1-8 7.1-7 -1.4-6	1:33
hamming-7-	1793 128;	4 5 4.2	4.26666667 1	4.26666645 1	4.1-12 6.6-8 2.6-8	01
hamming-8-	16129 256;	4 4 4.8	2.56000007 1	2.55999960 1	2.8-9 2.1-7 9.0-8	02
hamming-9-	53761 512;	4 6 6.5	8.53333333 1	8.53333311 1	1.4-11 3.9-8 1.3-8	10
brock200-1	5067 200;	20 24 12.6	2.74566402 1	2.74566367 1	1.2-7 6.7-7 6.3-8	06
brock200-4	6812 200;	18 23 13.0	2.12934757 1	2.12934727 1	1.1-7 5.8-7 6.8-8	06
brock400-1	20078 400;	21 25 10.6	3.97018902 1	3.97018916 1	5.4-7 9.9-7 -1.7-8	26
keller4	5101 171;	17 21 15.9	1.40122390 1	1.40122386 1	1.3-7 4.4-7 1.3-8	05
p-hat300-1	33918 300;	20 84 38.7	1.00679674 1	1.00679561 1	5.5-7 9.4-7 5.3-7	1:45
G43	9991 1000;	18 27 11.6	2.80624585 2	2.80624562 2	3.0-8 4.6-7 4.2-8	1:33
G44	9991 1000;	18 28 11.1	2.80583335 2	2.80583149 2	3.6-7 9.2-7 3.3-7	2:59
G45	9991 1000;	17 26 11.5	2.80185131 2	2.80185100 2	3.6-8 5.8-7 5.6-8	2:51
G46	9991 1000;	18 26 11.4	2.79837027 2	2.79836899 2	3.2-7 9.1-7 2.3-7	2:53
G47	9991 1000;	17 27 11.4	2.81893976 2	2.81893904 2	7.0-8 9.3-7 1.3-7	2:54
1dc.64	544 64;	22 87 61.1	1.00000038 1	9.99998513 0	6.9-7 9.2-7 8.9-7	06
1et.64	265 64;	13 16 10.0	1.87999993 1	1.88000161 1	1.2-7 7.2-7 -4.3-7	01
1tc.64	193 64;	14 25 14.1	2.00000028 1	1.99999792 1	5.5-7 9.2-7 5.7-7	01
1dc.128	1472 128;	26 160 78.3	1.68422941 1	1.68420185 1	6.4-6 6.5-7 7.9-6	31
1et.128	673 128;	14 25 11.5	2.92308767 1	2.92308940 1	7.6-7 4.5-7 -2.9-7	02
1tc.128	513 128;	12 33 10.7	3.79999935 1	3.79999915 1	1.6-7 8.5-7 2.6-8	02

Table 5: Results for the SDPNAL algorithm on computing $\theta(G)$ in (80) for the maximum stable set problems.

problem	$m \mid n_s; n_l$	it itsub pcg	$\langle C, X \rangle$	$b^T y$	$R_P \mid R_D \mid \text{gap}$	time
1zc.128	1121 128;	10 16 8.2	2.06666622 1	2.06666556 1	1.1-7 5.9-7 1.6-7	02
1dc.256	3840 256;	22 131 46.5	3.00000152 1	2.99999982 1	5.1-7 1.1-8 2.8-7	1:05
1et.256	1665 256;	22 105 30.5	5.51142859 1	5.51142381 1	3.2-7 5.3-7 4.3-7	52
1tc.256	1313 256;	29 211 82.2	6.34007911 1	6.33999101 1	7.4-6 4.8-7 6.9-6	2:30
1zc.256	2817 256;	13 17 8.5	3.79999847 1	3.79999878 1	9.5-8 4.9-7 -4.1-8	05
1dc.512	9728 512;	30 181 75.7	5.30311533 1	5.30307418 1	2.0-6 4.2-7 3.8-6	12:07
1et.512	4033 512;	16 90 40.1	1.04424062 2	1.04424003 2	9.9-7 7.9-7 2.8-7	3:48
1tc.512	3265 512;	28 316 83.4	1.13401460 2	1.13400320 2	3.3-6 6.9-7 5.0-6	28:53
2dc.512	54896 512;	27 258 61.3	1.17732077 1	1.17690636 1	2.4-5 5.0-7 1.7-4	32:16
1zc.512	6913 512;	12 21 10.6	6.87499484 1	6.87499880 1	9.0-8 3.7-7 -2.9-7	44
1dc.1024	24064 1024;	26 130 64.0	9.59854968 1	9.59849281 1	1.4-6 4.9-7 2.9-6	41:26
1et.1024	9601 1024;	19 117 76.8	1.84226899 2	1.84226245 2	2.5-6 3.5-7 1.8-6	1:01:14
1tc.1024	7937 1024;	30 250 79.1	2.06305257 2	2.06304344 2	1.7-6 6.3-7 2.2-6	1:48:04
1zc.1024	16641 1024;	15 22 12.2	1.28666659 2	1.28666651 2	2.8-8 3.0-7 3.3-8	4:15
2dc.1024	169163 1024;	28 219 68.0	1.86426368 1	1.86388392 1	7.8-6 6.8-7 9.9-5	2:57:56
1dc.2048	58368 2048;	27 154 82.5	1.74729647 2	1.74729135 2	7.7-7 4.0-7 1.5-6	6:11:11
1et.2048	22529 2048;	22 138 81.6	3.42029313 2	3.42028707 2	6.9-7 6.3-7 8.8-7	7:13:55
1tc.2048	18945 2048;	26 227 78.5	3.74650769 2	3.74644820 2	3.3-6 3.7-7 7.9-6	9:52:09
1zc.2048	39425 2048;	13 24 14.0	2.37400485 2	2.37399909 2	1.5-7 7.3-7 1.2-6	45:16
2dc.2048	504452 2048;	27 184 67.1	3.06764717 1	3.06737001 1	3.7-6 4.5-7 4.4-5	15:13:19

Table 6: Results for the SDPNAL algorithm on computing $\theta_+(G)$ in (81) for the maximum stable set problems.

problem	$m - n_l \mid n_s; n_l$	it itsub pcg	$\langle C, X \rangle$	$b^T y$	$R_P \mid R_D \mid \text{gap}$	time
theta4	1949 200; 20100	20 67 31.3	4.98690157 1	4.98690142 1	4.6-8 7.9-7 1.4-8	33
theta42	5986 200; 20100	18 41 26.0	2.37382088 1	2.37382051 1	5.7-7 9.8-7 7.6-8	22
theta6	4375 300; 45150	15 61 27.7	6.29618432 1	6.29618399 1	2.9-8 7.6-7 2.6-8	1:03
theta62	13390 300; 45150	16 38 22.4	2.93779448 1	2.93779378 1	4.0-7 6.6-7 1.2-7	44
theta8	7905 400; 80200	13 52 29.8	7.34078436 1	7.34078372 1	2.8-7 7.3-7 4.3-8	1:54
theta82	23872 400; 80200	13 45 28.6	3.40643550 1	3.40643458 1	4.0-7 9.9-7 1.3-7	2:09
theta83	39862 400; 80200	13 40 23.0	2.01671070 1	2.01671031 1	1.8-7 4.5-7 9.4-8	1:50
theta10	12470 500; 125250	12 54 32.0	8.31489963 1	8.31489897 1	1.3-7 8.0-7 4.0-8	3:35
theta102	37467 500; 125250	15 44 27.6	3.80662551 1	3.80662486 1	4.5-7 9.1-7 8.4-8	3:31
theta103	62516 500; 125250	12 38 26.5	2.23774200 1	2.23774190 1	1.0-7 9.3-7 2.3-8	3:28
theta104	87245 500; 125250	14 35 22.0	1.32826023 1	1.32826068 1	8.1-7 8.4-7 -1.6-7	2:35
theta12	17979 600; 180300	12 53 33.9	9.20908140 1	9.20908772 1	6.5-7 6.6-7 -3.4-7	5:38
theta123	90020 600; 180300	15 43 29.2	2.44951438 1	2.44951497 1	7.7-7 8.5-7 -1.2-7	6:44
theta162	127600 800; 320400	14 42 26.2	3.67113362 1	3.67113729 1	8.1-7 4.5-7 -4.9-7	11:24
MANN-a27	703 378; 71631	7 26 21.5	1.32762850 2	1.32762894 2	2.1-7 6.8-7 -1.6-7	35
johnson8-4	561 70; 2485	5 6 7.0	1.39999984 1	1.40000110 1	2.2-8 5.8-7 -4.4-7	01
johnson16-	1681 120; 7260	6 7 7.0	7.99999871 0	8.00000350 0	5.3-8 4.3-7 -2.8-7	01
san200-0.7	5971 200; 20100	16 33 14.5	3.00000135 1	2.99999957 1	5.9-7 4.0-7 2.9-7	11
c-fat200-1	18367 200; 20100	7 48 42.1	1.20000008 1	1.19999955 1	1.3-7 9.5-7 2.1-7	36
hamming-6-	1313 64; 2080	6 7 7.0	4.00000050 0	3.99999954 0	5.7-9 6.2-8 1.1-7	01
hamming-8-	11777 256; 32896	8 10 7.2	1.59999978 1	1.59999873 1	8.5-9 3.7-7 3.2-7	05
hamming-9-	2305 512; 131328	3 8 8.4	2.24000002 2	2.24000016 2	4.6-8 5.9-7 -3.1-8	18
hamming-10	23041 1024; 524800	8 17 10.6	8.53334723 1	8.53334002 1	6.0-8 7.9-7 4.2-7	4:35
hamming-7-	1793 128; 8256	12 26 8.2	3.59999930 1	3.60000023 1	3.8-8 1.3-7 -1.3-7	03
hamming-8-	16129 256; 32896	6 7 7.0	2.56000002 1	2.56000002 1	2.0-9 5.1-9 -2.7-10	05
hamming-9-	53761 512; 131328	11 18 10.6	5.86666682 1	5.86666986 1	1.1-7 4.4-7 -2.6-7	42
brock200-1	5067 200; 20100	17 48 30.7	2.71967178 1	2.71967126 1	3.8-7 7.0-7 9.3-8	27
brock200-4	6812 200; 20100	18 40 23.4	2.11210736 1	2.11210667 1	5.4-8 9.9-7 1.6-7	21
brock400-1	20078 400; 80200	14 42 26.4	3.93309197 1	3.93309200 1	9.5-7 6.5-7 -3.5-9	1:45
keller4	5101 171; 14706	18 73 43.3	1.34658980 1	1.34659082 1	6.1-7 9.7-7 -3.7-7	43
p-hat300-1	33918 300; 45150	21 123 73.5	1.00202172 1	1.00202006 1	8.7-7 7.2-7 7.9-7	6:50
G43	9991 1000; 500500	9 126 52.2	2.79735847 2	2.79735963 2	9.1-7 8.1-7 -2.1-7	52:00
G44	9991 1000; 500500	8 122 51.4	2.79746110 2	2.79746078 2	3.3-7 6.2-7 5.7-8	49:32
G45	9991 1000; 500500	9 124 52.0	2.79317531 2	2.79317544 2	9.3-7 8.6-7 -2.4-8	50:25
G46	9991 1000; 500500	8 112 52.2	2.79032493 2	2.79032511 2	3.5-7 9.6-7 -3.3-8	44:38
G47	9991 1000; 500500	9 102 53.1	2.80891719 2	2.80891722 2	4.7-7 6.0-7 -5.1-9	40:27

Table 6: Results for the SDPNAL algorithm on computing $\theta_+(G)$ in (81) for the maximum stable set problems.

problem	$m - n_l \mid n_s; n_l$	it itsub pcg	$\langle C, X \rangle$	$b^T y$	$R_P \mid R_D \mid \text{gap}$	time
1dc.64	544 64; 2080	12 107 39.6	9.99999884 0	9.99998239 0	1.2-7 9.9-7 7.8-7	09
1et.64	265 64; 2080	12 24 17.0	1.88000008 1	1.87999801 1	3.2-8 6.6-7 5.4-7	02
1tc.64	193 64; 2080	12 54 37.9	1.99999995 1	1.99999784 1	7.9-8 9.3-7 5.2-7	05
1dc.128	1472 128; 8256	28 277 117.4	1.66790646 1	1.66783087 1	5.4-5 2.6-8 2.2-5	3:16
1et.128	673 128; 8256	12 41 26.9	2.92309168 1	2.92308878 1	8.3-7 6.6-7 4.9-7	08
1tc.128	513 128; 8256	14 51 28.0	3.80000025 1	3.79999965 1	2.3-7 4.4-7 7.9-8	09
1zc.128	1121 128; 8256	14 23 12.9	2.06667715 1	2.06666385 1	8.5-7 9.3-7 3.1-6	04
1dc.256	3840 256; 32896	21 131 39.3	2.99999987 1	3.00000004 1	4.3-8 1.7-8 -2.8-8	2:24
1et.256	1665 256; 32896	21 195 108.4	5.44706489 1	5.44652433 1	2.3-5 4.0-7 4.9-5	8:37
1tc.256	1313 256; 32896	23 228 137.5	6.32416075 1	6.32404374 1	1.5-5 7.5-7 9.2-6	11:17
1zc.256	2817 256; 32896	17 40 13.6	3.73333432 1	3.73333029 1	1.7-7 8.2-7 5.3-7	21
1dc.512	9728 512; 131328	24 204 72.9	5.26955154 1	5.26951392 1	2.7-6 5.4-7 3.5-6	36:48
1et.512	4033 512; 131328	17 181 147.4	1.03625531 2	1.03555196 2	1.3-4 5.8-7 3.4-4	51:10
1tc.512	3265 512; 131328	28 396 143.9	1.12613099 2	1.12538820 2	9.3-5 7.9-7 3.3-4	2:14:55
2dc.512	54896 512; 131328	33 513 106.2	1.13946331 1	1.13857125 1	2.1-4 7.7-7 3.8-4	2:25:15
1zc.512	6913 512; 131328	11 57 37.3	6.80000034 1	6.79999769 1	4.3-7 7.6-7 1.9-7	6:09
1dc.1024	24064 1024; 524800	24 260 81.4	9.55539508 1	9.55512205 1	1.4-5 6.9-7 1.4-5	5:03:49
1et.1024	9601 1024; 524800	20 198 155.0	1.82075477 2	1.82071562 2	4.8-6 7.0-7 1.1-5	6:45:50
1tc.1024	7937 1024; 524800	27 414 124.6	2.04591268 2	2.04236122 2	1.5-4 7.3-7 8.7-4	10:37:57
1zc.1024	16641 1024; 524800	11 67 38.1	1.27999936 2	1.27999977 2	6.4-7 5.7-7 -1.6-7	40:13
2dc.1024	169163 1024; 524800	28 455 101.8	1.77416130 1	1.77149535 1	1.6-4 6.2-7 7.3-4	11:57:25
1dc.2048	58368 2048; 2098176	20 320 73.0	1.74292685 2	1.74258827 2	1.9-5 7.1-7 9.7-5	35:52:44
1et.2048	22529 2048; 2098176	22 341 171.5	3.38193695 2	3.38166811 2	6.3-6 5.7-7 4.0-5	80:48:17
1tc.2048	18945 2048; 2098176	24 381 150.2	3.71592017 2	3.70575527 2	3.5-4 7.9-7 1.4-3	73:56:01
1zc.2048	39425 2048; 2098176	11 38 29.3	2.37400054 2	2.37399944 2	2.5-7 7.9-7 2.3-7	2:13:04
2dc.2048	504452 2048; 2098176	27 459 53.4	2.89755241 1	2.88181157 1	1.3-4 7.2-7 2.7-3	45:21:42

7 Applications to Quadratic Assignment and Binary Integer Quadratic Programming Problems

In this section, we apply our SDPNAL algorithm to compute lower bounds for quadratic assignment problems (QAPs) and binary integer quadratic (BIQ) problems through SDP relaxations. Our purpose here is to demonstrate that the SDPNAL algorithm can potentially be very efficient in solving large SDPs (and hence in computing bounds) arising from hard combinatorial problems.

Let Π be the set of $n \times n$ permutation matrices. Given matrices $A, B \in \mathbb{R}^{n \times n}$, the quadratic assignment problem is:

$$v_{\text{QAP}}^* := \min\{\langle X, AXB \rangle : X \in \Pi\}. \quad (82)$$

For a matrix $X = [x_1, \dots, x_n] \in \mathbb{R}^{n \times n}$, we will identify it with the n^2 -vector $x = [x_1; \dots; x_n]$. For a matrix $Y \in \mathbb{R}^{n^2 \times n^2}$, we let Y^{ij} be the $n \times n$ block corresponding to $x_i x_j^T$ in the matrix xx^T . It is shown in [23] that v_{QAP}^* is bounded below by the following number:

$$\begin{aligned} v &:= \min \quad \langle B \otimes A, Y \rangle \\ \text{s.t.} \quad &\sum_{i=1}^n Y^{ii} = I, \langle I, Y^{ij} \rangle = \delta_{ij} \quad \forall 1 \leq i \leq j \leq n, \\ &\langle E, Y^{ij} \rangle = 1, \quad \forall 1 \leq i \leq j \leq n, \\ &Y \succeq 0, \quad Y \geq 0, \end{aligned} \quad (83)$$

where E is the matrix of ones, and $\delta_{ij} = 1$ if $i = j$, and 0 otherwise. There are $3n(n+1)/2$ equality constraints in (83). But two of them are actually redundant, and we remove them when solving the standard SDP generated from (83). Note that [23] actually used the constraint $\langle E, Y \rangle = n^2$ in place of the last set of the equality constraints in (83). But we prefer to use the formulation here because the associated SDP has slightly better numerical behavior. Note also that the SDP problems (83) typically do not satisfy the constraint nondegenerate conditions (37) and (59) at the optimal solutions.

In our experiment, we apply the SDPNAL algorithm to the dual of (83) and hence any dual feasible solution would give a lower bound for (83). But in practice, our algorithm only delivers an approximately feasible dual solution \tilde{y} . We therefore apply the procedure given in [14, Theorem 2] to \tilde{y} to construct a valid lower bound for (83), which we denote by \underline{v} .

Table 7 lists the results of the SDPNAL algorithm on the quadratic assignment instances (83). The details of the table are the same as for Table 1 except that the objective values are replaced by the best known upper bound on (82) under the column “best upper bound” and the lower bound \underline{v} . The entries under the column under “%gap” are calculated as follows:

$$\%gap = \frac{\text{best upper bound} - \underline{v}}{\text{best upper bound}} \times 100\%.$$

We compare our results with those obtained in [5] which used a dedicated augmented Lagrangian algorithm to solve the SDP arising from applying the lift-and-project procedure of Lovász and Schrijver to (82). As the augmented Lagrangian algorithm in [5] is designed specifically for the SDPs arising the lift-and-project procedure, the details of that algorithm is very different from our SDPNAL algorithm. Note that the algorithm in [5] was implemented in C (with LAPACK library) and the results reported were obtained from a 2.4 GHz Pentium 4 PC with 1 GB of RAM (which is about 50% slower than our PC). By comparing the results in Table 7 against those in [5, Tables 6 and 7], we can safely conclude that the SDPNAL algorithm applied to (83) is superior in terms of CPU time and the accuracy of the approximate optimal solution computed. Take for example the SDPs corresponding to the QAPs `nug30` and `tai35b`, the SDPNAL algorithm obtains the lower bounds with %gap of 2.939 and 5.318 in 15,729 and 37,990 seconds respectively, whereas the the algorithm in [5] computes the bounds with %gap of 3.10 and 15.42 in 127,011 and 430,914 seconds respectively.

The paper [5] also solved the lift-and-project SDP relaxations for the maximum stable set problems (denoted as N_+ and is known to be at least as strong as θ_+) using a dedicated augmented Lagrangian algorithm. By comparing the results in Table 6 against those in [5, Table 4], we can again conclude that the SDPNAL algorithm applied to (81) is superior in terms of CPU time and the accuracy of the approximate optimal solution computed. Take for example the SDPs corresponding to the graphs `p-hat300-1` and `c-fat200-1`, the SDPNAL algorithm obtains the upper bounds of $\theta_+ = 10.0202$ and $\theta_+ = 12.0000$ in 410 and 36 seconds respectively, whereas the the algorithm in [5] computes the bounds of $N_+ = 18.6697$ and $N_+ = 14.9735$ in 322,287 and 126,103 seconds respectively.

The BIQ problem we consider is the following:

$$v_{\text{BIQ}}^* := \min\{x^T Q x : x \in \{0, 1\}^n\}, \quad (84)$$

where Q is a symmetric matrix (non positive semidefinite) of order n . A natural SDP relaxation of (84) is the following:

$$\begin{aligned} \min \quad & \langle Q, Y \rangle \\ \text{s.t.} \quad & \text{diag}(Y) - y = 0, \quad \alpha = 1, \\ & \begin{bmatrix} Y & y \\ y^T & \alpha \end{bmatrix} \succeq 0, \quad Y \geq 0, y \geq 0. \end{aligned} \quad (85)$$

Table 8 lists the results obtained by the SDPNAL algorithm on the SDPs (85) arising from the BIQ instances described in [43]. It is interesting to note that the lower bound obtained from (85) is within 10% of the optimal value v_{BIQ}^* for all the instances tested, and for the instances **gka1b–gka9b**, the lower bounds are actually equal to v_{BIQ}^* .

Table 7: Results for the SDPNAL algorithm on the quadratic assignment problems. The entries under the column “%gap” are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - n_l$	$n_s; n_l$	it itsub pcg	best upper bound	lower bound \underline{v}	R_P R_D — %gap	time
bur26a	1051	676; 228826	27 389 105.9	5.42667000 6	5.42577700 6	2.9-3 2.8-7 0.016	4:28:43
bur26b	1051	676; 228826	25 358 92.3	3.81785200 6	3.81663900 6	2.3-3 6.1-7 0.032	3:23:39
bur26c	1051	676; 228826	26 421 107.5	5.42679500 6	5.42593600 6	3.9-3 4.7-7 0.016	4:56:09
bur26d	1051	676; 228826	27 424 102.3	3.82122500 6	3.81982900 6	3.8-3 5.0-7 0.037	4:21:32
bur26e	1051	676; 228826	27 573 100.0	5.38687900 6	5.38683200 6	7.5-3 1.7-7 0.001	5:34:39
bur26f	1051	676; 228826	25 534 100.9	3.78204400 6	3.78184600 6	3.1-3 6.2-7 0.005	5:32:51
bur26g	1051	676; 228826	24 422 91.0	1.01171720 7	1.01167630 7	3.8-3 6.6-7 0.004	3:33:58
bur26h	1051	676; 228826	24 450 96.8	7.09865800 6	7.09856700 6	2.0-3 2.3-7 0.001	3:53:22
chr12a	232	144; 10440	24 314 82.5	9.55200000 3	9.55200000 3	4.6-7 4.2-12 0.000	3:02
chr12b	232	144; 10440	23 374 106.6	9.74200000 3	9.74200000 3	4.3-7 5.9-12 0.000	4:12
chr12c	232	144; 10440	25 511 103.7	1.11560000 4	1.11560000 4	1.7-3 5.6-7 0.000	3:33
chr15a	358	225; 25425	27 505 110.9	9.89600000 3	9.88800000 3	3.3-3 3.1-7 0.081	19:51
chr15b	358	225; 25425	23 385 94.0	7.99000000 3	7.99000000 3	1.9-4 3.1-8 0.000	11:42
chr15c	358	225; 25425	21 382 82.4	9.50400000 3	9.50400000 3	2.2-4 2.4-8 0.000	10:39
chr18a	511	324; 52650	32 660 111.7	1.10980000 4	1.10960000 4	8.1-3 1.7-7 0.018	57:06
chr18b	511	324; 52650	25 308 136.1	1.53400000 3	1.53400000 3	9.9-5 6.9-7 0.000	35:25
chr20a	628	400; 80200	32 563 117.8	2.19200000 3	2.19200000 3	4.3-3 2.9-8 0.000	1:28:45
chr20b	628	400; 80200	25 375 98.2	2.29800000 3	2.29800000 3	1.1-3 1.5-7 0.000	54:09
chr20c	628	400; 80200	30 477 101.0	1.41420000 4	1.41400000 4	5.5-3 5.4-7 0.014	57:26
chr22a	757	484; 117370	26 467 116.7	6.15600000 3	6.15600000 3	2.3-3 9.3-8 0.000	1:50:37
chr22b	757	484; 117370	26 465 106.4	6.19400000 3	6.19400000 3	1.8-3 6.9-8 0.000	1:47:16
chr25a	973	625; 195625	26 462 84.7	3.79600000 3	3.79600000 3	1.9-3 1.4-7 0.000	3:20:35
els19	568	361; 65341	28 554 99.5	1.72125480 7	1.72112340 7	1.0-4 6.5-7 0.008	51:52
esc16a	406	256; 32896	24 251 106.3	6.80000000 1	6.40000000 1	9.3-5 5.3-7 5.882	10:48
esc16b	406	256; 32896	26 321 80.7	2.92000000 2	2.89000000 2	5.0-4 4.9-7 1.027	10:10
esc16c	406	256; 32896	27 331 77.5	1.60000000 2	1.53000000 2	6.6-4 5.6-7 4.375	10:42
esc16d	406	256; 32896	20 62 70.8	1.60000000 1	1.30000000 1	6.1-7 8.0-7 18.750	1:45
esc16e	406	256; 32896	19 61 70.1	2.80000000 1	2.70000000 1	9.7-8 9.4-7 3.571	1:42
esc16g	406	256; 32896	23 106 109.8	2.60000000 1	2.50000000 1	2.9-7 4.7-7 3.846	4:26
esc16h	406	256; 32896	29 319 90.0	9.96000000 2	9.76000000 2	1.4-4 5.8-7 2.008	10:52
esc16i	406	256; 32896	20 106 117.4	1.40000000 1	1.20000000 1	8.6-7 6.9-7 14.286	4:51
esc16j	406	256; 32896	15 67 104.8	8.00000000 0	8.00000000 0	1.6-7 4.1-7 0.000	2:41
esc32a	1582	1024; 524800	26 232 101.9	† 1.30000000 2	1.04000000 2	2.5-5 7.8-7 20.000	4:48:55
esc32b	1582	1024; 524800	22 201 99.4	† 1.68000000 2	1.32000000 2	1.7-4 7.8-7 21.429	3:52:36
esc32c	1582	1024; 524800	30 479 140.2	† 6.42000000 2	6.16000000 2	6.5-4 2.1-7 4.050	11:12:30

Table 7: Results for the SDPNAL algorithm on the quadratic assignment problems. The entries under the column “%gap” are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - n_l \mid n_s; n_l$	it itsub pcg	best upper bound	lower bound \underline{v}	$R_P \mid R_D$ — %gap	time
esc32d	1582 1024; 524800	25 254 132.0	† 2.00000000 2	1.91000000 2	5.3-7 5.6-7 4.500	5:43:54
esc32e	1582 1024; 524800	15 46 58.2	2.00000000 0	2.00000000 0	2.2-7 1.1-7 0.000	31:11
esc32f	1582 1024; 524800	15 46 58.2	2.00000000 0	2.00000000 0	2.2-7 1.1-7 0.000	31:13
esc32g	1582 1024; 524800	15 38 50.7	6.00000000 0	6.00000000 0	1.7-7 3.2-7 0.000	23:25
esc32h	1582 1024; 524800	30 403 113.3	† 4.38000000 2	4.23000000 2	9.9-4 3.0-7 3.425	8:05:32
had12	232 144; 10440	23 457 93.8	1.65200000 3	1.65200000 3	2.2-4 1.4-7 0.000	5:17
had14	313 196; 19306	28 525 99.5	2.72400000 3	2.72400000 3	1.5-3 7.6-7 0.000	13:03
had16	406 256; 32896	27 525 98.7	3.72000000 3	3.72000000 3	1.4-3 1.1-7 0.000	22:37
had18	511 324; 52650	29 458 104.3	5.35800000 3	5.35800000 3	1.5-3 4.0-7 0.000	44:30
had20	628 400; 80200	32 568 96.7	6.92200000 3	6.92200000 3	3.8-3 2.6-7 0.000	1:24:06
kra30a	1393 900; 405450	27 313 68.0	8.89000000 4	8.64280000 4	4.5-4 6.5-7 2.781	4:08:17
kra30b	1393 900; 405450	28 289 68.9	9.14200000 4	8.74500000 4	3.1-4 3.6-7 4.343	3:50:35
kra32	1582 1024; 524800	31 307 78.6	8.89000000 4	8.52980000 4	4.6-4 6.0-7 4.052	6:43:41
lipa20a	628 400; 80200	18 243 70.1	3.68300000 3	3.68300000 3	5.5-7 2.9-9 0.000	24:29
lipa20b	628 400; 80200	14 116 56.2	2.70760000 4	2.70760000 4	1.7-5 6.5-7 0.000	10:10
lipa30a	1393 900; 405450	20 252 78.2	1.31780000 4	1.31780000 4	2.5-7 1.1-10 0.000	3:41:44
lipa30b	1393 900; 405450	18 83 80.8	1.51426000 5	1.51426000 5	6.9-7 3.3-8 0.000	1:23:34
lipa40a	2458 1600; 1280800	22 324 81.7	3.15380000 4	3.15380000 4	4.1-7 4.6-11 0.000	21:02:51
lipa40b	2458 1600; 1280800	19 121 76.6	4.76581000 5	4.76581000 5	3.9-6 1.3-8 0.000	7:24:25
nug12	232 144; 10440	22 266 69.6	5.78000000 2	5.68000000 2	1.2-4 3.6-7 1.730	2:27
nug14	313 196; 19306	24 337 62.3	1.01400000 3	1.00800000 3	3.1-4 8.0-7 0.592	5:50
nug15	358 225; 25425	27 318 62.6	1.15000000 3	1.13800000 3	3.0-4 7.5-7 1.043	7:32
nug16a	406 256; 32896	25 346 80.4	1.61000000 3	1.59700000 3	3.3-4 6.6-7 0.807	14:15
nug16b	406 256; 32896	28 315 64.5	1.24000000 3	1.21600000 3	2.8-4 4.2-7 1.935	10:20
nug17	457 289; 41905	26 302 60.6	1.73200000 3	1.70400000 3	2.0-4 7.7-7 1.617	12:38
nug18	511 324; 52650	26 287 59.5	1.93000000 3	1.89100000 3	2.2-4 3.5-7 2.021	15:39
nug20	628 400; 80200	26 318 65.1	2.57000000 3	2.50400000 3	1.5-4 5.2-7 2.568	31:49
nug21	691 441; 97461	27 331 62.5	2.43800000 3	2.37800000 3	1.9-4 6.6-7 2.461	40:22
nug22	757 484; 117370	28 369 86.0	3.59600000 3	3.52200000 3	3.1-4 5.9-7 2.058	1:21:58
nug24	898 576; 166176	29 348 63.7	3.48800000 3	3.39600000 3	1.8-4 3.6-7 2.638	1:33:59
nug25	973 625; 195625	27 335 60.2	3.74400000 3	3.62100000 3	1.8-4 3.0-7 3.285	1:41:49
nug27	1132 729; 266085	29 380 80.1	5.23400000 3	5.12400000 3	1.3-4 4.5-7 2.102	3:31:50
nug28	1216 784; 307720	26 329 80.5	5.16600000 3	5.02000000 3	2.4-4 6.3-7 2.826	3:36:38
nug30	1393 900; 405450	27 360 61.4	6.12400000 3	5.94400000 3	1.3-4 3.3-7 2.939	4:22:09
rou12	232 144; 10440	25 336 106.3	2.35528000 5	2.35434000 5	4.6-4 1.6-7 0.040	4:50
rou15	358 225; 25425	26 238 64.0	3.54210000 5	3.49544000 5	2.5-4 4.0-7 1.317	5:48
rou20	628 400; 80200	26 250 69.9	7.25522000 5	6.94397000 5	1.5-4 7.5-7 4.290	27:26
scr12	232 144; 10440	19 255 99.9	3.14100000 4	3.14080000 4	4.3-4 7.5-7 0.006	3:16
scr15	358 225; 25425	19 331 91.7	5.11400000 4	5.11400000 4	1.3-7 2.8-7 0.000	9:42
scr20	628 400; 80200	28 353 65.2	1.10030000 5	1.06472000 5	2.6-4 4.9-7 3.234	34:32
ste36a	1996 1296; 840456	26 318 93.8	9.52600000 3	9.23600000 3	1.7-4 4.1-7 3.044	15:09:10
ste36b	1996 1296; 840456	29 348 101.0	1.58520000 4	1.56030000 4	1.8-3 4.3-7 1.571	19:05:19
ste36c	1996 1296; 840456	28 360 105.3	8.23911000 6	8.11864500 6	6.3-4 4.0-7 1.462	19:56:15
tail2a	232 144; 10440	15 180 59.8	2.24416000 5	2.24416000 5	1.8-6 7.6-8 0.000	1:28
tail2b	232 144; 10440	29 596 112.2	3.94649250 7	3.94649080 7	3.7-4 9.3-9 0.000	7:40
tail5a	358 225; 25425	23 196 65.1	3.88214000 5	3.76608000 5	1.3-4 5.0-7 2.990	4:58
tail5b	358 225; 25425	29 409 102.2	5.17652680 7	5.17609220 7	1.5-3 7.0-7 0.008	16:04
tail7a	457 289; 41905	23 168 69.7	4.91812000 5	4.75893000 5	1.4-4 5.0-7 3.237	8:21
tai20a	628 400; 80200	27 220 73.3	7.03482000 5	6.70827000 5	1.9-4 4.2-7 4.642	25:32
tai20b	628 400; 80200	31 485 91.6	1.22455319 8	1.22452095 8	2.9-3 1.4-7 0.003	54:05
tai25a	973 625; 195625	27 194 77.3	1.16725600 6	1.01301000 6	8.0-7 7.9-7 13.214	1:17:54
tai25b	973 625; 195625	29 408 70.4	3.44355646 8	3.33685462 8	2.6-3 6.2-7 3.099	2:33:26
tai30a	1393 900; 405450	27 207 82.4	† 1.81814600 6	1.70578200 6	8.1-5 2.0-7 6.180	3:35:03
tai30b	1393 900; 405450	30 421 71.6	6.37117113 8	5.95926267 8	1.4-3 4.9-7 6.465	6:26:30
tai35a	1888 1225; 750925	28 221 81.0	2.42200200 6	2.21523000 6	1.5-4 5.0-7 8.537	8:09:44
tai35b	1888 1225; 750925	28 401 58.3	2.83315445 8	2.68328155 8	8.7-4 6.4-7 5.290	10:33:10
tai40a	2458 1600; 1280800	27 203 85.1	3.13937000 6	2.84184600 6	7.5-5 5.3-7 9.477	15:25:52
tai40b	2458 1600; 1280800	30 362 74.1	6.37250948 8	6.06880822 8	1.7-3 4.9-7 4.766	23:32:56
tho30	1393 900; 405450	27 315 61.1	1.49936000 5	1.43267000 5	2.4-4 7.3-7 4.448	3:41:26
tho40	2458 1600; 1280800	27 349 60.9	† 2.40516000 5	2.26161000 5	2.0-4 6.5-7 5.968	17:13:24

Table 8: Results for the SDPNAL algorithm on the BIQ problems. The entries under the column “%gap” are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - n_l \mid n_s; n_l$	it itsub pcg	best upper bound	lower bound \underline{v}	$R_P \mid R_D \mid \%$ gap	time
be100.1	101 101; 5151	27 488 70.5	-1.94120000 4	-2.00210000 4	8.6-7 5.7-7 3.137	1:45
be100.2	101 101; 5151	25 378 78.5	-1.72900000 4	-1.79880000 4	8.3-7 7.6-7 4.037	1:32
be100.3	101 101; 5151	27 432 96.3	-1.75650000 4	-1.82310000 4	3.7-7 7.0-7 3.792	2:08
be100.4	101 101; 5151	27 505 101.2	-1.91250000 4	-1.98410000 4	2.4-6 7.7-7 3.744	2:37
be100.5	101 101; 5151	25 355 78.5	-1.58680000 4	-1.68880000 4	8.6-7 8.8-7 6.428	1:28
be100.6	101 101; 5151	26 440 94.4	-1.73680000 4	-1.81480000 4	4.7-6 6.3-7 4.491	2:06
be100.7	101 101; 5151	27 219 92.3	-1.86290000 4	-1.97000000 4	1.3-7 4.9-7 5.749	1:01
be100.8	101 101; 5151	25 265 47.1	-1.86490000 4	-1.99460000 4	5.1-7 5.9-7 6.955	40
be100.9	101 101; 5151	28 526 72.6	-1.32940000 4	-1.42630000 4	6.4-7 5.3-7 7.289	2:01
be100.10	101 101; 5151	27 493 52.0	-1.53520000 4	-1.64080000 4	6.7-7 5.8-7 6.879	1:25
be120.3.1	121 121; 7381	26 384 112.4	-1.30670000 4	-1.38030000 4	5.9-6 4.9-7 5.633	2:57
be120.3.2	121 121; 7381	27 410 117.9	-1.30460000 4	-1.36260000 4	4.6-6 4.1-7 4.446	3:16
be120.3.3	121 121; 7381	26 210 89.2	-1.24170000 4	-1.29870000 4	2.9-7 4.4-7 4.582	1:19
be120.3.4	121 121; 7381	27 391 64.8	-1.38670000 4	-1.45110000 4	6.6-7 5.5-7 4.644	1:49
be120.3.5	121 121; 7381	27 489 99.0	-1.14030000 4	-1.19910000 4	7.8-6 2.9-7 5.157	3:21
be120.3.6	121 121; 7381	26 386 111.2	-1.29150000 4	-1.34320000 4	7.9-7 4.3-7 4.003	2:57
be120.3.7	121 121; 7381	27 412 111.9	-1.40680000 4	-1.45640000 4	1.0-4 5.1-7 3.526	3:16
be120.3.8	121 121; 7381	27 426 108.5	-1.47010000 4	-1.53030000 4	8.1-5 4.0-7 4.095	3:10
be120.3.9	121 121; 7381	27 418 89.2	-1.04580000 4	-1.12410000 4	7.5-5 6.3-7 7.487	2:39
be120.3.10	121 121; 7381	30 611 84.0	-1.22010000 4	-1.29300000 4	1.1-6 2.9-7 5.975	3:36
be120.8.1	121 121; 7381	26 384 71.5	-1.86910000 4	-2.01940000 4	4.3-7 6.6-7 8.041	1:53
be120.8.2	121 121; 7381	26 402 113.9	-1.88270000 4	-2.00740000 4	4.9-5 4.4-7 6.623	3:11
be120.8.3	121 121; 7381	27 267 96.2	-1.93020000 4	-2.05050000 4	5.1-7 5.1-7 6.233	1:48
be120.8.4	121 121; 7381	26 399 96.6	-2.07650000 4	-2.17790000 4	3.4-6 4.2-7 4.883	2:42
be120.8.5	121 121; 7381	27 452 120.1	-2.04170000 4	-2.13160000 4	8.3-7 5.3-7 4.403	3:48
be120.8.6	121 121; 7381	29 459 90.6	-1.84820000 4	-1.96770000 4	1.3-6 6.3-7 6.466	2:53
be120.8.7	121 121; 7381	28 457 52.5	-2.21940000 4	-2.37320000 4	2.0-7 4.9-7 6.930	1:46
be120.8.8	121 121; 7381	27 151 66.1	-1.95340000 4	-2.12040000 4	8.0-7 9.7-7 8.549	43
be120.8.9	121 121; 7381	27 301 60.4	-1.81950000 4	-1.92840000 4	2.3-7 4.1-7 5.985	1:17
be120.8.10	121 121; 7381	27 307 102.7	-1.90490000 4	-2.00240000 4	4.1-7 4.1-7 5.118	2:14
be150.3.1	151 151; 11476	27 538 84.7	-1.88890000 4	-1.98490000 4	1.3-5 5.3-7 5.082	4:57
be150.3.2	151 151; 11476	28 499 89.3	-1.78160000 4	-1.88640000 4	1.1-5 6.0-7 5.882	4:51
be150.3.3	151 151; 11476	26 514 101.8	-1.73140000 4	-1.80430000 4	1.8-6 7.6-7 4.210	5:37
be150.3.4	151 151; 11476	27 233 98.2	-1.98840000 4	-2.06520000 4	4.9-7 6.0-7 3.862	2:28
be150.3.5	151 151; 11476	28 507 90.4	-1.68170000 4	-1.77680000 4	1.6-5 4.1-7 5.655	4:53
be150.3.6	151 151; 11476	27 517 95.5	-1.67800000 4	-1.80500000 4	6.7-6 5.0-7 7.569	5:18
be150.3.7	151 151; 11476	27 470 73.5	-1.80010000 4	-1.91010000 4	6.8-7 9.1-7 6.111	3:42
be150.3.8	151 151; 11476	27 377 84.7	-1.83030000 4	-1.96980000 4	1.3-5 6.3-7 7.622	3:25
be150.3.9	151 151; 11476	26 292 58.0	-1.28380000 4	-1.41030000 4	3.8-7 8.8-7 9.854	1:52
be150.3.10	151 151; 11476	27 438 121.3	-1.79630000 4	-1.92300000 4	1.6-5 3.7-7 7.053	5:39
be150.8.1	151 151; 11476	28 661 78.0	-2.70890000 4	-2.91430000 4	9.4-7 6.6-7 7.582	5:36
be150.8.2	151 151; 11476	27 272 87.4	-2.67790000 4	-2.88210000 4	3.5-7 7.6-7 7.625	2:34
be150.8.3	151 151; 11476	27 435 77.9	-2.94380000 4	-3.10600000 4	3.5-7 8.3-7 5.510	3:37
be150.8.4	151 151; 11476	26 310 89.5	-2.69110000 4	-2.87290000 4	8.9-7 8.6-7 6.756	3:01
be150.8.5	151 151; 11476	27 500 113.9	-2.80170000 4	-2.94820000 4	9.4-7 3.7-7 5.229	6:06
be150.8.6	151 151; 11476	27 415 115.6	-2.92210000 4	-3.14370000 4	5.2-6 6.8-7 7.584	4:56
be150.8.7	151 151; 11476	27 446 127.2	-3.12090000 4	-3.32520000 4	2.8-5 2.5-7 6.546	6:06
be150.8.8	151 151; 11476	28 462 109.0	-2.97300000 4	-3.16000000 4	5.8-6 6.7-7 6.290	5:23
be150.8.9	151 151; 11476	28 370 110.7	-2.53880000 4	-2.71100000 4	2.6-7 5.3-7 6.783	4:20
be150.8.10	151 151; 11476	26 288 95.7	-2.83740000 4	-3.00480000 4	5.2-7 4.7-7 5.900	2:58
be200.3.1	201 201; 20301	29 615 89.7	-2.54530000 4	-2.77160000 4	5.6-7 5.0-7 8.891	10:29
be200.3.2	201 201; 20301	29 307 93.2	-2.50270000 4	-2.67600000 4	3.5-7 5.3-7 6.925	5:38
be200.3.3	201 201; 20301	29 507 120.8	-2.80230000 4	-2.94780000 4	5.6-5 5.7-7 5.192	12:09
be200.3.4	201 201; 20301	29 523 102.1	-2.74340000 4	-2.91060000 4	4.7-6 5.4-7 6.095	10:41
be200.3.5	201 201; 20301	28 466 116.2	-2.63550000 4	-2.80730000 4	1.4-6 5.5-7 6.519	10:38
be200.3.6	201 201; 20301	29 639 60.1	-2.61460000 4	-2.79280000 4	9.5-7 3.7-7 6.816	7:36
be200.3.7	201 201; 20301	29 534 93.9	-3.04830000 4	-3.16200000 4	1.1-6 5.8-7 3.730	9:43
be200.3.8	201 201; 20301	29 308 100.7	-2.73550000 4	-2.92440000 4	6.4-7 9.0-7 6.906	5:59
be200.3.9	201 201; 20301	28 482 87.1	-2.46830000 4	-2.64370000 4	3.2-5 3.7-7 7.106	8:28
be200.3.10	201 201; 20301	29 539 98.7	-2.38420000 4	-2.57600000 4	5.8-6 4.4-7 8.045	10:25
be200.8.1	201 201; 20301	28 489 97.5	-4.85340000 4	-5.08690000 4	3.7-5 6.2-7 4.811	9:41
be200.8.2	201 201; 20301	29 192 74.7	-4.08210000 4	-4.43360000 4	6.1-7 7.3-7 8.611	2:46
be200.8.3	201 201; 20301	28 476 116.1	-4.32070000 4	-4.62540000 4	5.8-7 9.2-7 7.052	10:53

Table 8: Results for the SDPNAL algorithm on the BIQ problems. The entries under the column “%gap” are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - n_l \mid n_s; n_l$	it itsub pcg	best upper bound	lower bound \underline{v}	$R_P \mid R_D \mid$ %gap	time
be200.8.4	201 201; 20301	29 267 93.3	-4.37570000 4	-4.66210000 4	8.4-7 7.2-7 6.545	4:55
be200.8.5	201 201; 20301	28 521 93.8	-4.14820000 4	-4.42710000 4	1.7-5 7.7-7 6.723	9:53
be200.8.6	201 201; 20301	28 556 87.4	-4.94920000 4	-5.12190000 4	2.7-5 4.4-7 3.489	9:48
be200.8.7	201 201; 20301	27 248 92.6	-4.68280000 4	-4.93530000 4	4.7-7 6.8-7 5.392	4:30
be200.8.8	201 201; 20301	28 314 94.3	-4.45020000 4	-4.76890000 4	7.0-7 7.7-7 7.161	5:49
be200.8.9	201 201; 20301	29 543 115.6	-4.32410000 4	-4.54950000 4	5.8-6 3.8-7 5.213	12:16
be200.8.10	201 201; 20301	29 485 107.9	-4.28320000 4	-4.57430000 4	6.9-6 5.5-7 6.796	10:15
be250.1	251 251; 31626	29 532 94.7	-2.40760000 4	-2.51190000 4	4.0-5 4.6-7 4.332	16:41
be250.2	251 251; 31626	28 519 113.6	-2.25400000 4	-2.36810000 4	3.1-5 6.4-7 5.062	18:51
be250.3	251 251; 31626	28 561 95.7	-2.29230000 4	-2.40000000 4	2.9-5 6.0-7 4.698	17:17
be250.4	251 251; 31626	30 577 112.2	-2.46490000 4	-2.57200000 4	4.8-5 4.7-7 4.345	20:42
be250.5	251 251; 31626	29 463 98.1	-2.10570000 4	-2.23740000 4	9.3-5 4.4-7 6.254	14:30
be250.6	251 251; 31626	30 567 93.6	-2.27350000 4	-2.40180000 4	2.0-5 4.3-7 5.643	16:39
be250.7	251 251; 31626	28 507 84.7	-2.40950000 4	-2.51190000 4	5.9-5 7.1-7 4.250	14:00
be250.8	251 251; 31626	28 620 84.1	-2.38010000 4	-2.50200000 4	1.6-5 7.5-7 5.122	16:50
be250.9	251 251; 31626	28 589 85.8	-2.00510000 4	-2.13970000 4	1.1-4 6.2-7 6.713	17:13
be250.10	251 251; 31626	29 591 88.9	-2.31590000 4	-2.43550000 4	3.4-5 4.8-7 5.164	16:48
bqp50-1	51 51; 1326	25 463 119.9	-2.09800000 3	-2.14300000 3	7.1-6 6.7-7 2.145	1:12
bqp50-2	51 51; 1326	26 387 72.7	-3.70200000 3	-3.74200000 3	2.3-5 5.8-7 1.080	39
bqp50-3	51 51; 1326	24 343 84.3	-4.62600000 3	-4.63700000 3	8.9-7 9.9-7 0.238	40
bqp50-4	51 51; 1326	28 486 106.6	-3.54400000 3	-3.58300000 3	2.5-4 5.2-7 1.100	1:08
bqp50-5	51 51; 1326	23 319 82.7	-4.01200000 3	-4.07700000 3	3.3-5 6.9-7 1.620	37
bqp50-6	51 51; 1326	20 338 95.8	-3.69300000 3	-3.71100000 3	1.1-5 4.4-7 0.487	44
bqp50-7	51 51; 1326	26 275 44.0	-4.52000000 3	-4.64900000 3	2.9-7 6.2-7 2.854	18
bqp50-8	51 51; 1326	26 289 73.3	-4.21600000 3	-4.26900000 3	8.5-7 6.5-7 1.257	29
bqp50-9	51 51; 1326	21 225 57.5	-3.78000000 3	-3.92100000 3	8.3-7 9.0-7 3.730	19
bqp50-10	51 51; 1326	27 191 52.2	-3.50700000 3	-3.62600000 3	4.4-7 6.5-7 3.393	14
bqp100-1	101 101; 5151	25 443 80.5	-7.97000000 3	-8.38000000 3	2.7-7 8.2-7 5.144	1:49
bqp100-2	101 101; 5151	23 374 97.1	-1.10360000 4	-1.14890000 4	5.4-4 4.8-7 4.105	1:53
bqp100-3	101 101; 5151	26 451 122.4	-1.27230000 4	-1.31530000 4	9.9-7 7.3-7 3.380	2:40
bqp100-4	101 101; 5151	26 420 129.4	-1.03680000 4	-1.07310000 4	3.5-5 6.5-7 3.501	2:42
bqp100-5	101 101; 5151	28 515 84.5	-9.08300000 3	-9.48700000 3	5.0-5 3.3-7 4.448	2:16
bqp100-6	101 101; 5151	28 524 88.4	-1.02100000 4	-1.08240000 4	6.7-7 4.6-7 6.014	2:22
bqp100-7	101 101; 5151	28 572 81.9	-1.01250000 4	-1.06890000 4	8.5-7 3.9-7 5.570	2:19
bqp100-8	101 101; 5151	26 440 107.4	-1.14350000 4	-1.17700000 4	2.4-5 7.8-7 2.930	2:25
bqp100-9	101 101; 5151	27 482 101.7	-1.14550000 4	-1.17330000 4	5.0-5 6.1-7 2.427	2:31
bqp100-10	101 101; 5151	25 415 110.4	-1.25650000 4	-1.29800000 4	3.9-5 5.7-7 3.303	2:18
bqp250-1	251 251; 31626	28 483 117.7	-4.56070000 4	-4.76630000 4	3.9-7 6.6-7 4.508	17:42
bqp250-2	251 251; 31626	30 554 93.5	-4.48100000 4	-4.72220000 4	4.4-5 4.1-7 5.383	16:23
bqp250-3	251 251; 31626	28 296 116.4	-4.90370000 4	-5.10770000 4	9.9-7 7.9-7 4.160	10:36
bqp250-4	251 251; 31626	29 607 88.9	-4.12740000 4	-4.33120000 4	1.8-5 4.5-7 4.938	17:37
bqp250-5	251 251; 31626	28 570 103.7	-4.79610000 4	-5.00040000 4	4.4-5 6.9-7 4.260	19:03
bqp250-6	251 251; 31626	28 477 113.1	-4.10140000 4	-4.36690000 4	1.9-5 7.7-7 6.473	17:11
bqp250-7	251 251; 31626	30 429 126.3	-4.67570000 4	-4.89220000 4	8.2-7 5.9-7 4.630	16:36
bqp250-8	251 251; 31626	28 748 73.5	-3.57260000 4	-3.87800000 4	6.3-7 8.8-7 8.548	17:34
bqp250-9	251 251; 31626	29 453 117.0	-4.89160000 4	-5.14970000 4	3.7-7 3.9-7 5.276	16:12
bqp250-10	251 251; 31626	28 691 76.7	-4.04420000 4	-4.30140000 4	8.1-7 5.1-7 6.360	16:29
bqp500-1	501 501; 125751	30 357 117.8	-1.16586000 5	-1.25965000 5	2.9-7 5.5-7 8.045	1:00:59
bqp500-2	501 501; 125751	30 637 94.7	-1.28223000 5	-1.36012000 5	7.9-5 7.2-7 6.075	1:31:17
bqp500-3	501 501; 125751	30 363 118.9	-1.30812000 5	-1.38454000 5	4.4-7 4.0-7 5.842	1:01:47
bqp500-4	501 501; 125751	30 663 79.9	-1.30097000 5	-1.39329000 5	3.7-6 4.3-7 7.096	1:16:35
bqp500-5	501 501; 125751	30 539 119.6	-1.25487000 5	-1.34092000 5	4.5-5 2.5-7 6.857	1:36:43
bqp500-6	501 501; 125751	30 485 124.4	-1.21772000 5	-1.30765000 5	4.1-7 5.1-7 7.385	1:28:49
bqp500-7	501 501; 125751	31 648 87.7	-1.22201000 5	-1.31492000 5	8.1-5 5.5-7 7.603	1:25:26
bqp500-8	501 501; 125751	31 412 126.3	-1.23559000 5	-1.33490000 5	8.6-7 4.5-7 8.037	1:14:37
bqp500-9	501 501; 125751	30 612 92.7	-1.20798000 5	-1.30289000 5	9.5-5 7.3-7 7.857	1:24:40
bqp500-10	501 501; 125751	30 454 130.5	-1.30619000 5	-1.38535000 5	7.0-7 6.4-7 6.060	1:24:23
gka1a	51 51; 1326	20 309 57.9	-3.41400000 3	-3.53700000 3	7.7-7 6.0-7 3.603	26
gka2a	61 61; 1891	24 281 57.3	-6.06300000 3	-6.17100000 3	1.4-7 4.9-7 1.781	27
gka3a	71 71; 2556	25 398 68.3	-6.03700000 3	-6.38600000 3	6.6-7 9.5-7 5.781	51
gka4a	81 81; 3321	25 567 106.2	-8.59800000 3	-8.88100000 3	4.2-6 6.3-7 3.291	2:09
gka5a	51 51; 1326	24 284 55.9	-5.73700000 3	-5.89700000 3	7.7-7 7.8-7 2.789	23
gka6a	31 31; 496	25 175 46.8	-3.98000000 3	-4.10300000 3	4.4-7 7.2-7 3.090	10
gka7a	31 31; 496	26 145 47.2	-4.54100000 3	-4.63800000 3	3.9-7 5.5-7 2.136	08

Table 8: Results for the SDPNAL algorithm on the BIQ problems. The entries under the column “%gap” are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - n_l \mid n_s; n_l$	it itsub pcg	best upper bound	lower bound \underline{v}	$R_P \mid R_D \mid \%$ gap	time
gka8a	101 101; 5151	27 543 94.1	-1.11090000 4	-1.11970000 4	3.8-5 6.6-7 0.792	2:39
gka1b	21 21; 231	7 42 23.8	-1.33000000 2	-1.33000000 2	9.8-7 5.4-7 0.000	02
gka2b	31 31; 496	15 241 101.1	-1.21000000 2	-1.21000000 2	8.8-5 7.7-7 0.000	25
gka3b	41 41; 861	12 85 25.6	-1.18000000 2	-1.18000000 2	2.9-7 2.4-8 0.000	04
gka4b	51 51; 1326	14 88 25.9	-1.29000000 2	-1.29000000 2	2.8-7 1.2-9 0.000	04
gka5b	61 61; 1891	12 86 26.0	-1.50000000 2	-1.50000000 2	7.6-8 1.7-8 0.000	05
gka6b	71 71; 2556	13 123 34.6	-1.46000000 2	-1.46000000 2	3.3-7 8.1-10 0.000	10
gka7b	81 81; 3321	19 193 33.8	-1.60000000 2	-1.60000000 2	8.9-7 5.3-7 0.000	16
gka8b	91 91; 4186	15 198 47.0	-1.45000000 2	-1.45000000 2	5.9-7 2.3-9 0.000	28
gka9b	101 101; 5151	18 252 50.9	-1.37000000 2	-1.37000000 2	3.7-7 1.2-10 0.000	44
gka10b	126 126; 8001	17 298 94.5	-1.54000000 2	-1.55000000 2	1.6-4 3.4-7 0.649	2:14
gka1c	41 41; 861	24 371 103.7	-5.05800000 3	-5.11300000 3	1.5-5 3.8-7 1.087	45
gka2c	51 51; 1326	27 358 72.0	-6.21300000 3	-6.32000000 3	2.5-7 5.6-7 1.722	35
gka3c	61 61; 1891	25 305 60.0	-6.66500000 3	-6.81300000 3	3.1-7 9.6-7 2.221	31
gka4c	71 71; 2556	27 476 114.7	-7.39800000 3	-7.56500000 3	9.7-7 4.5-7 2.257	1:38
gka5c	81 81; 3321	28 304 94.6	-7.36200000 3	-7.57600000 3	1.2-6 3.9-7 2.907	1:03
gka6c	91 91; 4186	27 427 108.4	-5.82400000 3	-5.96100000 3	3.0-5 6.2-7 2.352	1:58
gka7c	101 101; 5151	26 396 82.2	-7.22500000 3	-7.31600000 3	1.9-4 6.0-7 1.260	1:43
gka1d	101 101; 5151	27 439 96.5	-6.33300000 3	-6.52800000 3	1.1-5 2.5-7 3.079	2:09
gka2d	101 101; 5151	27 523 84.1	-6.57900000 3	-6.99000000 3	1.7-6 6.9-7 6.247	2:15
gka3d	101 101; 5151	26 467 96.9	-9.26100000 3	-9.73400000 3	1.4-5 4.8-7 5.107	2:21
gka4d	101 101; 5151	28 375 104.9	-1.07270000 4	-1.12780000 4	1.4-6 4.7-7 5.137	1:56
gka5d	101 101; 5151	26 422 91.5	-1.16260000 4	-1.23980000 4	2.3-6 6.9-7 6.640	1:57
gka6d	101 101; 5151	27 338 102.4	-1.42070000 4	-1.49290000 4	1.9-6 5.2-7 5.082	1:42
gka7d	101 101; 5151	27 177 75.3	-1.44760000 4	-1.53750000 4	6.2-7 5.8-7 6.210	40
gka8d	101 101; 5151	26 271 118.4	-1.63520000 4	-1.70050000 4	2.0-7 7.1-7 3.993	1:35
gka9d	101 101; 5151	26 351 63.9	-1.56560000 4	-1.65330000 4	7.2-7 6.1-7 5.602	1:10
gka10d	101 101; 5151	26 213 78.5	-1.91020000 4	-2.01080000 4	2.0-7 7.2-7 5.266	52
gka1e	201 201; 20301	29 530 97.3	-1.64640000 4	-1.70690000 4	5.2-5 7.9-7 3.675	10:36
gka2e	201 201; 20301	29 367 103.4	-2.33950000 4	-2.49170000 4	4.7-7 4.3-7 6.506	7:23
gka3e	201 201; 20301	30 559 91.5	-2.52430000 4	-2.68980000 4	1.6-5 2.9-7 6.556	10:22
gka4e	201 201; 20301	29 512 113.0	-3.55940000 4	-3.72250000 4	1.2-5 4.2-7 4.582	11:25
gka5e	201 201; 20301	28 510 95.2	-3.51540000 4	-3.80020000 4	3.9-5 5.1-7 8.101	9:46
gka1f	501 501; 125751	30 563 102.8	†-6.11940000 4	-6.55590000 4	9.9-5 5.2-7 7.133	1:28:54
gka2f	501 501; 125751	30 624 93.6	†-1.00161000 5	-1.07932000 5	6.6-5 5.7-7 7.759	1:28:11
gka3f	501 501; 125751	30 523 120.4	†-1.38035000 5	-1.50152000 5	2.8-5 6.7-7 8.778	1:31:34
gka4f	501 501; 125751	32 571 128.8	†-1.72771000 5	-1.87089000 5	8.7-6 4.0-7 8.287	1:44:43
gka5f	501 501; 125751	31 665 90.5	†-1.90507000 5	-2.06916000 5	6.6-6 7.1-7 8.613	1:25:48

8 Conclusion

In this paper, we introduced a Newton-CG augmented Lagrangian algorithm for solving semidefinite programming problems (D) and (P) and analyzed its convergence and rate of convergence. Our convergence analysis is based on classical results of proximal point methods [32, 33] along with recent developments on perturbation analysis of the problems under consideration. Extensive numerical experiments conducted on a variety of large scale SDPs demonstrated that our algorithm is very efficient. This opens up a way to attack problems in which a fast solver for large scale SDPs is crucial, for example, in applications within a branch-and-bound algorithm for solving hard combinatorial problems such as the quadratic assignment problems.

Acknowledgements

We thank Brian Borchers for providing us the SDP data for the maximum stable set problems collected by N. Sloane [35]. We also thank the two referees and the associate editor for their helpful comments and suggestions for improving the quality of this paper.

References

- [1] F. Alizadeh, J. P. A. Haeberly, and O. L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Mathematical Programming, 77 (1997), pp. 111-128.
- [2] V. I. Arnold, *On matrices depending on parameters*, Russian Mathematical Surveys, 26 (1971), pp. 29-43.
- [3] Z. J. Bai, D. Chu and D. F. Sun, *A dual optimization approach to inverse quadratic eigenvalue problems with partial eigenstructure*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2531-2561.
- [4] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimizaiton Problems*, Springer-Verlag, New York (2000).
- [5] S. Burer, and D. Vandenbussche, *Solving Lift-and-Project relaxations of binary integer programs*, SIAM Journal on Optimization, 16 (2006), pp. 726-750.
- [6] S. Burer, R. Monterio, and Y. Zhang, *A computational study of a gradient-based log-barrier algorithm for a class of large scale SDPs*, Math. Program., 95 (2003), pp. 359-379.
- [7] S. Butenko, P. Pardalos, I. Sergienko, V. Shylo, and P. Stetsyuk, *Finding maximum independent sets in graphs arising from coding theory*, Symposium on Applied Computing, Proceedings of the 2002 ACM symposium on Applied computing, 2002, pp. 542-546.
- [8] Z. X. Chan and D. F. Sun, *Constraint nondegeneracy, strong regularity and nonsingularity in semidefinite programming*, SIAM Journal on optimization, 19 (2008), pp. 370-396.
- [9] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York (1983).
- [10] A. Eisenblätter, M. Grötschel, and A. M. C. A. Koster, *Frequency planning and ramification of coloring*, Discuss. Math. Graph Theory, 22 (2002), pp. 51-88.
- [11] F. Facchinei, *Minimization of SC^1 functions and the Maratos effect*, Operation Research Letters, 17 (1995), 131 - 137.
- [12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore and London, 3rd edition, 1996.

- [13] C. Helmberg, *Numerical Evaluation of SBmethod*, Mathematical Programming, 95 (2003), pp. 381–406.
- [14] C. Jansson, *Termination and verification for ill-posed semidefinite programs*, Technical Report, Informatik III, TU Hamburg-Harburg, Hamburg, June 2005.
- [15] D. Johnson, G. Pataki, and Farid Alizadeh, *The Seventh DIMACS Implementation Challenge: Semidefinite and Related Optimization Problems*, Rutgers University, <http://dimacs.rutgers.edu/Challenges/>, (2000).
- [16] M. Kočvara and M. Stingl, *PENNON - a code for convex nonlinear and semidefinite programming*, Optimization Methods and Software, 18 (2003), pp. 317–333.
- [17] M. Kočvara and M. Stingl, *On the solution of large-scale SDP problems by the modified barrier method using iterative solvers*, Mathematical Programming, 109 (2007), pp. 413–444.
- [18] B. Kummer, *Newton's method for non-differentiable functions*, In: J. Guddat et al.(eds.), Advances in Mathematical Optimization, Akademie-Verlag, Berlin, (1988), pp. 114–125.
- [19] J. Malick, J. Povh, F. Rendl and A. Wiecele, *Regularization methods for semidefinite programming*, preprint, 2007.
- [20] H. Mittelmann, *An independent benchmarking of SDP and SOCP solvers*, Mathematical Programming, 95 (2003), pp. 407–430.
- [21] J. S. Pang and L. Qi, *A globally convergent Newton method of convex SC^1 minimization problems*, Journal of Optimization Theory and Application, 85 (1995), pp. 633–648.
- [22] J. S. Pang, D. F. Sun and J. Sun, *Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems*, Mathematics of Operations Research, 28 (2003), pp. 272–284.
- [23] J. Povh, and F. Rendl, *Copositive and semidefinite relaxations of the quadratic assignment problem*, Technical Report, Klagenfurt University, July 2006.
- [24] J. Povh, F. Rendl, and A. Wiecele, *A boundary point method to solve semidefinite programs*, Computing, 78 (2006), pp. 277–286.
- [25] H. D. Qi and D. F. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 360–385.
- [26] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Mathematical Programming, 58 (1993), pp. 353–367.

- [27] S. M. Robinson, *Local structure of feasible sets in nonlinear programming, Part II: nondegeneracy*, Mathematical Programming Study, 22 (1984), pp. 217-230.
- [28] S. M. Robinson, *Local structure of feasible sets in nonlinear programming, Part III: stability and sensitivity*, Mathematical Programming Study, 30 (1987), pp. 45-66.
- [29] R. T. Rockafellar, *Convex Analysis*, Princenton University Press, New Jersey, 1970.
- [30] R. T. Rockafellar, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Mathematical Programming, 5 (1973), pp. 354-373.
- [31] R. T. Rockafellar, *Conjugate Duality and Optimization*, Regional Conference Series in Applied Math., 16 (1974), SIAM Publication.
- [32] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877-898.
- [33] R. T. Rockafellar, *Augmented Lagrangains and applications of the proximal point algorithm in convex programming*, Mathematics of Operation Research, 1 (1976), pp. 97-116.
- [34] R. T. Rockafellar and R. J. -B. Wets, *Variational Analysis*, Springer, New York, 1998.
- [35] N. Sloane, *Challenge Problems: Independent Sets in Graphs*, <http://www.research.att.com/~njas/doc/graphs.html>.
- [36] D. F. Sun and J. Sun, *Semismooth matrix valued functions*, Mathematics of Operations Research, 27 (2002), pp. 150-169.
- [37] D. F. Sun, J. Sun and L. W. Zhang, *The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming*, Mathematical Programming, 114 (2008), pp. 349-391.
- [38] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia (1997).
- [39] K. C. Toh, and M. Kojima, *Solving some large scale semidefinite programs via the conjugate residual method*, SIAM Journal on Optimization, 12 (2002), pp. 669-691.
- [40] K. C. Toh, *Solving large scale semidefinite programs via an iterative solver on the augmented systems*, SIAM Journal on Optimization, 14 (2004), pp. 670-698.
- [41] K.C. Toh, *An inexact primal-dual path-following algorithm for convex quadratic SDP*, Mathematical Programming, 112 (2007), pp. 221-254.
- [42] M. Trick, V. Chvatal, W. Cook, D. Johnson, C. McGeoch, and R. Tarjan, *The Second DIMACS Implementation Challenge: NP Hard Problems: Maximum Clique, Graph Coloring, and Satisfiability*, Rutgers University, <http://dimacs.rutgers.edu/Challenges/> (1992).

- [43] A. Wiegele, *Biq Mac library – a collection of Max-Cut and quadratic 0-1 programming instances of medium size*, Preprint, 2007.
- [44] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory I and II*, Contributions to Nonlinear Functional Analysis (E. H. Zarantonello, ed.), Academic Press, New York (1971), pp. 237-424.