SDPNAL+: A Majorized Semismooth Newton-CG Augmented Lagrangian Method for Semidefinite Programming with Nonnegative Constraints

Liuqin Yang*, Defeng Sun†, Kim-Chuan Toh‡

May 28, 2014; Revised on Mar 10, 2015

Abstract

In this paper, we present a majorized semismooth Newton-CG augmented Lagrangian method, called SDPNAL+, for semidefinite programming (SDP) with partial or full nonnegative constraints on the matrix variable. SDPNAL+ is a much enhanced version of SDPNAL introduced by Zhao, Sun and Toh [SIAM Journal on Optimization, 20 (2010), pp. 1737–1765] for solving generic SDPs. SDPNAL works very efficiently for nondegenerate SDPs but may encounter numerical difficulty for degenerate ones. Here we tackle this numerical difficulty by employing a majorized semismooth Newton-CG augmented Lagrangian method coupled with a convergent 3-block alternating direction method of multipliers introduced recently by Sun, Toh and Yang [arXiv preprint arXiv:1404.5378, (2014)]. Numerical results for various large scale SDPs with or without nonnegative constraints show that the proposed method is not only fast but also robust in obtaining accurate solutions. It outperforms, by a significant margin, two other competitive publicly available first order methods based codes: (1) an alternating direction method of multipliers based solver called SDPAD by Wen, Goldfarb and Yin [Mathematical Programming Computation, 2 (2010), pp. 203–230] and (2) a two-easy-block-decomposition hybrid proximal extragradient method called 2EBD-HPE by Monteiro, Ortiz and Svaiter

*Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (`yangliuqin@u.nus.edu`).

†Department of Mathematics and Risk Management Institute, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (`matsundf@nus.edu.sg`). This author's research was supported in part by the Academic Research Fund under Grant R-146-000-149-112.

‡Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (`mattohkc@nus.edu.sg`). Research supported in part by the Ministry of Education, Singapore, Academic Research Fund under Grant R-146-000-194-112.

[Mathematical Programming Computation, (2013), pp. 1–48]. In contrast to these two codes, we are able to solve all the 95 difficult SDP problems arising from the relaxations of quadratic assignment problems tested in SDPNAL to an accuracy of 10^{-6} efficiently, while SDPAD and 2EBD-HPE successfully solve 30 and 16 problems, respectively. In addition, SDPNAL+ appears to be the only viable method currently available to solve large scale SDPs arising from rank-1 tensor approximation problems constructed by Nie and Wang [arXiv preprint arXiv:1308.6562, (2013)]. The largest rank-1 tensor approximation problem we solved (in about 14.5 hours) is `nonsym(21,4)`, in which its resulting SDP problem has matrix dimension $n = 9,261$ and the number of equality constraints $m = 12,326,390$.

Keywords: semidefinite programming, degeneracy, augmented Lagrangian, semismooth Newton-CG method

AMS subject classifications: 90C06, 90C22, 90C25, 65F10

1 Introduction

Let $\mathcal{K}$ be a pointed closed convex cone whose interior $int(\mathcal{K}) \neq \emptyset$ and $\mathcal{P}$ be a polyhedral convex cone in a finite-dimensional Euclidean space $\mathcal{X}$ such that $\mathcal{K} \cap \mathcal{P}$ is non-empty. For any cone $\mathcal{C} \subseteq \mathcal{X}$, we denote the dual cone of $\mathcal{C}$ by $\mathcal{C}^*$. For any closed convex set $\mathcal{C} \subseteq \mathcal{X}$, we denote the metric projection of $\mathcal{X}$ onto $\mathcal{C}$ by $\Pi_{\mathcal{C}}(\cdot)$ and the tangent cone of $\mathcal{C}$ at $X \in \mathcal{C}$ by $\mathcal{T}_{\mathcal{C}}(X)$, respectively. We will make extensive use of the Moreau decomposition theorem in [11], which states that $X = \Pi_{\mathcal{C}}(X) - \Pi_{\mathcal{C}^*}(-X)$ for any $X \in \mathcal{X}$ and any closed convex cone $\mathcal{C} \subseteq \mathcal{X}$. Let $\mathcal{S}^n$ be the space of $n \times n$ real symmetric matrices and $\mathcal{S}^n_+$ be the cone of positive semidefinite matrices in $\mathcal{S}^n$. In this paper, we focus on the case where $\mathcal{X} = \mathcal{S}^n$, $\mathcal{K} = \mathcal{K}^* = \mathcal{S}^n_+$. We are particularly interested in the case where $\mathcal{P} = \mathcal{S}^n_{\geq 0}$, the cone of $n \times n$ real symmetric matrices whose elements are all nonnegative, though the algorithm which we will design later is also applicable to other cases. For any matrix $X \in \mathcal{S}^n$, we use $X \succ 0$ to indicate that X is a real symmetric positive definite matrix.

Consider the semidefinite programming (SDP) with an additional polyhedral cone constraint, which we name as SDP+:

$$(\text{P}) \quad \max\Big\{\langle -C,\, X\rangle \mid \mathcal{A}(X) = b,\ X \in \mathcal{K},\ X \in \mathcal{P}\Big\}, \tag{1}$$

where $b \in \Re^m$ and $C \in \mathcal{X}$ are given data, $\mathcal{A} : \mathcal{X} \to \Re^m$ is a given linear map whose adjoint is denoted as $\mathcal{A}^*$. Note that $\mathcal{P} = \mathcal{X}$ is allowed in (1), in which case there is no additional polyhedral cone constraint imposed on X. We assume that the matrix $\mathcal{A}\mathcal{A}^*$ is invertible, i.e., $\mathcal{A}$ is surjective. The dual of (P) is given by

$$(\text{D}) \quad \min\Big\{\langle -b,\, y\rangle \mid \mathcal{A}^*(y) + S + Z = C,\ S \in \mathcal{K}^*,\ Z \in \mathcal{P}^*\Big\}. \tag{2}$$

The optimality conditions (KKT conditions) for (P) and (D) can be written as follows:

$$\begin{cases} \mathcal{A}(X) - b = 0, \quad \mathcal{A}^*(y) + S + Z - C = 0, \\ \langle X,\, S\rangle = 0,\ X \in \mathcal{K},\ S \in \mathcal{K}^*,\ \langle X,\, Z\rangle = 0,\ X \in \mathcal{P},\ Z \in \mathcal{P}^*. \end{cases} \tag{3}$$

In order for the KKT conditions (3) to have solutions, throughout this paper we make the following blanket assumption.

Assumption 1. (a) *For problem* (P)*, there exists a feasible solution* $X_0 \in \mathcal{S}^n_+$ *such that*

$$\mathcal{A}(X_0) = b,\ X_0 \succ \mathbf{0},\ X_0 \in \mathcal{P}. \tag{4}$$

(b) *For problem* (D)*, there exists a feasible solution* $(y_0, S_0, Z_0) \in \Re^m \times \mathcal{S}^n_+ \times \mathcal{S}^n$ *such that*

$$\mathcal{A}^*(y_0) + S_0 + Z_0 = C,\ S_0 \succ 0,\ Z_0 \in \mathcal{P}^*. \tag{5}$$

It is known from convex analysis (e.g, [3, Corollary 5.3.6]) that under Assumption 1, the strong duality for (P) and (D) holds and the KKT conditions (3) have solutions.

For a given $\sigma > 0$, define the augmented Lagrangian function for the dual problem (D) as follows:

$$\begin{aligned} L_\sigma(y, S, Z; X) &= \langle -b,\, y\rangle + \langle X,\, \mathcal{A}^*y + S + Z - C\rangle + \frac{\sigma}{2}\|\mathcal{A}^*y + S + Z - C\|^2 \\ &= \langle -b,\, y\rangle + \frac{\sigma}{2}\|\mathcal{A}^*y + S + Z + \sigma^{-1}X - C\|^2 - \frac{1}{2\sigma}\|X\|^2, \end{aligned} \tag{6}$$

where $X \in \mathcal{X}, y \in \Re^m, S \in \mathcal{K}^*, Z \in \mathcal{P}^*$. We can consider the following inexact augmented Lagrangian method to solve (D). Specifically, given $\sigma_0 > 0$, $(y^0, S^0, Z^0) \in \Re^m \times \mathcal{K}^* \times \mathcal{P}^*$, perform the following steps at the $(k+1)$-th iteration:

$$\begin{cases} (y^{k+1}, S^{k+1}, Z^{k+1}) \approx \arg\min\{L_{\sigma_k}(y, S, Z; X^k) \mid y \in \Re^m,\ S \in \mathcal{K}^*,\ Z \in \mathcal{P}^*\}, & \text{(7a)} \\ X^{k+1} = X^k + \sigma_k(\mathcal{A}^*y^{k+1} + S^{k+1} + Z^{k+1} - C), & \text{(7b)} \end{cases}$$

where $\sigma_k \in (0, +\infty)$, $k = 0, 1, \ldots$. For a general discussion on the convergence of the augmented Lagrangian method for solving convex optimization problems and beyond, see [18, 19].

Note that problem (P) can be reformulated as a standard SDP in the primal form by replacing the constraint $X \in \mathcal{P}$ with two constraints $X - Y = 0$ and $Y \in \mathcal{P}$. In [26], SDPNAL introduced by Zhao, Sun and Toh is applied to solve such a reformulated problem. It works quite well for nondegenerate SDPs, especially those without the constraint $X \in \mathcal{P}$. However, many of the tested SDPs (with the constraint $X \in \mathcal{P}$) in [26] are degenerate and SDPNAL is unable to solve those problems efficiently. Motivated by our desire to overcome the aforementioned difficulty in solving degenerate SDPs and to improve the performance of SDPNAL, we present here a majorized semismooth Newton-CG augmented Lagrangian method by directly working on (P) instead of its reformulated problem. We call this new method SDPNAL+ since it is a much enhanced version of SDPNAL and it is designed for SDP+ problems (P). We should emphasize that the technique of majorization plays a pivotal role in solving the inner problem (7a), which leads to an alternating minimization between the blocks (y, S) and Z.

The remaining parts of this paper are organized as follows. In Section 2, we introduce a majorized semismooth Newton-CG method for solving the inner minimization problems of the augmented Lagrangian method and analyze the convergence for solving these inner problems. Section 3 presents the SDPNAL+ dual approach. Section 4 is on numerical issues. There we report numerical results for a variety of SDP+ and SDP problems. We make an extensive numerical comparison with two other competitive first order methods based codes: (1) an alternating direction method of multiplier (ADMM) based solver called SDPAD by Wen et al. [25] and (2) a two-easy-block-decomposition hybrid proximal extragradient method called 2EBD-HPE by Monteiro et al. [10]. Numerical results show that SDPNAL+ is both fast and robust in achieving accurate solutions.

For the first time, we are able to solve all the 95 difficult SDP problems arising from the relaxations of quadratic assignment problems (QAPs) tested in SDPNAL to an accuracy of 10^{-6} efficiently, while SDPAD and 2EBD-HPE successfully solve 30 and 16 problems, respectively. In addition, SDPNAL+ appears to be the only viable method currently available to solve large scale SDPs arising from rank-1 tensor approximation problems constructed by Nie and Wang [12]. The largest rank-1 tensor approximation problem solved is `nonsym(21,4)`, in which its resulting SDP problem has matrix dimension $n = 9,261$ and the number of equality constraints $m = 12,326,390$. Finally, in order to demonstrate the power of the proposed majorized semismooth Newton-CG procedure, we list the numerical results by only running the convergent ADMM with 3-block constraints (ADMM+ in short) introduced by Sun et al. [22]. As one may observe, although ADMM+ outperforms both SDPAD and 2EBD-HPE, it can still encounter numerical difficulty in solving some hard problems such as those arising from QAPs to high accuracy. The superior numerical performance of SDPNAL+ over solvers based purely on first order methods such as SDPAD and 2EBD-HPE clearly shows the necessity of exploiting second order methods such as the semismooth Newton-CG method in order to solve hard SDP+ and SDP problems to high accuracy efficiently. While there has been a recent focus on using first order methods such as those based on ADMM or accelerated proximal gradient methods to solve structured convex optimization problems arising from machine learning and statistics, the extensive numerical results we obtained here for matrix conic programming problems serve to demonstrate that second order methods with good local convergence property are essential, if used wisely, for mitigating the inherent slow local convergence of first order methods, especially on difficult problems.

2 A Majorized Semismooth Newton-CG Method for Inner Problems

Let $\sigma > 0$ and $\widetilde{X} \in \mathcal{S}^n$ be fixed. In this section we will present a majorized semismooth Newton-CG method for solving the following inner problems involved in the augmented Lagrangian method (7a):

$$\min \Big\{ \phi(y, S, Z) := L_\sigma(y, S, Z; \widetilde{X}) \mid y \in \Re^m,\ S \in \mathcal{K}^*,\ Z \in \mathcal{P}^* \Big\}. \tag{8}$$

Note that problem (8) is the dual of the following problem:

$$\max \Big\{ \langle -C, X \rangle - \frac{1}{2\sigma} \|X - \widetilde{X}\|^2 \mid \mathcal{A}(X) = b,\ X \in \mathcal{K},\ X \in \mathcal{P} \Big\}. \tag{9}$$

Since the objective function in (9) is strongly concave, (9) has a unique optimal solution. In order for its dual problem (8) to have a bounded solution set, we need the following generalized Slater condition.

Assumption 2. *There exists a positive definite matrix $X_0 \in \mathcal{S}^n_+ \cap relint(\mathcal{P})$ such that*

$$\mathcal{A}(\mathcal{T}_{\mathcal{P}}(X_0)) = \Re^m,\ X_0 \succ 0, \tag{10}$$

where $relint(\mathcal{P})$ denotes the relative interior of $\mathcal{P}$.

Note that in (10), $\mathcal{T}_{\mathcal{P}}(X_0)$ is actually a linear subspace of $\mathcal{S}^n$ as X_0 is assumed to be in the relative interior part of the polyhedral cone $\mathcal{P}$. When $\mathcal{P} = \mathcal{S}^n$, Assumption 2 is equivalent to saying that

$$\begin{cases} \mathcal{A} : \mathcal{S}^n \to \Re^m \text{ is onto,} \\ \exists\, X_0 \in \mathcal{S}^n_+ \text{ such that } \mathcal{A}(X_0) = b,\ X_0 \succ \mathbf{0}. \end{cases} \tag{11}$$

From [17, Theorems 17 and 18], we have the following useful lemma.

Lemma 2.1. *Suppose that Assumption 2 holds. Then for any $\alpha \in \Re$, the level set $\mathcal{L}_\alpha := \{(y, S, Z) \in \Re^m \times \mathcal{K}^* \times \mathcal{P}^* \mid \phi(y, S, Z) \le \alpha\}$ is a closed and bounded convex set.*

2.1 A Majorized Semismooth Newton-CG Method

Consider $(\tilde{y}, \widetilde{S}, \widetilde{Z}) \in \arg\min\{\phi(y, S, Z) \mid y \in \Re^m, S \in \mathcal{K}^*, Z \in \mathcal{P}^*\}$. Let

$$\widehat{C} = C - \sigma^{-1}\widetilde{X}.$$

Then we must have $\widetilde{Z} = \Pi_{\mathcal{P}^*}(\widehat{C} - \mathcal{A}^*\tilde{y} - \widetilde{S})$. Therefore, problem (8) is equivalent to the following optimization problem:

$$\min \Big\{ \varPhi(y, S) := \langle -b,\, y \rangle + \frac{\sigma}{2} \|\Pi_{\mathcal{P}}(\mathcal{A}^*y + S - \widehat{C})\|^2 \mid y \in \Re^m,\ S \in \mathcal{K}^* \Big\}. \tag{12}$$

In order to introduce our majorized semismooth Newton-CG method for solving (12), we need to majorize the second part of the objective function in (12) by a convex, but not necessarily strongly convex, quadratic function. Specifically, for given $(y^l, S^l) \in \Re^m \times \mathcal{K}^*$ and $l \geq 0$, since

$$\begin{aligned}\|\Pi_{\mathcal{P}}(\mathcal{A}^*y + S - \widehat{C})\|^2 &\leq \|\Pi_{\mathcal{P}}(\mathcal{A}^*y^l + S^l - \widehat{C})\|^2 + \|\mathcal{A}^*y + S - \mathcal{A}^*y^l - S^l\|^2 \\ &\quad + 2\langle \Pi_{\mathcal{P}}(\mathcal{A}^*y^l + S^l - \widehat{C}),\, \mathcal{A}^*y + S - \mathcal{A}^*y^l - S^l\rangle \\ &= \|\mathcal{A}^*y + S + Z^l - \widehat{C}\|^2,\end{aligned}$$

where $Z^l := \Pi_{\mathcal{P}^*}(\widehat{C} - \mathcal{A}^*y^l - S^l)$, we know that for $(y, S) \in \Re^m \times \mathcal{S}^n$,

$$\begin{aligned}\Phi(y, S) &\leq \langle -b,\, y\rangle + \frac{\sigma}{2}\|\mathcal{A}^*y + S + Z^l - \widehat{C}\|^2 \\ &= \Psi_l(y, S) := \langle -b,\, y\rangle + \frac{\sigma}{2}\|\mathcal{A}^*y + S + \sigma^{-1}\widetilde{X} - \widetilde{C}^l\|^2, \qquad (13)\end{aligned}$$

where $\widetilde{C}^l := C - Z^l$. Thus Ψ_l is a majorization function of Φ at (y^l, S^l) because $\Psi_l(y^l, S^l) = \Phi(y^l, S^l)$ and $\Psi_l(y, S) \geq \Phi(y, S)$ $\forall (y, S) \in \Re^m \times \mathcal{S}^n$. In order to find an optimal solution for problem (12), for $l = 0, 1, \ldots$, we solve the following problem

$$\min \Big\{ \Psi_l(y, S) \mid y \in \Re^m,\, S \in \mathcal{K}^* \Big\}. \qquad (14)$$

Observe that if $(\tilde{y}, \widetilde{S}) \in \arg\min\{\Psi_l(y, S) \mid y \in \Re^m,\, S \in \mathcal{K}^*\}$, then we must have $\widetilde{S} = \Pi_{\mathcal{K}^*}(\widetilde{C}^l - \mathcal{A}^*\tilde{y} - \sigma^{-1}\widetilde{X})$. Thus we can compute y^{l+1} and S^{l+1} simultaneously as follows:

$$\begin{cases} y^{l+1} \in \arg\min \Big\{ \varphi_l(y) := \langle -b,\, y\rangle + \frac{\sigma}{2}\|\Pi_{\mathcal{K}}(\mathcal{A}^*y + \sigma^{-1}\widetilde{X} - \widetilde{C}^l)\|^2 \mid y \in \Re^m \Big\}, & (15a) \\ S^{l+1} = \Pi_{\mathcal{K}^*}(\widetilde{C}^l - \mathcal{A}^*y^{l+1} - \sigma^{-1}\widetilde{X}). & (15b)\end{cases}$$

Note that we can only solve problem (15a) inexactly by an iterative method. Here we will introduce a semismooth Newton-CG (SNCG) method for solving (15a). Specifically, for fixed $\widetilde{X}, \widetilde{C} \in \mathcal{S}^n$, we need to consider the following problem of the form

$$\min \Big\{ \varphi(y) := \langle -b,\, y\rangle + \frac{\sigma}{2}\|\Pi_{\mathcal{K}}(\mathcal{A}^*y + \sigma^{-1}\widetilde{X} - \widetilde{C})\|^2 \mid y \in \Re^m \Big\}. \qquad (16)$$

The objective function in (16) is continuously differentiable and solving (16) is equivalent to solving the following nonsmooth equation:

$$\nabla\varphi(y) = \mathcal{A}\Pi_{\mathcal{K}}(\widetilde{X} + \sigma(\mathcal{A}^*y - \widetilde{C})) - b = 0, \quad y \in \Re^m. \qquad (17)$$

Since $\Pi_{\mathcal{K}}(\cdot)$ is strongly semismooth [21], we can design a SNCG method as in [26] to solve (17), and expect fast superlinear or even quadratic convergence.

Let $\tilde{y} \in \Re^m$ be fixed. Consider the following eigenvalue decomposition:

$$\widetilde{X} + \sigma(\mathcal{A}^*\tilde{y} - \widetilde{C}) = Q\,\Gamma_{\tilde{y}}\,Q^{\mathrm{T}}, \tag{18}$$

where $Q \in \mathcal{R}^{n\times n}$ is an orthogonal matrix whose columns are eigenvectors, and $\Gamma_{\tilde{y}}$ is the diagonal matrix of eigenvalues with the diagonal elements arranged in the nonincreasing order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Define the following index sets

$$\alpha := \{i \mid \lambda_i > 0\}, \quad \bar{\alpha} := \{i \mid \lambda_i \leq 0\}.$$

We define the operator $W^0_{\tilde{y}} : \mathcal{S}^n \to \mathcal{S}^n$ by

$$W^0_{\tilde{y}}(H) := Q(\Sigma \circ (Q^{\mathrm{T}}HQ))Q^{\mathrm{T}}, \quad H \in \mathcal{S}^n, \tag{19}$$

where "$\circ$" denotes the Hadamard product of two matrices and

$$\Sigma = \begin{bmatrix} E_{\alpha\alpha} & \nu_{\alpha\bar{\alpha}} \\ \nu^{\mathrm{T}}_{\alpha\bar{\alpha}} & 0 \end{bmatrix}, \quad \nu_{ij} := \frac{\lambda_i}{\lambda_i - \lambda_j}, \; i \in \alpha, j \in \bar{\alpha}, \tag{20}$$

where $E_{\alpha\alpha} \in \mathcal{S}^{|\alpha|}$ is the matrix of ones. Define $V^0_{\tilde{y}} : \Re^m \to \mathcal{S}^n$ by

$$V^0_{\tilde{y}}d := \sigma\mathcal{A}[\,Q(\Sigma \circ (\,Q^{\mathrm{T}}(\mathcal{A}^*d)\,Q))\,Q^{\mathrm{T}}], \quad d \in \Re^m. \tag{21}$$

For any $y \in \Re^m$, define

$$\hat{\partial}^2\varphi(y) := \sigma\mathcal{A}\,\partial\Pi_{\mathcal{K}}(\widetilde{X} + \sigma(\mathcal{A}^*y - \widetilde{C}))\mathcal{A}^*, \tag{22}$$

where $\partial\Pi_{\mathcal{K}}(\widetilde{X} + \sigma(\mathcal{A}^*y - \widetilde{C}))$ is the Clarke subdifferential of $\Pi_{\mathcal{K}}(\cdot)$ at $\widetilde{X} + \sigma(\mathcal{A}^*y - \widetilde{C})$. Note that from [9], we know that

$$\hat{\partial}^2\varphi(\tilde{y})h = \partial^2\varphi(\tilde{y})h \quad \forall\, h \in \Re^m, \tag{23}$$

where $\partial^2\varphi(\tilde{y})$ denotes the generalized Hessian of φ at $\tilde{y}$, i.e., the Clarke subdifferential of $\nabla\varphi$ at $\tilde{y}$. However, note that (23) does not mean that $\hat{\partial}^2\varphi(\tilde{y}) = \partial^2\varphi(\tilde{y})$. Actually, it is unclear to us whether the latter holds. Fortunately, from Pang, Sun, and Sun [14, Lemma 11] we know that

$$W^0_{\tilde{y}} \in \partial\Pi_{\mathcal{K}}(\widetilde{X} + \sigma(\mathcal{A}^*\tilde{y} - \widetilde{C}))$$

and thus $V^0_{\tilde{y}} = \sigma\mathcal{A}W^0_{\tilde{y}}\mathcal{A}^* \in \hat{\partial}^2\varphi(\tilde{y})$.

Now we will introduce the SNCG algorithm for solving (16). Choose $y^0 \in \Re^m$. Then the algorithm can be stated as follows.

Algorithm SNCG: A Semismooth Newton-CG Algorithm (SNCG$(y^0, \widetilde{X}, \sigma)$).

Given $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, $\tau_1, \tau_2 \in (0, 1)$, and $\delta \in (0, 1)$. Perform the jth iteration as follows.

Step 1. Given a maximum number of CG iterations $N_j > 0$, compute

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

Apply the conjugate gradient (CG) algorithm ($CG(\eta_j, N_j)$), to find an approximation solution d^j to

$$(V_j + \varepsilon_j I)\, d = -\nabla\varphi(y^j), \tag{24}$$

where $V_j \in \hat{\partial}^2\varphi(y^j)$ is defined as in (21) and $\varepsilon_j := \tau_1 \min\{\tau_2, \|\nabla\varphi(y^j)\|\}$.

Step 2. Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\varphi(y^j + \delta^m d^j) \leq \varphi(y^j) + \mu\delta^m \langle \nabla\varphi(y^j), d^j \rangle. \tag{25}$$

Step 3. Set $y^{j+1} = y^j + \alpha_j\, d^j$.

The convergence results for the above SNCG algorithm are stated in Theorems 2.2 and 2.3 below. We shall omit the proofs as they can be proved in the same fashion as in [26, Theorems 3.4 and 3.5].

Theorem 2.2. *Suppose that Assumption* 2 *holds. Then Algorithm SNCG generates a bounded sequence* $\{y^j\}$ *and any accumulation point* $\hat{y}$ *of* $\{y^j\}$ *is an optimal solution to problem* (16).

Theorem 2.3. *Suppose that Assumption* 2 *holds. Let* $\hat{y}$ *be an accumulation point of the infinite sequence* $\{y^j\}$ *generated by Algorithm SNCG for solving the problem* (16)*. Suppose that at each step* $j \geq 0$*, when the CG algorithm terminates, the tolerance* η_j *is achieved (e.g., when* $N_j = m + 1$*), i.e.,*

$$\|\nabla\varphi(y^j) + (V_j + \varepsilon_j I)\, d^j\| \leq \eta_j. \tag{26}$$

Assume that the constraint nondegenerate condition

$$\mathcal{A}\, \mathrm{lin}(\mathcal{T}_{\mathcal{K}}(\widehat{W})) = \Re^m \tag{27}$$

holds at $\widehat{W} := \Pi_{\mathcal{K}}(\widetilde{X} + \sigma(\mathcal{A}^*\hat{y} - \widetilde{C}))$*, where* $\mathrm{lin}(\mathcal{T}_{\mathcal{K}}(\widehat{W}))$ *denotes the lineality space of* $\mathcal{T}_{\mathcal{K}}(\widehat{W})$*. Then the whole sequence* $\{y^j\}$ *converges to* $\hat{y}$ *and*

$$\|y^{j+1} - \hat{y}\| = O(\|y^j - \hat{y}\|^{1+\tau}). \tag{28}$$

For notational convenience, for any $y \in \Re^m$ and $S \in \mathcal{S}^n$, let $Y := (y, S)$. Define the linear map $\mathcal{M} : \Re^m \times \mathcal{S}^n \to \mathcal{S}^n$ by

$$\mathcal{M}Y := \mathcal{A}^* y + S \quad \forall\, Y = (y, S) \in \Re^m \times \mathcal{S}^n. \tag{29}$$

Let $B := (b, 0) \in \Re^m \times \mathcal{S}^n$ and $\mathcal{C} := \Re^m \times \mathcal{K}^*$. Then problem (12) is equivalent to

$$\min \left\{ \Phi(Y) := \langle -B,\, Y\rangle + \frac{\sigma}{2}\|\Pi_{\mathcal{P}}(\mathcal{M}Y + \sigma^{-1}\widetilde{X} - C)\|^2 \;\mid\; Y \in \mathcal{C} \right\} \tag{30}$$

and the function $\Psi_l(y, S)$ in (13) can be rewritten as

$$\Psi_l(Y) = \Psi_l(y, S) = -\langle B,\, Y\rangle + \frac{\sigma}{2}\|\mathcal{M}Y + \sigma^{-1}\widetilde{X} - C + Z^l\|^2.$$

Furthermore, $\widehat{Y} = (\hat{y}, \widehat{S})$ is an optimal solution of

$$\min \left\{ \Psi_l(Y) \;\mid\; Y \in \mathcal{C} \right\} \tag{31}$$

if and only if $\hat{y}$ is an optimal solution of problem (15a) and $\widehat{S} = \Pi_{\mathcal{K}^*}(\widetilde{C}^l - \mathcal{A}^*\hat{y} - \sigma^{-1}\widetilde{X})$. Given $\xi_1 \in (0, 1)$ and $\xi_2 \in (0, \infty)$, we will use the following stopping criteria for terminating Algorithm SNCG:

(A1) $\Psi_l(Y^{l+1}) \le \Psi_l(Y^l) - \frac{\xi_1}{2}\left|\langle \nabla\Psi_l(Y^l),\, Y^{l+1} - Y^l\rangle\right|$,

(A2) $\|Y^{l+1} - \Pi_{\mathcal{C}}(Y^{l+1} - \nabla\Psi_l(Y^{l+1}))\| \le \xi_2(\Psi_l(Y^l) - \Psi_l(Y^{l+1}))^{\frac{1}{2}}$.

We can now state our majorized semismooth Newton-CG method for solving (12) as follows:

Algorithm MSNCG: A Majorized Semismooth Newton-CG Algorithm (MSNCG($y^0, S^0, Z^0, \widetilde{X}, \sigma$)).

Given $\xi_1 \in (0, 1)$, $\xi_2 \in (0, +\infty)$. Perform the lth iteration as follows.

Step 1. Starting with y^l as the initial point, apply Algorithm SNCG to minimize $\varphi_l(\cdot)$ to find $y^{l+1} = \text{SNCG}(y^l, \widetilde{X}, \sigma)$ and $S^{l+1} := \Pi_{\mathcal{K}^*}(C - \mathcal{A}^* y^{l+1} - Z^l - \sigma^{-1}\widetilde{X})$ satisfying (A1) and (A2).

Step 2. Compute $Z^{l+1} := \Pi_{\mathcal{P}^*}(C - \mathcal{A}^* y^{l+1} - S^{l+1} - \sigma^{-1}\widetilde{X})$.

Next, we establish the convergence of Algorithm MSNCG.

Lemma 2.4. *Suppose that Assumption* 2 *holds. Then for Algorithm MSNCG,* (A1) *and* (A2) *are achievable.*

Proof. If $Y^l - \Pi_{\mathcal{C}}(Y^l - \nabla\Psi_l(Y^l)) = 0$, then one can take $Y^{l+1} = Y^l$ to satisfy (A1) and (A2). Next, we assume that $Y^l - \Pi_{\mathcal{C}}(Y^l - \nabla\Psi_l(Y^l)) \neq 0$. Then Y^l is not an optimal solution of problem (31). Let $\widehat{Y}$ be an arbitrary optimal solution of problem (31). Then $\widehat{Y} = \Pi_{\mathcal{C}}(\widehat{Y} - \nabla\Psi_l(\widehat{Y}))$. So $\langle Y^l - \widehat{Y},\, (\widehat{Y} - \nabla\Psi_l(\widehat{Y})) - \widehat{Y}\rangle \leq 0$, i.e., $\langle \nabla\Psi_l(\widehat{Y}),\, Y^l - \widehat{Y}\rangle \geq 0$, which implies

$$\langle \nabla\Psi_l(Y^l),\, Y^l - \widehat{Y}\rangle \geq \langle \nabla\Psi_l(Y^l) - \nabla\Psi_l(\widehat{Y}),\, Y^l - \widehat{Y}\rangle = \sigma\|\mathcal{M}(\widehat{Y} - Y^l)\|^2. \tag{32}$$

Since

$$\Psi_l(Y^l) > \Psi_l(\widehat{Y}) = \Psi_l(Y^l) + \langle \nabla\Psi_l(Y^l),\, \widehat{Y} - Y^l\rangle + \frac{\sigma}{2}\|\mathcal{M}(\widehat{Y} - Y^l)\|^2, \tag{33}$$

we obtain that $\langle \nabla\Psi_l(Y^l),\, \widehat{Y} - Y^l\rangle + \frac{\sigma}{2}\|\mathcal{M}(\widehat{Y} - Y^l)\|^2 < 0$. This implies

$$\langle \nabla\Psi_l(Y^l),\, \widehat{Y} - Y^l\rangle < 0. \tag{34}$$

Then by using (32), (33), (34) and the fact that $\widehat{Y} = \Pi_{\mathcal{C}}(\widehat{Y} - \nabla\Psi_l(\widehat{Y}))$, we know that for given $\xi_1 \in (0,1)$ and $\xi_2 \in (0,\infty)$, there exists $\delta > 0$ such that

$$\begin{aligned} &\Psi_l(Y) \leq \Psi_l(Y^l) - \frac{\xi_1}{2}\left|\langle \nabla\Psi_l(Y^l),\, Y - Y^l\rangle\right|, \\ &\|Y - \Pi_{\mathcal{C}}(Y - \nabla\Psi_l(Y))\| \leq \xi_2(\Psi_l(Y^l) - \Psi_l(Y))^{\frac{1}{2}}, \end{aligned}$$

for all $Y \in \mathcal{C}$ satisfying $\|Y - \widehat{Y}\| < \delta$. Let $\{\tilde{y}^j\}_{j=0}^{+\infty}$ be the sequence generated by SNCG$(\tilde{y}^0, \widetilde{X}, \sigma)$ with $\tilde{y}^0 := y^l$. For each $j \geq 0$, let $\widetilde{S}^j := \Pi_{\mathcal{K}^*}(C - \mathcal{A}^*\tilde{y}^j - Z^l - \sigma^{-1}\widetilde{X})$. Then by Theorem 2.2, we know that $\{(\tilde{y}^j, \widetilde{S}^j)\}$ is a bounded sequence and any accumulation point of $\{(\tilde{y}^j, \widetilde{S}^j)\}$, say $\widehat{Y} := (\hat{y}, \widehat{S})$, is an optimal solution to problem (31). Thus there exists a sufficiently large j such that $Y^{l+1} := (\tilde{y}^j, \widetilde{S}^j)$ satisfying (A1) and (A2). □

Theorem 2.5. *Suppose that Assumption 2 holds. Let Algorithm MSNCG be executed with stopping criteria* (A1) *and* (A2)*. Then it generates a bounded sequence* $\{(y^l, S^l, Z^l)\}$ *and any accumulation point* $(\hat{y}, \widehat{S})$ *of* $\{(y^l, S^l)\}$ *is an optimal solution to problem* (12) *and hence* $(\hat{y}, \widehat{S}, \widehat{Z})$ *is an optimal solution to problem* (8)*, where* $\widehat{Z} := \Pi_{\mathcal{P}^*}(C - \mathcal{A}^*\hat{y} - \widehat{S} - \sigma^{-1}\widetilde{X})$*. Furthermore,* $\|Z^{l+1} - Z^l\| \to 0$ *as* $l \to +\infty$*.*

Proof. By (A1), we have $\Phi(Y^{l+1}) \leq \Psi_l(Y^{l+1}) \leq \Psi_l(Y^l) - \frac{\xi_1}{2}\left|\langle \nabla\Psi_l(Y^l),\, Y^{l+1} - Y^l\rangle\right| = \Phi(Y^l) - \frac{\xi_1}{2}\left|\langle \nabla\Psi_l(Y^l),\, Y^{l+1} - Y^l\rangle\right|$. Hence, the sequence $\{\Phi(Y^l)\}$ is nonincreasing.

By Lemma 2.1, we know that the level set $\mathcal{L} := \{Y \in \mathcal{C} \mid \Phi(Y) \leq \Phi(Y^0)\}$ is a closed and bounded convex set. Then the sequence $\{Y^l\}$ is bounded and so is the sequence $\{Z^l\}$. Let $\widehat{Y}$ be an accumulation point of $\{Y^l\}$. Then $\Phi(Y^l) \to \Phi(\widehat{Y})$ and $\langle \nabla\Psi_l(Y^l),\, Y^{l+1} - Y^l\rangle \to 0$ as $l \to \infty$. Furthermore, $\Psi_l(Y^l) - \Psi_l(Y^{l+1}) \to 0$ as $l \to \infty$.

By noting that

$$\Psi_l(Y^{l+1}) = \Psi_l(Y^l) + \langle \nabla\Psi_l(Y^l),\, Y^{l+1} - Y^l\rangle + \frac{\sigma}{2}\|\mathcal{M}(Y^{l+1} - Y^l)\|^2, \tag{35}$$

we get from (A1) that

$$\langle \nabla \Psi_l(Y^l),\, Y^{l+1} - Y^l \rangle + \frac{\sigma}{2}\|\mathcal{M}(Y^{l+1} - Y^l)\|^2 \leq -\frac{\xi_1}{2}\left|\langle \nabla \Psi_l(Y^l),\, Y^{l+1} - Y^l \rangle\right| \leq 0. \quad (36)$$

Since $\langle \nabla \Psi_l(Y^l),\, Y^{l+1} - Y^l \rangle \to 0$ as $l \to \infty$, we obtain from (36) that

$$\|\mathcal{M}(Y^{l+1} - Y^l)\| \to 0 \quad \text{as} \quad l \to \infty. \quad (37)$$

For any $l \geq 0$, denote $\Delta_l := Y^l - \Pi_{\mathcal{C}}(Y^l - \nabla \Phi(Y^l))$. Then we have

$$\begin{aligned} \|\Delta_{l+1}\| &\leq \|Y^{l+1} - \Pi_{\mathcal{C}}(Y^{l+1} - \nabla \Psi_l(Y^{l+1}))\| \\ &\quad + \|\Pi_{\mathcal{C}}(Y^{l+1} - \nabla \Psi_l(Y^{l+1})) - \Pi_{\mathcal{C}}(Y^{l+1} - \nabla \Phi(Y^{l+1}))\| \\ &\leq \xi_2(\Psi_l(Y^l) - \Psi_l(Y^{l+1}))^{\frac{1}{2}} + \|\nabla \Psi_l(Y^{l+1}) - \nabla \Phi(Y^{l+1})\|. \end{aligned}$$

By direct computations, we have for $l \geq 1$,

$$\begin{aligned} &\|\nabla \Psi_l(Y^{l+1}) - \nabla \Phi(Y^{l+1})\| \\ &= \|\sigma \mathcal{M}^*(\mathcal{M}Y^{l+1} + \sigma^{-1}\widetilde{X} - C + Z^l) - \sigma \mathcal{M}^*(\Pi_{\mathcal{P}}(\mathcal{M}Y^{l+1} + \sigma^{-1}\widetilde{X} - C))\| \\ &= \|\sigma \mathcal{M}^*(Z^l - Z^{l+1})\| \leq \sigma \|\mathcal{M}^*\| \|\mathcal{M}(Y^{l+1} - Y^l)\|, \end{aligned}$$

where we have used the fact that

$$\begin{aligned} \|Z^{l+1} - Z^l\| &= \|\Pi_{\mathcal{P}^*}(C - \sigma^{-1}\widetilde{X} - \mathcal{M}Y^{l+1}) - \Pi_{\mathcal{P}^*}(C - \sigma^{-1}\widetilde{X} - \mathcal{M}Y^l)\| \\ &\leq \|\mathcal{M}(Y^{l+1} - Y^l)\|. \end{aligned} \quad (38)$$

Thus, by (37) and the fact that $(\Psi_l(Y^l) - \Psi_l(Y^{l+1}))^{\frac{1}{2}} \to 0$ as $l \to \infty$, we derive that $\|\Delta_{l+1}\| \to 0$ as $l \to \infty$. Since $\widehat{Y}$ is an accumulation point of $\{Y^l\}$, we obtain that $\widehat{Y} - \Pi_{\mathcal{C}}(\widehat{Y} - \nabla \Phi(\widehat{Y})) = 0$. By the convexity of Φ, $\widehat{Y}$ is an optimal solution of problem (30).

Finally, by using (37) and (38), we know that $\|Z^{l+1} - Z^l\| \to 0$ as $l \to \infty$. □

3 A Majorized Semismooth Newton-CG Augmented Lagrangian Method

For any $k \geq 0$ and $(y, S, Z) \in \Re^m \times \mathcal{S}^n \times \mathcal{S}^n$, denote

$$\phi_k(y, S, Z) := L_{\sigma_k}(y, S, Z; X^k), \quad (39)$$

$$\hat{\phi}_k(y, S, Z) := \begin{cases} L_{\sigma_k}(y, S, Z; X^k) & \text{if } (y, S, Z) \in \Omega := \Re^m \times \mathcal{K}^* \times \mathcal{P}^*, \\ +\infty & \text{otherwise.} \end{cases} \quad (40)$$

Since the inner problems in (8) are solved inexactly, we will use the following standard stopping criteria considered in [19, 18] to terminate Algorithm MSNCG:

(B1) $\hat{\phi}_k(y^{k+1}, S^{k+1}, Z^{k+1}) - \inf \hat{\phi}_k \leq \epsilon_k^2$, $\epsilon_k \geq 0$, $\sum_{k=0}^{\infty} \epsilon_k < \infty$.

(B2) $\hat{\phi}_k(y^{k+1}, S^{k+1}, Z^{k+1}) - \inf \hat{\phi}_k \leq (\delta_k^2/2\sigma_k)\|X^{k+1} - X^k\|$, $\delta_k \geq 0$, $\sum_{k=0}^{\infty} \delta_k < \infty$.

(B3) $\mathrm{dist}(0, \partial\hat{\phi}_k(y^{k+1}, S^{k+1}, Z^{k+1})) \leq (\delta_k^{'}/\sigma_k)\|X^{k+1} - X^k\|$, $0 \leq \delta_k^{'} \to 0$.

Just like SDPNAL, each iteration of the MSNCG algorithm can be quite expensive. Thus it is crucial for us to find a reasonably good initial point to warm start Algorithm SDPNAL+. We can certainly do so by solving the inner problem (8) by using any gradient descent type method. However, for this purpose we find that ADMM+ introduced by Sun, Toh and Yang [22] is usually more efficient than other choices. Now we can present our SDPNAL+ algorithm as follows.

Algorithm SDPNAL+: A Majorized Semismooth Newton-CG Augmented Lagrangian Algorithm (SDPNAL+$(y^0, S^0, Z^0, X^0, \sigma_0)$)

Stage 1. Use ADMM+ to generate an initial point

$$(y^0, S^0, Z^0, X^0, \sigma_0) \leftarrow \text{ADMM+}(y^0, S^0, Z^0, X^0, \sigma_0).$$

Stage 2. For $k = 0, \ldots,$ perform the kth iteration as follows:

(a) Using (y^k, S^k, Z^k) as the initial point, apply Algorithm MSNCG to minimize $\hat{\phi}_k(\cdot)$ to find $(y^{k+1}, S^{k+1}, Z^{k+1}) = \text{MSNCG}(y^k, S^k, Z^k, X^k, \sigma_k)$ and $X^{k+1} = X^k + \sigma_k(\mathcal{A}^* y^{k+1} + S^{k+1} + Z^{k+1} - C)$ satisfying (B1), (B2) or (B3).

(b) Update $\sigma_{k+1} = \rho\sigma_k$ for some $\rho > 1$ or $\sigma_{k+1} = \sigma_k$.

Remark 3.1. *(a) As mentioned in the introduction, if* (P) *is reformulated as a standard SDP and Algorithm SDPNAL+ is applied to this reformulated form, then SDPNAL+ reduces to SDPNAL proposed in [26].*

(b) Note that numerically it is difficult to compute $\mathrm{dist}(0, \partial\hat{\phi}_k(W^{k+1}))$ *in the criterion (B3) for terminating Algorithm MSNCG directly, where* $W^{k+1} = (y^{k+1}, S^{k+1}, Z^{k+1})$*. Fortunately, we have from [18] that*

$$\begin{aligned}
(\mathrm{dist}(0, \partial\hat{\phi}_k(W^{k+1})))^2 &= \|\Pi_{\mathcal{T}_\Omega(W^{k+1})}(-\nabla\phi_k(W^{k+1}))\|^2 \\
&= \|\Pi_{\mathcal{T}_{\Re^m}(y^{k+1})}(-\nabla_y\phi_k(W^{k+1}))\|^2 + \|\Pi_{\mathcal{T}_{\mathcal{K}^*}(S^{k+1})}(-\nabla_S\phi_k(W^{k+1}))\|^2 \\
&\quad + \|\Pi_{\mathcal{T}_{\mathcal{P}^*}(Z^{k+1})}(-\nabla_Z\phi_k(W^{k+1}))\|^2 \\
&= \|\mathcal{A}(X^k + \sigma_k R_D^{k+1}) - b\|^2 + \|\Pi_{\mathcal{T}_{\mathcal{K}^*}(S^{k+1})}(-\sigma_k R_D^{k+1} - X^k)\|^2 \\
&\quad + \|\Pi_{\mathcal{T}_{\mathcal{P}^*}(Z^{k+1})}(-\sigma_k R_D^{k+1} - X^k)\|^2
\end{aligned}$$

where $R_D^{k+1} = \mathcal{A}^* y^{k+1} + S^{k+1} + Z^{k+1} - C$. *Observe that the first term in the last equality can readily be evaluated. The third term can also be computed easily since* $\mathcal{P}$ *is a polyhedral cone. The second term is again computable as we shall show next. Let* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ *be the eigenvalues of* S^{k+1} *being arranged in a nonincreasing order. Denote* $\alpha := \{i \mid \lambda_i > 0,\ i = 1, \ldots, n\}$ *and* $\bar{\alpha} := \{1, \ldots, n\} \backslash \alpha$. *Then there exists an orthogonal matrix* $P \in \Re^{n \times n}$ *such that*

$$S^{k+1} = P \begin{bmatrix} \Lambda_\alpha & 0 \\ 0 & 0 \end{bmatrix} P^T,$$

where Λ_α *is the diagonal matrix whose diagonal entries are* λ_i *for* $i \in \alpha$. *Write* $P = [P_\alpha\ P_{\bar{\alpha}}]$ *with* $P_\alpha \in \Re^{n \times |\alpha|}$ *and* $P_{\bar{\alpha}} \in \Re^{n \times |\bar{\alpha}|}$. *From [1], we know that the tangent cone of* $\mathcal{S}_+^n$ *at* $S^{k+1} \in \mathcal{S}_+^n$ *can be characterized as* $\mathcal{T}_{\mathcal{S}_+^n}(S^{k+1}) = \{B \in \mathcal{S}^n \mid P_{\bar{\alpha}}^T B P_{\bar{\alpha}} \succeq 0\}$. *Let* $H = P^T(-\sigma_k R_D^{k+1} - X^k)P$. *Then*

$$\Pi_{\mathcal{T}_{\mathcal{K}^*}(S^{k+1})}(-\sigma_k R_D^{k+1} - X^k) = P \begin{pmatrix} H_{\alpha\alpha} & H_{\alpha\bar{\alpha}} \\ H_{\bar{\alpha}\alpha} & \Pi_{\mathcal{S}_+^{|\bar{\alpha}|}}(H_{\bar{\alpha}\bar{\alpha}}) \end{pmatrix} P^T.$$

We can obtain similar theorems on the convergence of SDPNAL+ as SDPNAL ([26, Theorems 4.1 and 4.2]). The global convergence of Algorithm SDPNAL+ follows from Rockafellar [19, Theorem 1] and [18, Theorem 4] without much difficulty.

Theorem 3.1. *Suppose that Assumption* 2 *holds. Let Algorithm SDPNAL+ be executed with stopping criterion* (B1). *If there exists* $(y_0, S_0, Z_0) \in \Re^m \times \mathcal{S}_+^n \times \mathcal{S}^n$ *such that*

$$\mathcal{A}^*(y_0) + S_0 + Z_0 = C,\ S_0 \succ 0,\ Z_0 \in relint(\mathcal{P}^*), \tag{41}$$

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

If $\{X^k\}$ *is bounded, then the sequence* $\{(y^k, S^k, Z^k)\}$ *is also bounded, and all of its accumulation points of the sequence* $\{(y^k, S^k, Z^k)\}$ *are optimal solutions to* (D).

Next we state the local linear convergence of Algorithm SDPNAL+.

Theorem 3.2. *Suppose that Assumption* 2 *holds. Let Algorithm SDPNAL+ be executed with stopping criteria* (B1) *and* (B2). *Assume that* (D) *satisfies condition* (41). *If the second order sufficient conditions (in the sense of the conditions in* [2, Theorem 3.137]*) holds at* $\overline{X}$, *where* $\overline{X}$ *is an optimal solution to* (P), *then the generated sequence* $\{X^k\} \subset \mathcal{P}$ *is bounded and* $\{X^k\}$ *converges to the unique optimal solution* $\overline{X}$ *with* $\max(\mathrm{P}) = \min(\mathrm{D})$, *and*

$$\|X^{k+1} - \overline{X}\| \leq \theta_\infty \|X^k - \overline{X}\| \quad \forall\ k \text{ sufficiently large},$$

for some $\theta_\infty \in [0, 1)$ *with the property that* $\theta_\infty \ll 1$ *if* $\sigma_k \to \sigma_\infty$ *for any sufficiently large* σ_∞. *The conclusions of Theorem* 3.1 *about* $\{(y^k, S^k, Z^k)\}$ *are also valid.*

Proof. The conclusions of Theorem 3.2 follow from the results in [19, Theorem 2] and [18, Theorem 5 and Proposition 3] combined with [2, Theorem 3.137]. □

4 Numerical Experiments

4.1 SDP+ and SDP Problem Sets

In our numerical experiments, we test the following SDP+ and SDP problem sets.

(i) SDP+ problems coming from the relaxation of a binary integer nonconvex quadratic (BIQ) programming:

$$\min\left\{\frac{1}{2}x^TQx + \langle c,\, x\rangle \mid x \in \{0,1\}^{n-1}\right\}. \tag{42}$$

This problem has been shown in [4] that under some mild assumptions, it can equivalently be reformulated as the following completely positive programming (CPP) problem:

$$\min\left\{\frac{1}{2}\langle Q,\, X_0\rangle + \langle c,\, x\rangle \mid \operatorname{diag}(X_0) = x, X = [X_0, x; x^T, 1] \in \mathcal{C}_{pp}^n\right\}, \tag{43}$$

where $\mathcal{C}_{pp}^n$ denotes the n-dimensional completely positive cone. It is well known that even though $\mathcal{C}_{pp}^n$ is convex, it is computationally intractable. To solve the CPP problem, one would typically relax $\mathcal{C}_{pp}^n$ to $\mathcal{S}_+^n \cap \mathcal{S}_{\geq 0}^n$, and the relaxed problem has the form (P):

$$\begin{array}{ll} \min & \frac{1}{2}\langle Q,\, X_0\rangle + \langle c,\, x\rangle \\ \text{s.t.} & \operatorname{diag}(X_0) - x = 0,\ \alpha = 1, \quad X = \begin{bmatrix} X_0 & x \\ x^T & \alpha \end{bmatrix} \in \mathcal{S}_+^n, \quad X \in \mathcal{P}, \end{array} \tag{44}$$

where the polyhedral cone $\mathcal{P} = \{X \in \mathcal{S}^n \mid X \geq 0\}$. In our numerical experiments, the test data for Q and c are taken from Biq Mac Library maintained by Wiegele, which is available at `http://biqmac.uni-klu.ac.at/biqmaclib.html`.

(ii) SDP and SDP+ problems arising from the relaxation of maximum stable set problems. Given a graph G with edge set $\mathcal{E}$, the SDP and SDP+ relaxation $\theta(G)$ and $\theta_+(G)$ of the maximum stable set problem are given by

$$\theta(G) = \max\{\langle ee^T,\, X\rangle \,:\, \langle E_{ij},\, X\rangle = 0, (i,j) \in \mathcal{E}, \langle I,\, X\rangle = 1, X \in \mathcal{S}_+^n\}, \tag{45}$$

$$\theta_+(G) = \max\{\langle ee^T,\, X\rangle \,:\, \langle E_{ij},\, X\rangle = 0, (i,j) \in \mathcal{E}, \langle I,\, X\rangle = 1, X \in \mathcal{S}_+^n, X \in \mathcal{P}\}, \tag{46}$$

where $E_{ij} = e_i e_j^{\mathrm{T}} + e_j e_i^{\mathrm{T}}$ and e_i denotes the ith column of the identity matrix, $\mathcal{P} = \{X \in \mathcal{S}^n \mid X \geq 0\}$. In our numerical experiments, we test the graph instances G considered in [20], [23], and [24].

(iii) SDP+ relaxation for computing lower bounds for quadratic assignment problems (QAPs). Let Π be the set of $n \times n$ permutation matrices. Given matrices $A, B \in \mathcal{S}^n$, the QAP is given by

$$v_{\mathrm{QAP}}^* := \min\{\langle X, AXB\rangle \,:\, X \in \Pi\}. \tag{47}$$

For a matrix $X = [x_1, \ldots, x_n] \in \Re^{n\times n}$, we will identify it with the n^2-vector $x = [x_1; \ldots; x_n]$. For a matrix $Y \in R^{n^2\times n^2}$, 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 [16] that $v^*_{\rm QAP}$ is bounded below by the following number generated from the SDP+ relaxation of (47):

$$\begin{array}{rll} v := & \min & \langle B \otimes A,\, Y\rangle \\ & \text{s.t.} & \sum_{i=1}^n Y^{ii} = I,\ \langle I,\, Y^{ij}\rangle = \delta_{ij} \quad \forall\, 1 \le i \le j \le n, \\ & & \langle E,\, Y^{ij}\rangle = 1 \quad \forall\, 1 \le i \le j \le n, \quad Y \in \mathcal{S}^n_+,\ Y \in \mathcal{P}, \end{array} \tag{48}$$

where the sign $\otimes$ stands for the Kronecker product, E is the matrix of ones, and $\delta_{ij} = 1$ if $i = j$, and 0 otherwise, $\mathcal{P} = \{X \in \mathcal{S}^{n^2} \mid X \ge 0\}$. In our numerical experiments, the test instances (A, B) are taken from the QAP Library [8].

(iv) SDP+ relaxations of clustering problems (RCPs) described in [15, eq. (13), up to a constant]:

$$\min \Big\{ \langle -W,\, X\rangle \mid Xe = e, \langle I,\, X\rangle = K, X \in \mathcal{S}^n_+, X \in \mathcal{P} \Big\}, \tag{49}$$

where W is the so-called affinity matrix whose entries represent the similarities of the objects in the dataset, e is the vector of ones, and K is the number of clusters, $\mathcal{P} = \{X \in \mathcal{S}^n \mid X \ge 0\}$. All the data sets we tested are from the UCI Machine Learning Repository (available at `http://archive.ics.uci.edu/ml/datasets.html`). For some large data instances, we only select the first n rows. For example, the original data instance "`spambase`" has 4601 rows, we select the first 1500 rows to obtain the test problem "`spambase-large.2`" for which the number "2" means that there are $K = 2$ clusters.

(v) SDP+ problems arising from the SDP relaxation of frequency assignment problems (FAPs) [7]. Given a network represented by a graph G and an edge-weight matrix W, a certain type of frequency assignment problem on G can be relaxed into the following SDP (see [5, eq. (5)]):

$$\begin{array}{ll} \max & \langle (\frac{k-1}{2k}) L(G, W) - \frac{1}{2}\mathrm{Diag}(We),\, X\rangle \\ \text{s.t.} & \mathrm{diag}(X) = e, X \in \mathcal{S}^n_+, \\ & -E^{ij} \bullet X = 2/(k-1) \quad \forall (i,j) \in U \subseteq E, \\ & -E^{ij} \bullet X \le 2/(k-1) \quad \forall (i,j) \in E \setminus U, \end{array} \tag{50}$$

where $k > 1$ is an integer, $L(G, W) := \mathrm{Diag}(We) - W$ is the Laplacian matrix, $E^{ij} = e_i e_j^T + e_j e_i^T$ with $e_i \in \Re^n$ being the ith standard unit vector and $e \in \Re^n$ is the vector of all ones. Define $M_{ij} = -\frac{1}{k-1}$ if $(i,j) \in E$, and $M_{ij} = 0$ otherwise. Then (50) is equivalent to

$$\begin{array}{ll} \max & \langle (\frac{k-1}{2k}) L(G, W) - \frac{1}{2}\mathrm{Diag}(We),\, X\rangle \\ \text{s.t.} & \mathrm{diag}(X) = e, X \in \mathcal{S}^n_+, X - M \in \mathcal{P}, \end{array} \tag{51}$$

where $\mathcal{P} = \{X \in \mathcal{S}^n \mid X_{ij} = 0, \forall (i,j) \in U; X_{ij} \geq 0, \forall (i,j) \in E \setminus U\}$.

We should mention that we can easily extend our algorithm to handle the following more general SDP+ problem:

$$\min \Big\{ \langle C,\, X \rangle \mid \mathcal{A}(X) = b,\ X \in \mathcal{K},\ X - M \in \mathcal{P} \Big\}, \tag{52}$$

where $M \in \mathcal{X}$ is a given matrix. Thus (51) can also be solved by our proposed algorithm.

(vi) SDP relaxations for rank-1 tensor approximations (R1TA) [13]:

$$\max \Big\{ \langle f,\, y \rangle \mid M(y) \in \mathcal{S}^n_+,\ \langle g,\, y \rangle = 1 \Big\}, \tag{53}$$

where $y \in \Re^{\mathbb{N}^n_m}$, $M(y)$ is a linear pencil in y. The dual of (53) is given by

$$\min \Big\{ \gamma \mid \gamma g - f = M^*(X),\ X \in \mathcal{S}^n_+ \Big\}. \tag{54}$$

It is shown in [12] that (54) can be transformed into a standard SDP (up to a constant):

$$\min \Big\{ \langle C,\, X \rangle \mid \mathcal{A}(X) = b,\ X \in \mathcal{S}^n_+ \Big\}, \tag{55}$$

where C is a constant matrix and $\mathcal{A}$ is a linear map, which depend on M, f, g.

4.2 Numerical Results

In this subsection, we compare the performance of our SDPNAL+ algorithm with two other competitive publicly available first order methods based codes for solving large-scale SDP+ and SDP problems: an ADMM based solver, called SDPAD (release-beta2, released in December 2012) developed in [25] and a two-easy-block-decomposition hybrid proximal extragradient method, which was called 2EBD-HPE[1] (v0.2, released on May 31, 2013) and we call it 2EBD here, introduced in [10]. Since we use the convergent ADMM with 3-block constraints introduced by Sun et al. [22] (which was called ADMM3c but we call it ADMM+ here to indicate that it is an enhanced version of ADMM with convergence guantantee) to warm start SDPNAL+, we also list the numerical results obtained by running ADMM+ alone for the purpose of demonstrating the power and the importance of the proposed majorized semismooth Newton-CG algorithm for solving difficult SDP+ and SDP problems.

All our computational results for the tested SDP+ and SDP problems are obtained by running MATLAB on a Linux server (6-core, Intel Xeon X5650 @ 2.67GHz, 32G RAM).

[1] `www2.isye.gatech.edu/~cod3/CamiloOrtiz/Software_files/2EBD-HPE_v0.2/2EBD-HPE_v0.2.zip`

In our numerical experiments, we measure the accuracy of an approximate optimal solution (X, y, S, Z) for (P) and (D) by using the following relative residual:

$$\eta = \max\{\eta_P, \eta_D, \eta_{\mathcal{K}}, \eta_{\mathcal{P}}, \eta_{\mathcal{K}^*}, \eta_{\mathcal{P}^*}, \eta_{C1}, \eta_{C2}\}, \tag{56}$$

where $\eta_P = \frac{\|\mathcal{A}X-b\|}{1+\|b\|}$, $\eta_D = \frac{\|\mathcal{A}^*y+S+Z-C\|}{1+\|C\|}$, $\eta_{\mathcal{K}} = \frac{\|\Pi_{\mathcal{K}^*}(-X)\|}{1+\|X\|}$, $\eta_{\mathcal{P}} = \frac{\|\Pi_{\mathcal{P}^*}(-X)\|}{1+\|X\|}$, $\eta_{\mathcal{K}^*} = \frac{\|\Pi_{\mathcal{K}}(-S)\|}{1+\|S\|}$, $\eta_{\mathcal{P}^*} = \frac{\|\Pi_{\mathcal{P}}(-Z)\|}{1+\|Z\|}$, $\eta_{C1} = \frac{|\langle X,S\rangle|}{1+\|X\|+\|S\|}$, $\eta_{C2} = \frac{|\langle X,Z\rangle|}{1+\|X\|+\|Z\|}$. Additionally, we compute the relative gap by

$$\eta_g = \frac{\langle C,X\rangle-\langle b,y\rangle}{1+|\langle C,X\rangle|+|\langle b,y\rangle|}. \tag{57}$$

Let $\varepsilon > 0$ be a given accuracy tolerance. We terminate both SDPNAL+ and ADMM+ when $\eta < \varepsilon$.

Note that SDPAD can be used to solve SDP+ problems of form (P) with $\mathcal{P} = \mathcal{S}^n_{\geq 0}$ directly and we stop SDPAD when $\eta < \varepsilon$, where η is defined as in (56). However, it is shown recently that the direct extension of ADMM to the multi-block case is not necessarily convergent [6]. Hence SDPAD, which is essentially an implementation of the direct extension of ADMM with the step length set at 1.618 for solving the dual of SDP+ problems, does not have convergence guarantee in theory.

The implementation of 2EBD including its termination, along with ADMM+ and SDPAD, is done in the same way as in [22]. For 2EBD, we reformulate QAP, RCP and R1TA problems as SDP problems in the standard form as these problems do not appear to have obvious two-easy blocks structures.

In our numerical experiments, we also use a restart strategy for SDPNAL+ if it is not able to achieve the required accuracy for the tested SDP+ problems. For some problems, even though η_P and η_D can reach the required accuracy tolerance, $\eta_{\mathcal{K}}$ or η_{C1} may stay above the required tolerance or stagnate. This may happen, as in the case for SDPNAL, because many of these SDP+ problems are degenerate at the optimal solutions. One way to overcome this difficulty is to apply ADMM+ to (P) using the most recently computed (y, S, Z, X, σ) to restart SDPNAL+ when its progress is not satisfactory. From this point of view, our proposed algorithm is quite flexible. In addition, the penalty parameter σ is dynamically adjusted according to the progress of the algorithm. A greater σ ensures faster convergence in theory but it leads to a more difficult inner problem (7a). Hence we adjust σ in order to balance this dilemma. However, the exact details on the restart and adjustment strategies are too tedious to be presented here. Note that MSNCG can be viewed as an alternating minimization between the blocks (y, S) and Z for solving problem (7a). Naturally we can try to alternately minimize the blocks (y, Z) and S, for which the computational cost of the generalized Newton system is cheaper due to the simple structure of $\mathcal{P}$. However, the overall cost can be much more expensive because the piecewise linear structure of $\Pi_{\mathcal{P}}(\cdot)$ generally does not give rise to a well-conditioned generalized Hessian, which leads to a slower convergence for solving the problem (7a).

Table 1 shows the number of problems that have been successfully solved to the accuracy of 10^{-6} in η by each of the four solvers SDPNAL+, ADMM+, SDPAD and 2EBD, with the maximum number of iterations set at 25000 or the maximum computation time set at 99 hours. As can be seen, only SDPNAL+ can solve all the problems to the accuracy of 10^{-6}. In particular, for the first time, we are able to solve all the 95 difficult SDP+ problems arising from QAP problems to an accuracy of 10^{-6} efficiently, while ADMM+, SDPAD and 2EBD can successfully solve 39, 30 and 16 problems, respectively.

Table 1: Number of problems which are solved to the accuracy of 10^{-6} in η.

problem set (No.) \ solver	SDPNAL+	ADMM+	SDPAD	2EBD
θ (58)	58	56	53	53
θ_+ (58)	58	58	58	56
FAP (7)	7	7	7	7
QAP (95)	95	39	30	16
BIQ (134)	134	134	134	134
RCP (120)	120	120	114	109
R1TA (55)	55	42	47	18
Total (527)	527	456	443	393

Tables 2 and 3 show the numerical results obtained by SDPNAL+ with the tolerance $\varepsilon = 10^{-6}$ for a subset of of the tested problems (the full set of numerical results can be found at `http://www.math.nus.edu.sg/∼mattohkc/publist.html/`). The first three columns of each table 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 total number of iterations for ADMM+, and the objective values $\langle C, X\rangle$ and $\langle b, y\rangle$. The relative infeasibilities and gap, as well as times (in the format hours:minutes:seconds) are listed in the last eight columns. It is interesting to note that all the tested problems (especially the QAPs) can be solved to the required accuracy 10^{-6} efficiently by SDPNAL+.

Tables 4 and 5 compare SDPNAL+, ADMM+, SDPAD and 2EBD on a subset of the tested SDP+ and SDP problems, respectively, using the tolerance $\varepsilon = 10^{-6}$. We terminate ADMM+, SDPAD, 2EBD after 25000 iterations or 99 hours. As can be seen, except for SDPNAL+, the required accuracy is not achieved for most of the tested QAPs after 25000 iterations for the solvers ADMM+, SDPAD and 2EBD. For the last three solvers, they typically converge very slowly when η falls below the range of 10^{-4}–10^{-5}. For R1TA problems, SDPNAL+ is significantly faster than the other 3 methods and it seems that only SDPNAL+ can solve those large scale ($n \geq 2000$) problems efficiently.

We observe that although ADMM+ and SDPAD perform similar steps in each iteration cycle (except that the former perform one extra update on the variable y to ensure the convergence of the algorithm), the former can be more efficient than latter on many tested instances. The main factors to account for the difference in the performance could be (a)

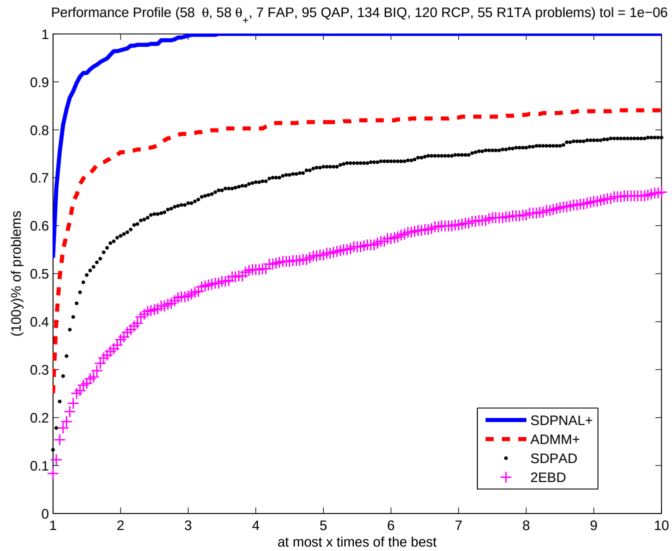

Figure 1: Performance profiles of SDPNAL+, ADMM+, SDPAD and 2EBD on $[1, 10]$

ADMM+ can take a larger step length for updating the multiplier (it is shown in [22] that the step length can be taken to be larger than $(1+\sqrt{5})/2$ when a certain checkable criterion holds); (b) it uses a more effective adjustment strategy for updating the penalty parameter σ; (c) in addition, ADMM+ also performs rescaling of the SDP data and restarting the algorithm whenever the ratio in $\|X^k\|$ and $\max\{\|S^k\|, \|Z^k\|, \|\mathcal{A}^* y^k\|\}$ deviate, say more than 20% from 1.

Figure 1 shows the performance profiles of SDPNAL+, ADMM+, SDPAD and 2EBD for all the 527 tested problems. We recall that a point (x, y) is in the performance profile curve of a method if and only if it can be solved exactly $(100y)\%$ of all the tested problems at most x times slower than any other method. It can be seen that SDPNAL+ outperforms the other 3 methods by a significant margin.

Acknowledgements

The authors would like to thank Jiawang Nie and Li Wang for sharing their codes on semidefinite relaxations of rank-1 tensor approximation problems.

References

[1] V. I. Arnold, *On matrices depending on parameters*, Russian Mathematical Surveys, 26 (1971), pp. 29–43.

[2] J. F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*, Springer, 2000.

[3] J. M. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization: theory and examples*, vol. 3, Springer, 2006.

[4] S. Burer, *On the copositive representation of binary and continuous nonconvex quadratic programs*, Mathematical Programming, 120 (2009), pp. 479–495.

[5] S. Burer, R. D. Monteiro, and Y. Zhang, *A computational study of a gradient-based log-barrier algorithm for a class of large-scale sdps*, Mathematical Programming, 95 (2003), pp. 359–379.

[6] C. Chen, B. He, Y. Ye, and X. Yuan, *The direct extension of admm for multi-block convex minimization problems is not necessarily convergent*, Mathematical Programming, (to appear).

[7] A. Eisenblätter, M. Grötschel, and A. M. Koster, *Frequency planning and ramifications of coloring*, Discussiones Mathematicae Graph Theory, 22 (2002), pp. 51–88.

[8] P. Hahn and M. Anjos, *QAPLIB – a quadratic assignment problem library*. `http://www.seas.upenn.edu/qaplib`.

[9] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with* $C^{1,1}$ *data*, Applied mathematics and optimization, 11 (1984), pp. 43–56.

[10] R. Monteiro, C. Ortiz, and B. Svaiter, *A first-order block-decomposition method for solving two-easy-block structured semidefinite programs*, Mathematical Programming Computation, (2013), pp. 1–48.

[11] J. J. Moreau, *Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires*, C. R. Acad. Sci., 255 (1962), pp. 238–240.

[12] J. Nie and L. Wang, *Regularization methods for sdp relaxations in large-scale polynomial optimization*, SIAM Journal on Optimization, 22 (2012), pp. 408–428.

[13] ——, *Semidefinite relaxations for best rank-1 tensor approximations*, SIAM J. on Matrix Analysis and Applications, 35 (2014), pp. 1155–1179.

[14] 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. 39–63.

[15] J. Peng and Y. Wei, *Approximating k-means-type clustering via semidefinite programming*, SIAM Journal on Optimization, 18 (2007), pp. 186–205.

[16] J. Povh and F. Rendl, *Copositive and semidefinite relaxations of the quadratic assignment problem*, Discrete Optimization, 6 (2009), pp. 231–241.

[17] R. T. Rockafellar, *Conjugate duality and optimization*, vol. 14, SIAM, 1974.

[18] ——, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of operations research, 1 (1976), pp. 97–116.

[19] ——, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[20] N. Sloane, *Challenge problems: Independent sets in graphs.* `http://www.research.att.com/njas/doc/graphs.html`, 2005.

[21] D.-F. Sun and J. Sun, *Semismooth matrix-valued functions*, Mathematics of Operations Research, 27 (2002), pp. 150–169.

[22] D.-F. Sun, K.-C. Toh, and L.-Q. Yang, *A convergent 3-block semi-proximal alternating direction method of multipliers for conic programming with 4-type of constraints*, SIAM J. Optimization, (to appear).

[23] K. C. Toh, *Solving large scale semidefinite programs via an iterative solver on the augmented systems*, SIAM J. Optimization, 14 (2004), pp. pp. 670–698.

[24] M. Trick, V. Chvatal, B. Cook, D. Johnson, C. McGeoch, and R. Tarjan, *The second dimacs implementation challenge — np hard problems: Maximum clique, graph coloring, and satisfiability.* `http://dimacs.rutgers.edu/Challenges/`, 1992.

[25] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Mathematical Programming Computation, 2 (2010), pp. 203–230.

[26] X.-Y. Zhao, D.-F. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. on Optimization, 20 (2010), pp. 1737–1765.

Table 2: Performance of SDPNAL+ on θ_+, FAP, QAP, BIQ and RCP problems
($\varepsilon = 10^{-6}$)

problem	m	$n_s; n_l$	it	itsub	itA	*pobj*	*dobj*	η_P	η_D	$\eta_{\mathcal{K}_1}$	$\eta_{\mathcal{K}_2}$	η_{C1}	η_{C2}	η_g	time
theta10	12470	500;	0	0	354	8.31485880 1	8.31490052 1	4.1-14	8.5-7	2.2-10	6.5-9	1.7-7	2.5-8	-2.5-6	46
theta102	37467	500;	0	0	157	3.80662264 1	3.80662724 1	1.1-13	9.5-7	8.5-9	7.0-9	1.7-8	9.6-10	-6.0-7	23
theta103	62516	500;	0	0	144	2.23774188 1	2.23774202 1	1.4-14	9.2-7	3.9-8	1.2-8	9.5-10	2.1-10	-3.0-8	22
theta104	87245	500;	0	0	169	1.32826073 1	1.32826099 1	1.7-14	9.3-7	1.5-8	2.6-9	2.1-9	6.9-10	-9.2-8	24
theta12	17979	600;	0	0	362	9.20905068 1	9.20909180 1	8.7-13	9.0-7	0	6.1-9	1.2-7	1.4-8	-2.2-6	1:15
theta123	90020	600;	0	0	156	2.44951466 1	2.44951496 1	6.6-14	9.3-7	1.3-8	4.1-9	1.4-9	3.8-10	-6.0-8	34
hamming-9-8	2305	512;	11	11	500	2.24000000 2	2.24000020 2	1.1-10	9.5-7	1.7-10	0.0-16	1.4-11	0.0-16	-4.4-8	44
hamming-10-2	23041	1024;	0	0	657	8.53345652 1	8.53332571 1	6.9-12	8.7-7	2.1-7	3.2-8	2.3-7	3.5-8	7.6-6	3:09
hamming-9-5-6	53761	512;	0	0	461	5.86651695 1	5.86665968 1	3.3-13	7.5-7	0	0	9.5-7	2.6-7	-1.2-5	45
G43	9991	1000;	21	21	973	2.79738220 2	2.79735897 2	5.3-12	8.9-7	1.7-7	2.7-8	2.1-7	1.6-8	4.1-6	12:32
G44	9991	1000;	21	21	942	2.79743352 2	2.79746225 2	5.7-12	9.9-7	1.2-8	1.5-8	2.6-7	5.0-8	-5.1-6	12:00
G45	9991	1000;	21	21	888	2.79313957 2	2.79317590 2	3.6-12	9.8-7	6.0-8	3.9-8	3.3-7	1.1-7	-6.5-6	11:43
G46	9991	1000;	21	21	887	2.79026490 2	2.79032512 2	3.6-12	9.7-7	5.8-8	4.8-8	5.6-7	1.4-7	-1.1-5	11:35
G47	9991	1000;	21	21	1042	2.80888870 2	2.80891901 2	9.7-12	9.4-7	7.2-10	1.5-8	2.7-7	2.4-8	-5.4-6	13:10
G51	5910	1000;	1	2	5672	3.49000718 2	3.49000007 2	8.6-10	9.9-7	4.5-7	4.9-7	4.2-8	1.8-8	1.0-6	1:15:12
G52	5917	1000;	5	5	10840	3.48386577 2	3.48386413 2	3.4-9	9.9-7	3.9-7	2.3-7	1.5-8	2.3-8	2.4-7	2:21:46
G53	5915	1000;	4	4	13260	3.48213570 2	3.48211536 2	3.3-9	9.9-7	5.0-7	3.0-7	1.2-7	4.9-8	2.9-6	2:48:21
G54	5917	1000;	8	8	4278	3.40999655 2	3.41000193 2	4.0-10	9.9-7	1.3-7	1.3-7	2.0-8	9.3-9	-7.9-7	51:18
1dc.1024	24064	1024;	0	0	2620	9.55514182 1	9.55511660 1	3.2-13	9.9-7	2.8-7	6.4-8	3.9-8	4.0-9	1.3-6	31:46
1et.1024	9601	1024;	0	0	1144	1.82071997 2	1.82071515 2	3.1-12	9.9-7	2.5-7	3.5-8	4.5-8	2.2-9	1.3-6	12:43
1tc.1024	7937	1024;	0	0	2732	2.04205928 2	2.04204076 2	7.4-13	9.9-7	6.3-7	5.1-7	1.5-7	2.9-8	4.5-6	31:34
1zc.1024	16641	1024;	0	0	711	1.28001293 2	1.27999917 2	3.9-12	7.7-7	1.6-7	2.0-8	1.8-7	1.7-8	5.4-6	7:12
2dc.1024	169163	1024;	0	0	4135	1.77104688 1	1.77099903 1	4.6-13	6.5-7	6.2-7	9.9-7	1.6-7	1.7-7	1.3-5	44:55
1dc.2048	58368	2048;	0	0	4153	1.74258920 2	1.74257466 2	1.1-11	9.9-7	3.7-7	2.9-7	9.5-8	1.2-8	4.2-6	5:50:06
1et.2048	22529	2048;	0	0	3039	3.38165943 2	3.38165218 2	1.2-11	9.9-7	1.8-7	1.7-7	2.7-8	8.6-9	1.1-6	4:01:54
1tc.2048	18945	2048;	0	0	2876	3.70489820 2	3.70488730 2	1.4-11	9.9-7	2.8-7	1.8-7	3.9-8	2.2-9	1.5-6	3:50:43
2dc.2048	504452	2048;	0	0	2997	2.87872690 1	2.87867849 1	1.9-12	9.9-7	2.9-7	6.1-7	7.3-8	7.0-8	8.3-6	3:54:58
fap11	252	252;	5	5	1180	2.97984952-2	2.98707469-2	8.1-12	9.6-7	4.9-7	4.5-15	2.4-7	6.5-15	-6.8-5	39
fap12	369	369;	15	15	1768	2.73312482-1	2.73415267-1	9.4-14	9.9-7	2.6-8	6.0-7	7.0-9	4.6-7	-6.6-5	1:56
fap25	2118	2118;	11	11	2268	1.28781491 1	1.28803442 1	2.2-7	9.1-7	0.8-15	4.5-7	4.7-16	9.2-7	-8.2-5	3:58:21
fap36	4110	4110;	4	4	2033	6.98573185 1	6.98607976 1	8.4-10	9.5-7	8.5-7	7.0-15	2.3-8	1.1-13	-2.5-5	23:07:56
bur26a	1051	676;	137	222	10228	5.42644954 6	5.42664508 6	7.6-11	9.9-7	2.6-7	7.9-7	5.1-7	2.5-9	-1.8-5	1:48:05
bur26c	1051	676;	247	441	21498	5.42685366 6	5.42707215 6	5.3-11	9.9-7	1.6-7	7.5-7	2.5-7	2.0-9	-2.0-5	2:03:12
bur26d	1051	676;	173	306	13287	3.82088460 6	3.82098028 6	1.5-10	9.9-7	9.4-8	8.0-7	2.8-7	3.5-10	-1.3-5	1:59:20
bur26e	1051	676;	129	361	14705	5.38699884 6	5.38711462 6	2.3-10	8.8-7	1.1-7	9.4-7	1.4-8	2.5-8	-1.1-5	1:18:35
bur26f	1051	676;	107	248	11272	3.78211965 6	3.78219580 6	2.1-15	9.9-7	4.1-7	2.7-7	3.8-11	4.1-9	-1.0-5	1:45:13
bur26g	1051	676;	250	392	10817	1.01172603 7	1.01177521 7	2.3-11	9.9-7	5.0-7	5.6-7	2.3-9	4.2-9	-2.4-5	1:32:44
bur26h	1051	676;	146	360	10658	7.09871739 6	7.09844876 6	1.5-10	9.5-7	7.4-7	9.9-7	2.4-9	4.7-9	1.9-5	1:25:45
chr22a	757	484;	104	250	6940	6.15599997 3	6.15591877 3	7.8-9	3.5-7	2.6-10	0.2-16	3.5-11	0.0-16	6.6-6	22:57
chr22b	757	484;	89	189	5620	6.19399919 3	6.19397816 3	2.0-7	3.1-7	1.6-7	1.9-16	6.3-10	0.0-16	1.7-6	12:46
chr25a	973	625;	53	200	5151	3.79659302 3	3.79236524 3	2.5-11	8.6-7	6.5-8	3.7-8	4.2-8	2.0-7	**5.6-4**	21:04
esc32a	1582	1024;	46	78	1664	1.03320457 2	1.03320681 2	3.3-13	9.5-7	9.9-7	8.1-7	6.6-9	7.5-9	-1.1-6	32:32
esc32b	1582	1024;	52	100	2196	1.31863123 2	1.31876533 2	7.2-12	9.8-7	3.7-7	9.6-7	6.7-8	9.3-7	-5.1-5	45:50
esc32c	1582	1024;	46	139	3562	6.15172917 2	6.15177999 2	3.0-7	9.7-7	8.6-13	3.2-7	3.3-13	1.4-7	-4.1-6	1:06:19
esc32d	1582	1024;	0	0	678	1.90223554 2	1.90227139 2	3.5-12	9.9-7	2.2-7	2.7-7	3.1-7	2.3-7	-9.4-6	9:40
esc32e	1582	1024;	40	47	1248	1.90001450 0	1.89997253 0	1.6-8	9.9-7	1.3-15	1.4-8	1.8-16	6.4-9	8.7-6	22:00
esc32f	1582	1024;	40	47	1248	1.90001450 0	1.89997253 0	1.6-8	9.9-7	1.3-15	1.4-8	1.8-16	6.4-9	8.7-6	21:31
esc32g	1582	1024;	0	0	520	5.83333959 0	5.83331567 0	2.5-13	9.3-7	9.7-9	8.0-8	2.1-10	4.4-9	1.9-6	7:17
kra30a	1393	900;	49	72	3208	8.68155154 4	8.68268686 4	1.2-10	7.5-7	1.2-7	9.9-7	3.4-7	1.9-7	-6.5-5	52:13
kra30b	1393	900;	81	101	3080	8.78355000 4	8.78468655 4	7.1-11	7.0-7	1.4-7	9.9-7	3.7-7	1.7-7	-6.5-5	55:48
kra32	1582	1024;	67	83	2946	8.57529179 4	8.57648412 4	1.3-11	8.4-7	1.4-7	9.9-7	3.5-7	9.3-7	-7.0-5	1:07:22
lipa30a	1393	900;	443	1216	1300	1.31780000 4	1.31780000 4	4.4-8	1.0-7	1.0-8	1.9-16	2.5-13	0.0-16	-4.3-10	47:33
lipa30b	1393	900;	4	9	820	1.51426000 5	1.51426111 5	2.3-9	8.2-9	3.3-14	4.7-16	4.1-14	0.1-16	-3.7-7	11:08
lipa40a	2458	1600;	153	546	3732	3.15379935 4	3.15379159 4	5.5-7	1.2-7	7.7-8	1.4-16	5.6-10	0.0-16	1.2-6	3:01:05
lipa40b	2458	1600;	5	18	991	4.76581311 5	4.76581129 5	4.2-7	1.5-7	3.3-7	5.2-16	3.7-9	0.0-16	1.9-7	1:02:59
nug24	898	576;	43	66	2359	3.40071083 3	3.40090464 3	1.0-10	9.6-7	1.1-7	9.9-7	2.6-7	2.5-7	-2.8-5	14:12
nug25	973	625;	48	76	2708	3.62566278 3	3.62577908 3	9.7-11	7.9-7	6.0-8	9.9-7	1.1-7	7.2-8	-1.6-5	19:25
nug27	1132	729;	49	86	3300	5.12924842 3	5.12946060 3	1.5-10	9.0-7	6.2-8	9.9-7	1.2-7	3.2-7	-2.1-5	36:53
nug28	1216	784;	50	77	3190	5.02532199 3	5.02552528 3	1.6-10	8.7-7	7.2-8	9.9-7	1.3-7	4.4-7	-2.0-5	40:26

Table 2: Performance of SDPNAL+ on θ_+, FAP, QAP, BIQ and RCP problems ($\varepsilon = 10^{-6}$)

problem	m	$n_s; n_l$	it	itsub	itA	*pobj*	*dobj*	η_P	η_D	$\eta_{\mathcal{K}_1}$	$\eta_{\mathcal{K}_2}$	η_{C1}	η_{C2}	η_g	time
nug30	1393	900;	44	68	2463	5.94890136 3	5.94920345 3	3.8-11	9.9-7	1.3-7	9.6-7	3.3-7	1.1-7	-2.5-5	45:02
ste36a	1996	1296;	122	189	7344	9.25661751 3	9.25811948 3	4.3-11	9.9-7	1.6-7	9.9-7	3.9-7	6.8-8	-8.1-5	6:29:21
ste36b	1996	1296;	173	242	11851	1.56582438 4	1.56653460 4	1.5-10	9.9-7	1.8-7	9.9-7	5.4-7	4.2-8	**-2.3-4**	9:45:58
ste36c	1996	1296;	143	202	10008	8.13267040 6	8.13407147 6	4.6-11	9.9-7	1.8-7	9.5-7	4.8-7	3.8-9	-8.6-5	8:06:50
tai25a	973	625;	33	42	2630	1.11336476 6	1.11525360 6	8.5-12	9.9-7	5.1-8	9.4-7	3.0-9	2.7-9	**-8.5-4**	14:52
tai25b	973	625;	296	344	18325	3.37687430 8	3.37871783 8	1.7-10	9.9-7	1.7-7	9.9-7	9.4-7	5.1-8	**-2.7-4**	1:18:04
tai30a	1393	900;	39	39	1614	1.70671520 6	1.70679434 6	2.8-11	7.4-7	3.4-7	9.9-7	6.5-7	2.2-7	-2.3-5	29:11
tai30b	1393	900;	236	342	16584	5.98852630 8	5.99068570 8	9.0-7	9.9-7	6.9-14	8.5-7	6.1-15	2.3-9	**-1.8-4**	2:52:00
tai35a	1888	1225;	38	38	3467	2.21649346 6	2.21657164 6	4.8-11	6.6-7	2.9-7	9.9-7	5.1-7	2.4-7	-1.8-5	1:56:18
tai35b	1888	1225;	142	214	10915	2.69644456 8	2.69710521 8	2.6-10	9.9-7	1.7-7	9.8-7	7.7-7	6.7-8	**-1.2-4**	8:01:01
tai40a	2458	1600;	33	33	3395	2.84310602 6	2.84321095 6	7.6-11	3.5-7	2.8-7	9.9-7	6.0-7	2.0-7	-1.8-5	3:56:34
tai40b	2458	1600;	101	146	7124	6.09005347 8	6.09143489 8	5.6-10	9.9-7	2.5-7	9.9-7	9.9-7	2.4-8	**-1.1-4**	10:55:44
tho30	1393	900;	44	74	2925	1.43549788 5	1.43563445 5	6.3-11	9.9-7	1.8-7	9.9-7	5.1-7	6.9-8	-4.8-5	1:03:01
tho40	2458	1600;	24	51	3998	2.26485088 5	2.26503953 5	2.0-10	9.9-7	2.0-7	9.9-7	5.2-7	2.7-8	-4.2-5	5:08:15
be250.1	251	251;	122	123	2800	-2.51194635 4	-2.51194398 4	1.1-7	9.9-7	1.3-7	0.2-16	1.8-8	4.6-16	-4.7-7	1:13
be250.2	251	251;	121	121	2842	-2.36814919 4	-2.36814545 4	1.2-12	9.9-7	1.1-8	2.0-8	1.5-8	2.3-8	-7.9-7	1:12
be250.3	251	251;	84	89	2200	-2.40000031 4	-2.39999662 4	1.3-7	9.9-7	1.2-7	0.1-16	1.2-8	1.4-16	-7.7-7	59
be250.4	251	251;	208	209	3850	-2.57203185 4	-2.57202544 4	9.7-9	9.9-7	5.5-8	0.0-16	2.0-8	0.2-16	-1.2-6	1:42
be250.5	251	251;	115	127	2791	-2.23747084 4	-2.23746795 4	4.1-12	9.9-7	3.1-9	3.4-8	8.2-9	2.9-7	-6.5-7	1:15
bqp500-1	501	501;	138	171	2499	-1.25964032 5	-1.25964547 5	1.6-11	9.9-7	6.5-9	6.8-9	9.1-7	1.6-7	2.0-6	5:20
bqp500-2	501	501;	142	194	2390	-1.36011042 5	-1.36011154 5	6.8-8	9.9-7	8.2-8	0.1-16	4.8-9	0.3-16	4.1-7	5:29
bqp500-3	501	501;	135	180	2390	-1.38453338 5	-1.38453549 5	2.0-8	9.7-7	3.8-7	0.5-16	6.6-8	2.8-16	7.6-7	6:31
bqp500-4	501	501;	128	174	2390	-1.39328333 5	-1.39328503 5	2.3-7	9.9-7	7.1-8	0.2-16	2.1-9	6.0-16	6.1-7	6:08
bqp500-5	501	501;	169	206	2910	-1.34092095 5	-1.34092382 5	4.5-8	9.9-7	4.8-8	0.0-16	1.0-8	0.2-16	1.1-6	7:25
gka1f	501	501;	166	203	2780	-6.55590598 4	-6.55591369 4	1.3-8	9.8-7	1.6-7	0.3-16	1.3-8	6.3-16	5.9-7	6:32
gka2f	501	501;	205	242	3541	-1.07931739 5	-1.07932064 5	1.0-11	9.9-7	6.0-9	9.1-9	1.6-7	8.1-8	1.5-6	7:54
gka3f	501	501;	174	216	2954	-1.50150987 5	-1.50151193 5	6.4-12	9.9-7	1.5-8	3.4-8	5.2-8	6.6-7	6.8-7	6:51
gka4f	501	501;	183	222	3101	-1.87087878 5	-1.87087908 5	4.2-12	9.9-7	4.2-9	2.4-8	2.4-8	2.0-7	8.2-8	7:10
gka5f	501	501;	142	187	2520	-2.06914264 5	-2.06914258 5	1.8-7	9.9-7	7.1-8	0.2-16	1.3-8	1.9-16	-1.5-8	5:53
soybean-large.2	308	307;	2	2	1171	5.46342122 3	5.46342235 3	8.1-13	9.2-7	8.0-9	9.9-7	1.8-8	6.6-7	-1.0-7	29
soybean-large.3	308	307;	2	2	934	4.57580592 3	4.57580844 3	4.6-13	7.2-7	1.7-8	2.7-7	4.5-8	9.3-8	-2.8-7	25
soybean-large.4	308	307;	52	52	1506	4.04637305 3	4.04637422 3	1.0-13	7.7-7	2.8-7	8.7-7	2.7-8	3.0-7	-1.4-7	52
soybean-large.5	308	307;	2	2	814	3.63158072 3	3.63158133 3	2.6-13	9.8-7	0	9.6-7	1.7-8	2.0-7	-8.4-8	22
soybean-large.6	308	307;	0	0	413	3.26767677 3	3.26767798 3	4.1-12	9.4-7	1.3-7	5.7-7	4.3-7	1.1-7	-1.9-7	12
spambase-large.2	1501	1500;	0	0	535	4.71138593 8	4.71150439 8	9.9-7	9.9-7	1.6-15	3.0-7	4.3-16	2.0-7	-1.3-5	11:07
spambase-large.3	1501	1500;	8	8	1844	2.36009657 8	2.36013239 8	2.5-10	8.9-7	2.3-7	9.9-7	6.0-7	5.3-8	-7.6-6	1:40:31
spambase-large.4	1501	1500;	8	8	4519	1.39698995 8	1.39699718 8	8.7-10	9.8-7	0	9.9-7	6.1-9	6.7-8	-2.6-6	2:49:39
spambase-large.5	1501	1500;	8	8	9184	1.02748129 8	1.02754393 8	3.8-13	9.7-7	2.6-8	5.7-7	2.5-8	9.6-7	-3.0-5	4:49:37
spambase-large.6	1501	1500;	8	8	2798	7.27756732 7	7.27685611 7	8.0-12	9.9-7	0	9.1-7	4.5-7	8.6-7	4.9-5	2:07:59
abalone-large.2	1001	1000;	0	0	576	5.52269325 3	5.52256503 3	9.9-7	5.2-7	1.4-15	2.2-7	1.8-15	1.0-7	1.2-5	5:01
abalone-large.3	1001	1000;	21	21	762	2.81040989 3	2.81042183 3	2.1-13	9.2-7	7.6-7	6.4-7	3.9-8	2.2-7	-2.1-6	7:29
abalone-large.4	1001	1000;	0	0	545	1.72764378 3	1.72763706 3	2.7-11	4.9-7	0	2.0-7	9.9-7	7.9-8	1.9-6	6:43
abalone-large.5	1001	1000;	38	38	797	1.21466288 3	1.21471600 3	3.3-12	9.5-7	1.3-7	4.6-7	7.6-7	2.6-9	-2.2-5	11:45
abalone-large.6	1001	1000;	8	8	781	9.17362617 2	9.17389149 2	6.7-11	9.9-7	0	6.6-7	5.2-7	2.3-7	-1.4-5	9:12
segment-large.2	1001	1000;	8	8	1191	1.47176055 7	1.47174710 7	9.7-12	9.4-7	0	1.6-7	9.9-7	6.5-8	4.6-6	9:16
segment-large.3	1001	1000;	0	0	373	1.03929738 7	1.03929372 7	6.1-12	9.9-7	0	9.7-7	3.9-7	1.1-7	1.8-6	2:43
segment-large.4	1001	1000;	2	2	1879	8.16944543 6	8.16945493 6	7.0-12	9.0-7	1.3-9	9.9-7	3.7-9	2.1-7	-5.8-7	13:52
segment-large.5	1001	1000;	8	8	2449	6.98489394 6	6.98490266 6	1.2-11	9.9-7	2.3-9	9.7-7	6.8-9	2.2-7	-6.2-7	19:06
segment-large.6	1001	1000;	8	8	3158	6.09809592 6	6.09811370 6	2.5-11	8.8-7	0	9.9-7	3.7-9	2.6-7	-1.5-6	24:00
housing.2	507	506;	8	8	3373	5.76086706 6	5.76093491 6	9.9-7	9.9-7	1.4-15	8.7-8	3.2-15	3.8-8	-5.9-6	4:50
housing.3	507	506;	8	8	1576	3.00980144 6	3.00979147 6	4.5-12	8.6-7	0	7.0-8	9.7-7	1.2-7	1.7-6	3:20
housing.4	507	506;	8	8	1645	1.79283384 6	1.79284813 6	7.5-12	9.9-7	2.8-8	2.3-8	8.3-8	8.5-9	-4.0-6	2:50
housing.5	507	506;	8	8	1918	1.38028143 6	1.38019123 6	7.4-12	9.9-7	0	7.5-8	9.5-7	1.6-7	3.3-5	3:30
housing.6	507	506;	11	11	533	1.11181933 6	1.11182191 6	6.5-13	9.9-7	4.4-7	9.6-7	8.2-7	1.8-7	-1.2-6	1:06

Table 3: Performance of SDPNAL+ on θ and R1TA problems ($\varepsilon = 10^{-6}$)

problem	m	$n_s; n_l$	it	itsub	itA	*pobj*	*dobj*	η_P	η_D	$\eta_{\mathcal{K}_1}$	$\eta_{\mathcal{K}_2}$	η_{C1}	η_{C2}	η_g	time
theta10	12470	500;	11	11	200	8.38059601 1	8.38059488 1	4.4-8	7.6-7	6.5-16	0	4.0-16	0	6.7-8	32
theta102	37467	500;	11	11	84	3.83905392 1	3.83905464 1	8.1-8	6.8-7	3.4-16	0	0.4-16	0	-9.3-8	21
theta103	62516	500;	20	20	64	2.25285667 1	2.25285686 1	2.8-7	8.1-7	1.8-16	0	7.6-16	0	-4.1-8	38
theta104	87245	500;	43	47	63	1.33361259 1	1.33361411 1	3.9-7	4.4-7	1.3-16	0	1.5-16	0	-5.5-7	53
theta12	17979	600;	13	13	200	9.28016817 1	9.28016713 1	2.3-8	3.7-7	0.9-15	0	1.3-16	0	5.6-8	51
theta123	90020	600;	12	12	70	2.46686527 1	2.46686518 1	6.0-8	7.1-7	2.1-16	0	1.1-16	0	1.9-8	36
hamming-9-8	2305	512;	12	12	200	2.24000000 2	2.24000041 2	1.1-9	2.5-7	7.1-16	0	4.4-16	0	-9.2-8	21
hamming-10-2	23041	1024;	10	10	200	1.02399926 2	1.02400047 2	1.2-8	4.8-7	1.3-16	0	0.8-16	0	-5.9-7	2:54
hamming-9-5-6	53761	512;	6	6	200	8.53333333 1	8.53333618 1	3.4-14	6.2-7	5.9-16	0	0.2-16	0	-1.7-7	20
G43	9991	1000;	27	27	200	2.80625087 2	2.80624586 2	6.8-7	9.8-7	1.6-15	0	1.6-14	0	8.9-7	3:32
G44	9991	1000;	30	30	200	2.80583223 2	2.80583220 2	1.1-7	6.1-7	2.5-15	0	0.8-16	0	5.9-9	3:48
G45	9991	1000;	26	26	200	2.80185148 2	2.80185127 2	1.3-7	6.4-7	3.2-15	0	2.4-15	0	3.8-8	3:31
G46	9991	1000;	28	30	200	2.79837009 2	2.79836974 2	1.9-7	7.9-7	5.7-15	0	5.4-15	0	6.2-8	3:49
G47	9991	1000;	30	31	200	2.81894037 2	2.81893954 2	1.8-7	3.7-7	1.4-14	0	1.4-14	0	1.5-7	3:52
G51	5910	1000;	148	584	200	3.48999920 2	3.48999975 2	7.3-7	5.9-7	9.2-14	0	4.6-14	0	-7.9-8	41:06
G52	5917	1000;	458	1619	200	3.48387855 2	3.48386488 2	7.6-7	9.3-7	3.0-15	0	2.4-14	0	2.0-6	3:53:19
G53	5915	1000;	425	1183	200	3.48348615 2	3.48347655 2	4.7-7	9.9-7	8.9-14	0	3.2-15	0	1.4-6	2:16:35
G54	5917	1000;	123	462	200	3.41000018 2	3.40999990 2	2.5-7	9.3-7	9.3-15	0	5.7-16	0	4.2-8	23:17
1dc.1024	24064	1024;	48	74	200	9.59856502 1	9.59849891 1	9.1-7	5.9-7	5.5-14	0	3.8-14	0	3.4-6	14:32
1et.1024	9601	1024;	64	129	200	1.84226960 2	1.84226147 2	5.9-7	3.3-7	5.3-15	0	6.1-14	0	2.2-6	35:33
1tc.1024	7937	1024;	156	417	200	2.06304654 2	2.06304284 2	7.5-7	6.2-7	3.1-14	0	7.5-16	0	8.9-7	1:22:24
1zc.1024	16641	1024;	16	16	200	1.28666672 2	1.28666665 2	1.8-8	8.4-7	0.8-15	0	3.8-15	0	2.7-8	4:56
2dc.1024	169163	1024;	148	376	200	1.86381938 1	1.86378972 1	8.2-7	9.2-7	1.1-14	0	4.0-15	0	7.8-6	2:21:20
1dc.2048	58368	2048;	62	112	200	1.74731330 2	1.74729390 2	9.7-7	6.2-7	5.3-14	0	1.6-14	0	5.5-6	1:55:31
1et.2048	22529	2048;	228	658	200	3.42031726 2	3.42029072 2	8.7-7	8.0-7	1.1-14	0	4.8-15	0	3.9-6	5:29:46
1tc.2048	18945	2048;	509	1725	200	3.74644012 2	3.74643271 2	4.7-7	8.2-7	1.6-14	0	9.6-15	0	9.9-7	22:10:45
2dc.2048	504452	2048;	167	385	200	3.06739144 1	3.06728102 1	9.8-7	8.2-7	3.2-15	0	2.7-15	0	1.8-5	14:22:06
nonsym(8,4)	46655	512;	13	13	200	5.74083101 0	5.74083615 0	1.5-7	2.4-8	0	0	1.0-16	0	-4.1-7	29
nonsym(9,4)	91124	729;	25	33	200	1.06613340 0	1.06611027 0	8.5-8	4.2-7	0	0	3.0-16	0	7.4-6	2:00
nonsym(10,4)	166374	1000;	17	21	200	1.69471772 0	1.69472878 0	7.9-7	1.6-7	0.8-16	0	0.5-16	0	-2.5-6	3:37
nonsym(11,4)	287495	1331;	19	22	200	2.91348466 0	2.91343357 0	7.5-8	2.5-7	0.2-16	0	0.6-16	0	7.5-6	7:14
nonsym(5,5)	50624	625;	30	31	200	3.08257539 0	3.08254910 0	2.4-7	1.7-7	1.8-16	0	1.3-16	0	3.7-6	1:18
nonsym(6,5)	194480	1296;	26	28	200	3.09572024 0	3.09557416 0	6.4-7	6.6-7	0.8-16	0	0.1-16	0	2.0-5	6:39
sym_rd(3,35)	82250	666;	43	46	72	1.82999132 0	1.83002862 0	5.4-7	3.3-7	0	0	0.8-16	0	-8.0-6	1:27
sym_rd(3,40)	135750	861;	37	39	82	1.99315221 0	1.99323089 0	1.7-7	5.7-7	0.3-16	0	1.0-16	0	-1.6-5	2:18
sym_rd(3,45)	211875	1081;	31	31	87	2.14076548 0	2.14073540 0	5.5-7	2.2-7	0	0	1.1-16	0	5.7-6	4:31
sym_rd(3,50)	316250	1326;	37	37	87	2.06951100 0	2.06938546 0	5.1-7	7.6-7	0	0	0.1-16	0	2.4-5	8:24
sym_rd(4,35)	73814	630;	46	51	77	1.09833279 1	1.09831864 1	6.3-8	2.2-7	0	0	0.7-16	0	6.2-6	1:45
sym_rd(4,40)	123409	820;	60	76	86	1.15471518 1	1.15473381 1	5.4-8	6.6-7	1.6-13	0	3.2-16	0	-7.7-6	4:56
sym_rd(4,45)	194579	1035;	45	62	90	1.18424653 1	1.18425819 1	5.7-8	3.5-7	1.0-13	0	1.3-16	0	-4.7-6	7:44
sym_rd(4,50)	292824	1275;	45	62	91	1.30418148 1	1.30421731 1	6.8-8	7.9-7	1.0-15	0	1.4-16	0	-1.3-5	12:44
sym_rd(5,15)	54263	816;	41	43	111	3.49345457 0	3.49338911 0	5.3-8	2.4-7	0	0	0.3-16	0	8.2-6	2:53
sym_rd(5,20)	230229	1771;	29	35	139	4.17928037 0	4.17942269 0	3.1-7	2.9-7	0	0	0.8-16	0	-1.5-5	27:28
sym_rd(6,15)	38759	680;	32	35	137	2.70987911 1	2.70973348 1	4.7-7	6.4-7	0	0	1.2-16	0	2.6-5	1:33
sym_rd(6,20)	177099	1540;	26	37	185	3.15086704 1	3.15084788 1	6.4-7	4.2-7	5.3-16	0	1.2-16	0	3.0-6	22:48
nsym_rd([20,25,25])	68249	500;	47	51	66	2.78568871 0	2.78562272 0	2.4-7	6.0-7	0	0	2.0-16	0	1.0-5	58
nsym_rd([25,20,25])	68249	500;	49	50	77	2.77557008 0	2.77563681 0	8.4-8	7.0-7	0.3-16	0	3.8-16	0	-1.0-5	58
nsym_rd([25,25,20])	68249	500;	14	14	129	2.87657149 0	2.87658217 0	8.8-8	9.0-8	0	0	1.4-16	0	-1.6-6	34
nsym_rd([25,25,25])	105624	625;	56	64	95	2.83000532 0	2.83011896 0	1.5-7	9.0-7	0	0	1.7-16	0	-1.7-5	1:33
nsym_rd([30,30,30])	216224	900;	30	33	122	3.03772558 0	3.03770676 0	6.2-7	4.6-7	0.3-16	0	0.0-16	0	2.7-6	3:52
nsym_rd([35,35,35])	396899	1225;	45	49	141	3.07047975 0	3.07050501 0	1.4-7	2.0-7	0	0	1.4-16	0	-3.5-6	9:14
nsym_rd([40,40,40])	672399	1600;	33	35	93	3.87873078 0	3.87863899 0	2.1-7	3.3-7	0.2-16	0	0.1-16	0	1.0-5	14:23
nsym_rd([8,8,8,8])	46655	512;	11	11	200	2.83768958 0	2.83773386 0	1.0-7	3.3-7	0	0	1.0-16	0	-6.6-6	28
nsym_rd([9,9,9,9])	91124	729;	14	14	200	3.10895856 0	3.10890841 0	3.1-7	2.6-7	2.6-16	0	2.0-16	0	6.9-6	1:07
nonsym(12,4)	474551	1728;	5	17	200	5.92161950 0	5.92162092 0	2.8-8	4.5-9	0.1-16	0	0.2-16	0	-1.1-7	16:55
nonsym(13,4)	753570	2197;	15	55	200	7.27450656 0	7.27450942 0	5.2-7	5.6-9	1.0-16	0	0.8-16	0	-1.8-7	1:51:34
nonsym(7,5)	614655	2401;	32	43	200	5.10582689 0	5.10572890 0	2.4-7	2.0-7	0	0	2.7-16	0	8.7-6	53:29
nonsym(8,5)	1679615	4096;	14	22	200	5.77855140 0	5.77862086 0	5.2-7	1.0-7	0	0	4.8-16	0	-5.5-6	2:46:20
nonsym(18,4)	5000210	5832;	13	55	200	1.53963123 1	1.53954727 1	5.8-7	3.7-7	0	0	1.6-16	0	2.6-5	8:50:14
nonsym(20,4)	9260999	8000;	7	17	200	1.77231047 1	1.77233375 1	5.2-8	7.4-8	0	0	2.2-15	0	-6.4-6	8:26:40
nonsym(21,4)	12326390	9261;	7	21	200	2.03462783 1	2.03463278 1	5.7-8	1.3-8	2.9-15	0	2.9-15	0	-1.2-6	14:22:05

Table 4: Performance of SDPNAL+ (a), ADMM+ (b), SDPAD (c) and 2EBD (d) on θ_+, FAP, QAP, BIQ and RCP problems ($\varepsilon = 10^{-6}$)

problem	m	$n_s; n_l$	iteration a	b	c	d	η a	b	c	d	η_g a	b	c	d	time a	b	c	d
theta10	12470	500;	0, 0;354	354	351	490	8.5-7	8.5-7	9.9-7	9.9-7	-2.5-6	-2.5-6	-7.8-7	1.5-6	46	45	42	1:11
theta102	37467	500;	0, 0;157	157	130	355	9.5-7	9.5-7	9.0-7	9.9-7	-6.0-7	-6.0-7	2.1-6	2.2-6	23	23	23	54
theta103	62516	500;	0, 0;144	144	108	323	9.2-7	9.2-7	9.8-7	9.8-7	-3.0-8	-3.0-8	5.9-7	2.6-6	22	22	21	49
theta104	87245	500;	0, 0;169	169	123	338	9.3-7	9.3-7	9.8-7	9.9-7	-9.2-8	-9.2-8	1.1-6	3.2-6	24	24	20	51
theta12	17979	600;	0, 0;362	362	366	494	9.0-7	9.0-7	8.8-7	9.2-7	-2.2-6	-2.2-6	1.0-6	1.3-6	1:15	1:14	1:11	1:52
theta123	90020	600;	0, 0;156	156	107	345	9.3-7	9.3-7	9.9-7	9.9-7	-6.0-8	-6.0-8	5.1-7	2.6-6	34	35	33	1:26
hamming-9-8	2305	512;	11,11;500	2413	3100	938	9.5-7	9.6-7	9.6-7	9.0-7	-4.4-8	-1.2-5	-6.9-7	5.6-6	44	3:07	4:20	1:36
hamming-10-2	23041	1024;	0, 0;657	657	651	902	8.7-7	8.7-7	9.4-7	8.8-7	7.6-6	7.6-6	-2.6-6	3.4-5	3:09	3:05	5:17	3:47
hamming-9-5-6	53761	512;	0, 0;461	461	507	563	9.5-7	9.5-7	9.5-7	8.9-7	-1.2-5	-1.2-5	-1.9-6	8.0-6	45	45	54	58
G43	9991	1000;	21,21;973	1154	1147	934	8.9-7	9.8-7	9.4-7	9.9-7	4.1-6	-3.1-6	1.7-6	2.0-6	12:32	13:04	10:20	13:00
G44	9991	1000;	21,21;942	1151	1144	968	9.9-7	9.3-7	9.9-7	9.9-7	-5.1-6	-2.9-6	1.6-6	1.6-6	12:00	12:13	10:11	13:15
G45	9991	1000;	21,21;888	1175	1185	966	9.8-7	9.5-7	9.4-7	9.9-7	-6.5-6	2.9-6	-1.0-6	1.6-6	11:43	13:24	10:36	13:28
G46	9991	1000;	21,21;887	1199	1180	943	9.7-7	9.9-7	9.8-7	9.9-7	-1.1-5	-3.2-6	-1.0-6	1.4-6	11:35	12:55	10:42	12:58
G47	9991	1000;	21,21;1042	1186	1137	992	9.4-7	9.5-7	9.5-7	9.9-7	-5.4-6	2.9-6	-9.4-7	1.2-6	13:10	13:18	10:28	13:50
G51	5910	1000;	1, 2;5672	6207	10361	9586	9.9-7	9.9-7	9.9-7	9.9-7	1.0-6	3.7-7	2.6-7	5.6-7	1:15:12	1:21:52	2:11:03	2:31:30
G52	5917	1000;	5, 5;10840	11463	14163	12124	9.9-7	9.9-7	9.9-7	9.9-7	2.4-7	4.2-7	4.5-7	6.9-7	2:21:46	2:26:28	2:46:25	3:15:11
G53	5915	1000;	4, 4;13260	13289	23865	20623	9.9-7	9.9-7	9.9-7	9.9-7	2.9-6	2.6-6	2.9-6	4.2-6	2:48:21	2:49:53	4:48:56	5:49:06
G54	5917	1000;	8, 8;4278	3262	7542	5136	9.9-7	9.7-7	9.9-7	9.9-7	-7.9-7	3.1-6	4.6-7	1.3-6	51:18	38:42	1:26:47	1:17:01
1dc.1024	24064	1024;	0, 0;2620	2620	2681	3641	9.9-7	9.9-7	9.9-7	9.9-7	1.3-6	1.3-6	3.4-6	4.0-6	31:46	32:22	45:21	53:12
1et.1024	9601	1024;	0, 0;1144	1144	2563	2609	9.9-7	9.9-7	9.9-7	9.9-7	1.3-6	1.3-6	5.6-6	5.9-6	12:43	12:54	39:53	35:35
1tc.1024	7937	1024;	0, 0;2732	2732	6545	6675	9.9-7	9.9-7	9.9-7	9.9-7	4.5-6	4.5-6	4.5-6	4.2-6	31:34	32:08	1:48:31	1:40:06
1zc.1024	16641	1024;	0, 0;711	711	770	25000	7.7-7	7.7-7	9.9-7	3.1-5	5.4-6	5.4-6	2.0-6	**7.9-4**	7:12	7:19	12:18	7:48:20
2dc.1024	169163	1024;	0, 0;4135	4135	1896	1891	9.9-7	9.9-7	9.9-7	9.9-7	1.3-5	1.3-5	1.0-5	1.5-5	44:55	45:59	29:02	24:59
1dc.2048	58368	2048;	0, 0;4153	4153	7277	8476	9.9-7	9.9-7	9.9-7	9.9-7	4.2-6	4.2-6	6.4-6	6.5-6	5:50:06	5:47:45	13:59:49	16:04:13
1et.2048	22529	2048;	0, 0;3039	3039	4422	4739	9.9-7	9.9-7	9.9-7	9.9-7	1.1-6	1.1-6	4.8-6	7.8-6	4:01:54	4:04:34	8:47:18	8:28:46
1tc.2048	18945	2048;	0, 0;2876	2876	7329	7482	9.9-7	9.9-7	9.9-7	9.9-7	1.5-6	1.5-6	5.5-6	5.6-6	3:50:43	3:50:16	13:29:15	13:50:32
2dc.2048	504452	2048;	0, 0;2997	2997	2147	1849	9.9-7	9.9-7	9.9-7	9.9-7	8.3-6	8.3-6	1.0-5	2.2-5	3:54:58	3:52:42	4:13:47	3:07:46
fap11	252	252;	5, 5;1180	1559	2585	2771	9.6-7	5.3-7	9.9-7	9.7-7	-6.8-5	-1.9-5	**-2.2-4**	**-1.1-4**	39	50	1:07	1:18
fap12	369	369;	15,15;1768	1830	3394	3325	9.9-7	8.4-7	9.9-7	9.9-7	-6.6-5	-2.6-5	**-2.2-4**	**-1.3-4**	1:56	1:55	3:32	3:08
fap25	2118	2118;	11,11;2268	5799	5495	4498	9.2-7	9.9-7	9.9-7	9.9-7	-8.2-5	-3.2-5	**-1.1-4**	-7.1-5	3:58:21	10:55:33	13:26:47	8:11:50
fap36	4110	4110;	4, 4;2033	2824	4445	3500	9.5-7	9.9-7	9.9-7	9.8-7	-2.5-5	-1.7-5	-3.0-5	-2.8-5	23:07:56	30:57:53	78:43:03	43:37:44
bur26a	1051	676;	137,222;10228	25000	25000	25000	9.9-7	5.6-6	1.1-5	8.9-6	-1.8-5	-6.3-5	-7.7-5	-8.2-5	1:48:05	2:05:11	2:07:44	2:38:24
bur26b	1051	676;	100,208;8605	25000	25000	25000	9.9-7	6.8-6	1.1-5	9.3-6	-1.8-5	-5.7-5	-8.0-5	-7.5-5	1:32:52	2:07:13	1:57:30	2:49:59
bur26c	1051	676;	247,441;21498	25000	25000	25000	9.9-7	4.2-6	1.4-5	1.4-5	-2.0-5	-4.5-5	**-1.2-4**	**-1.8-4**	2:03:12	2:05:11	2:02:35	2:50:08
bur26d	1051	676;	173,306;13287	25000	25000	25000	9.9-7	6.4-6	1.5-5	1.3-5	-1.3-5	-8.4-5	**-1.2-4**	**-1.4-4**	1:59:20	2:02:24	1:51:20	2:53:07
bur26e	1051	676;	129,361;14705	25000	25000	25000	9.4-7	3.1-6	6.4-6	1.4-5	-1.1-5	-2.8-5	-3.6-5	**-1.9-4**	1:18:35	2:03:18	2:28:06	2:46:03
bur26f	1051	676;	107,248;11272	20887	25000	25000	9.9-7	9.9-7	8.1-6	1.2-5	-1.0-5	-1.0-5	-4.8-5	-7.5-5	1:45:13	1:45:08	2:09:11	2:44:28
bur26g	1051	676;	250,392;10817	17910	25000	25000	9.9-7	8.6-7	1.6-6	7.8-6	-2.4-5	-6.3-6	-4.0-5	-6.9-5	1:32:44	1:29:22	1:57:13	2:46:34
bur26h	1051	676;	146,360;10658	23208	25000	25000	9.9-7	9.4-7	1.4-6	2.3-5	1.9-5	-1.4-6	-2.3-5	**-1.7-4**	1:25:45	1:57:33	2:01:12	2:54:20
chr22a	757	484;	104,250;6940	6457	22364	25000	3.5-7	8.8-7	9.9-7	2.5-5	6.6-6	**3.9-4**	**-2.7-4**	**-2.6-3**	22:57	12:29	50:45	1:17:44
chr22b	757	484;	89,189;5620	7211	25000	25000	3.1-7	9.7-7	1.5-5	2.1-5	1.7-6	**3.4-4**	**-1.4-3**	**-1.8-3**	12:46	13:52	1:07:51	1:15:34
chr25a	973	625;	53,200;5151	7127	25000	25000	8.6-7	8.6-7	3.7-5	3.4-5	**5.6-4**	**7.8-4**	**-9.5-3**	**-6.8-3**	21:04	26:03	2:10:29	2:23:01
esc32a	1582	1024;	46,78;1664	4931	2247	9630	9.9-7	9.9-7	9.9-7	9.9-7	-1.1-6	-2.0-6	-1.1-6	-2.7-6	32:32	1:06:25	27:49	2:44:37

Table 4: Performance of SDPNAL+ (a), ADMM+ (b), SDPAD (c) and 2EBD (d) on θ_+, FAP, QAP, BIQ and RCP problems ($\varepsilon = 10^{-6}$)

	iteration	η	η_g	time
problem | m | $n_s; n_l$	a| b| c| d	a| b| c| d	a| b| c| d	a| b| c| d
esc32b | 1582 | 1024;	52,100;2196 | 25000 | 25000 | 25000	9.8-7 | 3.7-6 | 2.4-6 | 8.3-6	-5.1-5 | **-2.4-4** | **-2.2-4** | **-4.0-4**	45:50 | 5:18:19 | 4:31:51 | 7:59:47
esc32c | 1582 | 1024;	46,139;3562 | 25000 | 25000 | 25000	9.7-7 | 5.3-6 | 5.1-6 | 1.6-5	-4.1-6 | -5.4-5 | -5.2-5 | -9.8-5	1:06:19 | 4:58:47 | 4:00:23 | 8:01:19
esc32d | 1582 | 1024;	0, 0;678 | 678 | 799 | 1412	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	-9.4-6 | -9.4-6 | -1.7-5 | -2.2-7	9:40 | 9:39 | 8:01 | 25:40
esc32e | 1582 | 1024;	40,47;1248 | 1108 | 905 | 784	9.9-7 | 9.8-7 | 8.6-7 | 9.7-7	8.7-6 | -3.8-7 | -9.2-6 | -3.1-6	22:00 | 16:09 | 8:32 | 14:30
esc32f | 1582 | 1024;	40,47;1248 | 1108 | 905 | 784	9.9-7 | 9.8-7 | 8.6-7 | 9.7-7	8.7-6 | -3.8-7 | -9.2-6 | -3.1-6	21:31 | 15:45 | 8:25 | 12:57
esc32g | 1582 | 1024;	0, 0;520 | 520 | 588 | 981	9.3-7 | 9.3-7 | 9.2-7 | 9.9-7	1.9-6 | 1.9-6 | -3.3-6 | 3.6-7	7:17 | 7:14 | 5:41 | 17:19
esc32h | 1582 | 1024;	97,236;4959 | 25000 | 25000 | 25000	9.9-7 | 9.4-6 | 1.2-5 | 2.8-5	-4.4-5 | **-4.1-4** | **-5.0-4** | **-7.4-4**	1:42:34 | 5:01:41 | 4:13:31 | 7:30:56
kra30a | 1393 | 900;	49,72;3208 | 25000 | 25000 | 25000	9.9-7 | 1.3-5 | 1.4-5 | 1.7-6	-6.5-5 | **-4.2-4** | **-5.7-4** | -5.9-5	52:13 | 3:45:29 | 5:11:22 | 5:42:45
kra30b | 1393 | 900;	81,101;3080 | 25000 | 25000 | 25000	9.9-7 | 1.1-5 | 1.1-5 | 1.8-5	-6.5-5 | **-3.7-4** | **-4.9-4** | **-6.1-4**	55:48 | 3:56:14 | 5:15:18 | 5:45:45
kra32 | 1582 | 1024;	67,83;2946 | 25000 | 25000 | 25000	9.9-7 | 9.1-6 | 1.1-5 | 1.7-5	-7.0-5 | **-3.2-4** | **-4.0-4** | **-5.0-4**	1:07:22 | 5:07:43 | 6:57:49 | 8:11:14
lipa30a | 1393 | 900;	443,1216;1300 | 3533 | 11683 | 16167	1.0-7 | 8.9-7 | 9.5-7 | 9.9-7	-4.3-10 | 8.3-6 | 3.2-5 | -3.8-6	47:33 | 26:59 | 1:52:11 | 3:46:31
lipa30b | 1393 | 900;	4, 9;820 | 2700 | 7516 | 25000	8.2-9 | 9.1-7 | 9.9-7 | **1.6-4**	-3.7-7 | 4.3-5 | 4.7-5 | **1.1-2**	11:08 | 16:33 | 52:22 | 5:31:56
lipa40a | 2458 | 1600;	153,546;3732 | 6483 | 18785 | 25000	5.5-7 | 8.8-7 | 9.8-7 | 9.5-6	1.2-6 | 4.2-5 | -4.1-5 | **-1.5-4**	3:01:05 | 3:32:47 | 19:15:19 | 26:56:27
lipa40b | 2458 | 1600;	5,18;991 | 4878 | 5970 | 25000	4.2-7 | 9.0-7 | 9.8-7 | **4.6-4**	1.9-7 | **1.1-4** | -6.4-5 | **-4.1-2**	1:02:59 | 2:20:09 | 4:11:06 | 23:33:11
nug24 | 898 | 576;	43,66;2359 | 25000 | 25000 | 25000	9.9-7 | 9.1-6 | 1.1-5 | 1.6-5	-2.8-5 | **-1.9-4** | **-2.3-4** | **-2.7-4**	14:12 | 1:17:49 | 1:40:05 | 1:57:03
nug25 | 973 | 625;	48,76;2708 | 25000 | 25000 | 25000	9.9-7 | 1.2-5 | 1.0-5 | 1.7-5	-1.6-5 | **-2.0-4** | **-2.0-4** | **-2.5-4**	19:25 | 1:35:46 | 1:53:26 | 2:16:27
nug27 | 1132 | 729;	49,86;3300 | 25000 | 25000 | 25000	9.9-7 | 1.0-5 | 1.3-5 | 1.7-5	-2.1-5 | **-2.0-4** | **-2.6-4** | **-2.8-4**	36:53 | 2:21:06 | 2:51:26 | 3:28:20
nug28 | 1216 | 784;	50,77;3190 | 25000 | 25000 | 25000	9.9-7 | 9.3-6 | 1.2-5 | 1.7-5	-2.0-5 | **-1.8-4** | **-2.2-4** | **-2.6-4**	40:26 | 2:47:04 | 3:27:54 | 4:02:11
nug30 | 1393 | 900;	44,68;2463 | 25000 | 25000 | 25000	9.9-7 | 8.7-6 | 1.1-5 | 1.7-5	-2.5-5 | **-1.6-4** | **-1.9-4** | **-2.2-4**	45:02 | 3:48:43 | 4:58:12 | 5:39:31
ste36a | 1996 | 1296;	122,189;7344 | 25000 | 25000 | 25000	9.9-7 | 9.7-6 | 1.3-5 | 1.6-5	-8.1-5 | **-5.8-4** | **-6.8-4** | **-6.7-4**	6:29:21 | 9:38:26 | 12:37:18 | 14:09:11
ste36b | 1996 | 1296;	173,242;11851 | 25000 | 25000 | 25000	9.9-7 | 1.2-5 | 1.8-5 | 1.3-5	**-2.3-4** | **-1.5-3** | **-2.0-3** | **-2.1-3**	9:45:58 | 9:19:24 | 12:10:09 | 14:23:33
ste36c | 1996 | 1296;	143,202;10008 | 25000 | 25000 | 25000	9.9-7 | 1.2-5 | 1.5-5 | 1.6-5	-8.6-5 | **-5.8-4** | **-7.3-4** | **-7.2-4**	8:06:50 | 9:26:42 | 12:22:19 | 14:23:52
tai25a | 973 | 625;	33,42;2630 | 2201 | 1845 | 25000	9.9-7 | 9.5-7 | 9.9-7 | 1.7-6	**-8.5-4** | **-8.0-4** | **-7.2-4** | **-1.8-3**	14:52 | 8:38 | 9:24 | 2:27:04
tai25b | 973 | 625;	296,344;18325 | 25000 | 25000 | 25000	9.9-7 | 2.9-5 | 3.7-5 | 4.2-5	**-2.7-4** | **-2.0-3** | **-2.4-3** | **-2.5-3**	1:18:04 | 1:28:33 | 1:55:04 | 2:21:35
tai30a | 1393 | 900;	39,39;1614 | 25000 | 25000 | 25000	9.9-7 | 4.7-6 | 4.6-6 | 1.3-5	-2.3-5 | -6.3-5 | -7.3-5 | **-1.3-4**	29:11 | 3:53:48 | 6:09:25 | 6:00:13
tai30b | 1393 | 900;	236,342;16584 | 25000 | 25000 | 25000	9.9-7 | 2.0-5 | 2.4-5 | 2.6-5	**-1.8-4** | **-1.0-3** | **-1.2-3** | **-1.2-3**	2:52:00 | 3:42:12 | 4:28:02 | 5:38:24
tai35a | 1888 | 1225;	38,38;3467 | 25000 | 25000 | 25000	9.9-7 | 3.9-6 | 4.0-6 | 1.3-5	-1.8-5 | -4.8-5 | -5.6-5 | **-1.0-4**	1:56:18 | 9:21:21 | 15:00:46 | 12:53:01
tai35b | 1888 | 1225;	142,214;10915 | 25000 | 25000 | 25000	9.9-7 | 2.1-5 | 2.4-5 | 2.8-5	**-1.2-4** | **-9.1-4** | **-1.0-3** | **-1.1-3**	8:01:01 | 8:51:20 | 11:15:52 | 12:51:27
tai40a | 2458 | 1600;	33,33;3395 | 25000 | 25000 | 25000	9.9-7 | 3.7-6 | 4.0-6 | 1.4-5	-1.8-5 | -4.6-5 | -5.3-5 | **-1.0-4**	3:56:34 | 20:22:53 | 31:45:29 | 26:00:47
tai40b | 2458 | 1600;	101,146;7124 | 25000 | 25000 | 25000	9.9-7 | 1.9-5 | 2.5-5 | 3.1-5	**-1.1-4** | **-7.2-4** | **-8.1-4** | **-8.5-4**	10:55:44 | 17:50:19 | 23:17:25 | 25:23:31
tho30 | 1393 | 900;	44,74;2925 | 25000 | 25000 | 25000	9.9-7 | 1.1-5 | 1.5-5 | 2.2-5	-4.8-5 | **-2.6-4** | **-3.4-4** | **-4.0-4**	1:03:01 | 3:46:49 | 4:46:03 | 5:44:33
tho40 | 2458 | 1600;	24,51;3998 | 25000 | 25000 | 25000	9.9-7 | 9.3-6 | 1.3-5 | 2.0-5	-4.2-5 | **-2.1-4** | **-2.7-4** | **-3.2-4**	5:08:15 | 17:12:50 | 24:35:42 | 26:05:11
be250.1 | 251 | 251;	122,123;2800 | 4327 | 5345 | 3537	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	-4.7-7 | -2.0-7 | -3.6-7 | -4.1-7	1:13 | 1:16 | 1:35 | 1:37
be250.2 | 251 | 251;	121,121;2842 | 3827 | 5108 | 3044	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	-7.9-7 | -8.6-7 | -5.3-7 | -8.0-7	1:12 | 1:08 | 1:28 | 1:22
be250.3 | 251 | 251;	84,89;2200 | 3796 | 4331 | 2592	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	-7.7-7 | -1.1-6 | -7.3-7 | -1.1-6	59 | 1:11 | 1:18 | 1:11
be250.4 | 251 | 251;	208,209;3850 | 8023 | 8350 | 6453	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	-1.2-6 | -1.1-6 | -1.1-6 | -2.9-7	1:42 | 2:23 | 2:24 | 2:53
be250.5 | 251 | 251;	115,127;2791 | 4460 | 5089 | 3174	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	-6.5-7 | -7.5-7 | -6.4-7 | -7.2-7	1:15 | 1:23 | 1:31 | 1:26
bqp500-1 | 501 | 501;	138,171;2499 | 6473 | 6932 | 4086	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	2.0-6 | -1.4-6 | -3.4-7 | -1.6-6	5:20 | 8:35 | 9:45 | 9:13
bqp500-2 | 501 | 501;	142,194;2390 | 8008 | 10582 | 4862	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	4.1-7 | -4.2-7 | -8.6-8 | -1.2-6	5:29 | 10:46 | 14:42 | 10:52
bqp500-3 | 501 | 501;	135,180;2390 | 8192 | 8915 | 4965	9.7-7 | 9.9-7 | 9.9-7 | 9.9-7	7.6-7 | -1.5-6 | 3.7-7 | -5.8-7	6:31 | 12:53 | 12:25 | 11:22
bqp500-4 | 501 | 501;	128,174;2390 | 7188 | 9012 | 4031	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	6.1-7 | -1.0-6 | -3.8-7 | -1.2-6	6:08 | 10:37 | 12:10 | 9:11
bqp500-5 | 501 | 501;	169,206;2910 | 6898 | 7641 | 4541	9.9-7 | 9.9-7 | 9.9-7 | 9.9-7	1.1-6 | -8.9-7 | -1.2-6 | -8.2-7	7:25 | 10:34 | 10:57 | 10:19
gka1f | 501 | 501;	166,203;2780 | 6717 | 8147 | 4600	9.8-7 | 9.9-7 | 9.9-7 | 9.9-7	5.9-7 | -1.3-6 | -1.2-6 | -5.6-7	6:32 | 9:38 | 11:28 | 10:31

Table 4: Performance of SDPNAL+ (a), ADMM+ (b), SDPAD (c) and 2EBD (d) on θ_+, FAP, QAP, BIQ and RCP problems ($\varepsilon = 10^{-6}$)

problem	m	$n_s; n_l$	iteration a	iteration b	iteration c	iteration d	η a	η b	η c	η d	η_g a	η_g b	η_g c	η_g d	time a	time b	time c	time d
gka2f	501	501;	205,242;3541	7519	8949	5403	9.9-7	9.9-7	9.9-7	9.9-7	1.5-6	-1.5-6	-1.4-6	-1.0-6	7:54	10:50	12:52	12:10
gka3f	501	501;	174,216;2954	6102	7037	3957	9.9-7	9.9-7	9.9-7	9.9-7	6.8-7	-1.1-6	-2.0-6	-1.6-7	6:51	9:07	10:46	9:07
gka4f	501	501;	183,222;3101	6673	7529	4070	9.9-7	9.9-7	9.9-7	9.9-7	8.2-8	-1.1-6	-4.2-7	-3.5-7	7:10	9:13	11:20	9:14
gka5f	501	501;	142,187;2520	6482	7023	4210	9.9-7	9.9-7	9.9-7	9.9-7	-1.5-8	-5.9-7	-9.4-7	-7.5-7	5:53	9:14	10:36	9:45
soybean-large.2	308	307;	2, 2;1171	1190	5050	2261	9.9-7	9.2-7	9.9-7	9.9-7	-1.0-7	-7.7-8	-1.2-7	-7.0-8	29	29	3:45	3:09
soybean-large.3	308	307;	2, 2;934	922	5993	2159	7.2-7	8.8-7	9.6-7	8.5-7	-2.8-7	-2.1-7	-5.4-9	1.4-7	25	24	4:47	3:54
soybean-large.4	308	307;	52,52;1506	1609	13512	3831	8.7-7	9.9-7	9.9-7	9.9-7	-1.4-7	-2.8-7	-1.2-7	-1.6-7	52	42	10:51	7:18
soybean-large.5	308	307;	2, 2;814	850	2974	1404	9.8-7	9.7-7	9.9-7	9.9-7	-8.4-8	-9.1-8	-7.9-8	-1.7-7	22	23	2:23	2:05
soybean-large.6	308	307;	0, 0;413	413	545	681	9.4-7	9.4-7	6.8-7	9.1-7	-1.9-7	-1.9-7	3.5-8	1.3-6	12	12	21	44
spambase-large.2	1501	1500;	0, 0;535	535	992	4429	9.9-7	9.9-7	9.9-7	9.9-7	-1.3-5	-1.3-5	-1.2-5	-1.3-5	11:07	11:17	22:17	3:12:36
spambase-large.3	1501	1500;	8, 8;1844	1705	1830	6617	9.9-7	9.8-7	9.9-7	9.9-7	-7.6-6	-7.6-6	-6.6-6	-3.3-6	1:40:31	35:47	58:10	6:13:50
spambase-large.4	1501	1500;	8, 8;4519	3761	7091	25000	9.9-7	9.9-7	9.9-7	**2.2-2**	-2.6-6	9.4-8	-5.2-7	**-10.0-1**	2:49:39	1:19:26	5:32:29	17:57:38
spambase-large.5	1501	1500;	8, 8;9184	8398	7510	25000	9.7-7	9.9-7	9.8-7	**1.1-2**	-3.0-5	-2.9-5	-2.4-5	**3.0-1**	4:49:37	3:26:14	3:21:11	18:14:24
spambase-large.6	1501	1500;	8, 8;2798	2031	2415	25000	9.9-7	9.9-7	9.9-7	**1.8-2**	4.9-5	-4.2-5	-5.8-5	**-10.0-1**	2:07:59	49:32	1:07:56	17:07:48
abalone-large.2	1001	1000;	0, 0;576	576	650	1493	9.9-7	9.9-7	9.9-7	9.9-7	1.2-5	1.2-5	6.6-6	-1.4-6	5:01	5:07	5:08	31:13
abalone-large.3	1001	1000;	21,21;762	765	796	1306	9.2-7	9.9-7	9.9-7	9.9-7	-2.1-6	-3.6-6	-9.9-7	-4.2-6	7:29	6:09	8:56	22:21
abalone-large.4	1001	1000;	0, 0;545	545	629	710	9.9-7	9.9-7	9.6-7	9.9-7	1.9-6	1.9-6	-6.9-6	-9.2-7	6:43	6:50	5:01	12:03
abalone-large.5	1001	1000;	38,38;797	834	1107	833	9.5-7	9.9-7	9.9-7	9.9-7	-2.2-5	-1.5-5	-2.1-5	-2.1-5	11:45	8:39	9:11	14:17
abalone-large.6	1001	1000;	8, 8;781	796	1101	950	9.9-7	9.9-7	9.9-7	9.9-7	-1.4-5	-1.4-5	-1.8-5	-1.9-5	9:12	8:21	8:49	15:24
segment-large.2	1001	1000;	8, 8;1191	1264	1080	1745	9.9-7	9.9-7	9.8-7	9.9-7	4.6-6	5.0-6	-4.7-6	-5.0-7	9:16	9:15	8:27	34:22
segment-large.3	1001	1000;	0, 0;373	373	412	1956	9.9-7	9.9-7	9.8-7	9.9-7	1.8-6	1.8-6	-7.1-7	-1.1-6	2:43	2:41	3:33	37:08
segment-large.4	1001	1000;	2, 2;1879	2024	19479	6354	9.9-7	9.9-7	9.9-7	9.9-7	-5.8-7	-5.5-7	-4.5-7	-5.0-7	13:52	14:50	5:23:13	3:07:06
segment-large.5	1001	1000;	8, 8;2449	2711	22003	8257	9.9-7	9.9-7	9.9-7	9.9-7	-6.2-7	-6.7-7	-6.0-7	-6.4-7	19:06	20:31	6:09:59	4:19:44
segment-large.6	1001	1000;	8, 8;3158	3262	25000	10211	9.9-7	9.9-7	1.3-6	9.9-7	-1.5-6	-1.5-6	-9.6-7	-1.0-6	24:00	24:06	7:10:04	5:25:59
housing.2	507	506;	8, 8;3373	3284	2679	2566	9.9-7	9.6-7	9.9-7	8.6-7	-5.9-6	-5.4-6	-5.2-6	-5.3-6	4:50	4:31	3:26	7:52
housing.3	507	506;	8, 8;1576	1247	1523	1338	9.7-7	9.9-7	9.9-7	9.8-7	1.7-6	8.0-6	-6.7-6	5.2-6	3:20	1:34	1:56	4:29
housing.4	507	506;	8, 8;1645	1368	1064	1090	9.9-7	9.9-7	9.9-7	8.4-7	-4.0-6	-3.5-6	-4.9-6	8.3-5	2:50	2:00	1:25	3:40
housing.5	507	506;	8, 8;1918	1319	1916	1451	9.9-7	9.6-7	9.3-7	8.8-7	3.3-5	-3.2-5	3.6-5	6.3-5	3:30	2:07	2:36	5:03
housing.6	507	506;	11,11;533	536	842	1958	9.9-7	9.9-7	9.8-7	9.5-7	-1.2-6	-9.7-6	5.9-6	6.3-5	1:06	53	1:20	6:29

Table 5: Performance of SDPNAL+ (a), ADMM+ (b), SDPAD (c) and 2EBD (d) on θ and R1TA problems ($\varepsilon = 10^{-6}$)

problem	m	$n_s; n_l$	iteration a	iteration b	iteration c	iteration d	η a	η b	η c	η d	η_g a	η_g b	η_g c	η_g d	time a	time b	time c	time d
theta10	12470	500;	11,11;200	396	333	422	7.6-7	9.1-7	9.9-7	9.8-7	6.7-8	-2.1-6	-1.4-6	1.2-6	32	51	36	59
theta102	37467	500;	11,11;84	159	127	312	6.8-7	9.1-7	9.2-7	9.9-7	-9.3-8	1.3-6	1.7-6	1.6-6	21	22	21	47
theta103	62516	500;	20,20;64	144	104	300	8.1-7	9.7-7	9.9-7	9.8-7	-4.1-8	-6.2-8	2.0-7	2.0-6	38	20	21	45
theta104	87245	500;	43,47;63	151	116	342	4.4-7	9.0-7	9.5-7	9.8-7	-5.5-7	-2.4-8	8.3-7	3.2-6	53	22	19	51
theta12	17979	600;	13,13;200	413	358	444	3.7-7	8.4-7	8.5-7	9.9-7	5.6-8	2.1-6	1.0-6	1.1-6	51	1:24	1:02	1:38
theta123	90020	600;	12,12;70	157	105	325	7.1-7	9.2-7	9.2-7	9.7-7	1.9-8	-6.8-8	2.2-7	2.5-6	36	35	34	1:19
hamming-9-8	2305	512;	12,12;200	2635	3129	1276	2.5-7	9.8-7	9.7-7	9.4-7	-9.2-8	-1.2-5	5.6-7	-5.8-6	21	3:17	4:07	2:02
hamming-10-2	23041	1024;	10,10;200	667	731	1066	4.8-7	6.9-7	9.6-7	9.9-7	-5.9-7	-1.5-5	2.2-6	2.5-5	2:54	7:19	9:39	12:05
hamming-9-5-6	53761	512;	6, 6;200	1022	1215	197	6.2-7	8.8-7	9.8-7	8.8-7	-1.7-7	-1.1-5	-1.8-6	-5.1-6	20	1:24	1:36	20
G43	9991	1000;	27,27;200	1237	1097	962	9.8-7	9.4-7	9.8-7	9.9-7	8.9-7	-3.5-6	-1.8-6	2.0-6	3:32	9:54	9:13	12:49
G44	9991	1000;	30,30;200	1236	1110	996	6.1-7	9.7-7	9.3-7	9.9-7	5.9-9	-3.6-6	-8.8-7	1.6-6	3:48	9:57	9:15	13:17
G45	9991	1000;	26,26;200	1261	1120	1007	6.4-7	9.9-7	9.6-7	9.9-7	3.8-8	3.2-6	1.8-6	1.6-6	3:31	10:04	9:21	13:35
G46	9991	1000;	28,30;200	1284	1142	974	7.9-7	9.6-7	9.9-7	9.9-7	6.2-8	-3.1-6	-1.6-6	1.3-6	3:49	10:11	9:21	12:55
G47	9991	1000;	30,31;200	1267	1088	1030	3.7-7	9.3-7	9.5-7	9.9-7	1.5-7	2.8-6	8.9-7	1.2-6	3:52	9:59	8:51	13:47
G51	5910	1000;	148,584;200	6151	10210	8746	7.3-7	9.9-7	9.9-7	9.9-7	-7.9-8	-1.7-7	8.6-8	1.9-7	41:06	53:54	1:33:48	1:55:48
G52	5917	1000;	458,1619;200	25000	25000	25000	9.3-7	1.6-6	3.5-6	2.9-6	2.0-6	7.1-6	1.4-5	1.5-5	3:53:19	3:24:55	3:45:22	5:38:42
G53	5915	1000;	425,1183;200	25000	25000	25000	9.9-7	1.5-6	3.7-6	3.7-6	1.4-6	5.9-6	1.5-5	1.8-5	2:16:35	3:12:47	3:24:18	5:30:23
G54	5917	1000;	123,462;200	3892	5633	5398	9.3-7	9.9-7	9.9-7	9.9-7	4.2-8	-2.8-6	3.3-7	3.2-7	23:17	33:11	49:11	1:13:51
1dc.1024	24064	1024;	48,74;200	5077	9728	15069	9.1-7	9.9-7	9.9-7	9.9-7	3.4-6	4.2-6	4.9-6	4.0-6	14:32	50:49	1:31:42	3:03:20
1et.1024	9601	1024;	64,129;200	3956	10174	17252	5.9-7	9.9-7	9.9-7	9.9-7	2.2-6	3.1-6	3.0-6	2.7-6	35:33	50:14	1:45:03	3:53:32
1tc.1024	7937	1024;	156,417;200	5775	25000	18474	7.5-7	9.9-7	2.3-6	9.9-7	8.9-7	3.8-6	3.3-6	2.3-6	1:22:24	54:35	4:13:43	3:47:44
1zc.1024	16641	1024;	16,16;200	884	734	4488	8.4-7	9.7-7	9.2-7	9.6-7	2.7-8	6.9-7	1.6-6	2.4-5	4:56	12:44	8:49	49:10
2dc.1024	169163	1024;	148,376;200	6951	14316	23007	9.2-7	9.9-7	9.9-7	9.9-7	7.8-6	3.1-5	2.6-5	2.6-5	2:21:20	1:24:20	4:20:36	5:32:26
1dc.2048	58368	2048;	62,112;200	5520	12938	20527	9.7-7	9.9-7	9.9-7	9.9-7	5.5-6	6.6-6	6.7-6	6.8-6	1:55:31	6:10:05	14:00:08	28:29:05
1et.2048	22529	2048;	228,658;200	3601	13985	25000	8.7-7	9.9-7	9.9-7	**6.1-3**	3.9-6	4.3-6	4.2-6	**2.2-2**	5:29:46	3:59:47	17:25:13	40:08:45
1tc.2048	18945	2048;	509,1725;200	6574	20819	25000	8.2-7	9.9-7	9.9-7	1.2-6	9.9-7	4.8-6	3.8-6	5.5-6	22:10:45	7:26:56	23:37:33	39:35:04
2dc.2048	504452	2048;	167,385;200	6293	25000	16945	9.8-7	9.9-7	3.7-6	9.9-7	1.8-5	2.0-5	3.5-5	2.8-5	14:22:06	8:52:47	47:15:41	28:09:34
nonsym(8,4)	46655	512;	13,13;200	6927	6871	25000	1.5-7	8.7-7	9.9-7	**1.9-2**	-4.1-7	5.7-6	2.1-5	**6.2-1**	29	8:40	8:54	1:10:07
nonsym(9,4)	91124	729;	25,33;200	25000	4584	25000	4.2-7	3.2-5	9.7-7	**1.9-2**	7.4-6	**5.2-4**	-1.7-5	**4.2-3**	2:00	1:18:21	14:29	2:41:00
nonsym(10,4)	166374	1000;	17,21;200	25000	6711	25000	7.9-7	2.8-5	9.9-7	**1.6-2**	-2.5-6	**-1.9-4**	2.6-5	**6.2-1**	3:37	2:57:30	48:32	5:52:44
nonsym(11,4)	287495	1331;	19,22;200	25000	16627	25000	2.5-7	**1.3-3**	9.9-7	**9.9-3**	7.5-6	**5.1-2**	3.0-5	**-2.2-1**	7:14	6:30:10	4:57:08	12:51:58
nonsym(5,5)	50624	625;	30,31;200	12638	4918	25000	2.4-7	9.6-7	9.9-7	**2.5-3**	3.7-6	9.0-6	1.8-5	**-4.8-2**	1:18	27:19	10:10	1:50:10
nonsym(6,5)	194480	1296;	26,28;200	25000	11981	25000	6.6-7	**1.6-4**	9.9-7	**1.6-3**	2.0-5	**-3.0-3**	-2.8-5	**-4.9-2**	6:39	5:57:28	2:59:30	11:24:24
sym_rd(3,35)	82250	666;	43,46;72	2964	2812	11937	5.4-7	9.7-7	9.9-7	9.5-7	-8.0-6	-1.3-6	1.8-5	-1.9-6	1:27	7:54	7:07	1:09:55
sym_rd(3,40)	135750	861;	37,39;82	3736	3356	25000	5.7-7	9.4-7	9.9-7	**3.9-3**	-1.6-5	-1.2-6	-2.1-5	**1.3-1**	2:18	19:34	16:55	4:37:17
sym_rd(3,45)	211875	1081;	31,31;87	4689	4498	25000	5.5-7	9.3-7	9.9-7	**5.5-3**	5.7-6	-3.5-6	-3.9-5	**-1.5-1**	4:31	46:05	41:11	8:09:46
sym_rd(3,50)	316250	1326;	37,37;87	4432	4161	25000	7.6-7	9.1-7	9.6-7	**3.2-3**	2.4-5	-3.7-6	4.2-5	**1.1-1**	8:24	1:13:12	1:07:37	13:45:47
sym_rd(4,35)	73814	630;	46,51;77	969	3400	25000	2.2-7	9.9-7	9.7-7	**5.1-4**	6.2-6	-4.2-6	-4.2-5	**-1.7-2**	1:45	4:44	9:10	1:58:12
sym_rd(4,40)	123409	820;	60,76;86	447	761	2396	6.6-7	9.9-7	9.9-7	9.9-7	-7.7-6	-1.3-5	-1.3-5	-1.3-5	4:56	3:37	7:06	21:58
sym_rd(4,45)	194579	1035;	45,62;90	462	737	2569	3.5-7	9.9-7	9.9-7	9.9-7	-4.7-6	-1.4-5	-1.4-5	-1.4-5	7:44	6:56	12:42	43:48
sym_rd(4,50)	292824	1275;	45,62;91	466	758	2824	7.9-7	9.9-7	9.9-7	9.9-7	-1.3-5	-1.6-5	-1.6-5	-1.6-5	12:44	12:11	22:43	1:21:38
sym_rd(5,15)	54263	816;	41,43;111	1549	1980	25000	2.4-7	8.8-7	9.7-7	**3.2-4**	8.2-6	7.1-5	-3.7-5	**1.6-2**	2:53	6:46	7:53	3:28:56
sym_rd(5,20)	230229	1771;	29,35;139	2832	3563	25000	3.1-7	9.3-7	9.8-7	**5.5-3**	-1.5-5	9.7-5	**-1.0-4**	**-3.1-1**	27:28	1:56:14	2:22:52	27:04:51

Table 5: Performance of SDPNAL+ (a), ADMM+ (b), SDPAD (c) and 2EBD (d) on θ and R1TA problems ($\varepsilon = 10^{-6}$)

	iteration	η	η_g	time
problem | m | $n_s; n_l$	a| b| c| d	a| b| c| d	a| b| c| d	a| b| c| d
sym_rd(6,15) | 38759 | 680;	32,35;137 | 1358 | 1652 | 13111	6.4-7 | 9.3-7 | 9.9-7 | 9.9-7	2.6-5 | 5.2-5 | -6.9-5 | -6.3-7	1:33 | 3:52 | 4:26 | 1:12:00
sym_rd(6,20) | 177099 | 1540;	26,37;185 | 3280 | 3576 | 25000	6.4-7 | 9.7-7 | 9.6-7 | **3.0-4**	3.0-6 | **1.5-4** | **-1.1-4** | **-2.5-2**	22:48 | 1:34:25 | 1:43:31 | 17:41:06
nsym_rd([20,25,25]) | 68249 | 500;	47,51;66 | 5883 | 6641 | 25000	6.0-7 | 9.8-7 | 9.7-7 | **6.1-3**	1.0-5 | -3.0-5 | 2.5-5 | **-1.2-1**	58 | 7:55 | 8:10 | 1:12:25
nsym_rd([25,20,25]) | 68249 | 500;	49,50;77 | 6113 | 6912 | 25000	7.0-7 | 9.9-7 | 9.9-7 | **5.5-3**	-1.0-5 | -3.0-5 | -1.3-5 | **-8.7-2**	58 | 7:26 | 8:25 | 1:11:35
nsym_rd([25,25,20]) | 68249 | 500;	14,14;129 | 6252 | 6898 | 25000	9.0-8 | 9.9-7 | 9.9-7 | **2.8-3**	-1.6-6 | -5.9-6 | 2.7-5 | **5.2-2**	34 | 8:11 | 8:31 | 1:10:17
nsym_rd([25,25,25]) | 105624 | 625;	56,64;95 | 11250 | 3705 | 25000	9.0-7 | 9.9-7 | 9.9-7 | **6.6-4**	-1.7-5 | -4.9-6 | -1.3-5 | **1.3-2**	1:33 | 26:20 | 8:09 | 2:04:05
nsym_rd([30,30,30]) | 216224 | 900;	30,33;122 | 25000 | 4829 | 25000	6.2-7 | 2.0-5 | 9.9-7 | **4.9-3**	2.7-6 | **5.5-4** | 3.2-5 | **1.5-1**	3:52 | 2:21:43 | 26:48 | 5:07:41
nsym_rd([35,35,35]) | 396899 | 1225;	45,49;141 | 25000 | 8788 | 25000	2.0-7 | **1.4-3** | 9.9-7 | **1.0-3**	-3.5-6 | **4.1-3** | -4.2-5 | **3.6-3**	9:14 | 5:18:55 | 1:57:42 | 11:13:11
nsym_rd([40,40,40]) | 672399 | 1600;	33,35;93 | 25000 | 25000 | 25000	3.3-7 | **1.1-4** | **3.7-4** | **5.1-4**	1.0-5 | **5.0-3** | **1.3-2** | **-1.9-2**	14:23 | 12:12:03 | 13:56:21 | 22:41:13
nsym_rd([8,8,8,8]) | 46655 | 512;	11,11;200 | 5325 | 5865 | 25000	3.3-7 | 9.4-7 | 9.9-7 | **1.2-3**	-6.6-6 | 6.9-6 | -1.3-5 | **1.3-2**	28 | 7:09 | 7:13 | 1:12:01
nsym_rd([9,9,9,9]) | 91124 | 729;	14,14;200 | 21833 | 4073 | 25000	3.1-7 | 9.0-7 | 9.6-7 | **1.6-2**	6.9-6 | -7.1-6 | -3.1-5 | **-2.5-1**	1:07 | 1:12:18 | 12:47 | 2:52:40
nonsym(12,4) | 474551 | 1728;	5,17;200 | 16473 | 25000 | 25000	2.8-8 | 8.8-7 | **1.2-2** | **1.2-2**	-1.1-7 | -2.7-6 | **-7.7-2** | **5.8-1**	16:55 | 9:04:03 | 15:26:14 | 24:24:33
nonsym(13,4) | 753570 | 2197;	15,55;200 | 25000 | 25000 | 25000	5.2-7 | **5.4-4** | **9.1-3** | **1.9-2**	-1.8-7 | **3.6-2** | **-1.6-1** | **2.7-1**	1:51:34 | 29:11:11 | 32:02:36 | 54:30:04
nonsym(7,5) | 614655 | 2401;	32,43;200 | 25000 | 25000 | 25000	2.4-7 | **1.4-3** | **1.2-2** | **1.8-2**	8.7-6 | **-5.7-2** | **-1.2-1** | **-1.1-1**	53:29 | 38:36:39 | 43:46:36 | 67:40:52
nonsym(8,5) | 1679615 | 4096;	14,22;200 | 12791 | 10732 | 7851	5.2-7 | **1.2-3** | **1.3-2** | **1.2-2**	-5.5-6 | **-3.7-2** | **2.5-1** | **-4.3-1**	2:46:20 | 99:00:46 | 99:01:08 | 99:03:37
nonsym(18,4) | 5000210 | 5832;	13,55;200 | 8748 | 7962 | 7017	5.8-7 | **3.5-4** | **7.8-3** | **1.4-2**	2.6-5 | **-1.3-2** | **-3.5-1** | **3.8-1**	8:50:14 | 99:02:10 | 99:01:13 | 99:05:37
nonsym(20,4) | 9260999 | 8000;	7,17;200 | 3231 | 3031 | 2645	7.4-8 | **4.7-4** | **9.6-3** | **2.2-2**	-6.4-6 | **5.7-2** | **-4.4-1** | **-3.5-1**	8:26:40 | 99:07:11 | 99:03:17 | 99:16:28
nonsym(21,4) | 12326390 | 9261;	7,21;200 | 1918 | 1904 | 1792	5.7-8 | **2.7-4** | **9.8-3** | **5.2-3**	-1.2-6 | **2.6-3** | **5.0-1** | **9.4-1**	14:22:05 | 99:09:25 | 99:05:25 | 99:29:18