https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–36
ISSN 0364-765X (print), ISSN 1526-5471 (online)

A Squared Smoothing Newton Method for Semidefinite Programming

Ling Liang,[a,*] **Defeng Sun,**[b] **Kim-Chuan Toh**[c]

[a] Department of Mathematics, University of Maryland, College Park, Maryland 20742; [b] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom 999077, Hong Kong; [c] Department of Mathematics and Institute of Operations Research and Analytics, National University of Singapore, Singapore 119076

*Corresponding author

Contact: liang.ling@u.nus.edu, https://orcid.org/0000-0003-0671-9561 (LL); defeng.sun@polyu.edu.hk, https://orcid.org/0000-0003-0481-272X (DS); mattohkc@nus.edu.sg, https://orcid.org/0000-0001-7204-8933 (K-CT)

Received: October 6, 2023
Revised: July 2, 2024; October 7, 2024
Accepted: October 15, 2024
Published Online in Articles in Advance: November 12, 2024

MSC2020 Subject Classifications: Primary: 90C06; Secondary: 90C22

https://doi.org/10.1287/moor.2023.0311

Abstract. This paper proposes a squared smoothing Newton method via the Huber smoothing function for solving semidefinite programming problems (SDPs). We first study the fundamental properties of the matrix-valued mapping defined upon the Huber function. Using these results and existing ones in the literature, we then conduct rigorous convergence analysis and establish convergence properties for the proposed algorithm. In particular, we show that the proposed method is well-defined and admits global convergence. Moreover, under suitable regularity conditions, that is, the primal and dual constraint nondegenerate conditions, the proposed method is shown to have a super-linear convergence rate. To evaluate the practical performance of the algorithm, we conduct extensive numerical experiments for solving various classes of SDPs. Comparison with the state-of-the-art SDP solvers demonstrates that our method is also efficient for computing accurate solutions of SDPs.

Funding: The research of D. Sun was supported in part by the Hong Kong Research Grants Council under Grant 15307523, and the research of K.-C. Toh was supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 3 grant call [MOE-2019-T3-1-010].

Keywords: semidefinite programming • smoothing Newton method • Huber function • nondegeneracy

1. Introduction

A standard primal linear semidefinite programming (SDP) problem is the problem of minimizing a linear function in the space $\mathbb{S}^n$ subject to m linear equality constraints and the essential positive semidefinite constraint $X \in \mathbb{S}^n_+$. Mathematically, the primal SDP has its standard form:

$$\min_{X \in \mathbb{S}^n} \langle C, X \rangle \quad \text{s.t.} \quad \mathcal{A}X = b,\ X \in \mathbb{S}^n_+, \tag{1}$$

where $C \in \mathbb{S}^n$, $b \in \mathbb{R}^m$ are given data, and $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a linear mapping given by $\mathcal{A}X := (\langle A_1, X \rangle \cdots \langle A_m, X \rangle)^T$ for all $X \in \mathbb{S}^n$, with given matrices $A_i \in \mathbb{S}^n$, for $i = 1, \ldots, m$, and $\langle \cdot, \cdot \rangle$ denoting the standard trace inner product. Note that the adjoint of $\mathcal{A}$, denoted by $\mathcal{A}^*$, is a linear mapping from $\mathbb{R}^m$ to $\mathbb{S}^n$ defined as $\mathcal{A}^* y := \sum_{i=1}^m y_i A_i$ for $y \in \mathbb{R}^m$. Associated with problem (1), the Lagrangian dual problem of (1) is given by

$$\max_{y \in \mathbb{R}^m, Z \in \mathbb{S}^n} \langle b, y \rangle \quad \text{s.t.} \quad \mathcal{A}^* y + Z = C,\ Z \in \mathbb{S}^n_+. \tag{2}$$

In this paper, we assume that the primal problem (1) admits at least one optimal solution and satisfies the Slater's condition; that is, there exists $\tilde{X} \in \mathbb{S}^n_{++}$ such that $\mathcal{A}\tilde{X} = b$. Under these assumptions, the dual problem (2) has an optimal solution, and the dual optimal value is equal to the primal optimal value.[1] Thus, the duality gap between (1) and (2) is zero. As a consequence, the following system of KKT optimality conditions, given as

$$\mathcal{A}X = b, \quad \mathcal{A}^* y + Z = C, \quad X \in \mathbb{S}^n_+, Z \in \mathbb{S}^n_+, \langle X, Z \rangle = 0, \tag{3}$$

admits a solution. We call an arbitrary triple $(\overline{X}, \overline{y}, \overline{Z}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ a KKT point if it satisfies the KKT conditions in (3). Let $\overline{X}$ be an optimal solution to the primal problem (1); for simplicity, we denote the set of associated Lagrangian multipliers $\mathcal{M}(\overline{X})$ as

$$\mathcal{M}(\overline{X}) := \{(y, Z) \in \mathbb{R}^m \times \mathbb{S}^n : (\overline{X}, y, Z) \text{ is a KKT point}\}. \tag{4}$$

Then, under the aforementioned conditions, $\mathcal{M}(\overline{X})$ is a nonempty set.

The research on SDPs has been active for decades and still receives strong attention to date. Indeed, SDP has become one of the fundamental modeling and optimization tools that encompasses a wide range of applications in different fields. The increasing interest in SDP has resulted in fruitful and impressive works in the literature. For theoretical developments and many important applications of SDP in engineering, finance, optimal control, statistics, machine learning, combinatorics, and beyond, we refer the reader to Bonnans and Shapiro [3], Chan and Sun [11], Majumdar et al. [33], Sun [57], Vandenberghe and Boyd [66], and Wolkowicz et al. [70], to mention just a few. In the next few paragraphs, we will briefly review some influential works on developing efficient solution methods, including interior point methods (IPMs), first-order methods (FOMs), augmented Lagrangian methods (ALMs), and smoothing Newton methods, for solving SDPs.

Perhaps the most notable and reliable algorithms for solving SDPs are based on the IPMs; see Todd [63] and Wolkowicz et al. [70] for comprehensive surveys.[2] IPMs admit polynomial worst-case complexity and are able to deliver highly accurate solutions. The essential idea of a primal-dual IPM is to solve the KKT conditions in (3) indirectly by solving a sequence of perturbed KKT conditions:

$$\mathcal{A}X = b, \quad \mathcal{A}^*y + Z = C, \quad XZ = \mu I, \quad \mu > 0,$$

with μ being driven to zero. At each iteration of the IPM, one usually needs to solve a linear system whose coefficient matrix (i.e., the Schur complement matrix) is of size $m \times m$ and generally fully dense and ill-conditioned. Thus, IPMs may not be suitable for some practical applications because SDPs arising from such applications are typically of large size. To alleviate these issues, many modifications and extensions of IPMs have been developed. Two main approaches are (a) investigating iterative linear system solvers for computing the search directions (Toh [64], Toh and Kojima [65]) in order to handle large-scale linear systems and (b) exploiting the underlying sparsity structures (such as the chordal structure of the aggregate sparsity pattern) in the SDP data for more efficiency (Kim et al. [29], Kočvara [31]). For (a), we note that iterative linear system solvers (such as the preconditioned conjugate gradient (PCG) method) require carefully designed preconditioners to handle the issue of ill-conditioning. However, finding an effective preconditioner is still a challenging task. For (b), we note that not all SDPs admit appealing sparsity structures, and hence, the range of applications for this approach remains limited. In summary, there exist several critical issues that may stop one from applying an IPM and its variants for solving large-scale[3] SDPs in real-world applications.

FOMs are at the forefront on developing scalable algorithms for solving large-scale optimization problems because of the low per-iteration complexity. For SDPs, we mention some relevant FOMs to capture the picture on recent developments along this direction. In the early 21st century, Helmberg and Rendl [21] applied a spectral bundle method to solve a special regularized problem for the dual SDP (2). These authors proved the convergence of the proposed method under the condition that the trace of any primal optimal solution is fixed. FOMs such as the (accelerated) proximal gradient method are also popular; see, for instance, the works of Jiang et al. [25] and Yang et al. [72] together with some numerical evaluations of their practical performance. Later Renegar [47], proposed an FOM that transforms the SDP into an equivalent convex optimization problem. But no numerical experiment was conducted in this work, and hence, the practical performance of the proposed algorithm is unclear. Meanwhile, operator splitting methods and the alternating direction method of multipliers (ADMM) and their variants (Povh et al. [41], Sun et al. [62], Zhao et al. [75], Zheng et al. [76]) have been demonstrated to be well suited for solving large-scale SDPs, although high accuracy solutions may not be achievable in general. More recently, in order to design storage-efficient algorithms, Yurtsever et al. [73] proposed a conditional-gradient-based augmented Lagrangian method equipped with random sketching techniques for solving SDPs with fixed trace constraints. The method has state-of-the-art performance in memory consumption, but whether it is also efficient in computational time for obtaining high-accuracy solutions of general SDPs (with or without fixed trace constraints) requires further investigation. Another similar storage-optimal framework can be found in Ding et al. [17]. For sparse SDPs, Jiang and Vandenberghe [24] recently proposed a Bregman distance-based Chambolle-Pock primal-dual algorithm (note that the classical Chambolle-Pock algorithm is an equivalent form of the so-called linearized ADMM; see Chambolle and Pock [10]) with provable convergence properties, which exploits data sparsity to reduce the computational cost for evaluating the Bregman proximal operator. Indeed, those existing works have shown that FOMs are scalable algorithms. However, a commonly accepted fact for FOMs is that they may not have favorable efficiency for computing accurate solutions, or they may even fail to deliver moderately accurate solutions. Therefore, when high-quality solutions are needed, FOMs may not be attractive.

Being studied for decades, augmented Lagrangian methods (ALMs) have been shown to be suitable for solving large-scale optimization problems (including SDPs) efficiently and accurately. Many efficient algorithms based on the ALM have been proposed. For example, Jarre and Rendl [23] developed an augmented primal-dual

method. Later, Malick et al. [34] proposed a Moreau-Yosida regularization method for the primal SDP (1). Both methods perform reasonably well on some SDPs with a relatively large number of linear constraints. Zhao et al. [75] developed a dual-based ALM (i.e., the ALM is applied to the dual problem (2)) whose design principle is fundamentally different from the one in Malick et al. [34] by relying on the deep connection between the primal proximal point algorithm and the dual-based ALM (Rockafellar [52], Rockafellar [53]) and the highly efficient semismooth Newton method (Qi and Sun [44]). The solver developed in Zhao et al. [75] has shown very promising practical performance for a large collection of SDPs compared with existing ones. Moreover, practical experience shows that when a good initial point is available, the performance of the algorithm can be further improved. This has motivated the same group of researchers to develop a hybrid framework, namely `SDPNAL+` (Yang et al. [71]), which combines the dual-based ALM (Zhao et al. [75]) and the ADMM method (Sun et al. [62]) with some majorization techniques. `SDPNAL+` has demonstrated excellent numerical performance, particularly when solving SDPs with large numbers of constraints (say more than 20,000), and it can handle additional polyhedral constraints. Lastly, we mention another popular ALM-based approach for solving low-rank SDPs, namely the Burer-Monteiro (BM) method (Burer and Monteiro [7]). The key ingredient of the BM method is to replace the essential constraint $X \in \mathbb{S}^n_+$ with the low-rank factorization $X = RR^T$ that reformulates (1) as a nonconvex optimization problem in $\mathbb{R}^{n\times r}$, where r is a prior bound on the rank of an optimal solution. Following the same research theme of Burer and Monteiro [7], recent works (Wang and Hu [67], Wang et al. [68]) demonstrate that one can expect more promising practical performance for solving low-rank SDPs via incorporating the matrix factorization technique with manifold optimization. Moreover, many important works on the connection between the factorized problem and the primal SDP (1) have also been published; see, for instance, Boumal et al. [4], Burer and Monteiro [8], and Pataki [37]. However, choosing a suitable r to balance a conservative (large) and an aggressive (small) choice is a delicate task because the former will lead to higher computational cost per iteration, whereas the latter may lead to convergence failure. As a consequence, the practical performance of the BM method is sensitive to the choice of r.

Different from IPMs, one could also consider solving the non-smooth reformulation of the KKT conditions in (3):

$$\mathcal{F}(X,y,Z) := \begin{pmatrix} \mathcal{A}X - b \\ -\mathcal{A}^* y - Z + C \\ X - \Pi_{\mathbb{S}^n_+}(X - Z) \end{pmatrix} = 0, \quad (X,y,Z) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n, \tag{5}$$

where the conditions $X \in \mathbb{S}^n_+$, $Z \in \mathbb{S}^n_+$, $\langle X, Z\rangle = 0$ are replaced by a single non-smooth equation $X - \Pi_{\mathbb{S}^n_+}(X - Z) = 0$, and $\Pi_{\mathbb{S}^n_+}(\cdot)$ is the projection operator onto $\mathbb{S}^n_+$. Obviously, the function $\mathcal{F}(\cdot)$ is non-smooth because of the non-smoothness of $\Pi_{\mathbb{S}^n_+}(\cdot)$. Thus, the classical Jacobian of $\mathcal{F}(\cdot)$ is not well-defined, and the classical Newton method is not applicable for solving (5). Because the B-subdifferential and the Clarke's generalized Jacobian (Clarke [16]) of $\Pi_{\mathbb{S}^n_+}(\cdot)$, namely, $\partial_B \Pi_{\mathbb{S}^n_+}$ and $\partial \Pi_{\mathbb{S}^n_+}$, are well-defined and $\Pi_{\mathbb{S}^n_+}(\cdot)$ is strongly semismooth (Sun and Sun [60]), $\partial_B \mathcal{F}$ and $\partial \mathcal{F}$ are both well-defined, and $\mathcal{F}(\cdot)$ is also strongly semismooth. Then, it is natural to apply the semismooth Newton (SSN) method (Qi and Sun [44]) that has a fast local convergence rate under suitable regularity conditions. However, the global convergence of the SSN method needs a valid merit function for which the line search procedure for computing a step size is well-defined. Typically, such a valid merit function should be continuously differentiable or satisfy other stronger conditions. Yet defining such a merit function from $\mathcal{F}(\cdot)$ directly turns out to be challenging. This motivates us to develop a smoothing Newton method because a suitable merit function can readily be obtained, and its global convergence can be established. Moreover, the smoothing Newton method inherits the strong local convergence properties from the classical Newton method and the semismooth Newton method under similar regularity conditions.

Existing smoothing Newton methods have been developed and studied extensively in different areas. To mention just a few of these works, the reader is referred to Chan and Sun [11], Chen and Tseng [13], Chen et al. [14], Chen et al. [15], Gao and Sun [18], Kanzow and Pieper [28], Qi and Chen [43], and Qi et al. [45]. These smoothing methods can be roughly divided into two groups, Jacobian smoothing Newton methods (Chen et al. [14], Kanzow and Pieper [28]), and squared smoothing Newton methods (Chan and Sun [11], Gao and Sun [18], Qi et al. [45]). The convergence analysis of Jacobian smoothing Newton methods depends strongly on the so-called Jacobian consistency property and other strong conditions for the underlying smoothing functions. However, many smoothing functions, such as those functions defined via normal maps (Robinson [50]), do not satisfy these conditions. Furthermore, to get stronger convergence results, it is often useful to add a small perturbation term to the smoothing function. In this case, conditions ensuring those stronger results for the Jacobian smoothing Newton methods are generally not satisfied.

Smoothing Newton methods rely on an appropriate smoothing function for the plus function

$$\rho(t) := \max\{0, t\}, \quad t \in \mathbb{R},$$

in order to deal with the non-smooth projector $\Pi_{\mathbb{S}^n_+}(\cdot)$. To the best of our knowledge, the most notable and commonly used smoothing function is the so-called Chen-Harker-Kanzow-Smale (CHKS) function (Chen and Harker [12], Kanzow [26], Smale [56]),

$$\xi(\epsilon, t) = (\sqrt{t^2 + 4\epsilon^2} + t)/2, \quad (\epsilon, t) \in \mathbb{R} \times \mathbb{R},$$

which has been studied extensively and used for solving SDPs and semidefinite linear and nonlinear complementarity problems. For other smoothing functions whose properties have also been well-studied, the reader is referred to Qi et al. [45] and Sun and Qi [58] for more details. However, it is easy to see that $\xi(\epsilon, t)$ maps any negative number t to a positive one when $\epsilon \neq 0$, even though $\rho(t) = 0$. Thus, it destroys the possible sparsity structure when evaluating the Jacobian of the merit function. Hence, smoothing Newton methods based on the CHKS function would require more computational effort when performing the matrix-vector multiplications needed in an iterative linear system solver for computing the search directions; see Section 4.3 for more detailed discussions. To resolve this issue, we propose to use the Huber smoothing function (Pinar and Zenios [40]), which maps any negative number to zero so that the underlying sparsity structure of the plus function is inherited. To use the Huber smoothing function, we then need to study some fundamental properties, including continuity, differentiability, and (strong) semi-smoothness of the matrix-valued mapping associated with the Huber function.

Although much progress has been made in developing efficient algorithms for solving SDPs in the literature, the efficiency and cost of these methods are still far from satisfactory. This motivates us to study and analyze the squared smoothing Newton method via the Huber smoothing function. To evaluate the efficiency, we implement the algorithm and conduct extensive numerical experiments by solving several classes of SDPs. Our theoretical analysis and numerical results show that the proposed method admits appealing convergence properties, and it has shown promising numerical performance for computing relatively accurate solutions. Compared with the ALM-based method in Zhao et al. [75] for solving an SDP problem, our smoothing Newton method has the advantage that it can also handle a relatively large number of linear constraints and has the feature that it is applied to a single nonlinear system of equations, whereas the ALM method needs to solve a sequence of subproblems for which the total number of Newton iterations is usually more than the former. Moreover, because the practical performance of the smoothing Newton method for solving SDPs has not been systematically evaluated in the literature, our theoretical and numerical studies can certainly help one to gain a better understanding of this class of algorithms. However, note that a comprehensive comparison of the practical performance of different smoothing functions is not the main focus of this research.

The rest of the paper is organized as follows. In Section 2, we summarize some existing results that will be used in later analysis. Then, we study the continuity, differentiability, and semi-smoothness of the matrix-valued mapping defined upon the Huber function in Section 3. Using these results, we are able to analyze the correctness, global convergence, and local fast convergence rate of the proposed squared smoothing Newton method in Section 4. In Section 5, we conduct numerical experiments for solving various classes of SDPs and evaluate the practical performance of the proposed method. Finally, we conclude the paper in Section 6.

2. Preliminary

Let $\mathbb{Y}$ and $\mathbb{Z}$ be any two finite dimensional real vector spaces, each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$. Assume that $\mathcal{O} \subseteq \mathbb{Y}$ is an open set and $\Theta : \mathcal{O} \to \mathbb{Z}$ is a function on $\mathcal{O}$. If Θ is locally Lipschitz continuous on $\mathcal{O}$, then it is Fréchet-differentiable almost everywhere on $\mathcal{O}$ (Rockafellar and Wets [54]), that is, for almost every $y \in \mathcal{O}$, there exists a linear operator depending on y, denoted as $\Theta'(y) : \mathbb{Y} \to \mathbb{Z}$, such that

$$\lim_{\|\Delta y\| \to 0} \frac{\|\Theta(y + \Delta y) - \Theta(y) - \Theta'(y)\Delta y\|}{\|\Delta y\|} = 0,$$

and $\Theta'(y)$ is called the F-derivative of Θ at $y \in \mathcal{O}$. Hence, the B-subdifferential (Qi [42]) of Θ at $y \in \mathcal{O}$, denoted by $\partial_B \Theta(y)$, is well-defined and given by

$$\partial_B \Theta(y) := \left\{ V : V = \lim_{k \to \infty} \Theta'(y^k), y^k \to y, \Theta'(y^k) \text{ is the F-derivative of } \Theta \text{ at } y^k \right\}.$$

Moreover, Clarke's generalized Jacobian (Clarke [16]) of Θ at $y \in \mathcal{O}$ is defined as $\partial\Theta(y) := \mathrm{conv}\{\partial_B\Theta(y)\}$, where "conv" denotes the convex hull of the underlying set. From Clarke [16], we know that $\partial_B\Theta(\cdot)$ is nonempty and upper semicontinuous for all $y \in \mathcal{O}$. Furthermore, if $\Theta'(y)$ exists for a certain $y \in \mathcal{O}$, then $\Theta'(y) \in \partial_B\Theta(y)$. The above statements hold true if one replaces $\partial_B\Theta(y)$ with $\partial\Theta(y)$.

2.1. Properties of $\Pi_{\mathbb{S}^n_+}(\cdot)$

Denote the matrix-valued mapping $\Omega_0 : \mathbb{R}^N \to \mathbb{S}^N$ for any integer $n > 0$ as

$$[\Omega_0(d)]_{ij} = \frac{\rho(d_i) + \rho(d_j)}{|d_i| + |d_j|}, \quad i,j = 1,\ldots,N, \quad \forall d \in \mathbb{R}^N, \tag{6}$$

with the convention that $0/0 := 1$. Let the spectral decomposition of $W \in \mathbb{S}^n$ be

$$W = PDP^T, \quad D = \mathrm{diag}(d), \quad d = (d_1,\ldots,d_n)^T \in \mathbb{R}^n, \quad d_1 \geq \cdots \geq d_n, \tag{7}$$

where $P \in \mathbb{R}^{n\times n}$ is orthogonal. For later usage, we also define the following three index sets: $\alpha := \{i : d_i > 0\}$, $\beta := \{i : d_i = 0\}$, and $\gamma := \{i : d_i < 0\}$, corresponding to the positive, zero, and negative eigenvalues of W, respectively.

It is clear that both $\partial_B\Pi_{\mathbb{S}^n_+}(W)$ and $\partial\Pi_{\mathbb{S}^n_+}(W)$ are well-defined for any $W \in \mathbb{S}^n$ (because $\Pi_{\mathbb{S}^n_+}(\cdot)$ is globally Lipschitz continuous with modulus 1 on $\mathbb{S}^n$). We then have the following useful lemma (see (17)) on $\partial_B\Pi_{\mathbb{S}^n_+}(W)$ and $\partial\Pi_{\mathbb{S}^n_+}(W)$, whose proof can be found in Sun [57, proposition 2.2]. Here we use "$\circ$" to denote the Hadamard product.

Lemma 1. *Suppose that $W \in \mathbb{S}^n$ has the spectral decomposition (7). Then, $V \in \partial_B\Pi_{\mathbb{S}^n_+}(W)$ (respectively, $\partial\Pi_{\mathbb{S}^n_+}(W)$) if and only if there exists $V_{|\beta|} \in \partial_B\Pi_{\mathbb{S}^{|\beta|}_+}(0)$ (respectively, $\partial\Pi_{\mathbb{S}^{|\beta|}_+}(0)$) such that*

$$V(H) = P\begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}^T_{\alpha\beta} & V_{|\beta|}(\tilde{H}_{\beta\beta}) & 0 \\ H^T_{\alpha\gamma} \circ [\Omega_0(d)]^T_{\alpha\gamma} & 0 & 0 \end{pmatrix} P^T,$$

where $\tilde{H} := P^THP$ for any $H \in \mathbb{S}^n$. In particular, $V_{|\beta|} \in \partial_B\Pi_{\mathbb{S}^{|\beta|}_+}(0)$ if and only if there exist an orthogonal matrix $U \in \mathbb{R}^{|\beta|\times|\beta|}$ and

$$\Omega_{|\beta|} \in \left\{\Omega \in \mathbb{S}^{|\beta|} : \Omega = \lim_{k\to\infty} \Omega_0(z^k), \mathbb{R}^{|\beta|} \ni z^k \to 0, z^k_1 \geq \cdots \geq z^k_{|\beta|}, z^k \neq 0\right\}$$

such that

$$V_{|\beta|}(Y) = U[\Omega_{|\beta|} \circ (U^TYU)]U^T, \quad \forall Y \in \mathbb{S}^{|\beta|}.$$

Another critical and fundamental concept of $\Pi_{\mathbb{S}^n_+}(\cdot)$ is its semi-smoothness, which plays an essential role in analyzing the convergence of Newton-type algorithms. Recall that a mapping $\Theta : \mathbb{Y} \to \mathbb{Z}$ is said to be directionally differentiable at the point $y \in \mathbb{Y}$ if the limit, defined by

$$\Theta(y;h) := \lim_{t\downarrow 0} \frac{\Theta(y+th) - \Theta(y)}{t},$$

exists for any $h \in \mathbb{Y}$. Then the definition of the semismoothness is given as follows.

Definition 1. Let $\Theta : \mathcal{O} \subseteq \mathbb{Y} \to \mathbb{Z}$ be a locally Lipschitz continuous function. Θ is said to be semismooth at $y \in \mathcal{O}$ if Θ is directionally differentiable at y and for any $V \in \partial\Theta(y+h)$,

$$\|\Theta(y+h) - \Theta(y) - Vh\| = o(\|h\|), \quad h \to 0.$$

Θ is said to be strongly semismooth at $y \in \mathbb{Y}$ if Θ is semismooth at y and for any $V \in \partial\Theta(y+h)$

$$\|\Theta(y+h) - \Theta(y) - Vh\| = O(\|h\|^2), \quad h \to 0.$$

We say that Θ is semismooth (respectively, strongly semismooth) if Θ is semismooth (respectively, strongly semismooth) at every $y \in \mathbb{Y}$.

Semi-smoothness was originally introduced in Mifflin [35] for functionals. Qi and Sun [44] extended it to vector-valued functions. It is also well known that the projector $\Pi_{\mathbb{S}^n_+}(\cdot)$ is strongly semismooth on $\mathbb{S}^n$ (Sun and

Sun [60]). Later, we will also show that the smoothed counterpart of $\Pi_{\mathbb{S}^n_+}(\cdot)$ is also strongly semismooth on $\mathbb{S}^n$ (see Proposition 1).

2.2. Equivalent Conditions

For the rest of this section, we describe several important conditions related to the theoretical analysis for SDPs and present some connections between these conditions. The presented materials are borrowed directly from the literature; see, for instance, Chan and Sun [11] and the references therein.

First, we recall the general concept of constraint nondegeneracy. Let $g : \mathbb{Y} \to \mathbb{Z}$ be a continuously differentiable function and $\mathcal{C}$ be a nonempty closed convex set in $\mathbb{Z}$. Consider the following feasibility problem:

$$g(\xi) \in \mathcal{C}, \quad \xi \in \mathbb{Y}. \tag{8}$$

Let $\overline{\xi} \in \mathbb{Y}$ be a feasible solution to the above feasibility problem. The tangent cone of $\mathcal{C}$ at the point $g(\overline{\xi})$ is denoted by $\mathcal{T}_{\mathcal{C}}(g(\overline{\xi}))$, and the largest linear subspace of $\mathbb{Y}$ contained in $\mathcal{T}_{\mathcal{C}}(g(\overline{\xi}))$ (i.e., the linearity space of $\mathcal{T}_{\mathcal{C}}(g(\overline{\xi}))$) is denoted by $\mathrm{lin}(\mathcal{T}_{\mathcal{C}}(g(\overline{\xi})))$.

Definition 2. A feasible solution $\overline{\xi}$ for (8) is constraint nondegenerate if

$$g'(\overline{\xi})\mathbb{Y} + \mathrm{lin}(\mathcal{T}_{\mathcal{C}}(g(\overline{\xi}))) = \mathbb{Z}.$$

The concept of degeneracy was introduced by Pereyra et al. [39] and Robinson [48], Robinson [49], and the term "constraint nondegeneracy" was also coined by Robinson [51]. For nonlinear programming (i.e., $\mathbb{Y}$ is the Euclidean space $\mathbb{R}^m$ and $\mathcal{C} = \{0\}^{m_1} \times \mathbb{R}^{m_2}_+$ with $m = m_1 + m_2$), the constraint nondegeneracy condition is equivalent to the well-known linear independence constraint qualification (LICQ) (Robinson [48]).

Let $\mathcal{I}$ denote the identity map on $\mathbb{S}^n$. Applying Definition 2, we see that the primal constraint nondegeneracy holds at a feasible solution $\overline{X} \in \mathbb{S}^n_+$ of the primal problem (1) if

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{I} \end{pmatrix} \mathbb{S}^n + \begin{pmatrix} \{0\} \\ \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\overline{X})) \end{pmatrix} = \begin{pmatrix} \mathbb{R}^m \\ \mathbb{S}^n \end{pmatrix}. \tag{9}$$

Similarly, the dual constraint nondegeneracy holds at a feasible solution $(\overline{y}, \overline{Z}) \in \mathbb{R}^n \times \mathbb{S}^n_+$ of the dual problem (2) if

$$\begin{pmatrix} \mathcal{A}^* & \mathcal{I} \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbb{R}^m \\ \mathbb{S}^n \end{pmatrix} + \begin{pmatrix} \{0\} \\ \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\overline{Z})) \end{pmatrix} = \begin{pmatrix} \mathbb{S}^n \\ \mathbb{S}^n \end{pmatrix}. \tag{10}$$

Without much difficulty, one can show that the primal constraint nondegeneracy (9) is equivalent to $\mathcal{A}\mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\overline{X})) = \mathbb{R}^m$, and the dual constraint nondegeneracy (10) is equivalent to $\mathcal{A}^*\mathbb{R}^m + \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\overline{Z})) = \mathbb{S}^n$. Moreover, if $\overline{X}$ is an optimal solution for (1), then under the primal constraint nondegeneracy condition, we can show that $\mathcal{M}(\overline{X})$ is a singleton (Alizadeh [1, theorem 2].

It is shown in theorem 18 of Chan and Sun [11] that the primal and dual constraint nondegeneracy conditions are both necessary and sufficient conditions for the non-singularity of elements in both $\partial_B \mathcal{F}(\overline{X}, \overline{y}, \overline{Z})$ and $\partial \mathcal{F}(\overline{X}, \overline{y}, \overline{Z})$, where $(\overline{X}, \overline{y}, \overline{Z})$ is a KKT point. Specifically, we have the following theorem.

Theorem 1 (Chan and Sun [11]). *Let $(\overline{X}, \overline{y}, \overline{Z}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ be a KKT point. Then, the following statements are equivalent:*

1. *The primal constraint nondegeneracy condition holds at $\overline{X}$, and the dual constraint nondegeneracy condition holds at $(\overline{y}, \overline{Z})$, respectively.*
2. *Every element in $\partial \mathcal{F}(\overline{X}, \overline{y}, \overline{Z})$ is nonsingular.*
3. *Every element in $\partial_B \mathcal{F}(\overline{X}, \overline{y}, \overline{Z})$ is nonsingular.*

We next present the concept of a strong second-order sufficient condition for linear SDPs. We note that the concept was introduced by Sun [57] for general nonlinear SDPs. To this end, let $(\overline{X}, \overline{y}, \overline{Z})$ be any KKT point. Write $W = \overline{X} - \overline{Z}$, and assume that W has the spectral decomposition (7). Partition P accordingly with respect to the index set α, β, and γ as $P = (P_\alpha \quad P_\beta \quad P_\gamma)$, where $P_\alpha \in \mathbb{R}^{n \times |\alpha|}$, $P_\beta \in \mathbb{R}^{n \times |\beta|}$, and $P_\gamma \in \mathbb{R}^{n \times |\gamma|}$. Then, by the fact that $\overline{X}$ and $\overline{Z}$ commute, it holds that

$$\overline{X} = P_\alpha \mathrm{diag}(d_1, \ldots, d_{|\alpha|}) P_\alpha^T, \quad \overline{Z} = -P_\gamma \mathrm{diag}(d_{1+|\alpha|+|\beta|}, \ldots, d_n) P_\gamma^T.$$

For any given matrix $B \in \mathbb{S}^n$, let $B^\dagger$ be the Moore-Penrose pseudoinverse of B. We define the linear-quadratic function $\Gamma_B : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{R}$ by

$$\Gamma_B(S,H) := 2\langle S, HB^\dagger H\rangle, \quad \forall (S,H) \in \mathbb{S}^n \times \mathbb{S}^n. \tag{11}$$

Using the above notation, the definition of the strong second-order sufficient condition is given as follows.

Definition 3. Let $\overline{X}$ be an optimal solution to the primal problem (1). We say that the strong second-order sufficient condition holds at $\overline{X}$ if $\sup_{(y,Z)\in\mathcal{M}(\overline{X})}\{-\Gamma_{\overline{X}}(-Z,H)\} > 0$ for any $0 \neq H \in \cap_{(y,Z)\in\mathcal{M}(\overline{X})}\mathrm{app}(y,Z)$, where $\mathrm{app}(y,Z) := \{B \in \mathbb{S}^n : \mathcal{A}B = 0, P_\beta^T B P_\gamma = 0, P_\gamma^T B P_\gamma = 0\}$.

Finally, we present a result that links the strong second-order sufficient condition and the dual constraint nondegeneracy condition; see proposition 15 in Chan and Sun [11] for a proof.

Lemma 2. *Let $(\overline{X},\overline{y},\overline{Z}) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ be a KKT point such that $\mathcal{M}(\overline{X}) = \{(\overline{y},\overline{Z})\}$. Then, the following two statements are equivalent:*

1. *The strong second-order sufficient condition holds at $\overline{X}$.*
2. *The dual constraint nondegeneracy condition holds at $(\overline{y},\overline{Z})$.*

3. The Huber Smoothing Function

Because the plus function $\rho(t) = \max\{0,t\}$, $t \in \mathbb{R}$ is not differentiable at $t = 0$, we consider its Huber smoothing (or approximation) function that is defined as follows:

$$h(\epsilon,t) = \begin{cases} t - \frac{|\epsilon|}{2}, & t > |\epsilon|, \\ \frac{t^2}{2|\epsilon|}, & 0 \le t \le |\epsilon|, \\ 0, & t < 0, \end{cases} \quad \forall (\epsilon,t) \in \mathbb{R}\backslash\{0\}\times\mathbb{R}, \quad h(0,t) = \rho(t), \quad \forall t \in \mathbb{R}. \tag{12}$$

Clearly, $h(\epsilon,t)$ is continuously differentiable for $\epsilon \neq 0$ and $t \in \mathbb{R}$. Moreover, one can easily check that h is directionally differentiable at $(0,t)$ for any $t \in \mathbb{R}$. Because $\Pi_{\mathbb{S}^n_+}(W) = P\mathrm{diag}(\rho(d_1),\dots,\rho(d_n))P^T$, we can compute $\Pi_{\mathbb{S}^n_+}(W)$ approximately by evaluating the matrix-valued mapping $\Phi(\epsilon,W)$ that is defined as

$$\Phi(\epsilon,W) := P\mathrm{diag}(h(\epsilon,d_1),\dots,h(\epsilon,d_n))P^T, \quad \forall (\epsilon,W) \in \mathbb{R}\times\mathbb{S}^n.$$

In this section, we shall study some fundamental properties of Φ. We note that the techniques used in our analysis are not new but mainly borrowed from those in the literature. However, existing techniques were mainly used for analyzing the CHKS-smoothing function. We will show in this section that the same analysis framework is applicable to the Huber smoothing function (12).

Before presenting our results, we need to introduce some useful notation. For any $(\epsilon,d) \in \mathbb{R}\backslash\{0\}\times\mathbb{R}^N$, where $N > 0$ is any dimension, we define the matrix-valued mappings $\Omega : \mathbb{R}\times\mathbb{R}^N \to \mathbb{S}^N$ and $\mathcal{D} : \mathbb{R}\times\mathbb{R}^N \to \mathbb{S}^N$ as follows:

$$[\Omega(\epsilon,d)]_{ij} := \begin{cases} \dfrac{h(\epsilon,d_i) - h(\epsilon,d_j)}{d_i - d_j} & d_i \neq d_j \\ h_2'(\epsilon,d_i) & d_i = d_j \end{cases}, \quad 1 \le i,j \le N,$$
$$\mathcal{D}(\epsilon,d) := \mathrm{diag}(h_1'(\epsilon,d_1),\dots,h_1'(\epsilon,d_N)). \tag{13}$$

Here, h_1' and h_2' denote the partial derivatives with respect to the first and the second arguments of h, respectively. Note that $0 \le [\Omega(\epsilon,d)]_{ij} \le 1$, $1 \le i,j \le N$, for any $(\epsilon,d) \in \mathbb{R}\times\mathbb{R}^N$, and that $0 \le |h_1'(\epsilon,d_i)| \le \frac{1}{2}$, $1 \le i \le N$, for any $\epsilon \neq 0$.

Let W have the spectral decomposition (7) and $\lambda_1 > \cdots > \lambda_r$ be the distinct eigenvalues of W with multiplicities $m_1,\dots,m_r$. Define $s_1 := 0$, $s_j := \sum_{i=1}^{j-1} m_j$ for $j = 2,\dots,r$ and $s_{r+1} := n$. For $j = 1,\dots,r$, denote the matrices $P_j := (p_{s_j+1} \cdots p_{s_{j+1}}) \in \mathbb{R}^{n\times m_j}$, where p_i denotes the i-th column of P for $i = 1,\dots,n$, and $Q_j := P_j P_j^T \in \mathbb{S}^n$. Then, it is clear that

$$W = \sum_{j=1}^r \lambda_j Q_j, \quad \Phi(\epsilon,W) = \sum_{j=1}^r h(\epsilon,\lambda_j) Q_j.$$

Now, consider $W + tH$, which admits a decomposition $W + tH = \sum_{i=1}^n d_i(t)p_i(t)p_i(t)^T$ with $d_1(t) \ge \cdots \ge d_n(t)$ and $\{p_1(t),\dots,p_n(t)\}$ forming an orthonormal basis for $\mathbb{R}^n$. Similarly, for $j = 1,\dots,r$, we can define the matrices $P_j(t) :=$

$(p_{s_j+1}(t)\ \cdots\ p_{s_{j+1}}(t)) \in \mathbb{R}^{n\times m_j}$ and $Q_j(t) := P_j(t)P_j(t)^T \in \mathbb{S}^n$. By the definition of directional differential, we may denote

$$d_i'(W;H) := \lim_{t\downarrow 0} \frac{d_i(t) - d_i}{t}, \quad \forall H \in \mathbb{S}^n,$$

if the limit exists, for any $i = 1,\ldots,n$.

Proposition 1. *Given any $W \in \mathbb{S}^n$ having the spectral decomposition (7), the following statements hold:*

1. *For any $\epsilon \neq 0$, Φ is continuously differentiable at (ϵ, W), and it holds that*

$$\Phi'(\epsilon,W)(\tau,H) = P[\Omega(\epsilon,d)\circ(P^THP) + \tau\mathcal{D}(\epsilon,d)]P^T, \quad \forall(\tau,H)\in\mathbb{R}\times\mathbb{S}^n, \tag{14}$$

where $\Omega(\epsilon,d)$, $\mathcal{D}(\epsilon,d)$ are defined in (13).

2. *Φ is locally Lipschitz continuous on $\mathbb{R}\times\mathbb{S}^n$.*
3. *Φ is directionally differentiable at $(0,W)$, and for any $(\tau,H)\in\mathbb{R}\times\mathbb{S}^n$, it holds that*

$$\begin{aligned}\Phi'((0,W);(\tau,H)) = {}& \frac{1}{2}\sum_{1\le k\neq j\le r}\frac{h(0,\lambda_k)-h(0,\lambda_j)}{\lambda_k-\lambda_j}(Q_jHQ_k + Q_kHQ_j)\\ &+\sum_{i\in\alpha}\left(d_i'(W;H) - \frac{|\tau|}{2}\right)p_ip_i^T + \sum_{i\in\beta}h(\tau,d_i'(W;H))p_ip_i^T,\end{aligned} \tag{15}$$

where for any $1\le i\le n$ with $d_i = \lambda_j$ for some $1\le j\le r$, the elements of $\{d_i'(W;H): i = s_j+1,\ldots,s_{j+1}\}$ are the eigenvalues of $P_j^THP_j$, arranged in decreasing order.

4. *Φ is strongly semismooth on $\mathbb{R}\times\mathbb{S}^n$.*

Proof. See Appendix A. Interested readers are referred to Gao and Sun [18] and Zhao 74] for similar results for other popular smoothing functions. Q.E.D.

Notice that the matrix P_j is defined up to an orthogonal transformation; that is, it can be replaced with P_jU, where $U\in\mathbb{R}^{m_j\times m_j}$ is an arbitrary orthogonal matrix. For any given $H\in\mathbb{S}^n$, consider the matrix $P_j^THP_j$, which admits the spectral decomposition

$$P_j^THP_j = U_j\mathrm{diag}(\mu_1,\ldots,\mu_{m_j})U_j^T, \quad \mu_1\ge\cdots\ge\mu_{m_j},$$

where $U_j\in\mathbb{R}^{m_j\times m_j}$ is orthogonal. Then, we may replace P_j by P_jU_j. In this way, $P_j^THP_j$ is always a diagonal matrix whose diagonal entries are arranged in decreasing order. As a consequence, we can easily verify from the third statement of Proposition 1 that

$$\begin{aligned}\Phi'((0,W);(\tau,H)) = {}& P\begin{pmatrix}\tilde H_{\alpha\alpha} & \tilde H_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma}\circ\tilde H_{\alpha\gamma}\\ \tilde H_{\alpha\beta}^T & \Phi_{|\beta|}(\tau,\tilde H_{\beta\beta}) & 0\\ [\Omega_0(d)]_{\alpha\gamma}^T\circ H_{\alpha\gamma}^T & 0 & 0\end{pmatrix}P^T\\ &-\frac{|\tau|}{2}\sum_{i\in\alpha}p_ip_i^T,\end{aligned} \tag{16}$$

$\forall(\tau,H)\in\mathbb{R}\times\mathbb{S}^n$, where $\tilde H := P^THP$ and the mapping $\Phi_{|\beta|}:\mathbb{R}\times\mathbb{S}^{|\beta|}\to\mathbb{S}^{|\beta|}$ is defined by replacing the dimension n in the definition of $\Phi:\mathbb{R}\times\mathbb{S}^n\to\mathbb{S}^n$ with $|\beta|$.

Define the mapping $\mathcal{L}:\mathbb{R}\times\mathbb{S}^n\to\mathbb{S}^n$ as

$$\mathcal{L}(\tau,H) := \Phi'((0,W);(\tau,H)), \quad (\tau,H)\in\mathbb{R}\times\mathbb{S}^n.$$

Using (16), we see that the mapping $\mathcal{L}(\cdot,\cdot)$ is F-differentiable at (τ,H) if and only if $\tau\neq 0$. Obviously, $\mathcal{L}$ is locally Lipschitz continuous everywhere. Hence, $\partial_B\mathcal{L}(0,0)$ is well defined. Then, we have the following useful result that builds an insightful connection between $\partial_B\Phi$ and $\partial_B\mathcal{L}$. To prove the result, we will basically follow the proof of Chan and Sun [11, lemma 4]. However, because of the second term on the right-hand side of (16), we need to modify the proof to make the paper self-contained; see Appendix B for more details.

Lemma 3. *Suppose that $W \in \mathbb{S}^n$ has the eigenvalue decomposition (7). Then, it holds that*

$$\partial_B \Phi(0, W) = \partial_B \mathcal{L}(0, 0).$$

Recall that $\Phi_{|\beta|}$ is also locally Lipschitz continuous. Hence, $\partial_B \Phi_{|\beta|}$ and $\partial \Phi_{|\beta|}$ are both well-defined. The following lemma provides an effective way to calculate $\partial_B \Phi(0, W)$ and $\partial \Phi(0, W)$.

Lemma 4. *For any $W \in \mathbb{S}^n$ with the spectral decomposition (7), we have that $V \in \partial_B \Phi(0, W)$ (respectively, $\partial \Phi(0, W)$) if and only if there exist $V_{|\beta|} \in \partial_B \Phi_{|\beta|}(0, 0)$ (respectively, $\partial \Phi_{|\beta|}(0, 0)$) and a scalar $v \in \{-1, 1\}$ (respectively, $[-1, 1]$) such that for all $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$,*

$$V(\tau, H) = \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & V_{|\beta|}(\tau, \tilde{H}_{\beta\beta}) & 0 \\ \tilde{H}_{\alpha\gamma}^T \circ [\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T + \frac{v\tau}{2} \sum_{i \in \alpha} p_i p_i^T,$$

where $\tilde{H} := P^T H P$. Moreover, $V_{|\beta|} \in \partial_B \Phi_{|\beta|}(0, 0)$ if and only if there exist an orthogonal matrix $U \in \mathbb{R}^{|\beta| \times |\beta|}$ and a symmetric matrix

$$\Omega_{|\beta|} \in \left\{ \Omega \in \mathbb{S}^{|\beta|} : \Omega_{|\beta|} = \lim_{k \to \infty} \Omega(\epsilon^k, z^k), (\epsilon^k, z^k) \to (0, 0), \epsilon^k \neq 0, z_1^k \geq \cdots \geq z_{|\beta|}^k \right\}$$

such that

$$V_{|\beta|}(0, Y) = U[\Omega_{|\beta|} \circ (U^T Y U)]U^T, \quad \forall Y \in \mathbb{S}^{|\beta|}.$$

Proof. See Appendix C. Q.E.D.

As a corollary of Lemma 1 and Lemma 4, we can verify that for any $V_0 \in \partial_B \Pi_{\mathbb{S}^n_+}(W)$, there exists $V \in \partial_B \Phi(0, W)$ such that

$$V_0(H) = V(0, H), \quad \forall H \in \mathbb{S}^n. \tag{17}$$

We end this section by presenting a useful inequality for elements in $\partial \Phi(0, W)$.

Lemma 5. *Let $W \in \mathbb{S}^n$ have the spectral decomposition (7). Then, for any $V \in \partial \Phi(0, W)$,*

$$\langle H - V(0, H), V(0, H) \rangle \geq 0, \quad \forall H \in \mathbb{S}^n.$$

Proof. See Appendix D. Q.E.D.

4. A Squared Smoothing Newton Method

In this section, we shall present our main algorithm, a squared smoothing Newton method via the Huber smoothing function. We then focus on analyzing its correctness, global convergence, and the fast local convergence rate under suitable regularity conditions. In the following, $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the space of real numbers, the nonnegative orthant and the positive orthant, respectively.

By the results presented in the previous section, we can define the smoothing function for $\mathcal{F}$ based on the smoothing function Φ for $\Pi_{\mathbb{S}^n_+}$. In particular, let $\mathcal{E} : \mathbb{R} \times \mathbb{X} \to \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$ be defined as

$$\mathcal{E}(\epsilon, X, y, Z) = \begin{pmatrix} \mathcal{A}X + \kappa_p |\epsilon| y - b \\ -\mathcal{A}^* y - Z + C \\ (1 + \kappa_c |\epsilon|) X - \Phi(\epsilon, X - Z) \end{pmatrix}, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}, \tag{18}$$

where $\kappa_p > 0$, $\kappa_c > 0$ are two given constants and $\mathbb{X} := \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$. Then, $\mathcal{E}$ is a continuously differentiable function around any (ϵ, X, y, Z) for any $\epsilon \neq 0$. Also, it satisfies

$$\mathcal{E}(\epsilon', X', y', Z') \to \mathcal{F}(X, y, Z), \quad \text{as } (\epsilon', X', y', Z') \to (0, X, y, Z).$$

Note that adding the perturbation terms $\kappa_p |\epsilon| y$ and $\kappa_c |\epsilon| X$ for constructing the smoothing function of $\mathcal{F}$ is crucial in our algorithmic design because it ensures that the proposed algorithm is well-defined (see Lemma 6).

Define the function $\hat{\mathcal{E}} : \mathbb{R} \times \mathbb{X} \to \mathbb{R} \times \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$ by

$$\hat{\mathcal{E}}(\epsilon, X, y, Z) = \begin{pmatrix} \epsilon \\ \mathcal{E}(\epsilon, X, y, Z) \end{pmatrix}, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}. \tag{19}$$

Then, solving the non-smooth equation $\mathcal{F}(X, y, Z) = 0$ is equivalent to solving the following system of nonlinear equations

$$\hat{\mathcal{E}}(\epsilon, X, y, Z) = 0. \tag{20}$$

Associated with the mapping $\hat{\mathcal{E}}$, we have the natural merit function $\psi : \mathbb{R} \times \mathbb{X} \to \mathbb{R}_+$ that is defined as

$$\psi(\epsilon, X, y, Z) := \|\hat{\mathcal{E}}(\epsilon, X, y, Z)\|^2, \quad \forall (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}. \tag{21}$$

Given $r \in (0, 1]$, $\hat{r} \in (0, \infty)$ and $\tau \in (0, 1)$, we can define two functions $\zeta : \mathbb{R} \times \mathbb{X} \to \mathbb{R}_+$ and $\eta : \mathbb{R} \times \mathbb{X} \to \mathbb{R}_+$ as follows:

$$\zeta(\epsilon, X, y, Z) = r \min\{1, \|\hat{\mathcal{E}}(\epsilon, X, y, Z)\|^{1+\tau}\}, \quad (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}, \tag{22}$$

and

$$\eta(\epsilon, X, y, Z) = \min\{1, \hat{r}\|\hat{\mathcal{E}}(\epsilon, X, y, Z)\|^{\tau}\}, \quad (\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}. \tag{23}$$

Then, the inexact smoothing Newton can be described in Algorithm 1.

Algorithm 1 (A Squared Smoothing Newton Method)

1: **Input:** $\hat{\epsilon} \in (0, \infty)$, $r \in (0, 1)$, $\hat{r} \in (0, \infty)$, $\hat{\eta} \in (0, 1)$ be such that $\delta := \sqrt{2} \max\{r\hat{\epsilon}, \hat{\eta}\} < 1$, $\rho \in (0, 1)$, $\sigma \in (0, 1/2)$, $\tau \in (0, 1]$, $\epsilon^0 = \hat{\epsilon}$, $(X^0, y^0, Z^0) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$.
2: **for** $k \geq 0$ **do**
3: **if** $\hat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k) = 0$, **then**
4: **Output:** $(\epsilon^k, X^k, y^k, Z^k)$.
5: **else**
6: Compute $\eta_k := \eta(\epsilon^k, X^k, y^k, Z^k)$ and $\zeta_k := \zeta(\epsilon^k, X^k, y^k, Z^k)$.
7: Solve the following equation

$$\hat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k) + \hat{\mathcal{E}}'(\epsilon^k, X^k, y^k, Z^k) \begin{pmatrix} \Delta\epsilon^k \\ \Delta X^k \\ \Delta y^k \\ \Delta Z^k \end{pmatrix} = \begin{pmatrix} \zeta_k \hat{\epsilon} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{24}$$

approximately such that

$$\|\mathcal{R}_k\| \leq \min\{\eta_k \|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'_\epsilon(\epsilon^k, X^k, y^k, Z^k)\Delta\epsilon^k\|, \hat{\eta}\|\hat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k)\|\}, \tag{25}$$

where $\Delta\epsilon^k := -\epsilon^k + \zeta_k\hat{\epsilon}$ and $\mathcal{R}_k := \mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'(\epsilon^k, X^k, y^k, Z^k) \begin{pmatrix} \Delta\epsilon^k \\ \Delta X^k \\ \Delta y^k \\ \Delta Z^k \end{pmatrix}$.

8: Compute ℓ_k as the smallest nonnegative integer ℓ satisfying

$$\psi(\epsilon^k + \rho^\ell \Delta\epsilon^k, X^k + \rho^\ell \Delta X^k, y^k + \rho^\ell \Delta y^k, Z^k + \rho^\ell \Delta Z^k) \leq [1 - 2\sigma(1 - \delta)\rho^\ell]\psi(\epsilon^k, X^k, y^k, Z^k).$$

9: Compute

$$(\epsilon^{k+1}, X^{k+1}, y^{k+1}, Z^{k+1}) = (\epsilon^k + \rho^{\ell_k}\Delta\epsilon^k, X^k + \rho^{\ell_k}\Delta X^k, y^k + \rho^{\ell_k}\Delta y^k, Z^k + \rho^{\ell_k}\Delta Z^k).$$

10: **end if**
11: **end for**

For the rest of this section, we shall show that Algorithm 1 is well-defined and analyze its convergence properties. Note that the global convergence and the fast local convergence rate under the non-singularity of $\partial_B\hat{\mathcal{E}}$ or $\partial\hat{\mathcal{E}}$ at solution points of nonlinear Equations (20) are studied extensively in the literature (see e.g., Chan and Sun [11] and Gao and Sun [18]). However, because we are using the Huber function, conditions that ensure the non-singularity conditions need to be redeveloped. Furthermore, Algorithm 1 allows one to specify the parameter $\tau \in$

$(0,1]$ (which is used to control the rate of convergence), whereas existing results focus mainly on the case when $\tau = 1$. Consequently, to make the paper self-contained, we also present the details of the analysis for our key results.

4.1. Global Convergence

We first show that the linear system (24) is well-defined and solvable for any $\epsilon^k > 0$. Hence, the inexactness conditions in (25) are always achievable. This objective can be accomplished by showing that the coefficient matrix of the linear system (24) is nonsingular for any $\epsilon^k > 0$. In the following exposition, $\mathcal{E}'$ and $\hat{\mathcal{E}}'$ denote the derivatives of $\mathcal{E}$ and $\hat{\mathcal{E}}$, respectively, and $\mathcal{E}'_\epsilon$ denotes the partial derivative of $\mathcal{E}$ with respect to the first argument ϵ.

Lemma 6. *For any $(\epsilon', X', y', Z') \in \mathbb{R}_{++} \times \mathbb{X}$, there exists an open neighborhood $\mathcal{U}$ of (ϵ', X', y', Z') such that $\hat{\mathcal{E}}'(\epsilon, X, y, Z)$ is nonsingular for any $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon \in \mathbb{R}_{++}$.*

Proof. See Appendix E. Q.E.D.

The next task is to show that the line search procedure is well-defined, that is, ℓ_k is finite for all $k \geq 0$. We will see that the inexactness condition $\|\mathcal{R}_k\| \leq \hat{\eta}\|\hat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k)\|$ plays a fundamental role in our analysis. We also note that the inexactness condition

$$\|\mathcal{R}_k\| \leq \eta_k\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'_\epsilon(\epsilon^k, X^k, y^k, Z^k)\Delta\epsilon^k\|$$

does not affect the correctness of Algorithm 1 but will be crucial in analyzing the local convergence rate of Algorithm 1.

Lemma 7. *For any $(\epsilon', X', y', Z') \in \mathbb{R}_{++} \times \mathbb{X}$, there exist an open neighborhood $\mathcal{U}$ of (ϵ', X', y', Z') and a positive scalar $\overline{\alpha} \in (0,1]$ such that for any $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon \in \mathbb{R}_{++}$ and $\alpha \in (0, \overline{\alpha}]$, it holds that*

$$\psi(\epsilon + \alpha\Delta\epsilon, X + \alpha\Delta X, y + \alpha\Delta y, Z + \alpha\Delta Z) \leq [1 - 2\sigma(1-\delta)\alpha]\psi(\epsilon, X, y, Z),$$

where $\mathbf{\Delta} := (\Delta\epsilon; \Delta X; \Delta y; \Delta Z) \in \mathbb{R} \times \mathbb{X}$ satisfies

$$\Delta\epsilon = -\epsilon + \zeta(\epsilon, X, y, Z)\hat{\epsilon}, \quad \|\mathcal{E}(\epsilon, X, y, Z) + \mathcal{E}'(\epsilon, X, y, Z)\mathbf{\Delta}\| \leq \hat{\eta}\|\hat{\mathcal{E}}(\epsilon, X, y, Z)\|.$$

Proof. See Appendix F. Q.E.D.

As shown before, the non-singularity of the coefficient matrix in (24) requires ϵ^k being positive. The next lemma shows that Algorithm 1 generates ϵ^k that is lower bounded by $\zeta(\epsilon^k, X^k, y^k, Z^k)\hat{\epsilon}$. Thus, as long as the optimal solution is not found, ϵ^k remains positive. To present the lemma, we need to define the following set:

$$\mathcal{N} := \{(\epsilon, X, y, Z) : \epsilon \geq \zeta(\epsilon, X, y, Z)\hat{\epsilon}\}.$$

Lemma 8. *Suppose that for a given $k \geq 0$, $\epsilon^k \in \mathbb{R}_{++}$ and $(\epsilon^k, X^k, y^k, Z^k) \in \mathcal{N}$. Then, for any $\alpha \in [0,1]$ such that*

$$\psi(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k) \leq [1 - 2\sigma(1-\delta)\alpha]\psi(\epsilon^k, X^k, y^k, Z^k),$$

it holds that

$$(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k) \in \mathcal{N}.$$

Proof. See Appendix G. Q.E.D.

With those previous results, we can now establish the global convergence of Algorithm 1.

Theorem 2. *Algorithm 1 is well-defined and generates an infinite sequence $\{(\epsilon^k, X^k, y^k, Z^k)\} \subseteq \mathcal{N}$ with the property that any accumulation point $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$ (it exists if the solution set to the KKT system is nonempty and bounded) of $\{(\epsilon^k, X^k, y^k, Z^k)\}$ is a solution of $\hat{\mathcal{E}}(\epsilon, X, y, Z) = 0$, $(\epsilon, X, y, Z) \in \mathbb{R} \times \mathbb{X}$.*

Proof. See Appendix H. Q.E.D.

4.2. Super-Linear Convergence Rate

We next establish the super-linear convergence rate of Algorithm 1 under certain regularity conditions. The following lemma is useful for characterizing the non-singularity of an element in $\partial\hat{\mathcal{E}}$ at any accumulation point.

Lemma 9. *Suppose that $(\overline{X}, \overline{y}, \overline{Z}) \in \mathbb{X}$ is a KKT point. Let $\overline{X} - \overline{Z} := \overline{W} \in \mathbb{S}^n$ and $\overline{V} \in \partial\Phi(0, \overline{W})$. Then, for any ΔX and ΔZ in $\mathbb{S}^n$ such that $\Delta X = \overline{V}(0, \Delta X - \Delta Z)$, it holds that $\langle \Delta X, \Delta Z\rangle \leq \Gamma_{\overline{X}}(-\overline{Z}, \Delta X)$, where $\Gamma_{\overline{X}}$ is defined as in (11).*

Proof. See Appendix I. Q.E.D.

Using the above lemma, we can establish the following equivalent relations.

Proposition 2. *Let $(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})$ be such that $\hat{\mathcal{E}}(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})=0$. Then, the following statements are equivalent to each other.*

1. *The primal constraint nondegenerate condition holds at $\overline{X}$, and the dual constraint nondegenerate condition holds at $(\overline{y},\overline{Z})$.*
2. *Every element in $\partial_B\hat{\mathcal{E}}(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})$ is nonsingular.*
3. *Every element in $\partial\hat{\mathcal{E}}(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})$ is nonsingular.*

Proof. See Appendix J. Q.E.D.

Typically, under certain regularity conditions, one is able to establish the local fast convergence rate of Newton-type methods. Indeed, we show in the next theorem that Algorithm 1 admits a super-linear convergence rate under the primal and dual constraint nondegenerate conditions.

Theorem 3. *Let $(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})$ be an accumulation point of the infinite sequence $\{(\epsilon^k,X^k,y^k,Z^k)\}$ generated by Algorithm 1. Suppose that the primal constraint nondegeneracy condition holds at $\overline{X}$, and the dual constraint nondegeneracy condition holds at $(\overline{y},\overline{Z})$. Then, the whole sequence $\{(\epsilon^k,X^k,y^k,Z^k)\}$ converges to $(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})$ super-linearly; that is,*

$$\|(\epsilon^{k+1}-\overline{\epsilon},X^{k+1}-\overline{X},y^{k+1}-\overline{y},Z^{k+1}-\overline{Z})\|=O(\|(\epsilon^k-\overline{\epsilon},X^k-\overline{X},y^k-\overline{y},Z^k-\overline{Z})\|^{1+\tau}),$$

where $\tau\in(0,1]$ controls how accurately one should solve the Newton system in (24).

Proof. See Appendix K. Q.E.D.

Analyzing the fast convergence rate of smoothing Newton type methods for solving SDPs has been a continuing research direction for decades. Early works (see, e.g., Chen and Tseng [13], Chen et al. [15], and Sun et al. [61]) assumed the strict complementarity (i.e., $\overline{X}+\overline{Z}\in\mathbb{S}^n_{++}$) and primal-dual nondegeneracy conditions for establishing the local fast convergence rate. Later works (Chan and Sun [11], Kanzow and Nagel [27]) attempted to drop the strict complementarity and establish the fast convergence rate merely under nondegeneracy conditions for smoothing Newton methods via the minimum smoothing function and the CHKS-smoothing function, respectively. Following the same research theme as Chan and Sun [11] and Kanzow and Nagel [27], our work establishes similar convergence properties for the Huber-based squared smoothing Newton method under the same nondegeneracy conditions without assuming the strict complementarity condition.

Remark 1 (Convergence Properties of IPMs). Here, we shall provide some comparisons on the singularity of the (generalized) Jacobian matrices arising from IPMs and those from the proposed smoothing Newton methods. Given an interior-point iterate (X,y,Z) that is close to some KKT point (X^*,y^*,Z^*), denote $\mu:=\langle X,Z\rangle/n$. If (X^*,Z^*) satisfies the strict complementarity condition (i.e., $X^*+Z^*>0$) and the primal and dual nondegeneracy conditions hold at the KKT point, then it is known that the Jacobian matrix of the KKT conditions has a bounded condition number even when $\mu>0$ approaches 0 (Alizadeh et al. [2]). Under the same conditions, the IPM with the so-called AHO direction is shown to have a super-linear or quadratic convergence rate (Alizadeh et al. [2]). The key difference between our results and those for IPMs is that we do not need to assume the strict complementarity holds.

4.3. Numerical Implementation

Given $\nu>0$, one can check that the condition $X-\Pi_{\mathbb{S}^n_+}(X-Z)=0$ is equivalent to the condition $X-\Pi_{\mathbb{S}^n_+}(X-\nu Z)=0$. Thus, in our implementation, we in fact solve the following nonlinear system (with a slight abuse of the notation $\mathcal{E}$):

$$\mathcal{E}(\epsilon,X,y,Z)=\begin{pmatrix}\mathcal{A}X+\kappa_p|\epsilon|y-b\\ -\mathcal{A}^*y-Z+C\\ (1+\kappa_c|\epsilon|)X-\Phi(\epsilon,X-\nu Z)\end{pmatrix},\quad\forall(\epsilon,X,y,Z)\in\mathbb{R}\times\mathbb{X}.$$

Our numerical experience shows that introducing the parameter ν is important because it balances the norms of X and Z, thus improving the performance of the algorithm. However, the convergence properties established in the previous sections are not affected.

Note that one of the key computational tasks in Algorithm 1 is to solve a system of linear equations for computing the search direction at each iteration. Here, we briefly explain how we solve the linear system of the

following form:

$$\begin{pmatrix} \mathcal{A} & \mu_p I_m & 0 \\ 0 & -\mathcal{A}^* & -\mathcal{I} \\ (1+\mu_c)\mathcal{I} - V & 0 & \nu V \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta Z \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}, \tag{26}$$

where $\mu_p := \kappa_p|\epsilon| > 0$, $\mu_c := \kappa_c|\epsilon| > 0$, $V := \Phi_2'(\epsilon, X - \nu Z)$, $\mathcal{I}$ is the identity mapping over $\mathbb{S}^n$, and $(R_1, R_2, R_3) \in \mathbb{R}^m \times \mathbb{S}^n \times \mathbb{S}^n$ is the right-hand-side vector constructed from the current iterate $(X, y, Z) \in \mathbb{X}$. From the last two equations of (26), we have

$$\Delta Z = -\mathcal{A}^*\Delta y - R_2, \quad \Delta X = [(1+\mu_c)\mathcal{I} - V]^{-1}(R_3 - \nu V \Delta Z),$$

which implies that $\Delta X = [(1+\mu_c)\mathcal{I} - V]^{-1}(R_3 + \nu V\mathcal{A}^*\Delta y + \nu V R_2)$. Substituting this equality into the first equation in (26), we get

$$\mathcal{A}[(1+\mu_c)\mathcal{I} - V]^{-1}(R_3 + \nu V\mathcal{A}^*\Delta y + \nu V R_2) + \mu_p \Delta y = R_1,$$

which leads to the following smaller $m \times m$ symmetric positive definite system:

$$(\mu_p I_m + \nu\mathcal{A}[(1+\mu_c)\mathcal{I} - V]^{-1}V\mathcal{A}^*)\Delta y = R_1 - \mathcal{A}[(1+\mu_c)\mathcal{I} - V]^{-1}(\nu V R_2 + R_3). \tag{27}$$

We then apply the preconditioned conjugate gradient (PCG) method with a simple approximated diagonal preconditioner to solve the last linear system to get Δy. After obtaining Δy, we can compute ΔX and ΔZ in terms of Δy. In our experiments, we set κ_p to be a small number, say $\kappa_p = 10^{-10}$, whereas κ_c should be larger and may be changed for different classes of problems. Note that when ϵ is small, the coefficient matrix in (27) can also be highly ill-conditioned, which is a critical issue that also arises in IPMs and ALMs (when the primal nondegeneracy condition is not satisfied). To alleviate this issue, an effective preconditioner is needed. However, to focus on the smoothing Newton method itself, we refrain from diving into this direction and leave it for future research.

Next, we shall see how to evaluate the matrix-vector products involving $[(1+\mu_c)\mathcal{I} - V]^{-1}V$ that are needed when solving (27). Suppose that (ϵ, X, y, Z) with $\epsilon > 0$ is given and that $W := X - \nu Z$ has the spectral decomposition in (7). Then, the linear mapping $V : \mathbb{S}^n \to \mathbb{S}^n$ takes the form $V(H) = P[\Omega(\epsilon, d) \circ (P^T H P)]P^T$, $\forall H \in \mathbb{S}^n$, where $\Omega(\epsilon, d) \in \mathbb{S}^n$ is defined in (13) and $[\Omega(\epsilon, d)]_{ij} \in [0, 1]$ for $1 \le i, j \le n$. Consider the following three index sets: $\alpha := \{i : d_i \ge \epsilon\}$, $\beta := \{i : 0 < d_i < \epsilon\}$, and $\gamma := \{i : d_i \le 0\}$. Then, we can simply write $\Omega(\epsilon, d)$ as

$$\Omega(\epsilon, d) = \begin{pmatrix} E_{\alpha\alpha} & \Omega_{\alpha\beta} & \Omega_{\alpha\gamma} \\ \Omega_{\alpha\beta}^T & \Omega_{\beta\beta} & \Omega_{\beta\gamma} \\ \Omega_{\alpha\gamma}^T & \Omega_{\beta\gamma}^T & 0 \end{pmatrix},$$

where $E_{\alpha\alpha} \in \mathbb{R}^{|\alpha|\times|\alpha|}$ is the matrix of all ones, and $\Omega_{\alpha\beta} \in \mathbb{R}^{|\alpha|\times|\beta|}$, $\Omega_{\beta\beta} \in \mathbb{R}^{|\beta|\times|\beta|}$, $\Omega_{\alpha\gamma} \in \mathbb{R}^{|\alpha|\times|\gamma|}$, and $\Omega_{\beta\gamma} \in \mathbb{R}^{|\beta|\times|\gamma|}$, with all their entries belonging to the interval 0, 1. We also partition the orthogonal matrix P as $P = (P_\alpha \quad P_\beta \quad P_\gamma)$ accordingly. Define the matrix $\hat{\Omega} \in \mathbb{S}^n$ as $[\hat{\Omega}]_{ij} := [\Omega(\epsilon, d)]_{ij}/(1 + \mu_c - [\Omega(\epsilon, d)]_{ij})$ for $i, j = 1, \ldots, n$, which takes the following form:

$$\hat{\Omega} = \begin{pmatrix} \frac{1}{\mu_c}E_{\alpha\alpha} & \hat{\Omega}_{\alpha\beta} & \hat{\Omega}_{\alpha\gamma} \\ \hat{\Omega}_{\alpha\beta}^T & \hat{\Omega}_{\beta\beta} & \hat{\Omega}_{\beta\gamma} \\ \hat{\Omega}_{\alpha\gamma}^T & \hat{\Omega}_{\beta\gamma}^T & 0 \end{pmatrix},$$

Then, one can check that $[(1+\mu_c)\mathcal{I} - V]^{-1}V(H) = P[\hat{\Omega} \circ (P^T H P)]P^T$, $\forall H \in \mathbb{S}^n$. If $|\alpha| + |\beta| \ll n$, one may use the following scheme to compute

$$\begin{aligned} [(1+\mu_c)\mathcal{I} - V]^{-1}V(H) = {} & \frac{1}{\mu_c}P_\alpha(P_\alpha^T H P_\alpha)P_\alpha^T + P_\beta(\hat{\Omega}_{\beta\beta} \circ (P_\beta^T H P_\beta))P_\beta^T \\ & + P_\alpha(\hat{\Omega}_{\alpha\beta} \circ (P_\alpha^T H P_\beta))P_\beta^T + (P_\alpha(\hat{\Omega}_{\alpha\beta} \circ (P_\alpha^T H P_\beta))P_\beta^T)^T \\ & + P_\alpha(\hat{\Omega}_{\alpha\gamma} \circ (P_\alpha^T H P_\gamma))P_\gamma^T + (P_\alpha(\hat{\Omega}_{\alpha\gamma} \circ (P_\alpha^T H P_\gamma))P_\gamma^T)^T \\ & + P_\beta(\hat{\Omega}_{\beta\gamma} \circ (P_\beta^T H P_\gamma))P_\gamma^T + (P_\beta(\hat{\Omega}_{\beta\gamma} \circ (P_\beta^T H P_\gamma))P_\gamma^T)^T. \end{aligned}$$

On the other hand, if $n - |\alpha| \ll n$, then one may consider using the following scheme to compute

$$[(1+\mu_c)\mathcal{I} - V]^{-1}V(H) = \frac{1}{\mu_c}H - P[\tilde{\Omega} \circ (P^T H P)]P^T,$$

where $\tilde{\Omega} := \frac{1}{\mu_c}E - \hat{\Omega}$ would have more zero entries than $\hat{\Omega}$. As a consequence, because the Huber function maps any nonpositive number to zero, we can exploit the sparsity structure in $\hat{\Omega}$ or $\tilde{\Omega}$ to cut down the computational cost. However, if the CHKS-smoothing function is used, we will get a dense counterpart, and the aforementioned sparsity structure will be lost. Theoretically, both Huber-based and CHKS-based smoothing Newton methods share the same convergence properties. From our numerical experience, the practical performances of both methods in terms of the total number of CG iterations are also similar. Thus, reducing the computational costs in matrix-vector multiplications definitely makes the Huber-based method more efficient than the CHKS-based method. This also explains why we choose the Huber function instead of the CHKS function to perform the smoothing of the projection operator. Other than the CHKS-smoothing function, one can also compare the practical performance of the Huber function with other smoothing functions, but we leave it as a topic for future investigation.

When compared with IPMs, the proposed algorithm demonstrates some advantages from the computational perspective. To see this, let $\mu = \langle X, Z\rangle/n$ be defined as in Remark 1. It is known that an IPM typically computes the search direction in each iteration via solving an $m \times m$ linear system whose coefficient matrix is defined as

$$\mathcal{M}_{\mathrm{IPM}}h := \mathcal{A}P(D \circ (P^T(\mathcal{A}^*h)P))P^T, \quad \forall h \in \mathbb{R}^m,$$

where $P \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{n\times n}_{++}$ are fully dense matrices, depending on the iterate (X, Z), and hence, also μ (Toh and Kojima [65]). In general, the matrix $\mathcal{M}_{\mathrm{IPM}}$ is asymptotically singular when $\mu \to 0$; that is, the condition number of $\mathcal{M}_{\mathrm{IPM}}$ will grow to infinity as $\mu \to 0$, even under strict complementarity and primal and dual nondegeneracy conditions (Alizadeh et al. [2]). Moreover, because P and D are fully dense, it is clear that computing the matrix-vector product involving $\mathcal{M}_{\mathrm{IPM}}$ would cost at least $8n^3$ arithmetic operations. Similar to IPMs, the $m \times m$ linear system (27) that is used to compute the search direction in the smoothing Newton method can also be asymptotically singular when $\epsilon \to 0$. The key difference is that the perturbation terms $\kappa_p|\epsilon|y$ and $\kappa_c|\epsilon|X$ introduced when constructing the smoothing function of $\mathcal{F}$ help to improve the conditioning of the underlying coefficient matrix. Moreover, from the viewpoint of computing the matrix-vector product, we emphasize that the coefficient matrix in (27) can exploit the low-rank (or high-rank) property of the matrix $X - \nu Z$ as mentioned above. For example, if $r := |\alpha| + |\beta| \ll n$, then the matrix-vector product only costs $O(n^2 r)$ arithmetic operations, which could be much cheaper than that of an IPM iteration.

We observe from our numerical tests that when the dual iterate (y^k, Z^k) does not make significant progress, it is helpful for us to project the primal iterate X^k onto the affine subspace $\mathcal{H}_k := \{X \in S^n : \mathcal{A}(X) = b, \langle C, X\rangle = \langle b, y^k\rangle\}$. To perform such a projection operation, we only need to perform a Cholesky factorization for a certain operator (depending only on $\mathcal{A}$ and C) once at the beginning of the Algorithm 1. However, in the case when the factorization fails (i.e., the operator is no positive definite) or m is too large, we will not perform such projections.

5. Numerical Experiments

To evaluate the practical performance of Algorithm 1 described in the last section, we conduct numerical experiments to solve various classes of SDPs that are commonly tested in the literature.

5.1. Experimental Settings and Implementation

Similar to Yang et al. [71], the following relative KKT residues are used as the termination criteria of our algorithm:

$$\eta_p := \frac{\|\mathcal{A}X - b\|}{1 + \|b\|}, \quad \eta_d := \frac{\|\mathcal{A}^*y + Z - C\|}{1 + \|C\|}, \quad \eta_c := \frac{\|X - \Pi_{\mathbb{S}^n_+}(X - Z)\|}{1 + \|X\| + \|Z\|}.$$

Particularly, we terminate the algorithm as long as $\eta_{KKT} := \max\{\eta_p, \eta_d, \eta_c\} \le \mathtt{tol}$, where $\mathtt{tol} > 0$ is a given tolerance. Moreover, we denote the maximum number of iterations of Algorithm 1 as `maxiter`. We also terminate the algorithm when the iteration count reaches this number. In our experiments, we set $\mathtt{tol} = 10^{-6}$ and $\mathtt{maxiter} = 50$.

For more efficiency, we apply a certain first-order method to generate a starting point (X^0, y^0, Z^0) to warmstart Algorithm 1. Our choice is the routine based on a semi-proximal ADMM method (Yang et al. [71]). The stopping tolerance (in terms of the maximal relative KKT residual, i.e., η_{KKT}) and the maximum number of iterations for

the warmstarting phase are denoted by $\texttt{tol}_0$ and $\texttt{maxiter}_0$, respectively. In our experiments, we set $\texttt{maxiter}_0 = 1000$, and the computational time spent in the warmstarting phase will be included in the total computational time. We also notice that the performance of Algorithm 1 depends sensitively on the choice of $\texttt{tol}_0$. Hence, we set the value of $\texttt{tol}_0$ differently for different classes of SDPs for more efficiency.

For the parameters required in Algorithm 1, we set $r = \hat{r} = 0.6$, $\eta = \tau = 0.2$, $\rho = 0.5$, and $\sigma = 10^{-8}$. However, because the initial smoothing parameter $\hat{\epsilon} > 0$ affects the performance of Algorithm 1, it is chosen differently for different classes of problems.

The baseline solvers to be compared with are `SDPLR` (Burer and Monteiro [7]), `ManiSDP` (Wang and Hu [67]), and `SDPNAL+` (Yang et al. [71]).[4] We use their default settings but only set the stopping tolerance to be 10^{-6}.

Finally, we should mention that our algorithm is implemented in MATLAB (R2023b), and all of the numerical experiments are conducted on a Windows PC having 13th Gen Intel cores (i7-13700K) and 64 GB of RAM.

5.2. Testing Examples

Example 1 (MaxCut-SDP). Consider the SDP relaxation (Goemans and Williamson [19]) of the maximum cut problem of a graph that takes the form of

$$\min_{X\in\mathbb{S}^n} \langle C, X\rangle \quad \text{s.t.} \quad \operatorname{diag}(X) = e,\ X \in \mathbb{S}^n_+,$$

where $e \in \mathbb{R}^n$ denotes the vector of all ones and $C := -(\operatorname{diag}(We_n) - W)/4$, with W being the weighted adjacency matrix of the underlying graph. The above SDP problem has been a commonly used test problem for evaluating the performance of different SDP solvers. A popular data set for this problem is the GSET collection of randomly generated graphs. The GSET is available at https://web.stanford.edu/~yyye/yyye/Gset/.

Example 2 (Theta-SDP). Let $G = (V, E)$ be a graph with n nodes V and edges E. A stable set of G is a subset of V containing no adjacent nodes. The stability number $\alpha(G)$ is the cardinality of a maximal stable set of G. More precisely, it holds that

$$\alpha(G) := \max_{x\in\mathbb{R}^n} \{e^T x : x_i x_j = 0, (i,j) \in E, x \in \{0,1\}^n\}.$$

However, computing $\alpha(G)$ is difficult. A notable lower bound of $\alpha(G)$ is called the Lovász theta number (Lovász [32]), which is defined as

$$\theta(G) := \max_{X\in\mathbb{S}^n} \langle ee^T, X\rangle \quad \text{s.t.} \quad \langle E_{ij}, X\rangle = 0, (i,j) \in E,\ \langle I, X\rangle = 1,\ X \in \mathbb{S}^n_+,$$

where $E_{ij} = e_i e_j^T + e_j e_i^T \in \mathbb{S}^n$. For our experiments, the test data sets are chosen from Zhao et al. [75, section 6.3].

Example 3 (BIQ-SDP). The NP-hard binary integer quadratic programming (BIQ) problem has the following form:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle \quad \text{s.t.} \quad x \in \{0,1\}^n.$$

It has many important practical applications because of its modeling power in representing the structure of graphs; see, for example, Kochenberger et al. [30] for a recent survey. Under some mild conditions, Burer [6] showed that the BIQ problem can be reformulated as a completely positive conic programming problem. Because the completely positive cone is numerically intractable, we consider its SDP relaxation:

$$\min_{X\in\mathbb{S}^n, x\in\mathbb{R}^n, \alpha\in\mathbb{R}} \frac{1}{2}\langle Q, X\rangle + \langle c, x\rangle \quad \text{s.t.} \quad \operatorname{diag}(X) = x,\ \alpha = 1, \begin{pmatrix} X & x^T \\ x & \alpha \end{pmatrix} \in \mathbb{S}^{n+1}_+.$$

In our experiments, the matrix Q and the vector c are obtained from the BIQMAC library (Wiegele [69]).

Example 4 (Clustering-SDP). The clustering problem is to group a set of data points into several clusters. The problem is in general NP-hard. According to Peng and Wei [38], we can consider the following SDP relaxation of the clustering problem:

$$\min_{X\in\mathbb{S}^n} -\langle W, X\rangle \quad \text{s.t.} \quad Xe_n = e_n,\ \langle I, X\rangle = k,\ X \in \mathbb{S}^n_+,$$

where W is the affinity matrix whose entries represent the pairwise similarities of the objects in the input data set. In our experiments, the test data sets are obtained from the UCI Machine Learning Repository:

https://archive.ics.uci.edu/datasets. For more information on how we generate the test SDP problems from the raw input data sets, readers are referred to Yang et al. [71].

Example 5 (Tensor-SDP). Consider the following SDP relaxation of a rank-1 tensor approximation problem (Nie and Wang [36]):

$$\max_{y\in\mathbb{R}^{\mathbb{N}^n_m}} \langle f, y\rangle \quad \text{s.t.} \quad M(y)\in\mathbb{S}^n_+,\ \langle g, y\rangle = 1,$$

where $\mathbb{N}^n_m := \{t=(t_1,\dots,t_n)\in\mathbb{N}^n : t_1+\cdots+t_n = m\}$ and $M(y)$ is a linear pencil in y. The dual of the above problem is given by

$$\min_{\gamma\in\mathbb{R},\, X\in\mathbb{S}^n} \gamma \quad \text{s.t.} \quad \gamma g - f = M^*(X),\ X\in\mathbb{S}^n_+,$$

where M^* is the adjoint of M. The above problem can be transformed into a standard SDP; see Yang et al. [71] for more details.

5.3. Computational Results

In this subsection, we present our numerical results for each class of SDP problems in detail. We use tables to summarize our results. We report the sizes (i.e., m and n) of the tested problems, number of iterations, total computational times (i.e., `cpu`) in seconds, relative KKT residues (i.e., η_p, η_d, and η_c), and objective function values (i.e., $\langle C, X\rangle$). In particular, under the column labeled "`Problem`," we first present the name of the tested problem, followed by the matrix dimension n and the number of linear constraints m, respectively. For the number of iterations associated with `SDPNAL+` and Algorithm 1, the first entry represents the number of iterations for the warm-starting phase, the second is the number of main iterations, and the last is the total number of PCG iterations needed for solving the linear systems when computing the search directions. For `SDPLR`, the first number stands for the total number of the ALM iterations, and the second number is the total number of iterations of the limited-memory BFGS method that is applied to solve the (nonconvex) ALM subproblems. Last, for `ManiSDP`, the first item represents the number of iterations for the outer loop, the second is the iteration counts for the Riemannian trust-region method used in `Manopt` (Boumal et al. [5]), and the third number is the total number of matrix-vector multiplications. Moreover, KKT residues that are larger than 10^{-5} are highlighted in bold text. As mentioned before, because the performance of our algorithm may depend sensitively on the choices of parameters $\texttt{tol}_0$, $\hat{\epsilon}$ and κ_c, we also report the values for these parameters in the captions of the presented tables. The computational results for Examples 1–5 are presented in Tables 1–5, respectively.

From the results in Tables 1–4, we observe that our proposed algorithm performs better than `SDPNAL+`. In fact, Algorithm 1 is usually several times more efficient than `SDPNAL+`. These results have indeed shown that Algorithm 1 is efficient. In terms of accuracy, we see that both algorithms are able to compute optimal solutions with $\eta < 10^{-6}$ for almost all of the tested problems, which further shows that both algorithms are robust and suitable for solving SDPs relatively accurately. We observe that for the SDPs in Tables 1–4, their optimal solutions are usually of very low rank. Consequently, the primal constraint nondegenerate condition usually fails to hold, and `SDPNAL+` requires more computational effort to converge than Algorithm 1. For those SDPs with low-rank solutions, one expects `SDPLR` and `ManiSDP` to have excellent performance. Indeed, we observe that `SDPLR` and `ManiSDP` share comparable performance and outperform `SDPNAL+` and Algorithm 1 on Example 1 consisting of max-cut SDPs. We note that the feasible set of the max-cut SDP defines a smooth manifold, and hence, `ManiSDP` essentially applies `Manopt` for solving an optimization over this manifold. For Example 2, we see that `ManiSDP` outperforms `SDPLR` but usually becomes less efficient than `SDPNAL+` and Algorithm 1. Also, we observe that `SDPLR` and `ManiSDP` typically output less accurate solutions in the sense that the KKT residues, especially η_c, are much worse than those of `SDPNAL+` and Algorithm 1. On Example 3, `SDPLR` performs better than `SDPNAL+` and `ManiSDP` but worse than Algorithm 1. Moreover, `ManiSDP` fails to solve all of the tested problems for Example 3. This could be due to the fact that there is no explicit manifold structure for `ManiSDP` to take advantage of.[5] On the other hand, for problems arising from Example 4, there is a fixed-trace constraint that leads to an explicit manifold structure. In this case, `ManiSDP` shows the best performance for problems with $n \le 1000$. However, `ManiSDP` fails to solve the problem `spambase-large.2` with $n = 1{,}500$. We also observe that `SDPLR` fails to solve half of the tested problems. As a conclusion, `SDPNAL+` and Algorithm 1 show excellent performance in terms of efficiency and robustness, whereas the efficiency and robustness of `SDPLR` and `ManiSDP` can be sensitive to the problem being solved.

From Table 5, we see that Algorithm 1 is able to solve all of the problems but perform much worse than `SDPNAL+` in terms of computational time. The reason is that for those SDPs, the ranks of the optimal solutions

Table 1. Computational results for Example 1 with $\texttt{tol}_0 = 10^{-3}$, $\hat{\epsilon} = 10^{-3}$, $\nu = 5 \times 10^3$, and $\kappa_c = 10^2$.

Problem	Solver	Iteration	Time	KKT	Objective
g1	SDPLR	20, 356	2.4	2.2e-08, 0.0e+00, 1.5e-06	−1.2083198e+04
	ManiSDP	3, 74, 991	0.9	2.3e-16, 0.0e+00, 1.1e-12	−1.2083198e+04
800	SDPNAL+	658, 12, 121	75.4	5.4e-14, 1.2e-06, 1.4e-06	−1.2083198e+04
800	Ours	73, 6, 425	11.4	0.0e+00, 7.2e-13, 2.1e-07	−1.2083197e+04
g2	SDPLR	20, 340	2.3	5.8e-08, 0.0e+00, 7.3e-06	−1.2089430e+04
	ManiSDP	3, 80, 1,230	0.9	2.5e-16, 0.0e+00, 4.6e-12	−1.2089430e+04
800	SDPNAL+	734, 11, 125	94.9	2.1e-14, 2.9e-06, 6.0e-07	−1.2089431e+04
800	Ours	73, 6, 430	23.4	0.0e+00, 6.9e-13, 6.2e-07	−1.2089427e+04
g3	SDPLR	19, 293	2.0	3.9e-08, 0.0e+00, 1.7e-06	−1.2084333e+04
	ManiSDP	3, 78, 1,290	1.0	1.8e-16, 0.0e+00, 2.5e-13	−1.2084333e+04
800	SDPNAL+	400, 15, 161	62.3	3.3e-07, 4.9e-07, 2.8e-13	−1.2084333e+04
800	Ours	72, 6, 429	23.2	0.0e+00, 7.1e-13, 5.7e-07	−1.2084330e+04
g4	SDPLR	20, 280	1.9	6.1e-08, 0.0e+00, 1.1e-06	−1.2111450e+04
	ManiSDP	3, 69, 1,289	1.0	2.3e-16, 0.0e+00, 1.4e-12	−1.2111451e+04
800	SDPNAL+	400, 14, 176	62.9	3.2e-07, 8.5e-07, 2.3e-13	−1.2111451e+04
800	Ours	74, 6, 407	23.4	0.0e+00, 7.3e-13, 1.2e-07	−1.2111450e+04
g5	SDPLR	18, 317	2.1	4.2e-08, 0.0e+00, **1.8e-05**	−1.2099887e+04
	ManiSDP	3, 73, 1,239	0.9	2.3e-16, 0.0e+00, 6.7e-14	−1.2099887e+04
800	SDPNAL+	810, 11, 131	104.9	6.2e-09, 1.3e-06, 5.6e-08	−1.2099887e+04
800	Ours	73, 6, 414	23.0	0.0e+00, 6.2e-13, 2.9e-07	−1.2099886e+04
g22	SDPLR	21, 396	4.6	3.5e-08, 0.0e+00, 2.1e-06	−1.4135946e+04
	ManiSDP	3, 76, 1,659	3.7	2.3e-16, 0.0e+00, 7.5e-13	−1.4135946e+04
2,000	SDPNAL+	900, 13, 346	803.0	1.9e-07, 7.0e-07, 3.7e-13	−1.4135946e+04
2,000	Ours	126, 7, 519	230.0	0.0e+00, 1.7e-14, 1.7e-07	−1.4135945e+04
g23	SDPLR	21, 705	7.8	4.3e-08, 0.0e+00, 8.7e-07	−1.4142108e+04
	ManiSDP	4, 83, 1,888	5.1	2.1e-16, 0.0e+00, 1.6e-13	−1.4142109e+04
2,000	SDPNAL+	814, 11, 287	712.1	1.6e-09, 4.8e-06, 7.4e-07	−1.4142107e+04
2,000	Ours	128, 8, 635	251.5	0.0e+00, 5.0e-16, 2.9e-07	−1.4142108e+04
g24	SDPLR	19, 430	5.0	2.9e-08, 0.0e+00, 7.5e-07	−1.4140856e+04
	ManiSDP	3, 68, 1,137	3.1	2.2e-16, 0.0e+00, 1.8e-13	−1.4140856e+04
2,000	SDPNAL+	808, 15, 261	674.1	2.8e-09, 3.1e-06, 1.0e-07	−1.4140855e+04
2,000	Ours	128, 6, 404	204.5	0.0e+00, 5.5e-13, 7.0e-07	−1.4140847e+04
g25	SDPLR	21, 473	5.3	3.4e-08, 0.0e+00, 3.0e-07	−1.4144245e+04
	ManiSDP	4, 90, 2,442	5.7	2.2e-16, 0.0e+00, 1.4e-12	−1.4144245e+04
2,000	SDPNAL+	800, 13, 234	704.0	7.3e-08, 6.2e-07, 2.9e-13	−1.4144245e+04
2,000	Ours	109, 6, 373	177.8	0.0e+00, 7.2e-13, 8.5e-07	−1.4144234e+04
g26	SDPLR	20, 660	7.2	2.5e-08, 0.0e+00, 1.2e-06	−1.4132871e+04
	ManiSDP	3, 81, 1,707	3.9	2.3e-16, 0.0e+00, 2.7e-13	−1.4132871e+04
2,000	SDPNAL+	800, 15, 254	612.8	3.9e-08, 3.5e-07, 5.1e-13	−1.4132871e+04
2,000	Ours	128, 8, 590	204.4	0.0e+00, 4.7e-16, 2.8e-07	−1.4132870e+04
g43	SDPLR	21, 375	1.7	1.3e-08, 0.0e+00, 7.5e-06	−7.0322218e+03
	ManiSDP	3, 69, 1,017	1.0	2.0e-16, 0.0e+00, 2.3e-13	−7.0322218e+03
1,000	SDPNAL+	400, 16, 197	96.6	6.4e-07, 2.5e-07, 4.2e-13	−7.0322218e+03
1,000	Ours	102, 5, 417	37.9	0.0e+00, 2.1e-11, 3.4e-07	−7.0322209e+03
g44	SDPLR	19, 313	1.7	3.2e-08, 0.0e+00, 8.6e-07	−7.0278846e+03
	ManiSDP	3, 68, 958	0.4	2.4e-16, 0.0e+00, 3.7e-12	−7.0278847e+03
1,000	SDPNAL+	400, 14, 187	68.1	9.3e-07, 9.9e-07, 2.6e-13	−7.0278849e+03
1,000	Ours	100, 5, 380	36.5	0.0e+00, 2.1e-11, 8.9e-07	−7.0278819e+03
g45	SDPLR	18, 313	1.4	5.6e-08, 0.0e+00, **1.1e-05**	−7.0247814e+03
	ManiSDP	3, 79, 1,214	1.1	1.9e-16, 0.0e+00, 1.2e-13	−7.0247814e+03
1,000	SDPNAL+	555, 14, 200	103.8	9.5e-07, 8.5e-07, 2.3e-07	−7.0247857e+03
1,000	Ours	99, 6, 456	19.4	0.0e+00, 6.3e-13, 2.0e-07	−7.0247812e+03
g46	SDPLR	19, 347	1.6	2.4e-08, 0.0e+00, 1.7e-06	−7.0299330e+03
	ManiSDP	3, 99, 3,958	2.5	2.4e-16, 0.0e+00, 8.3e-10	−7.0299331e+03
1,000	SDPNAL+	400, 16, 187	95.8	7.1e-07, 2.6e-07, 4.0e-13	−7.0299332e+03
1,000	Ours	102, 6, 419	38.7	0.0e+00, 6.4e-13, 3.7e-07	−7.0299329e+03
g47	SDPLR	19, 392	1.8	4.7e-08, 0.0e+00, 1.3e-06	−7.0366606e+03
	ManiSDP	3, 85, 2,496	1.8	2.0e-16, 0.0e+00, 3.2e-12	−7.0366605e+03
1,000	SDPNAL+	610, 14, 177	133.8	4.7e-14, 1.2e-06, 7.4e-07	−7.0366613e+03
1,000	Ours	100, 6, 473	40.6	0.0e+00, 6.5e-13, 3.4e-07	−7.0366600e+03
g48	SDPLR	19, 302	1.9	3.1e-08, 0.0e+00, 8.4e-08	−6.0000001e+03
	ManiSDP	2, 61, 434	1.8	1.7e-16, 0.0e+00, 7.8e-15	−6.0000000e+03

Table 1. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
3,000	SDPNAL+	2,141, 10, 22	3,343.5	1.6e-12, **3.8e-05**, **7.0e-04**	−6.0811714e+03
3,000	Ours	50, 11, 22	259.4	0.0e+00, 1.0e-19, 5.0e-07	−5.9999994e+03
g49	SDPLR	20, 408	2.3	3.3e-08, 0.0e+00, 1.0e-07	−6.0000000e+03
	ManiSDP	2, 62, 524	1.8	1.9e-16, 0.0e+00, 7.0e-15	−6.0000000e+03
3,000	SDPNAL+	1,093, 8, 18	828.0	3.6e-12, **1.1e-05**, **1.5e-05**	−6.0025751e+03
3,000	Ours	30, 13, 30	93.6	0.0e+00, 1.9e-16, 4.4e-08	−6.0000000e+03
g50	SDPLR	20, 545	3.0	2.4e-08, 0.0e+00, 6.6e-07	−5.9881721e+03
	ManiSDP	2, 72, 778	1.9	2.1e-16, 0.0e+00, 1.6e-15	−5.9881721e+03
3,000	SDPNAL+	1,442, 9, 28	1,150.8	7.5e-13, **2.5e-05**, **9.1e-04**	−6.0528634e+03
3,000	Ours	50, 10, 20	95.6	0.0e+00, 1.5e-16, 1.8e-07	−5.9881720e+03

Table 2. Computational results for Example 2 with $\texttt{tol}_0 = 10^{-4}$, $\hat{\epsilon} = 10^{-3}$, $\nu = 5 \times 10^3$, and $\kappa_c = 10^3$.

Problem	Solver	Iteration	Time	KKT	Objective
theta4	SDPLR	17, 34,750	20.3	1.0e-06, 0.0e+00, 2.0e-07	−5.0321375e+01
	ManiSDP	50, 200, 2,871	1.9	1.6e-07, 9.0e-18, 1.8e-06	−5.0321226e+01
200	SDPNAL+	451, 9, 352	3.5	3.9e-08, 1.6e-07, 2.7e-16	−5.0321223e+01
1,949	Ours	198, 2, 75	1.4	1.8e-19, 5.6e-08, 1.1e-07	−5.0321190e+01
theta42	SDPLR	18, 18,742	14.5	1.0e-06, 0.0e+00, **1.2e-04**	−2.3931747e+01
	ManiSDP	110, 440, 6,975	5.8	2.0e-07, 1.3e-17, **2.5e-05**	−2.3931705e+01
200	SDPNAL+	176, 4, 496	1.6	1.2e-07, 9.1e-08, 4.6e-16	−2.3931708e+01
5,986	Ours	79, 2, 99	0.6	2.1e-16, 7.8e-08, 1.3e-07	−2.3931708e+01
theta6	SDPLR	17, 46,024	97.4	1.0e-06, 0.0e+00, **2.0e-05**	−6.3477318e+01
	ManiSDP	40, 160, 2,188	2.3	4.9e-07, 7.7e-18, 4.3e-06	−6.3477018e+01
300	SDPNAL+	286, 4, 152	5.5	6.3e-07, 3.9e-07, 3.0e-16	−6.3477158e+01
4,375	Ours	203, 2, 46	3.8	2.7e-19, 7.9e-08, 9.1e-08	−6.3477053e+01
theta62	SDPLR	18, 36,237	119.5	1.0e-06, 0.0e+00, **3.5e-04**	−2.9641355e+01
	ManiSDP	40, 160, 1,988	3.1	4.2e-07, 1.3e-17, **2.1e-05**	−2.9641263e+01
300	SDPNAL+	180, 3, 365	4.4	3.8e-07, 8.0e-07, 5.0e-16	−2.9641251e+01
13,390	Ours	81, 2, 134	2.1	1.7e-16, 9.0e-08, 7.1e-08	−2.9641252e+01
theta8	SDPLR	17, 47,743	169.8	1.0e-06, 0.0e+00, **3.4e-05**	−7.3953750e+01
	ManiSDP	50, 200, 3,023	7.0	2.4e-07, 7.4e-18, 4.4e-06	−7.3953537e+01
400	SDPNAL+	299, 3, 211	10.8	2.2e-07, 7.7e-07, 3.9e-16	−7.3953562e+01
7,905	Ours	218, 2, 61	7.5	1.9e-16, 4.8e-08, 1.0e-07	−7.3953549e+01
theta82	SDPLR	17, 60,519	242.6	9.9e-07, 0.0e+00, 1.1e-06	−3.4367028e+01
	ManiSDP	50, 200, 2,530	11.4	3.9e-07, 1.2e-17, **1.9e-05**	−3.4366870e+01
400	SDPNAL+	179, 3, 351	8.2	1.0e-07, 2.1e-07, 5.8e-16	−3.4366894e+01
23,872	Ours	85, 2, 64	3.7	9.7e-17, 8.7e-08, 4.3e-08	−3.4366895e+01
theta83	SDPLR	18, 59,260	121.2	9.6e-07, 0.0e+00, 1.3e-06	−2.0301976e+01
	ManiSDP	60, 240, 2,418	19.4	6.7e-07, 1.3e-17, **3.1e-05**	−2.0301920e+01
400	SDPNAL+	204, 1, 296	5.2	6.6e-09, **2.7e-05**, 1.0e-05	−2.0386342e+01
39,862	Ours	57, 2, 154	2.2	4.9e-17, 8.8e-08, 1.0e-07	−2.0301922e+01
theta10	SDPLR	17, 57,594	251.3	9.9e-07, 0.0e+00, 2.7e-07	−8.3806226e+01
	ManiSDP	80, 320, 5,293	33.4	3.1e-07, 9.2e-18, 4.3e-06	−8.3805862e+01
500	SDPNAL+	200, 4, 596	8.3	1.5e-07, 5.1e-07, 4.2e-16	−8.3805970e+01
12,470	Ours	224, 2, 54	6.1	5.4e-17, 6.8e-08, 4.1e-08	−8.3805954e+01
theta102	SDPLR	19, 74,123	613.1	9.6e-07, 0.0e+00, 8.8e-07	−3.8390684e+01
	ManiSDP	60, 240, 2,996	29.9	4.5e-07, 1.2e-17, 1.0e-05	−3.8390562e+01
500	SDPNAL+	220, 1, 244	14.1	9.8e-08, **1.9e-05**, 9.4e-06	−3.7669207e+01
37,467	Ours	88, 2, 88	6.3	1.6e-16, 4.4e-08, 4.1e-08	−3.8390548e+01
theta103	SDPLR	18, 70,434	798.2	1.0e-06, 0.0e+00, 1.3e-06	−2.2528678e+01
	ManiSDP	90, 360, 4,934	67.5	6.4e-07, 1.2e-17, **3.0e-05**	−2.2528544e+01
500	SDPNAL+	165, 3, 510	16.0	6.6e-07, 1.7e-07, 7.4e-16	−2.2528560e+01
62,516	Ours	56, 3, 152	7.4	2.7e-16, 4.4e-08, 1.2e-07	−2.2528635e+01
theta104	SDPLR	18, 76,791	1,023.9	9.9e-07, 0.0e+00, 1.9e-04	−1.3336230e+01
	ManiSDP	90, 360, 3,915	62.9	9.0e-07, 1.1e-17, **5.1e-05**	−1.3336183e+01
500	SDPNAL+	166, 3, 651	16.6	9.1e-07, 2.6e-07, 7.4e-16	−1.3336135e+01
87,245	Ours	62, 3, 184	8.8	1.6e-16, 4.3e-08, 1.1e-07	−1.3336215e+01
theta12	SDPLR	17, 65,564	580.0	1.0e-06, 0.0e+00, 2.9e-07	−9.2801941e+01

Table 2. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
	ManiSDP	50, 200, 2,750	28.7	4.6e-07, 8.4e-18, 2.1e-06	−9.2801755e+01
600	SDPNAL+	400, 8, 428	38.4	1.7e-07, 5.0e-07, 4.4e-16	−9.2801687e+01
17,979	Ours	240, 2, 71	19.4	3.5e-16, 6.8e-08, 2.4e-07	−9.2801658e+01
theta123	SDPLR	18, 93,187	1,573.4	9.9e-07, 0.0e+00, 9.9e-07	−2.4668771e+01
	ManiSDP	60, 240, 2,488	53.8	7.8e-07, 1.2e-17, **1.6e-05**	−2.4668707e+01
600	SDPNAL+	166, 3, 455	11.3	1.6e-07, 6.1e-08, 6.4e-16	−2.4668652e+01
90,020	Ours	57, 3, 202	6.3	1.2e-16, 7.9e-08, 6.8e-08	−2.4668773e+01
theta162	SDPLR	13, 124,123	3,600.2	**1.0e-04**, 0.0e+00, **1.5e-05**	−3.7049199e+01
	ManiSDP	90, 360, 4,632	168.6	8.8e-07, 1.1e-17, **1.2e-05**	−3.7009840e+01
800	SDPNAL+	169, 2, 689	22.2	9.4e-09, 1.1e-06, 5.7e-16	−3.7009736e+01
127,600	Ours	71, 2, 66	7.0	1.4e-16, 5.8e-08, 4.4e-08	−3.7009811e+01
MANN-a27	SDPLR				
	ManiSDP	35, 139, 1,287	4.6	1.7e-07, 0.0e+00, 1.1e-07	−1.3276315e+02
378	SDPNAL+	300, 4, 25	8.8	4.7e-07, 1.7e-07, 2.0e-16	−1.3276116e+02
703	Ours	436, 2, 10	9.9	1.9e-16, 8.3e-08, 5.3e-08	−1.3276282e+02
san200-0.7-1	SDPLR	16, 1,737	2.5	9.3e-07, 0.0e+00, 2.6e-07	−3.0000125e+01
	ManiSDP	15, 60, 344	0.5	4.0e-07, 6.5e-18, 4.7e-08	−3.0000006e+01
200	SDPNAL+	300, 5, 671	3.6	1.3e-07, 3.6e-07, 4.9e-17	−2.9999993e+01
5,971	Ours	1,000, 3, 6	8.3	2.1e-16, 2.0e-08, 7.2e-08	−2.9999956e+01
sanr200-0.7	SDPLR	18, 24,032	35.7	9.9e-07, 0.0e+00, **1.7e-04**	−2.3836200e+01
	ManiSDP	40, 160, 2,235	3.6	2.3e-07, 1.3e-17, **3.0e-05**	−2.3836177e+01
200	SDPNAL+	175, 4, 563	2.5	7.2e-08, 2.3e-07, 4.4e-16	−2.3836157e+01
6,033	Ours	96, 2, 100	1.3	3.2e-19, 8.4e-08, 1.7e-07	−2.3836156e+01
c-fat200-1	SDPLR	16, 23,094	54.8	1.0e-06, 0.0e+00, **1.4e-05**	−1.2000024e+01
	ManiSDP	60, 240, 3,354	3.4	1.9e-07, 1.3e-17, **1.5e-05**	−1.2000017e+01
200	SDPNAL+	229, 2, 221	2.5	5.1e-08, 1.8e-07, 1.5e-16	−1.1999999e+01
18,367	Ours	257, 2, 22	2.3	7.0e-17, 7.6e-08, 3.7e-07	−1.1999981e+01
hamming-8-4	SDPLR	14, 834	2.2	9.8e-07, 0.0e+00, 4.7e-06	−1.5999965e+01
	ManiSDP	40, 159, 1,727	3.7	1.2e-07, 0.0e+00, 4.5e-06	−1.6000002e+01
256	SDPNAL+	172, 1, 2	2.5	2.7e-11, 6.1e-07, 1.3e-15	−1.6000000e+01
11,777	Ours	72, 2, 6	1.0	3.4e-21, 8.5e-08, 3.3e-08	−1.5999995e+01
hamming-9-8	SDPLR	13, 252	1.0	9.1e-07, 0.0e+00, 2.3e-06	−2.2399920e+02
	ManiSDP	22, 80, 326	2.8	1.6e-07, 0.0e+00, 7.4e-08	−2.2400000e+02
512	SDPNAL+	200, 3, 14	18.1	1.9e-07, 7.9e-08, 1.3e-16	−2.2400008e+02
2,305	Ours	1,000, 2, 7	45.7	3.8e-16, 8.1e-09, 8.3e-08	−2.2399982e+02
hamming-10-2	SDPLR	14, 746	17.5	9.9e-07, 0.0e+00, 1.5e-06	−1.0240013e+02
	ManiSDP	34, 127, 936	35.5	2.8e-08, 0.0e+00, 3.7e-07	−1.0240000e+02
1,024	SDPNAL+	200, 3, 12	54.9	2.8e-09, 1.8e-08, 1.6e-15	−1.0240000e+02
23,041	Ours	447, 2, 6	82.6	3.8e-17, 6.2e-08, 8.7e-09	−1.0240000e+02
hamming-7-5-6	SDPLR	13, 128	0.1	7.1e-07, 0.0e+00, 7.3e-07	−4.2666727e+01
	ManiSDP	13, 52, 181	0.2	8.2e-08, 0.0e+00, 3.8e-09	−4.2666667e+01
128	SDPNAL+	300, 2, 6	0.8	3.2e-08, 7.0e-07, 3.2e-16	−4.2666663e+01
1,793	Ours	336, 2, 7	1.2	2.3e-20, 8.3e-08, 1.6e-07	−4.2666639e+01
hamming-8-3-4	SDPLR	14, 197	0.6	9.5e-07, 0.0e+00, 3.0e-06	−2.5600151e+01
	ManiSDP	22, 88, 501	1.2	1.5e-08, 0.0e+00, 2.4e-07	−2.5600000e+01
256	SDPNAL+	117, 2, 6	1.6	5.8e-08, 3.8e-07, 3.9e-16	−2.5600004e+01
16,129	Ours	107, 2, 6	1.4	9.3e-21, 6.7e-08, 6.1e-08	−2.5599998e+01
hamming-9-5-6	SDPLR	11, 135	1.8	8.0e-07, 0.0e+00, 1.4e-06	−8.5333420e+01
	ManiSDP	11, 41, 96	1.3	1.0e-07, 0.0e+00, 2.6e-14	−8.5333333e+01
512	SDPNAL+	200, 3, 13	14.1	9.7e-08, 3.5e-07, 8.4e-17	−8.5333343e+01
53,761	Ours	684, 2, 7	33.5	5.5e-17, 8.3e-08, 9.6e-08	−8.5333287e+01
brock200-1	SDPLR	17, 25,568	36.1	9.3e-07, 0.0e+00, 4.4e-06	−2.7456696e+01
	ManiSDP	170, 680, 11,467	17.2	2.4e-07, 1.3e-17, **2.3e-05**	−2.7456635e+01
200	SDPNAL+	187, 4, 283	2.0	7.7e-07, 2.7e-07, 4.0e-16	−2.7456674e+01
5,067	Ours	100, 2, 98	1.1	7.0e-17, 8.2e-08, 1.3e-07	−2.7456636e+01
brock200-4	SDPLR	18, 20,463	16.1	9.9e-07, 0.0e+00, **6.1e-04**	−2.1293493e+01
	ManiSDP	130, 520, 8,155	13.6	2.6e-07, 1.2e-17, **3.5e-05**	−2.1293470e+01
200	SDPNAL+	273, 4, 666	2.4	2.4e-07, 2.3e-07, 4.6e-16	−2.1293476e+01
6,812	Ours	81, 2, 200	0.8	1.3e-19, 8.8e-08, 1.4e-07	−2.1293473e+01
brock400-1	SDPLR	17, 53,843	239.4	9.8e-07, 0.0e+00, **1.3e-04**	−3.9701980e+01
	ManiSDP	130, 520, 8,568	47.2	2.9e-07, 1.1e-17, **1.1e-05**	−3.9701882e+01

Table 2. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
400	SDPNAL+	194, 3, 347	4.9	1.6e-07, 3.9e-07, 5.6e-16	−3.9701901e+01
20,078	Ours	106, 2, 71	2.6	3.9e-16, 5.0e-08, 1.9e-07	−3.9701897e+01
p-hat300-1	SDPLR	18, 125,577	567.8	1.0e-06, 0.0e+00, 2.4e-06	−1.0068009e+01
	ManiSDP	160, 640, 13,115	46.6	1.7e-06, 1.8e-17, **3.0e-04**	−1.0067938e+01
300	SDPNAL+	191, 6, 6,728	25.5	8.7e-07, 5.9e-07, 5.9e-16	−1.0067987e+01
33,918	Ours	147, 3, 232	4.4	4.1e-19, 1.8e-08, 2.6e-07	−1.0068325e+01
1dc.128	SDPLR	19, 338,023	213.7	1.0e-06, 0.0e+00, **3.0e-05**	−1.6841834e+01
	ManiSDP	100, 400, 8,000	3.6	7.9e-06, 3.7e-18, **1.1e-02**	−1.6841905e+01
128	SDPNAL+	500, 13, 8,137	6.7	5.9e-07, 5.8e-06, 8.0e-07	−1.6839378e+01
1,472	Ours	733, 50, 1,226	4.9	5.7e-17, 1.6e-14, 7.6e-06	−1.6842320e+01
1et.128	SDPLR	17, 14,981	7.1	9.9e-07, 0.0e+00, 1.2e-06	−2.9230866e+01
	ManiSDP	190, 760, 13,432	6.5	1.8e-07, 6.4e-18, **2.6e-05**	−2.9230901e+01
128	SDPNAL+	273, 3, 69	1.0	5.4e-07, 8.0e-07, 7.9e-16	−2.9230893e+01
673	Ours	204, 2, 34	0.8	1.1e-16, 6.4e-08, 2.2e-07	−2.9230862e+01
1tc.128	SDPLR	17, 5,716	2.6	9.9e-07, 0.0e+00, 3.9e-07	−3.8000024e+01
	ManiSDP	29, 116, 959	0.4	2.5e-09, 7.1e-18, 2.5e-08	−3.8000000e+01
128	SDPNAL+	300, 5, 61	1.0	2.5e-07, 1.5e-07, 7.5e-17	−3.8000007e+01
513	Ours	628, 2, 11	2.3	2.8e-16, 7.8e-08, 6.5e-07	−3.7999594e+01
1zc.128	SDPLR	15, 5,583	3.0	9.2e-07, 0.0e+00, 1.7e-06	−2.0666670e+01
	ManiSDP	90, 360, 5,442	2.7	1.4e-07, 6.3e-18, **3.0e-05**	−2.0666672e+01
128	SDPNAL+	200, 3, 30	0.8	7.6e-07, 4.0e-07, 4.8e-16	−2.0666696e+01
1,121	Ours	120, 2, 10	0.5	1.7e-16, 8.3e-08, 9.2e-08	−2.0666657e+01
1dc.256	SDPLR	17, 79,800	123.3	9.9e-07, 0.0e+00, 7.2e-06	−2.9999957e+01
	ManiSDP	70, 280, 5,911	4.7	**1.3e-05**, 2.4e-18, **3.1e-04**	−2.9993623e+01
256	SDPNAL+	1,000, 10, 2,575	14.4	8.6e-10, 3.1e-07, 6.7e-17	−3.0000000e+01
3,840	Ours	1,000, 50, 1,205	18.2	3.9e-17, 2.2e-16, 1.1e-06	−3.0000267e+01
1et.256	SDPLR	19, 223,320	268.4	1.0e-06, 0.0e+00, 4.8e-07	−5.5114388e+01
	ManiSDP	80, 320, 5,545	4.5	4.2e-06, 3.1e-18, **8.8e-04**	−5.5114535e+01
256	SDPNAL+	816, 12, 7,049	23.0	2.0e-08, 9.9e-07, 1.4e-09	−5.5113816e+01
1,665	Ours	1,000, 2, 100	11.5	1.6e-16, 2.6e-09, 2.0e-07	−5.5129462e+01
1tc.256	SDPLR	18, 403,838	453.2	1.0e-06, 0.0e+00, 1.1e-06	−6.3399970e+01
	ManiSDP	110, 440, 8,557	6.3	3.4e-06, 7.6e-18, **4.7e-04**	−6.3400854e+01
256	SDPNAL+	508, 18, 13,383	26.8	7.1e-07, 2.9e-07, 1.9e-09	−6.3399050e+01
1,313	Ours	461, 23, 1,069	8.6	3.9e-16, 3.2e-16, 9.7e-07	−6.3403311e+01
1zc.256	SDPLR	18, 14,140	20.2	9.5e-07, 0.0e+00, 8.9e-07	−3.8000045e+01
	ManiSDP	130, 520, 7,586	6.9	1.3e-07, 0.0e+00, 6.7e-06	−3.7999995e+01
256	SDPNAL+	266, 1, 18	3.4	6.8e-10, 4.2e-06, 4.1e-06	−3.4463072e+01
2,817	Ours	179, 2, 10	2.1	5.1e-21, 8.4e-08, 4.5e-08	−3.7999987e+01
1dc.512	SDPLR	20, 618,683	3,600.1	1.3e-06, 0.0e+00, 1.6e-06	−5.3031234e+01
	ManiSDP	190, 760, 15,197	60.1	4.9e-06, 1.0e-18, **3.6e-05**	−5.3031630e+01
512	SDPNAL+	300, 19, 13,376	109.6	7.7e-07, 2.3e-06, 3.9e-08	−5.3025097e+01
9,728	Ours	1,000, 2, 85	50.7	5.5e-17, 5.3e-09, 7.5e-08	−5.3034003e+01
1et.512	SDPLR	18, 257,338	1,025.3	1.0e-06, 0.0e+00, 4.9e-07	−1.0442385e+02
	ManiSDP	400, 1,600, 34,062	183.3	7.1e-06, 1.2e-18, **1.9e-04**	−1.0441931e+02
512	SDPNAL+	300, 11, 7,042	69.6	2.1e-07, 8.6e-07, 9.1e-09	−1.0442327e+02
4,033	Ours	423, 2, 100	21.8	1.1e-16, 6.7e-08, 2.3e-07	−1.0444155e+02
1tc.512	SDPLR	17, 998,043	3,600.1	1.6e-06, 0.0e+00, 3.1e-07	−1.1340137e+02
	ManiSDP	320, 1,280, 27,992	154.7	6.8e-06, 5.3e-18, **2.6e-04**	−1.1340159e+02
512	SDPNAL+	300, 24, 19,872	174.3	5.9e-06, **1.4e-05**, 4.2e-16	−1.1340509e+02
3,265	Ours	1,000, 5, 435	56.0	4.9e-16, 5.7e-12, 8.6e-07	−1.1347915e+02
2dc.512	SDPLR	16, 344,809	3,600.1	**5.0e-05**, 0.0e+00, 1.4e-06	−1.1794474e+01
	ManiSDP	30, 120, 709	7.3	**4.1e-03**, 3.7e-17, **5.4e-01**	−1.1659025e+01
512	SDPNAL+	300, 12, 8,003	111.0	1.8e-13, 1.2e-06, 3.9e-08	−1.1775990e+01
54,896	Ours	1,000, 50, 935	92.9	7.1e-16, 9.1e-15, 2.9e-06	−1.1773819e+01
1zc.512	SDPLR	17, 63,237	300.8	1.0e-06, 0.0e+00, 2.2e-06	−6.8750317e+01
	ManiSDP	90, 360, 5,798	50.4	6.0e-07, 3.1e-18, 7.9e-06	−6.8750010e+01
512	SDPNAL+	200, 4, 228	15.2	2.8e-07, 1.9e-07, 7.8e-16	−6.8749972e+01
6,913	Ours	271, 2, 10	14.1	1.6e-16, 6.7e-08, 2.3e-08	−6.8749986e+01
1dc.1024	SDPLR	18, 162,116	3,600.2	**7.6e-04**, 0.0e+00, 9.6e-07	−9.7153056e+01
	ManiSDP	350, 1,400, 22,834	626.9	6.7e-07, 5.9e-19, 2.1e-06	−9.5985469e+01
1,024	SDPNAL+	200, 15, 8,785	307.6	7.5e-07, 9.5e-07, 1.8e-16	−9.5985603e+01

Table 2. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
24,064	Ours	1,000, 2, 76	190.9	2.7e-16, 3.8e-09, 4.0e-08	−9.5991058e+01
1et.1024	SDPLR	17, 221,471	3,600.2	**2.8e-04**, 0.0e+00, 1.3e-06	−1.8512627e+02
	ManiSDP	130, 519, 8,664	253.3	**1.5e-05**, 1.4e-18, **5.5e-05**	−1.8421860e+02
1,024	SDPNAL+	1,042, 26, 17,176	878.4	7.4e-07, 2.5e-06, 2.7e-08	−1.8418886e+02
9,601	Ours	396, 2, 100	81.8	1.4e-18, 6.6e-08, 5.3e-08	−1.8439511e+02
1tc.1024	SDPLR				
	ManiSDP	140, 559, 10,208	149.6	8.1e-06, 4.9e-18, **3.4e-05**	−2.0631534e+02
1,024	SDPNAL+	400, 20, 21,555	818.6	2.3e-06, 2.5e-06, 1.6e-16	−2.0630720e+02
7,937	Ours	1,000, 2, 100	196.1	3.1e-16, 1.1e-08, 1.5e-07	−2.0652662e+02
1zc.1024	SDPLR	16, 105,125	2,111.4	9.9e-07, 0.0e+00, 3.8e-07	−1.2866717e+02
	ManiSDP	94, 375, 5,589	103.4	4.8e-07, 0.0e+00, 9.2e-07	−1.2866664e+02
1,024	SDPNAL+	200, 7, 866	84.2	2.1e-07, 4.3e-07, 3.3e-16	−1.2866668e+02
16,641	Ours	386, 2, 10	72.2	1.5e-16, 8.3e-08, 1.3e-08	−1.2866667e+02
1dc.2048	SDPLR	19, 37,396	3,601.1	**1.4e-02**, 0.0e+00, 2.2e-06	−2.5493425e+02
	ManiSDP	450, 1,799, 29,653	1,638.1	5.5e-07, 0.0e+00, 7.0e-07	−1.7473052e+02
2,048	SDPNAL+	200, 16, 10,181	1,600.0	9.2e-07, 4.6e-07, 1.0e-16	−1.7473028e+02
58,368	Ours	1,000, 2, 100	1,109.6	3.7e-16, 1.7e-08, 3.7e-08	−1.7480655e+02
1et.2048	SDPLR				
	ManiSDP	220, 878, 14,582	972.7	**1.1e-05**, 3.0e-18, **8.1e-05**	−3.4205781e+02
2,048	SDPNAL+	586, 23, 19,822	3,866.9	8.0e-07, 7.4e-07, 6.9e-11	−3.4203646e+02
22,529	Ours	1,000, 2, 100	917.0	5.3e-16, 1.3e-08, 8.1e-08	−3.4289319e+02
1tc.2048	SDPLR				
	ManiSDP	170, 678, 9,201	585.1	**9.0e-05**, 3.2e-18, **7.2e-04**	−3.7470264e+02
2,048	SDPNAL+	2,105, 29, 28,120	6,624.0	9.1e-07, 7.5e-07, 9.3e-11	−3.7464678e+02
18,945	Ours	1,000, 2, 100	982.7	5.4e-17, 2.5e-08, 1.6e-07	−3.7582591e+02
1zc.2048	SDPLR	19, 41,500	3,601.1	**3.7e-03**, 0.0e+00, 2.4e-08	−2.5977665e+02
	ManiSDP	51, 202, 1,990	142.9	3.1e-07, 1.3e-18, 4.9e-07	−2.3740022e+02
2,048	SDPNAL+	200, 5, 1,046	573.3	2.3e-07, 6.7e-07, 6.7e-16	−2.3740004e+02
39,425	Ours	743, 2, 12	714.4	6.4e-16, 5.6e-08, 1.6e-08	−2.3739997e+02
1zc.4096	SDPLR	18, 8,435	3,609.5	**6.1e-03**, 0.0e+00, 1.6e-08	−5.5890705e+02
	ManiSDP	180, 719, 6,955	2,600.2	**2.9e-04**, 0.0e+00, **4.6e-04**	−4.4869891e+02
4,096	SDPNAL+	200, 7, 1,248	4,647.1	3.8e-08, 2.9e-07, 5.3e-16	−4.4916678e+02
92,161	Ours	300, 5, 120	2,859.3	1.6e-15, 3.5e-07, 9.9e-09	−4.4917208e+02

Table 3. Computational results for Example 3 with $\texttt{tol}_0 = 5 \times 10^{-1}$, $\hat{\epsilon} = 10^{-3}$, $\nu = 5 \times 10^3$, and $\kappa_c = 10^2$.

Problem	Solver	Iteration	Time	KKT	Objective
be150.3.1	SDPLR	20, 5,189	4.4	8.3e-07, 0.0e+00, 1.2e-08	−2.0175194e+04
	ManiSDP	290, 1,447, 16,714	5.0	**1.2e+01**, 3.0e-17, **2.9e-05**	−6.8971132e+04
151	SDPNAL+	655, 26, 5,808	7.7	1.3e-11, 4.8e-06, 9.8e-10	−2.0175328e+04
151	Ours	140, 11, 536	1.0	3.3e-16, 5.3e-12, 5.9e-07	−2.0174310e+04
be150.8.1	SDPLR	21, 10,903	1.3	5.8e-07, 0.0e+00, 4.4e-08	−2.9671661e+04
	ManiSDP	550, 2,750, 35,913	10.1	**2.5e+00**, 2.9e-17, 9.5e-06	−5.5643145e+04
151	SDPNAL+	800, 25, 6,219	9.4	3.8e-07, 7.6e-07, 2.2e-15	−2.9671637e+04
151	Ours	127, 9, 341	0.8	3.0e-16, 8.9e-12, 9.2e-08	−2.9671499e+04
be200.3.1	SDPLR	21, 8,561	14.6	1.0e-06, 0.0e+00, 1.8e-05	−2.8275524e+04
	ManiSDP	270, 1,350, 13,673	5.6	**2.0e+00**, 3.0e-17, **1.6e-05**	−3.1643403e+04
201	SDPNAL+	1,713, 25, 8,606	28.0	9.6e-09, 1.0e-06, 6.7e-11	−2.8275594e+04
201	Ours	157, 8, 399	1.4	3.2e-16, 1.7e-10, 3.1e-07	−2.8274789e+04
be200.8.1	SDPLR	21, 15,424	6.4	5.8e-07, 0.0e+00, 6.5e-09	−5.1718246e+04
	ManiSDP	130, 640, 3,014	1.9	**4.0e+00**, 3.9e-17, 8.0e-06	−1.7317299e+05
201	SDPNAL+	2,912, 24, 5,963	31.1	6.6e-09, 1.5e-06, 2.0e-10	−5.1718260e+04
201	Ours	181, 8, 416	1.6	3.6e-16, 1.1e-11, 8.4e-07	−5.1713602e+04
be250.1	SDPLR	21, 11,492	12.0	9.7e-07, 0.0e+00, 8.2e-09	−2.5551688e+04
	ManiSDP	170, 850, 7,181	3.8	**1.8e+00**, 6.0e-17, **1.6e-05**	−1.6307124e+04
251	SDPNAL+	2,100, 30, 10,855	44.7	4.4e-07, 4.7e-07, 9.5e-16	−2.5551657e+04
251	Ours	219, 8, 370	2.4	4.0e-16, 3.6e-13, 3.2e-07	−2.5550756e+04
bqp250-1	SDPLR	22, 8,778	9.3	5.9e-07, 0.0e+00, **1.3e-05**	−4.8732291e+04

Table 3. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
	ManiSDP	60, 300, 380	0.8	**5.0e+03**, 0.0e+00, **1.5e-03**	−1.7966528e+08
251	SDPNAL+	2,100, 31, 8,420	40.8	2.8e-07, 3.8e-07, 3.8e-16	−4.8732371e+04
251	Ours	217, 8, 411	2.5	3.9e-16, 3.8e-13, 9.3e-07	−4.8726367e+04
bqp500-1	SDPLR	23, 15,191	76.9	5.7e-07, 0.0e+00, 5.6e-06	−1.2840267e+05
	ManiSDP	260, 1,300, 10,182	20.9	**3.5e+01**, 3.1e-18, 9.5e-06	−1.5873482e+06
501	SDPNAL+	4,957, 34, 4,080	242.0	3.0e-14, 2.7e-06, 3.0e-11	−1.2840266e+05
501	Ours	344, 10, 545	11.8	5.6e-16, 5.3e-15, 8.3e-07	−1.2837740e+05
gka1e	SDPLR	21, 6,571	5.0	6.1e-07, 0.0e+00, 4.1e-08	−1.7386969e+04
	ManiSDP	490, 2,450, 32,950	11.6	**1.1e+01**, 4.4e-17, **4.3e-05**	−2.3390925e+05
201	SDPNAL+	1,714, 27, 9,269	27.0	7.5e-09, 1.1e-06, 1.1e-10	−1.7386969e+04
201	Ours	177, 8, 368	1.5	3.6e-16, 3.9e-13, 2.5e-07	−1.7386645e+04
gka1f	SDPLR	22, 14,233	72.2	9.3e-07, 0.0e+00, 2.6e-09	−6.6817691e+04
	ManiSDP	240, 1,200, 6,307	15.9	**2.2e+01**, 2.5e-17, **1.3e-05**	−2.8307927e+05
501	SDPNAL+	5,623, 37, 5,315	257.1	6.6e-14, 1.7e-06, 3.7e-10	−6.6819431e+04
501	Ours	347, 8, 458	11.3	5.2e-16, 1.4e-14, 4.7e-07	−6.6810416e+04

Table 4. Computational results for Example 4 with $\texttt{tol}_0 = 10^{-1}$, $\hat{\epsilon} = 10^{-1}$, $\nu = 5 \times 10^3$, and $\kappa_c = 5 \times 10^3$.

Problem	Solver	Iteration	Time	KKT	Objective
soybean-large.2	SDPLR	19, 184	0.2	1.3e-07, 0.0e+00, 5.9e-07	−1.0581914e+04
	ManiSDP	11, 44, 86	0.3	1.3e-08, 2.3e-17, 4.4e-15	−1.0581910e+04
307	SDPNAL+	300, 5, 23	4.6	1.1e-07, 4.7e-09, 7.5e-17	−1.0581910e+04
308	Ours	10, 10, 48	0.8	5.5e-16, 1.8e-13, 3.1e-07	−1.0581898e+04
spambase-small.2	SDPLR				
	ManiSDP	23, 92, 143	0.5	2.9e-07, 9.5e-16, 3.4e-13	−1.1477648e+08
300	SDPNAL+	300, 5, 12	4.7	4.8e-09, 1.7e-07, 3.4e-17	−1.1477648e+08
301	Ours	10, 8, 37	0.6	6.2e-16, 1.4e-11, 1.6e-07	−1.1477619e+08
spambase-medium.2	SDPLR				
	ManiSDP	37, 148, 232	6.8	8.7e-07, 1.1e-15, 1.1e-17	−6.5736462e+08
900	SDPNAL+	200, 7, 43	60.6	2.8e-09, 4.3e-07, 3.1e-17	−6.5736458e+08
901	Ours	10, 7, 33	4.8	1.1e-15, 5.0e-10, 9.8e-07	−6.5734525e+08
spambase-large.2	SDPLR				
	ManiSDP	190, 760, 1,276	101.7	**2.3e-05**, 1.4e-15, 1.4e-09	−1.3754449e+09
1,500	SDPNAL+	200, 11, 117	261.9	1.8e-08, 5.9e-07, 1.5e-17	−1.3754209e+09
1,501	Ours	10, 7, 34	11.9	1.3e-15, 5.1e-10, 8.2e-07	−1.3753748e+09
abalone-medium.2	SDPLR	19, 137	0.2	2.0e-09, 0.0e+00, 2.5e-10	−5.6760187e+04
	ManiSDP	9, 35, 40	0.4	9.5e-08, 1.3e-16, 7.4e-10	−5.6760186e+04
400	SDPNAL+	300, 5, 17	7.5	7.4e-10, 4.1e-07, 7.3e-17	−5.6760186e+04
401	Ours	10, 10, 43	1.4	6.4e-16, 1.4e-13, 4.5e-07	−5.6760101e+04
abalone-large.2	SDPLR	19, 158	1.1	7.4e-08, 0.0e+00, 2.5e-10	−1.3600035e+05
	ManiSDP	10, 40, 48	2.0	7.3e-08, 3.4e-16, 1.7e-10	−1.3600035e+05
1,000	SDPNAL+	200, 7, 33	42.8	1.2e-10, 5.5e-09, 4.0e-17	−1.3600035e+05
1,001	Ours	10, 10, 44	6.3	1.2e-15, 1.4e-13, 3.2e-07	−1.3600011e+05
segment-small.2	SDPLR	23, 578	0.7	1.2e-08, 0.0e+00, 1.1e-09	−1.8871137e+07
	ManiSDP	20, 63, 80	0.7	1.3e-07, 2.3e-16, 2.4e-10	−1.8871136e+07
400	SDPNAL+	300, 5, 17	6.2	9.9e-09, 3.9e-07, 3.0e-17	−1.8871137e+07
401	Ours	10, 10, 47	1.4	6.3e-16, 1.8e-13, 3.0e-07	−1.8871107e+07
segment-medium.2	SDPLR				
	ManiSDP	21, 73, 98	2.2	2.2e-07, 3.2e-16, 1.3e-09	−3.3223348e+07
700	SDPNAL+	200, 6, 37	14.3	1.4e-09, 2.2e-09, 2.0e-17	−3.3223351e+07
701	Ours	10, 10, 48	3.9	9.8e-16, 1.8e-13, 2.5e-07	−3.3223293e+07
segment-large.2	SDPLR				
	ManiSDP	21, 83, 127	4.7	2.2e-07, 5.4e-16, 1.0e-09	−4.7439909e+07
1,000	SDPNAL+	200, 5, 58	41.6	4.1e-08, 1.2e-08, 3.8e-17	−4.7439912e+07
1,001	Ours	10, 10, 48	6.6	1.2e-15, 1.8e-13, 2.2e-07	−4.7439823e+07
housing.2	SDPLR				
	ManiSDP	25, 100, 150	1.6	9.6e-07, 1.1e-16, 1.4e-13	−1.6740775e+08
506	SDPNAL+	200, 5, 20	8.1	1.4e-07, 6.6e-07, 4.6e-17	−1.6740779e+08
507	Ours	10, 12, 54	2.3	7.9e-16, 2.6e-16, 9.3e-08	−1.6740774e+08

Table 5. Computational results for Example 5 with $\texttt{tol}_0 = 10^{-4}$, $\hat{\epsilon} = 5 \times 10^{-3}$, $\nu = 5 \times 10^3$, and $\kappa_c = 10^3$.

Problem	Solver	Iteration	Time	KKT	Objective
nonsym(6,4)	SDPLR	17, 859	0.8	1.4e-07, 0.0e+00, **1.1e-05**	3.0767777e+00
	ManiSDP	120, 600, 8,022	5.2	**4.7e-04**, 0.0e+00, **2.7e-05**	3.0777468e+00
216	SDPNAL+	300, 4, 270	2.5	7.6e-08, 1.5e-08, 2.1e-15	3.0767784e+00
9,260	Ours	932, 8, 513	7.5	2.0e-16, 8.8e-17, 5.1e-08	3.0767803e+00
nonsym(7,4)	SDPLR	17, 527	0.8	6.7e-08, 0.0e+00, 6.3e-06	5.0740764e+00
	ManiSDP	110, 550, 5,894	6.6	**4.5e-04**, 0.0e+00, 8.7e-06	5.0741640e+00
343	SDPNAL+	300, 5, 207	4.8	6.7e-08, 5.3e-08, 2.6e-15	5.0740771e+00
21,951	Ours	1,000, 6, 421	16.3	2.9e-16, 1.2e-13, 9.2e-07	5.0741640e+00
nonsym(8,4)	SDPLR	17, 677	2.0	6.0e-08, 0.0e+00, 1.9e-06	5.7408318e+00
	ManiSDP	50, 250, 1,202	4.2	**6.3e-01**, 0.0e+00, **4.7e-03**	1.5255265e+01
512	SDPNAL+	200, 5, 269	7.8	1.0e-07, 6.5e-08, 2.7e-15	5.7408289e+00
46,655	Ours	1,000, 6, 365	35.1	2.5e-16, 2.2e-12, 7.0e-07	5.7409223e+00
nonsym(9,4)	SDPLR	19, 899	6.0	1.8e-07, 0.0e+00, **2.1e-05**	1.0661330e+00
	ManiSDP	350, 1,750, 9,852	78.5	**3.5e-04**, 0.0e+00, **2.5e-05**	1.0661584e+00
729	SDPNAL+	200, 6, 742	19.9	7.6e-08, 6.7e-08, 2.6e-15	1.0661335e+00
91,124	Ours	1,000, 7, 347	63.7	3.4e-16, 1.0e-13, 1.6e-07	1.0661381e+00
nonsym(10,4)	SDPLR	20, 2,740	41.0	2.0e-07, 0.0e+00, **1.4e-05**	1.6947133e+00
	ManiSDP	120, 600, 1,626	31.3	**3.3e+00**, 0.0e+00, **1.1e-02**	1.4687971e+01
1,000	SDPNAL+	200, 6, 1,047	56.3	1.0e-06, 3.5e-08, 2.6e-15	1.6947159e+00
166,374	Ours	1,000, 14, 1,303	169.6	3.2e-16, 1.5e-16, 6.3e-07	1.6947498e+00
nonsym(11,4)	SDPLR	19, 1,291	42.0	1.3e-07, 0.0e+00, 7.5e-06	2.9134845e+00
	ManiSDP	110, 550, 1,245	47.1	**4.4e+00**, 0.0e+00, **3.4e-02**	7.5444956e+00
1,331	SDPNAL+	200, 7, 1,036	174.4	9.2e-08, 1.8e-08, 3.1e-15	2.9134853e+00
287,495	Ours	1,000, 7, 260	215.6	4.9e-16, 1.5e-14, 3.3e-07	2.9135212e+00
nonsym(4,5)	SDPLR	17, 2,875	2.8	3.6e-07, 0.0e+00, **3.2e-05**	1.5174069e+00
	ManiSDP	370, 1,850, 28,164	19.4	**6.8e-04**, 0.0e+00, **7.7e-05**	1.5164330e+00
256	SDPNAL+	300, 4, 255	4.3	1.8e-07, 3.6e-07, 2.1e-15	1.5174093e+00
9,999	Ours	750, 12, 1,653	10.3	4.1e-16, 1.5e-16, 8.8e-07	1.5174316e+00
nonsym(5,5)	SDPLR	17, 942	3.7	2.3e-07, 0.0e+00, 8.0e-06	3.0825740e+00
	ManiSDP	400, 2,000, 26,577	96.5	**8.9e-03**, 0.0e+00, **5.5e-04**	3.0843595e+00
625	SDPNAL+	200, 6, 690	29.0	5.6e-07, 2.9e-08, 2.5e-15	3.0825745e+00
50,624	Ours	1,000, 8, 473	51.7	4.5e-16, 5.4e-16, 9.6e-08	3.0825817e+00
nonsym(6,5)	SDPLR	19, 1,443	36.7	1.6e-07, 0.0e+00, **1.0e-05**	3.0957169e+00
	ManiSDP	90, 450, 1,776	39.5	**4.4e-01**, 0.0e+00, **6.5e-03**	9.3481818e+00
1,296	SDPNAL+	200, 6, 948	150.2	8.3e-08, 1.9e-08, 2.8e-15	3.0957190e+00
194,480	Ours	1,000, 11, 1,302	328.2	2.6e-16, 9.4e-16, 6.6e-07	3.0957937e+00
sym_rd(3,25)	SDPLR	16, 1,750	2.6	8.1e-08, 0.0e+00, **1.5e-05**	1.6297464e+00
	ManiSDP	100, 500, 2,003	3.0	**1.7e-01**, 0.0e+00, **1.3e-02**	1.3563755e+01
351	SDPNAL+	237, 3, 97	6.8	1.9e-07, 3.9e-07, 2.0e-15	1.6297461e+00
23,750	Ours	428, 9, 3,019	16.1	1.2e-16, 4.1e-17, 8.7e-08	1.6297479e+00
sym_rd(3,30)	SDPLR	17, 2,235	6.1	5.5e-08, 0.0e+00, **7.1e-05**	1.8241660e+00
	ManiSDP	50, 250, 665	2.9	**1.3e+00**, 0.0e+00, **3.1e-02**	5.0945633e+01
496	SDPNAL+	300, 3, 294	19.5	4.3e-07, 6.0e-07, 2.1e-15	1.8241633e+00
46,375	Ours	916, 9, 3,115	54.6	1.3e-16, 1.5e-16, 1.2e-07	1.8241687e+00
sym_rd(3,35)	SDPLR	17, 6,460	34.0	5.0e-08, 0.0e+00, 4.0e-06	1.8299937e+00
	ManiSDP	110, 550, 1,749	11.5	**9.9e-01**, 0.0e+00, **3.3e-02**	4.0567666e+01
666	SDPNAL+	200, 3, 784	36.1	5.2e-07, 8.3e-07, 2.2e-15	1.8299940e+00
82,250	Ours	435, 12, 5,408	138.9	1.4e-16, 4.8e-16, 5.8e-08	1.8299936e+00
sym_rd(3,40)	SDPLR	16, 8,238	86.4	4.1e-08, 0.0e+00, 7.2e-06	1.9931548e+00
	ManiSDP	50, 250, 722	7.8	**2.3e+00**, 0.0e+00, **5.2e-02**	7.2802120e+01
861	SDPNAL+	200, 3, 659	62.1	4.3e-07, 7.7e-07, 2.2e-15	1.9931521e+00
135,750	Ours	385, 9, 4,309	213.2	1.3e-16, 3.7e-16, 8.5e-07	1.9931792e+00
sym_rd(3,45)	SDPLR	17, 8,035	152.7	3.5e-08, 0.0e+00, **4.1e-05**	2.1407708e+00
	ManiSDP	50, 250, 523	10.5	**1.9e+00**, 0.0e+00, **5.5e-02**	1.1688834e+02
1,081	SDPNAL+	200, 3, 1,134	119.1	5.1e-07, 8.1e-07, 2.3e-15	2.1407703e+00
211,875	Ours	923, 10, 5,009	510.1	1.4e-16, 1.5e-16, 9.1e-08	2.1407752e+00
sym_rd(3,50)	SDPLR	18, 9,616	315.0	3.4e-08, 0.0e+00, 6.6e-06	2.0694999e+00
	ManiSDP	70, 350, 798	22.5	**2.3e+00**, 0.0e+00, **6.5e-02**	1.5131076e+02
1,326	SDPNAL+	200, 5, 1,382	143.8	8.9e-08, 2.1e-07, 2.4e-15	2.0694992e+00
316,250	Ours	403, 10, 5,075	745.0	1.3e-16, 1.5e-16, 2.9e-07	2.0695115e+00
sym_rd(4,20)	SDPLR	14, 687	0.6	4.7e-08, 0.0e+00, 2.9e-06	8.6061260e+00
	ManiSDP	110, 550, 958	1.5	**2.3e+00**, 0.0e+00, **2.7e-02**	6.8431952e+02

Table 5. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
210	SDPNAL+	300, 3, 211	2.5	4.4e-07, 1.6e-07, 1.8e-15	8.6061230e+00
8,854	Ours	680, 8, 1,339	6.5	1.1e-16, 3.7e-16, 9.4e-08	8.6061276e+00
sym_rd(4,25)	SDPLR	15, 2,212	3.0	3.6e-08, 0.0e+00, 4.2e-06	8.5618442e+00
	ManiSDP	40, 200, 256	0.9	**1.4e+00**, 0.0e+00, **3.6e-01**	1.0824463e+00
325	SDPNAL+	236, 3, 588	5.9	6.8e-08, 7.4e-09, 2.1e-15	8.5618423e+00
20,474	Ours	342, 11, 4,625	17.4	1.3e-16, 3.0e-16, 3.6e-07	8.5618630e+00
sym_rd(4,30)	SDPLR	16, 3,488	8.1	2.3e-08, 0.0e+00, 8.2e-07	9.5602225e+00
	ManiSDP	70, 350, 678	3.5	**1.8e+00**, 0.0e+00, **6.3e-02**	9.9161450e+02
465	SDPNAL+	183, 3, 691	8.4	4.0e-07, 6.7e-08, 2.0e-15	9.5602133e+00
40,919	Ours	1,000, 9, 4,431	59.8	1.2e-16, 7.6e-17, 3.4e-07	9.5602492e+00
sym_rd(4,35)	SDPLR	17, 43,809	199.8	1.8e-08, 0.0e+00, 3.1e-06	1.0983323e+01
	ManiSDP	30, 150, 241	2.4	**5.1e+00**, 0.0e+00, **5.1e-01**	9.0881233e+00
630	SDPNAL+	181, 3, 927	16.5	6.6e-07, 3.5e-08, 2.2e-15	1.0983326e+01
73,814	Ours	1,000, 10, 5,400	137.3	1.1e-16, 1.5e-16, 7.7e-08	1.0983332e+01
sym_rd(4,40)	SDPLR	16, 57,711	517.1	1.4e-08, 0.0e+00, 2.5e-06	1.1560687e+01
	ManiSDP	70, 350, 672	8.9	**1.7e+00**, 0.0e+00, **9.4e-02**	9.6468247e+02
820	SDPNAL+	280, 12, 879	40.9	5.3e-08, 2.9e-08, 5.0e-15	1.1547151e+01
123,409	Ours	1,000, 9, 8,888	280.9	1.2e-16, 3.3e-16, 6.0e-07	1.1547175e+01
sym_rd(4,45)	SDPLR	13, 120,969	2,046.5	1.3e-08, 0.0e+00, 6.6e-06	1.2051365e+01
	ManiSDP	110, 550, 1,096	22.3	**3.9e+00**, 0.0e+00, **1.8e-01**	1.2627037e+03
1,035	SDPNAL+	278, 11, 461	57.2	8.8e-07, 6.0e-09, 1.2e-06	1.1842468e+01
194,579	Ours	1,000, 9, 7,780	438.9	1.2e-16, 3.0e-16, 5.0e-07	1.1842486e+01
sym_rd(4,50)	SDPLR	16, 106,834	3,151.3	1.0e-08, 0.0e+00, **1.2e-05**	1.3073209e+01
	ManiSDP	50, 250, 366	14.3	**5.7e+00**, 0.0e+00, **2.1e-01**	7.4617702e+00
1,275	SDPNAL+	178, 11, 420	60.1	3.9e-08, 9.8e-07, 6.3e-15	1.3041815e+01
292,824	Ours	1,000, 9, 5,061	523.2	1.2e-16, 3.3e-16, 4.9e-07	1.3041825e+01
sym_rd(5,10)	SDPLR	13, 1,983	1.8	1.4e-07, 0.0e+00, 5.0e-06	2.9812615e+00
	ManiSDP	40, 200, 264	0.7	**3.2e+00**, 0.0e+00, **1.9e-01**	5.4238299e+00
286	SDPNAL+	174, 1, 49	2.3	2.5e-09, 3.9e-06, **1.9e-03**	2.8290826e+00
8,007	Ours	435, 8, 1,314	7.9	1.4e-16, 1.5e-16, 6.0e-07	2.9812810e+00
sym_rd(5,15)	SDPLR	15, 14,037	85.2	5.4e-08, 0.0e+00, 2.9e-06	3.4934605e+00
	ManiSDP	320, 1,600, 7,314	54.4	**6.9e-01**, 0.0e+00, **9.5e-03**	1.0012877e+03
816	SDPNAL+	196, 3, 646	22.7	6.4e-07, 4.3e-08, 2.0e-15	3.4934739e+00
54,263	Ours	421, 14, 8,514	345.8	1.4e-16, 3.8e-17, 9.7e-08	3.4934646e+00
sym_rd(5,20)	SDPLR	14, 76,380	3,600.4	1.4e-07, 0.0e+00, 1.0e-05	5.4052691e+00
	ManiSDP	180, 900, 1,162	93.6	**4.0e+00**, 0.0e+00, **6.6e-01**	3.3692148e+00
1,771	SDPNAL+	200, 3, 1,801	208.3	1.6e-07, 7.3e-08, 2.2e-15	4.1792187e+00
230,229	Ours	1,000, 9, 4,415	1,554.5	1.4e-16, 6.3e-16, 3.0e-07	4.1792631e+00
sym_rd(6,10)	SDPLR	11, 1,026	0.8	1.2e-07, 0.0e+00, 2.0e-06	2.2737406e+01
	ManiSDP	210, 1,050, 1,180	2.7	**1.9e+00**, 0.0e+00, **2.5e-01**	5.9223302e+00
220	SDPNAL+	265, 3, 296	2.6	6.5e-08, 6.3e-07, 2.0e-15	2.2737299e+01
5,004	Ours	379, 8, 1,372	4.7	1.3e-16, 7.5e-17, 5.4e-07	2.2737450e+01
sym_rd(6,15)	SDPLR	13, 42,609	167.9	3.7e-08, 0.0e+00, 7.4e-07	2.7098938e+01
	ManiSDP	270, 1,350, 1,864	22.8	**3.5e+00**, 0.0e+00, **1.9e-01**	1.6048026e+01
680	SDPNAL+	200, 3, 1,604	24.2	1.2e-07, 6.8e-08, 2.0e-15	2.7098696e+01
38,759	Ours	613, 20, 7,038	179.3	1.3e-16, 7.4e-17, 1.4e-07	2.7098700e+01
sym_rd(6,20)	SDPLR	13, 77,794	2,521.5	1.6e-08, 0.0e+00, 3.9e-07	3.5598088e+01
	ManiSDP	30, 150, 182	11.1	**1.3e+00**, 0.0e+00, **5.1e-01**	1.1327177e+00
1,540	SDPNAL+	185, 3, 1,520	123.8	3.6e-08, 5.2e-07, 2.4e-15	3.1508324e+01
177,099	Ours	1,000, 10, 6,000	920.4	1.4e-16, 1.1e-16, 6.6e-07	3.1508498e+01
nsym_rd([15,15,15])	SDPLR	17, 1,068	1.0	6.6e-08, 0.0e+00, **1.8e-05**	2.4837923e+00
	ManiSDP	130, 650, 6,032	5.4	**2.5e-05**, 0.0e+00, 1.1e-06	2.4838368e+00
225	SDPNAL+	300, 3, 136	2.7	2.8e-07, 2.2e-07, 2.1e-15	2.4837930e+00
14,399	Ours	1,000, 8, 1,519	10.1	1.3e-16, 3.0e-16, 7.8e-08	2.4837946e+00
nsym_rd([20,20,20])	SDPLR	18, 730	1.6	3.8e-08, 0.0e+00, 8.0e-06	3.4777159e+00
	ManiSDP	190, 950, 8,473	20.1	**6.0e-05**, 0.0e+00, 3.6e-06	3.4778671e+00
400	SDPNAL+	300, 4, 855	8.6	2.2e-07, 2.3e-07, 2.4e-15	3.4777156e+00
44,099	Ours	1,000, 8, 1,279	28.3	1.4e-16, 1.8e-15, 3.1e-07	3.4777169e+00
nsym_rd([20,25,25])	SDPLR	19, 1,500	4.8	3.2e-08, 0.0e+00, **1.4e-05**	2.7856927e+00
	ManiSDP	230, 1,150, 8,157	30.3	**2.6e-03**, 0.0e+00, **5.2e-04**	2.7833029e+00
500	SDPNAL+	200, 4, 792	10.7	2.5e-07, 5.6e-07, 2.4e-15	2.7856953e+00
68,249	Ours	353, 12, 4,202	56.5	1.2e-16, 5.0e-17, 4.7e-07	2.7857188e+00

Table 5. (Continued)

Problem	Solver	Iteration	Time	KKT	Objective
nsym_rd([25,20,25])	SDPLR	18, 1,308	4.1	3.2e-08, 0.0e+00, 4.0e-06	2.7755704e+00
	ManiSDP	270, 1,350, 11,903	42.7	**2.9e-05**, 0.0e+00, 1.8e-06	2.7756442e+00
500	SDPNAL+	200, 4, 805	10.1	1.1e-07, 4.4e-08, 2.2e-15	2.7755718e+00
68,249	Ours	1,000, 8, 1,992	48.2	1.2e-16, 5.3e-16, 7.5e-07	2.7756101e+00
nsym_rd([25,25,20])	SDPLR	18, 2,112	6.6	3.3e-08, 0.0e+00, 1.0e-05	2.8765714e+00
	ManiSDP	230, 1,150, 9,090	33.1	**5.1e-04**, 0.0e+00, **2.2e-04**	2.8828557e+00
500	SDPNAL+	200, 4, 477	8.9	9.0e-08, 1.7e-08, 2.5e-15	2.8765721e+00
68,249	Ours	1,000, 9, 2,652	55.2	1.2e-16, 4.7e-17, 2.3e-07	2.8765741e+00
nsym_rd([25,25,25])	SDPLR	18, 3,101	16.6	2.6e-08, 0.0e+00, **1.5e-05**	2.8300019e+00
	ManiSDP	40, 200, 419	3.6	**2.0e+00**, 0.0e+00, **1.6e-01**	3.1847883e+00
625	SDPNAL+	200, 5, 1,855	24.4	6.1e-08, 5.3e-09, 2.4e-15	2.8300023e+00
105,624	Ours	337, 16, 10,189	205.0	1.3e-16, 2.2e-16, 7.3e-07	2.8300149e+00
nsym_rd([30,30,30])	SDPLR	19, 2,316	32.6	2.1e-08, 0.0e+00, 6.8e-06	3.0377560e+00
	ManiSDP	190, 950, 1,933	39.6	**4.0e-01**, 0.0e+00, **7.5e-03**	1.4129021e+01
900	SDPNAL+	200, 5, 1,010	35.5	1.4e-07, 6.5e-07, 2.6e-15	3.0377599e+00
216,224	Ours	1,000, 11, 3,955	285.6	1.5e-16, 1.5e-16, 1.4e-07	3.0377605e+00
nsym_rd([35,35,35])	SDPLR	20, 2,662	85.3	1.5e-08, 0.0e+00, 5.7e-06	3.0704750e+00
	ManiSDP	60, 300, 621	20.6	**2.3e+00**, 0.0e+00, **2.0e-01**	3.7383637e+00
1,225	SDPNAL+	200, 4, 1,186	79.0	6.6e-07, 3.7e-07, 2.7e-15	3.0704751e+00
396,899	Ours	540, 11, 4,667	569.0	1.2e-16, 3.7e-16, 4.0e-07	3.0704779e+00
nsym_rd([40,40,40])	SDPLR	21, 2,859	190.2	1.1e-08, 0.0e+00, 5.7e-06	3.8787371e+00
	ManiSDP	40, 200, 354	20.5	**5.6e+00**, 0.0e+00, **2.3e-01**	4.1093664e+00
1,600	SDPNAL+	200, 3, 461	105.0	7.2e-07, 5.2e-07, 2.9e-15	3.8787371e+00
672,399	Ours	1,000, 9, 3,272	861.7	3.1e-16, 7.4e-16, 5.7e-07	3.8788559e+00
nsym_rd([6,6,6,6])	SDPLR	16, 1,082	1.0	6.7e-08, 0.0e+00, **1.3e-05**	2.6823271e+00
	ManiSDP	830, 4,150, 64,117	42.3	**6.2e-05**, 0.0e+00, 3.8e-06	2.6822326e+00
216	SDPNAL+	300, 3, 158	2.6	3.5e-07, 4.7e-07, 2.1e-15	2.6823279e+00
9,260	Ours	520, 8, 1,298	5.7	1.1e-16, 8.9e-17, 4.8e-08	2.6823293e+00
nsym_rd([7,7,7,7])	SDPLR	17, 744	1.1	5.7e-08, 0.0e+00, 8.7e-06	3.3323703e+00
	ManiSDP	140, 700, 7,993	9.5	**9.1e-05**, 0.0e+00, 4.0e-06	3.3323037e+00
343	SDPNAL+	300, 4, 228	4.7	4.7e-07, 1.2e-07, 2.1e-15	3.3323703e+00
21,951	Ours	1,000, 7, 550	17.1	1.5e-16, 2.9e-14, 7.0e-08	3.3323720e+00
nsym_rd([8,8,8,8])	SDPLR	18, 937	2.7	3.5e-08, 0.0e+00, 4.8e-06	2.8376898e+00
	ManiSDP	170, 850, 8,328	25.0	**1.3e-04**, 0.0e+00, 8.5e-06	2.8379737e+00
512	SDPNAL+	200, 4, 377	8.6	5.9e-08, 3.3e-08, 2.4e-15	2.8376877e+00
46,655	Ours	1,000, 8, 1,055	42.5	1.1e-16, 3.6e-16, 7.3e-08	2.8376911e+00
nsym_rd([9,9,9,9])	SDPLR	18, 693	4.7	2.4e-08, 0.0e+00, **1.6e-05**	3.1089491e+00
	ManiSDP	80, 400, 1,089	10.7	**2.8e-01**, 0.0e+00, **2.3e-02**	5.8785058e+00
729	SDPNAL+	200, 4, 525	16.3	3.0e-07, 2.9e-07, 2.6e-15	3.1089487e+00
91,124	Ours	1,000, 8, 865	79.6	1.3e-16, 1.1e-15, 7.1e-08	3.1089533e+00

are high. In fact, we always observe that the rank is nearly $n-1$, where n denotes the matrix dimension. In such cases, the primal constraint nondegenerate condition usually holds, making SDPNAL+ have a very fast convergence rate. On the other hand, for these SDPs, because the dual constraint nondegeneracy condition typically fails to hold, Algorithm 1 would require more computational effort than SDPNAL+ to solve the problems. Surprisingly, although SDPLR is designed for solving SDPs with low-rank solutions, it sometimes shows promising performance when solving SDPs with high-rank solutions, as indicated by Table 5. In comparison, ManiSDP fails to solve most of the tested problems because of the fact that the solutions are of high rank and there is no explicit manifold structure to utilize.

Based on the relative performance of SDPNAL+ and Algorithm 1 in Tables 1–5, we may conclude that when the primal constraint nondegeneracy condition is likely to hold at the optimal solution, SDPNAL+ would be preferable. However, when such prior information is not available, Algorithm 1 could be a good choice based on the promising numerical performance demonstrated in Tables 1–4, even though the primal constraint nondegeneracy condition may not hold based on our empirical observation. Note finally that, for our algorithm, the errors η_p and η_d are usually very small, whereas η_c is driven to a value slightly smaller than $\texttt{tol} = 10^{-6}$. On the contrary, SDPNAL+ keeps η_c very small, whereas it progressively decreases η_p and η_d.

6. Concluding Remarks

We have analyzed and implemented a squared smoothing Newton method via the Huber smoothing function for solving semidefinite programming problems (SDPs). With a careful design of the algorithmic framework, our theoretical analysis has shown that the proposed algorithm is well-defined, guarantees global convergence, and admits a super-linear convergence rate under the primal and dual constraint nondegeneracy conditions. Besides establishing these convergence properties, we have also conducted extensive numerical experiments on solving various classes of SDPs to evaluate the practical performance of our algorithm. We have compared our method with the state-of-the-art SDP solvers, including `SDPLR`, `ManiSDP`, and `SDPNAL+`, and the numerical results have demonstrated the excellent efficiency of our algorithm. We note that the current implementation of the algorithm is not as mature as we would hope for because the performance may depend sensitively on some parameters. However, given the promising numerical results on the tested examples, we are inspired to conduct a more robust implementation in future work.

Acknowledgments

The authors thank the editors and referees for providing numerous valuable suggestions that have helped to improve the quality of this paper.

Appendix A. Proof of Proposition 1

Part 1 is a direct consequence of proposition 4.3 in Chen et al. [15]. We next prove part 2. Because $h(\cdot,\cdot)$ is locally Lipschitz continuous on $\mathbb{R}\times\mathbb{R}$, then by theorem 9.67 in Rockafellar and Wets [54], there exist continuously differentiable functions $h_\ell:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$, $\ell\geq 1$, converging uniformly to h and satisfying

$$\|h'_\ell(\tau,\xi)\|\leq L,\quad \forall(\tau,\xi)\in\mathcal{J}:=[\epsilon-\nu,\epsilon+\nu]\times\left(\bigcup_{i=1}^{n}[d_i-\nu_i,d_i+\nu_i]\right),$$

with some constants $L>0$, $\nu>0$, and $\nu_i>0$. For any $(\tau,H)\in\mathbb{R}\times\mathbb{S}^n$ with $H=\tilde{P}\mathrm{diag}(\tilde{d}_1,\ldots,\tilde{d}_n)\tilde{P}^T$, define

$$\Phi_\ell(\tau,H)=\tilde{P}\mathrm{diag}(h_\ell(\tau,\tilde{d}_1),\ldots,h_\ell(\tau,\tilde{d}_n))\tilde{P}^T,\quad \ell\geq 1.$$

From Horn and Johnson [22], we may assume that there exists a neighborhood of (ϵ,W), denoted by $\mathcal{U}$, such that $(\tau,\tilde{d}_i)\in\mathcal{J}$ for all $i=1,\ldots,n$ and $(\tau,H)\in\mathcal{U}$. Note that Φ_ℓ converges to Φ uniformly on $\mathcal{U}$. Fix $(\tau_1,H_1),(\tau_2,H_2)\in\mathcal{U}$ such that $(\tau_1,H_1)\neq(\tau_2,H_2)$. Then, for any $\hat{L}>0$ and ℓ sufficiently large, it holds that

$$\|\Phi_\ell(\tau,H)-\Phi(\tau,H)\|\leq\hat{L}\|(\tau_1,H_1)-(\tau_2,H_2)\|,\quad\forall(\tau,H)\in\mathcal{U}. \tag{A.1}$$

Using (A.1), for any $(\tau,H)\in\mathcal{U}$, it follows that

$$\begin{aligned}
&\|\Phi(\tau_1,H_1)-\Phi(\tau_2,H_2)\|\\
&\quad\leq\|\Phi(\tau_1,H_1)-\Phi_\ell(\tau_1,H_1)\|+\|\Phi_\ell(\tau_1,H_1)-\Phi_\ell(\tau_2,H_2)\|+\|\Phi_\ell(\tau_2,H_2)-\Phi(\tau_2,H_2)\|\\
&\quad\leq 2\hat{L}\|(\tau_1,H_1)-(\tau_2,H_2)\|+\left\|\int_0^1\Phi_\ell'(\tau_1+t(\tau_2-\tau_1),H_1+t(H_2-H_1))(\tau_1-\tau_2,H_2-H_1)dt\right\|\\
&\quad\leq(2\hat{L}+L)\|(\tau_1,H_1)-(\tau_2,H_2)\|
\end{aligned}$$

for $\ell\geq 1$ sufficiently large. Because $\hat{L}>0$ is arbitrary, we see that Φ is locally Lipschitz continuous with modulus $L>0$.

Then, we turn to prove part 3. Let us recall that $\Phi(0,W)=\sum_{j=1}^r h(0,\lambda_j)Q_j$ and write

$$\Phi(t\tau,W+tH)=\sum_{j=1}^{r}h(0,\lambda_j)Q_j(t)+\sum_{i=1}^{n}(h(t\tau,d_i(t))-h(0,d_i))p_i(t)p_i(t)^T.$$

Then, it follows that

$$\begin{aligned}
&\lim_{t\downarrow 0}\frac{1}{t}(\Phi(t\tau,W+tH)-\Phi(0,W))\\
&\quad=\sum_{j=1}^{r}h(0,\lambda_j)\lim_{t\downarrow 0}\frac{1}{t}(Q_j(t)-Q_j)+\sum_{i=1}^{n}\lim_{t\downarrow 0}\frac{1}{t}(h(t\tau,d_i(t))-h(0,d_i))p_i(t)p_i(t)^T.
\end{aligned}$$

By equation (2.9) in Shapiro [55], for any $j=1,\ldots,r$, the following holds:

$$\lim_{t\downarrow 0}\frac{1}{t}(Q_j(t)-Q_j)=Q'(0)(H)=\frac{1}{2}\sum_{1\leq k\neq j\leq r}\frac{h(0,\lambda_k)-h(0,\lambda_j)}{\lambda_k-\lambda_j}(Q_jHQ_k+Q_kHQ_j).$$

From proposition 3.1 in Shapiro [55], for any $i = 1, \ldots, n$, we know that $d_i'(W;H)$ is well-defined, and it holds that

$$\lim_{t \downarrow 0} \frac{1}{t}(h(t\tau, d_i(t)) - h(0, d_i)) = h'((0,d_i);(\tau, d_i'(W;H)))$$
$$= \sum_{i \in \alpha} \left(d_i'(W;H) - \frac{|\tau|}{2} \right) p_i p_i^T + \sum_{i \in \beta} h(\tau, d_i'(W;H)) p_i p_i^T,$$

where we have used the fact that h is directionally differentiable with

$$h'((0,u);(\tau,v)) = \begin{cases} v - \dfrac{|\tau|}{2} & u > 0 \\ h(\tau, v) & u = 0 \\ 0 & u < 0 \end{cases}, \quad \forall (\tau, v) \in \mathbb{R} \times \mathbb{R}.$$

Because $p_i(t) \to p_i$ as $t \downarrow 0$, $\lim_{t \downarrow 0} \frac{1}{t}(\Phi(t\tau, W + tH) - \Phi(0,W))$ exists. Hence, Φ is directionally differentiable at $(0,W)$, and (15) holds true. Finally, the explicit expression of the directional differential $d_i'(W;H)$ $(1 \le i \le n)$ is again obtained from proposition 3.1 in Shapiro [55].

Finally, we prove part 4. Based on part 1, Φ is continuously differentiable at any (ϵ, W) with $\epsilon \neq 0$. Hence, Φ is naturally strongly semismooth at these points. Thus, we need to show only that Φ is strongly semismooth at $(0,W)$ for any $W \in \mathbb{S}^n$. We have already known that Φ is locally Lipschitz continuous and directionally differentiable everywhere. By theorem 3.7 in Sun and Sun [60], we need to show only that for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$ with $\|(\tau, H)\| \to 0$,

$$\Phi(\tau, W + H) - \Phi(0,W) - \Phi'((\tau, W+H);(\tau,H)) = O(\|(\tau,H)\|^2). \tag{A.2}$$

First, similarly to the proof of part 3, one can show for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$,

$$\Phi'((\tau, W+H);(\tau,H)) = \sum_{j=1}^{r} h(\tau, \lambda_j(1)) Q_j'(1)(H)$$
$$+ \sum_{i=1}^{n} h'((\tau, d_i(1));(\tau, d_i'(W+H;H))) p_i(1) p_i(1)^T. \tag{A.3}$$

Then, by the fact that d_i is strongly semismooth everywhere (see for example proposition 3.2 in Shapiro [55]), we deduce that

$$\begin{aligned} &\Phi(\tau, W+H) - \Phi(0,W) \\ &= \sum_{j=1}^{r} h(\tau, \lambda_j(1))(Q_j(1) - Q_j) + \sum_{i=1}^{n} (h(\tau, d_i(1)) - h(0,d_i)) p_i p_i^T \\ &= \sum_{j=1}^{r} h(\tau, \lambda_j(1)) Q_j'(1)(H) + O(\|H\|^2) \\ &\quad + \sum_{i=1}^{n} (h'((\tau, d_i(1));(\tau, d_i'(W+H;H)))) p_i p_i^T + O(\|(\tau,H)\|^2) \\ &= \sum_{j=1}^{r} h(\tau, \lambda_j(1)) Q_j'(1)(H) + \sum_{i=1}^{n} (h'((\tau, d_i(1));(\tau, d_i'(W+H;H)))) p_i(1) p_i(1)^T + O(\|(\tau,H)\|^2), \end{aligned}$$

which together with (A.3) and the fact that p_i is analytic around W implies (A.2). Thus, the proof is completed.

Appendix B. Proof of Lemma 3

First, let $V \in \partial_B \Phi(0,W)$. By the definition of $\partial_B \Phi(0,W)$, there exists a sequence $\{(\epsilon^k, W^k)\}$ converging to $(0,W)$ with $\epsilon^k \neq 0$ such that $V = \lim_{k \to \infty} \Phi'(\epsilon^k, W^k)$. Let each W^k have the following spectral decomposition:

$$W^k = P^k D^k (P^k)^T, \quad D^k := \mathrm{diag}(D_\alpha^k, D_\beta^k, D_\gamma^k),$$

where $D_\alpha^k = \mathrm{diag}(d_1^k, \ldots, d_{|\alpha|}^k)$, $D_\beta^k = \mathrm{diag}(d_{|\alpha|+1}^k, \ldots, d_{|\alpha|+|\beta|}^k)$, and $D_\gamma^k = \mathrm{diag}(d_{|\alpha|+|\beta|+1}^k, \ldots, d_n^k)$, with $d_1^k \ge \cdots \ge d_n^k$. For simplicity, denote $d^k = (d_1^k, \ldots, d_n^k)^T \in \mathbb{R}^n$. By taking a subsequence if necessary, we may assume without loss of generality that (a) $\lim_{k \to \infty} D^k = D$, $\lim_{k \to \infty} P^k = P$, and (b) both sequences $\{\Omega(\epsilon^k, d^k)\}$ and $\{\mathcal{D}(\epsilon^k, d^k)\}$ converge, where $\Omega(\epsilon^k, d^k)$ and $\mathcal{D}(\epsilon^k, d^k)$ are defined as in (12). In particular, we see that $\{\Omega(\epsilon^k, d^k)_{\alpha\alpha}\}$, $\{\Omega(\epsilon^k, d^k)_{\alpha\beta}\}$ converge to two matrices of all ones of suitable sizes, respectively, $\{\Omega(\epsilon^k, d^k)_{\alpha\gamma}\}$ converges to $\Omega_0(d)_{\alpha\gamma}$, and the limit of the sequence $\{\Omega(\epsilon^k, d^k)_{\beta\beta}\}$ exists. Moreover, let

$$\mathcal{D}(\epsilon^k, d^k) = \mathrm{diag}(\mathcal{D}(\epsilon^k, d^k)_\alpha, \mathcal{D}(\epsilon^k, d^k)_\beta, \mathcal{D}(\epsilon^k, d^k)_\gamma).$$

It holds that $\{\mathcal{D}(\epsilon^k, d^k)_\gamma\}$ converges to the zero matrix, and the limits of $\{\mathcal{D}(\epsilon^k, d^k)_\alpha\}$ and $\{\mathcal{D}(\epsilon^k, d^k)_\beta\}$ exist.

From Proposition 1 and $\epsilon_k \neq 0$, for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$, we see that

$$\Phi'(\epsilon^k, W^k)(\tau, H) = P^k[\Omega(\epsilon^k, d^k) \circ \tilde{H}^k + \tau \mathcal{D}(\epsilon^k, d^k)](P^k)^T, \tag{B.1}$$

where $\tilde{H}_k := (P^k)^T H P^k$. Taking the limit of both sides in (B.1) yields that

$$V(\tau, H) = P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & \Omega_0(d) \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \lim\limits_{k\to\infty} \Omega(\epsilon^k, d^k)_{\beta\beta} \circ \tilde{H}_{\beta\beta} & 0 \\ \tilde{H}_{\alpha\gamma}^T \circ \Omega_0(d)^T & 0 & 0 \end{pmatrix} P^T$$
$$+ \tau P \begin{pmatrix} \lim\limits_{k\to\infty} \mathcal{D}(\epsilon^k, d^k)_\alpha & 0 & 0 \\ 0 & \lim\limits_{k\to\infty} \mathcal{D}(\epsilon^k, d^k)_\beta & 0 \\ 0 & 0 & 0 \end{pmatrix} P^T.$$

For each $k \geq 1$, define $Y^k := P\mathrm{diag}(0, D_\beta^k, 0)P^T$ and $\tilde{Y}^k := P^T Y^k P$. Note that the mapping $\mathcal{L}$ is F-differentiable at (ϵ^k, Y^k) because $\epsilon_k \neq 0$. For k sufficiently large, we have from (16) and Proposition 1 that

$$\mathcal{L}'(\epsilon^k, Y^k)(\tau, H) = \lim_{t \downarrow 0} \frac{\mathcal{L}(\epsilon^k + t\tau, Y^k + tH) - \mathcal{L}(\epsilon^k, Y^k)}{t}$$
$$= P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \Phi'_{|\beta|}(\epsilon^k, D_\beta^k)(\tau, \tilde{H}_{\beta\beta}) & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T \circ H_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T - \frac{\tau}{2}\mathrm{sgn}(\epsilon^k) \sum_{i \in \alpha} p_i p_i^T$$
$$= P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \Omega(\epsilon^k, d^k)_{\beta\beta} \circ \tilde{H}_{\beta\beta} & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T \circ H_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T$$
$$+ \tau P \begin{pmatrix} \mathcal{D}(\epsilon_k, d^k)_\alpha & 0 & 0 \\ 0 & \mathcal{D}(\epsilon^k, d^k)_\beta & 0 \\ 0 & 0 & 0 \end{pmatrix} P^T,$$

which implies that $V(\tau, H) = \lim_{k\to\infty} \mathcal{L}'(\epsilon^k, Y^k)(\tau, H)$. Hence, $V \in \partial_B \mathcal{L}(0, 0)$.

Conversely, choose $V \in \partial_B \mathcal{L}(0, 0)$. By definition, there exists a sequence $\{(\epsilon^k, Y^k)\}$ converging to (0, 0) with $\epsilon^k \neq 0$ such that $V = \lim_{k\to\infty} \mathcal{L}'(\epsilon^k, Y^k)$. Let $\tilde{Y}^k := P^T Y^k P$. Assume that $\tilde{Y}_{\beta\beta}^k$ has the spectral decomposition $\tilde{Y}_{\beta\beta}^k = U^k \tilde{D}_\beta^k (U^k)^T$, where $\tilde{D}_\beta^k = \mathrm{diag}(\tilde{z}^k)$, $\tilde{z}^k := (\tilde{z}_1^k, \ldots, \tilde{z}_{|\beta|}^k)^T \in \mathbb{R}^{|\beta|}$, $\tilde{z}_1^k \geq \cdots \geq \tilde{z}_{|\beta|}^k$, and $U^k \in \mathbb{R}^{|\beta|\times|\beta|}$ is orthogonal. Then, for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$ with $\tilde{H} := P^T H P$, we get from (16) and Proposition 1 that

$$\mathcal{L}'(\epsilon^k, Y^k)(\tau, H) = P \begin{pmatrix} \tilde{H}_{\alpha\alpha} & \tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \\ \tilde{H}_{\alpha\beta}^T & \hat{H}_{\beta\beta} & 0 \\ [\Omega_0(d)]_{\alpha\gamma}^T \circ H_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T - \frac{\tau}{2}\mathrm{sgn}(\epsilon^k) \sum_{i \in \alpha} p_i p_i^T,$$

where $\hat{H}_{\beta\beta} := U^k[\Omega(\epsilon^k, \tilde{z}^k) \circ ((U^k)^T \tilde{H}_{\beta\beta} U^k) + \tau \mathcal{D}(\epsilon^k, \tilde{z}^k)](U^k)^T$. For any $k \geq 1$, define

$$W^k = W + P \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{Y}_{\beta\beta}^k & 0 \\ 0 & 0 & 0 \end{pmatrix} P^T, \quad \tilde{W}^k = P^T W^k P = \begin{pmatrix} D_\alpha & 0 & 0 \\ 0 & \tilde{Y}_{\beta\beta}^k & 0 \\ 0 & 0 & D_\gamma \end{pmatrix}$$

where

$$D_\alpha = \mathrm{diag}(d_1, \ldots, d_{|\alpha|}), \quad D_\gamma = \mathrm{diag}(d_{|\alpha|+|\beta|+1}, \ldots, d_n).$$

Moreover, we partition P as $P = (P_\alpha, P_\beta, P_\gamma)$, set $P^k = (P_\alpha, P_\beta U^k, P_\gamma)$, and construct a vector $d^k \in \mathbb{R}^n$ as follows

$$d_i^k = \begin{cases} d_i & i \in \alpha \cup \gamma \\ \tilde{z}^k_{i-|\alpha|} & i \in \beta. \end{cases}$$

Clearly, it holds that $W^k = P^k \mathrm{diag}(d^k)(P^k)^T$, because $\epsilon^k \neq 0$, Φ is F-differentiable at (ϵ^k, W^k), and

$$\Phi'(\epsilon^k, W^k)(\tau, H) = P^k[\Omega(\epsilon^k, d^k) \circ ((P^k)^T H P^k) + \tau \mathcal{D}(\epsilon^k, d^k)](P^k)^T.$$

Let us now assume without loss of generality that the three sequences $\{U^k\}$, $\{\Omega(\epsilon^k, d^k)\}$ and $\{\mathcal{D}(\epsilon, d^k)\}$ converge (because they are all uniformly bounded). Then, simple calculations show that

$$\lim_{k\to\infty} [\Omega(\epsilon^k, d^k)]_{ij} = \begin{cases} 1 & i \in \alpha, j \in \alpha \cup \beta \\ \Omega_0(d) & i \in \alpha, j \in \gamma \\ 0 & i \in \beta \cup \gamma, j \in \gamma \\ [\lim_{k\to\infty} \Omega(\epsilon^k, \tilde{z}^k)]_{(i-|\alpha|)(j-|\alpha|)} & i \in \beta, j \in \beta, \end{cases}$$

and that

$$\lim_{k\to\infty} \mathcal{D}(\epsilon^k, d^k) = \begin{pmatrix} \lim_{k\to\infty} \mathcal{D}(\epsilon^k, d_\alpha) & 0 & 0 \\ 0 & \lim_{k\to\infty} \mathcal{D}(\epsilon^k, \tilde{z}^k) & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where $d_\alpha := (d_1, \ldots, d_{|\alpha|})^T \in \mathbb{R}^{|\alpha|}$. As a consequence, we get

$$\lim_{k\to\infty} (P^k)^T (\mathcal{L}'(\epsilon^k, Y^k)(\tau, H) - \Phi'(\epsilon^k, W^k)(\tau, H)) P^k = 0, \quad \forall (\tau, H) \in \mathbb{R} \times \mathbb{S}^n,$$

which further implies that $V(\tau, H) = \lim_{k\to\infty} \Phi'(\epsilon^k, W^k)(\tau, H)$ for all $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$. Then, $V \in \partial_B \Phi(0, W)$. Therefore, the proof is completed.

Appendix C. Proof of Lemma 4

We prove the case for the B-subdifferential only because the case for Clarke's generalized Jacobian can be proved similarly. Let the smooth mapping $\Psi : \mathbb{R} \times \mathbb{S}^n \to \mathbb{R} \times \mathbb{S}^n$ be defined as $\Psi(\tau, H) := (\tau, P^T H P)$ for any $(\tau, H) \in \mathbb{R} \times \mathbb{S}^n$. It is clear that $\Psi'(\tau, H) : \mathbb{R} \times \mathbb{S}^n \to \mathbb{R} \times \mathbb{S}^n$ is onto. Define the mapping $\Upsilon : \mathbb{R} \times \mathbb{S}^n \to \mathbb{S}^n$ as follows:

$$\Upsilon(\nu, Y) := P \begin{pmatrix} Y_{\alpha\alpha} & Y_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma} \circ Y_{\alpha\gamma} \\ Y_{\alpha\beta}^T & \Phi_{|\beta|}(\nu, Y_{\beta\beta}) & 0 \\ Y_{\alpha\gamma}^T \circ [\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix} P^T - \frac{|\nu|}{2} \sum_{i\in\alpha} p_i p_i^T.$$

Then, by (16), it holds that $\mathcal{L}(\tau, H) = \Upsilon(\Psi(\tau, H))$. Now, by lemma 1 in Chan and Sun [11] and Lemma 3, we have

$$\partial_B \Phi(0, W) = \partial_B \mathcal{L}(0, 0) = \partial_B \Upsilon(0, 0) \Psi'(0, 0),$$

which proves the first part of the lemma. The expression of $V_{|\beta|} \in \partial_B \Phi(0, 0)$ follows directly from its definition and Proposition 1. Thus, the proof is completed.

Appendix D. Proof of Lemma 5

We first consider $V \in \partial_B \Phi(0, W)$. Using Lemma 4, there exist an orthogonal matrix $U \in \mathbb{R}^{|\beta|\times|\beta|}$ and a symmetric matrix $\Omega_{|\beta|} \in \mathbb{S}^{|\beta|}$ such that

$$\begin{aligned} &\langle H - V(0, H), V(0, H) \rangle \\ &\quad = \langle P^T(H - V(0, H))P, P^T V(0, H) P \rangle \\ &\quad = 2\langle (E_{\alpha\gamma} - [\Omega_0(d)]_{\alpha\gamma}) \circ \tilde{H}_{\alpha\gamma}, [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \rangle \\ &\qquad + \langle \tilde{H}_{\beta\beta} - U[\Omega_{|\beta|} \circ (U^T \tilde{H}_{\beta\beta} U)]U^T, U[\Omega_{|\beta|} \circ (U^T \tilde{H}_{\beta\beta} U)]U^T \rangle \\ &\quad = 2\langle (E_{\alpha\gamma} - [\Omega_0(d)]_{\alpha\gamma}) \circ \tilde{H}_{\alpha\gamma}, [\Omega_0(d)]_{\alpha\gamma} \circ \tilde{H}_{\alpha\gamma} \rangle \\ &\qquad + \langle (E_{\beta\beta} - \Omega_{|\beta|}) \circ (U^T \tilde{H}_{\beta\beta} U), \Omega_{|\beta|} \circ (U^T \tilde{H}_{\beta\beta} U) \rangle, \end{aligned}$$

where $E_{\alpha\gamma}$ and $E_{\beta\beta}$ denote the matrices of all ones in $\mathbb{R}^{|\alpha|\times|\gamma|}$ and $\mathbb{R}^{|\beta|\times|\beta|}$, respectively. Notice that the elements in both matrices $\Omega_0(d)$ and $\Omega_{|\beta|}$ are all inside the interval $[0, 1]$. Hence, we conclude that $\langle H - V(0, H), V(0, H) \rangle \geq 0$ for any $V \in \partial_B \Phi(0, W)$ and $H \in \mathbb{S}^n$.

Next, we let $V \in \Phi(0,W)$. By Carathéodory's [9] theorem, there exist a positive integer q and $V^i \in \partial_B \Phi(0,W)$, $i = 1,\ldots,q$ such that V is the convex combination of $V^1,\ldots,V^q$. Therefore, there exist nonnegative scalars $t_1,\ldots,t_q$ such that $V = \sum_{i=1}^q t_i V^i$ with $\sum_{i=1}^q t_i = 1$. Define the convex function $\theta(X) = \langle X, X\rangle, X \in \mathbb{S}^n$. By the convexity of θ, we have

$$\langle V(0,H), V(0,H)\rangle = \theta(V(0,H)) = \theta\left(\sum_{i=1}^q t_i V^i(0,H)\right) \le \sum_{i=1}^q t_i \theta(V^i(0,H)) = \langle H, V(0,H)\rangle.$$

This completes the proof.

Appendix E. Proof of Lemma 6

Simple calculations give

$$\hat{\mathcal{E}}'(\epsilon,X,y,Z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \kappa_p y & \mathcal{A} & \kappa_p \epsilon I_m & 0 \\ 0 & 0 & -\mathcal{A}^* & -\mathcal{I} \\ \kappa_c X - \Phi_1'(\epsilon, X-Z) & (1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon, X-Z) & 0 & \Phi_2'(\epsilon, X-Z) \end{pmatrix},$$

where Φ_1' and Φ_2' denote the partial derivatives of Φ with respect to the first and second arguments, respectively, and $\mathcal{I}$ is the identity map over $\mathbb{S}^n$. To show that $\hat{\mathcal{E}}'(\epsilon,X,y,Z)$ is nonsingular, it suffices to show that the following system of linear equations,

$$\hat{\mathcal{E}}'(\epsilon,X,y,Z)(\Delta\epsilon,\Delta X,\Delta y,\Delta Z) = 0, \quad (\Delta\epsilon,\Delta X,\Delta y,\Delta Z) \in \mathbb{R}\times\mathbb{X},$$

has only the trivial solution $(\Delta\epsilon,\Delta X,\Delta y,\Delta Z) = (0,0,0,0)$. It is obvious that $\Delta\epsilon = 0$. Because $(1+\kappa_c\epsilon)\Delta X - \Phi_2'(\epsilon,X-Z)\Delta X + \Phi_2'(\epsilon,X-Z)\Delta Z = 0$, it follows that

$$\Delta X = -((1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon,X-Z))^{-1}\Phi_2'(\epsilon,X-Z)\Delta Z. \tag{E.1}$$

By the equality $-\mathcal{A}^*\Delta y - \Delta Z = 0$, it follows that $\Delta Z = -\mathcal{A}^*\Delta y$, which together with (E.1) implies that

$$\Delta X = ((1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon,X-Z))^{-1}\Phi_2'(\epsilon,X-Z)\mathcal{A}^*\Delta y. \tag{E.2}$$

Using the fact that $\mathcal{A}\Delta X + \kappa_p\epsilon\Delta y = 0$ and (E.2), we get

$$(\kappa_p\epsilon I_m + \mathcal{A}((1+\kappa_c\epsilon)\mathcal{I} - \Phi_2'(\epsilon,X-Z))^{-1}\Phi_2'(\epsilon,X-Z)\mathcal{A}^*)\Delta y = 0.$$

Because $0 \preceq \Phi_2'(\epsilon,X-Z) \preceq \mathcal{I}$, it is easy to show that the coefficient matrix in the above linear system is symmetric positive definite. Thus, $\Delta y = 0$, which further implies that $\Delta X = \Delta Z = 0$. Therefore, the proof is completed.

Appendix F. Proof of Lemma 7

Because $\epsilon' \in \mathbb{R}_{++}$, by Lemma 6, $\hat{\mathcal{E}}'(\epsilon',X',y',Z')$ is nonsingular. Because $\hat{\mathcal{E}}$ is continuously differentiable around (ϵ',X',y',Z'), there exists an open neighborhood $\mathcal{U}$ of (ϵ',X',y',Z') such that for any (ϵ,X,y,Z), $\hat{\mathcal{E}}'(\epsilon,X,y,Z)$ is nonsingular. Therefore, the existence of $\mathbf{\Delta}$ is guaranteed.

Denote $\mathcal{R}(\epsilon,X,y,Z) := \mathcal{E}(\epsilon,X,y,Z) + \mathcal{E}'(\epsilon,X,y,Z)\mathbf{\Delta}$. Then, one can verify that $(\Delta\epsilon,\Delta X,\Delta y,\Delta Z)$ is the unique solution of the following equation

$$\hat{\mathcal{E}}(\epsilon,X,y,Z) + \hat{\mathcal{E}}'(\epsilon,X,y,Z)\mathbf{\Delta} = \begin{pmatrix} \zeta(\epsilon,X,y,Z)\hat{\epsilon} \\ \mathcal{R}(\epsilon,X,y,Z) \end{pmatrix}.$$

Hence, it holds that

$$\begin{aligned}
\langle \nabla\psi(\epsilon,X,y,Z), \mathbf{\Delta}\rangle &= \langle 2\nabla\hat{\mathcal{E}}(\epsilon,X,y,Z)\hat{\mathcal{E}}(\epsilon,X,y,Z), \mathbf{\Delta}\rangle \\
&= \left\langle 2\hat{\mathcal{E}}(\epsilon,X,y,Z), \begin{pmatrix} \zeta(\epsilon,X,y,Z)\hat{\epsilon} \\ \mathcal{R}(\epsilon,X,y,Z) \end{pmatrix} - \hat{\mathcal{E}}(\epsilon,X,y,Z)\right\rangle \\
&= -2\psi(\epsilon,X,y,Z) + 2\zeta(\epsilon,X,y,Z)\epsilon\hat{\epsilon} + 2\langle \mathcal{R}(\epsilon,X,y,Z), \mathcal{E}(\epsilon,X,y,Z)\rangle \\
&\le -2\psi(\epsilon,X,y,Z) + 2r\min\{1, \psi(\epsilon,X,y,Z)^{(1+\tau)/2}\}\epsilon\hat{\epsilon} + 2\hat{\eta}\psi(\epsilon,X,y,Z)^{1/2}\|\mathcal{E}(\epsilon,X,y,Z)\|.
\end{aligned} \tag{F.1}$$

We consider two possible cases when $\psi(\epsilon,X,y,Z) > 1$ and $\psi(\epsilon,X,y,Z) \le 1$.

If $\psi(\epsilon,X,y,Z)>1$, then (F.1) implies that

$$\begin{aligned}
&\langle\nabla\psi(\epsilon,X,y,Z),\mathbf{\Delta}\rangle\\
&\le -2\psi(\epsilon,X,y,Z)+2r\epsilon\hat{\epsilon}+2\hat{\eta}\psi(\epsilon,X,y,Z)^{1/2}\sqrt{\psi(\epsilon,X,y,Z)-\epsilon^2}\\
&\le -2\psi(\epsilon,X,y,Z)+2\max\{r\hat{\epsilon},\hat{\eta}\}\left(\epsilon+\psi(\epsilon,X,y,Z)^{1/2}\sqrt{\psi(\epsilon,X,y,Z)-\epsilon^2}\right)\\
&\le -2\psi(\epsilon,X,y,Z)+2\max\{r\hat{\epsilon},\hat{\eta}\}\psi(\epsilon,X,y,Z)\\
&=2(\sqrt{2}\max\{r\hat{\epsilon},\hat{\eta}\}-1)\psi(\epsilon,X,y,Z).
\end{aligned}$$

If $\psi(\epsilon,X,y,Z)\le 1$, then (F.1) implies that

$$\begin{aligned}
&\langle\nabla\psi(\epsilon,X,y,Z),\mathbf{\Delta}\rangle\\
&\le -2\psi(\epsilon,X,y,Z)+2r\psi(\epsilon,X,y,Z)^{(1+\tau)/2}\epsilon\hat{\epsilon}+2\hat{\eta}\psi(\epsilon,X,y,Z)^{1/2}\sqrt{\psi(\epsilon,X,y,Z)-\epsilon^2}\\
&\le -2\psi(\epsilon,X,y,Z)+2r\psi(\epsilon,X,y,Z)\hat{\epsilon}+2\hat{\eta}\psi(\epsilon,X,y,Z)^{1/2}\sqrt{\psi(\epsilon,X,y,Z)-\epsilon^2}\\
&\le -2\psi(\epsilon,X,y,Z)+2\max\{r\hat{\epsilon},\hat{\eta}\}\psi(\epsilon,X,y,Z)^{1/2}\left(\epsilon\psi(\epsilon,X,y,Z)^{1/2}+\sqrt{\psi(\epsilon,X,y,Z)-\epsilon^2}\right)\\
&\le -2\psi(\epsilon,X,y,Z)+2\max\{r\hat{\epsilon},\hat{\eta}\}\psi(\epsilon,X,y,Z)\\
&=2(\sqrt{2}\max\{r\hat{\epsilon},\hat{\eta}\}-1)\psi(\epsilon,X,y,Z).
\end{aligned}$$

For both cases, we always have that

$$\langle\nabla\psi(\epsilon,X,y,Z),\mathbf{\Delta}\rangle\le 2(\sqrt{2}\max\{r\hat{\epsilon},\hat{\eta}\}-1)\psi(\epsilon,X,y,Z). \tag{F.2}$$

Because $\nabla\psi(\cdot)$ is uniformly continuous on $\mathcal{U}$, for all $(\epsilon,X,y,Z)\in\mathcal{U}$ with $\epsilon>0$, we have from the Taylor expansion that

$$\psi(\epsilon+\alpha\Delta\epsilon,X+\alpha\Delta X,y+\alpha\Delta y,Z+\alpha\Delta Z)=\psi(\epsilon,X,y,Z)+\alpha\langle\nabla\psi(\epsilon,X,y,Z),\mathbf{\Delta}\rangle+o(\alpha). \tag{F.3}$$

Combining (F.2) and (F.3), it holds that

$$\begin{aligned}
\psi(\epsilon+\alpha\Delta\epsilon,X+\alpha\Delta X,y+\alpha\Delta y,Z+\alpha\Delta Z)&=\psi(\epsilon,X,y,Z)+2\alpha(\delta-1)\psi(\epsilon,X,y,Z)+o(\alpha)\\
&=[1-2\sigma(1-\delta)\alpha]\psi(\epsilon,X,y,Z)+o(\alpha),
\end{aligned}$$

and the proof is completed.

Appendix G. Proof of Lemma 8

Because $(\epsilon^k,X^k,y^k,Z^k)\in\mathcal{N}$, we have directly from the definition of $\mathcal{N}$ that $\epsilon^k\ge\zeta(\epsilon^k,X^k,y^k,Z^k)\hat{\epsilon}$. Recall that $\Delta\epsilon^k=-\epsilon^k+\zeta(\epsilon^k,X^k,y^k,Z^k)\hat{\epsilon}\le 0$. It holds that

$$\begin{aligned}
&\epsilon^k+\alpha\Delta\epsilon^k-\zeta(\epsilon^k+\alpha\Delta\epsilon^k,X^k+\alpha\Delta X^k,y^k+\alpha\Delta y^k,Z^k+\alpha\Delta Z^k)\hat{\epsilon}\\
&\ge \epsilon^k+\Delta\epsilon^k-\zeta(\epsilon^k+\alpha\Delta\epsilon^k,X^k+\alpha\Delta X^k,y^k+\alpha\Delta y^k,Z^k+\alpha\Delta Z^k)\hat{\epsilon}\\
&= \zeta(\epsilon^k,X^k,y^k,Z^k)\hat{\epsilon}-\zeta(\epsilon^k+\alpha\Delta\epsilon^k,X^k+\alpha\Delta X^k,y^k+\alpha\Delta y^k,Z^k+\alpha\Delta Z^k)\hat{\epsilon}\\
&\ge 0,
\end{aligned}$$

which indeed implies that $(\epsilon^k+\alpha\Delta\epsilon^k,X^k+\alpha\Delta X^k,y^k+\alpha\Delta y^k,Z^k+\alpha\Delta Z^k)\in\mathcal{N}$. This completes the proof.

Appendix H. Proof Theorem 2

It follows from Lemma 7 and Lemma 8 that Algorithm 1 is well-defined and generates an infinite sequence containing in $\mathcal{N}$. Because the line-search scheme is well-defined, it is obvious that

$$\psi(\epsilon^{k+1},X^{k+1},y^{k+1},Z^{k+1})<\psi(\epsilon^k,X^k,y^k,Z^k),\quad\forall k\ge 0.$$

The monotonically decreasing property of the sequence $\{\psi(\epsilon^k,X^k,y^k,Z^k)\}$ then implies that $\{\zeta_k\}$ is also monotonically decreasing. Hence, there exist $\overline{\psi}$ and $\overline{\zeta}$ such that

$$\psi(\epsilon^k,X^k,y^k,Z^k)\to\overline{\psi},\quad\zeta_k\to\overline{\zeta},\quad k\to\infty.$$

Note that $\overline{\psi}\ge 0$. We now prove that $\overline{\psi}=0$ by contradiction. To this end, we assume that $\overline{\psi}>0$, and hence, $\overline{\zeta}>0$. By the fact that the sequence $\{(\epsilon^k,X^k,y^k,Z^k)\}$ is contained in $\mathcal{N}$, we see that there exists an $\epsilon'>0$ such that $\epsilon^k>\epsilon'>0$ for all $k\ge 0$. Also note that $\epsilon^k\le\psi(\epsilon^k,X^k,y^k,Z^k)^{1/2}\le\psi(\epsilon^0,X^0,y^0,Z^0)^{1/2}$. Therefore, there exists an $\epsilon''>\epsilon'$ such that $\epsilon^k\in[\epsilon',\epsilon'']$ for all $k\ge 0$.

We next claim that $\{(X^k, y^k, Z^k)\}$ is bounded. For $k \geq 0$, let

$$R_1^k := \mathcal{A}X^k + \kappa_p \epsilon^k y^k - b,\ R_2^k := -\mathcal{A}^* y^k - Z^k + C,\ R_3^k := (1 + \kappa_c \epsilon^k)X^k - \Phi(\epsilon^k, X^k - Z^k).$$

Because $\{\psi(\epsilon^k, X^k, y^k, Z^k)\}$ is bounded, we have that $\{R_1^k\}$, $\{R_2^k\}$ and $\{R_3^k\}$ are bounded. Using the first two equalities in the above, Z^k can be expressed as

$$Z^k = -\mathcal{A}^* y^k + C - R_2^k = \frac{1}{\kappa_p \epsilon^k}\mathcal{A}^*\mathcal{A}X^k + C^k,$$

where $\left\{C^k := -\frac{1}{\kappa_p \epsilon^k}\mathcal{A}^*(b + R_1^k) + C - R_2^k\right\}$ is bounded and does not depend on $\{X^k\}$. Substituting this expression into the third equality yields that

$$(1 + \kappa_c \epsilon^k)X^k - \Phi\left(\epsilon^k, X^k - \frac{1}{\kappa_p \epsilon^k}\mathcal{A}^*\mathcal{A}X^k - C^k\right) = R_3^k.$$

Recall that Φ is a globally Lipschitz continuous mapping, and $\Phi(0, \cdot) = \Pi_{\mathbb{S}_+^n}(\cdot)$. We see that there exists a bounded sequence $\{R_4^k\}$ such that

$$\Phi\left(\epsilon^k, X^k - \frac{1}{\kappa_p \epsilon^k}\mathcal{A}^*\mathcal{A}X^k - C^k\right) = \Pi_{\mathbb{S}_+^n}\left(X^k - \frac{1}{\kappa_p \epsilon^k}\mathcal{A}^*\mathcal{A}X^k\right) + R_4^k, \quad \forall k \geq 0.$$

Consequently, it holds that

$$X^k - \Pi_{\mathbb{S}_+^n}\left(\frac{1}{1 + \kappa_c \epsilon^k}X^k - \frac{1}{\kappa_p \epsilon^k(1 + \kappa_c \epsilon^k)}\mathcal{A}^*\mathcal{A}X^k\right) = R_5^k,$$

where $\left\{R_5^k := (1 + \kappa_c \epsilon^k)^{-1}(R_3^k - R_4^k)\right\}$ is also bounded. Notice that $\kappa_p > 0, \kappa_c > 0$, and for any $\epsilon \in [\epsilon', \epsilon'']$, it holds that

$$\frac{1}{1 + \kappa_c \epsilon}X - \frac{1}{\kappa_p \epsilon(1 + \kappa_c \epsilon)}\mathcal{A}^*\mathcal{A}X = X - \frac{1}{\kappa_p \epsilon''(1 + \kappa_c \epsilon'')}\mathcal{A}^*\mathcal{A}X - \frac{\kappa_c \epsilon}{1 + \kappa_c \epsilon}X - \left(\frac{1}{\kappa_p \epsilon(1 + \kappa_c \epsilon)} - \frac{1}{\kappa_p \epsilon''(1 + \kappa_c \epsilon'')}\right)\mathcal{A}\mathcal{A}^* X.$$

One can derive from theorems 2.8 and 4.4 in Sun and Qi [59] that for any positive constant c the set

$$\left\{X \in \mathbb{S}^n\ :\ \left\|X - \Pi_{\mathbb{S}_+^n}\left(\frac{1}{1 + \kappa_c \epsilon}X - \frac{1}{\kappa_p \epsilon(1 + \kappa_c \epsilon)}\mathcal{A}^*\mathcal{A}X\right)\right\| \leq c, \epsilon \in [\epsilon', \epsilon'']\right\}$$

is bounded. Because there exists a sufficiently large $c > 0$ such that the sequence $\{X^k\}$ remains within the associated bounded set, we conclude that the sequence $\{X^k\}$ is bounded, which in turn implies that $\{(\epsilon^k, X^k, y^k, Z^k)\}$ is also bounded. In this case, an accumulation point must exist.

Let $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$ be any accumulation point of $\{(\epsilon^k, X^k, y^k, Z^k)\}$. By taking a subsequence if necessary, we may assume that $\{(\epsilon^k, X^k, y^k, Z^k)\}$ converges to $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$. Then, by the continuity of $\psi(\cdot)$, it holds that

$$\overline{\psi} = \psi(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z}), \quad \overline{\zeta} = \zeta(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z}), \quad (\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z}) \in \mathcal{N}.$$

By the fact that $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z}) \in \mathcal{N}$, we see that $\overline{\epsilon} \in \mathbb{R}_{++}$. Therefore, by Lemma 6, we see that there exists a neighborhood of $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$, denoted by $\mathcal{U}$, such that $\hat{\mathcal{E}}'(\epsilon, X, y, Z)$ is nonsingular for any $(\epsilon, X, y, Z) \in \mathcal{U}$ with $\epsilon > 0$. Note that for k sufficiently large, we have that $(\epsilon^k, X^k, y^k, Z^k)$ belongs to $\mathcal{U}$ with $\epsilon^k > 0$. Furthermore, by Lemma 7, there also exists $\overline{\alpha} \in (0, 1]$ such that for any $\alpha \in (0, \overline{\alpha}]$,

$$\psi(\epsilon^k + \alpha\Delta\epsilon^k, X^k + \alpha\Delta X^k, y^k + \alpha\Delta y^k, Z^k + \alpha\Delta Z^k) \leq [1 - 2\sigma(1 - \delta)\alpha]\psi(\epsilon^k, X^k, y^k, Z^k)$$

for $k \geq 0$ sufficiently large. The existence of the fixed number $\overline{\alpha} \in (0, 1]$ further indicates that there exists a nonnegative integer ℓ such that $\rho^\ell \in (0, \overline{\alpha}]$ and $\rho^{\ell_k} \geq \rho^\ell$ for all k sufficiently large. Therefore, it holds that

$$\begin{aligned}\psi(\epsilon^{k+1}, X^{k+1}, y^{k+1}, Z^{k+1}) &\leq [1 - 2\sigma(1 - \delta)\rho^{\ell_k}]\psi(\epsilon^k, X^k, y^k, Z^k)\\ &\leq [1 - 2\sigma(1 - \delta)\rho^{\ell}]\psi(\epsilon^k, X^k, y^k, Z^k),\end{aligned}$$

for all sufficiently large k. The above inequality implies that $\overline{\psi} \leq 0$, which is a contradiction to the assumption $\overline{\psi} > 0$. Hence, $\overline{\psi} = 0$, which implies that $\hat{\mathcal{E}}(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z}) = 0$.

Last, we assume that the solution set to the KKT system is nonempty and bounded. One can check that $\hat{\mathcal{E}}(\cdot)$ is a weakly univalent function (Gowda and Sznajder [20]). Then, from Ravindran and Gowda ([46]), one can verify that $\hat{\mathcal{E}}^{-1}(0)$ is also nonempty and bounded, which further implies the boundedness of the sequence $\{(\epsilon^k, X^k, y^k, Z^k)\}$. Thus, the proof is completed.

Appendix I. Proof of Lemma 9

Suppose that $\overline{W}$ has the spectral decomposition (7). Denote $\Delta\tilde{X} := P^T\Delta X P$ and $\Delta\tilde{Z} = P^T\Delta Z P$. For the index set β, define $\Phi_{|\beta|}$ as before. Then, by Lemma 4, there exists $V_{|\beta|} \in \partial\Phi_{|\beta|}(0,0)$ such that

$$V(0,\Delta X-\Delta Z) = P\begin{pmatrix} \Delta\tilde{H}_{\alpha\alpha} & \Delta\tilde{H}_{\alpha\beta} & [\Omega_0(d)]_{\alpha\gamma}\circ\Delta\tilde{H}_{\alpha\gamma} \\ \Delta\tilde{H}_{\alpha\beta}^T & V_{|\beta|}(0,\Delta\tilde{H}_{\beta\beta}) & 0 \\ \Delta\tilde{H}_{\alpha\gamma}^T\circ[\Omega_0(d)]_{\alpha\gamma}^T & 0 & 0 \end{pmatrix}P^T,$$

where $\Delta\tilde{H} = \Delta\tilde{X}-\Delta\tilde{Z}$. Comparing both sides of the relation $\Delta X = V(0,\Delta X-\Delta Z)$ yields that

$$\Delta\tilde{Z}_{\alpha\alpha}=0,\quad \Delta\tilde{Z}_{\alpha\beta}=0,\quad \Delta\tilde{X}_{\beta\gamma}=0,\quad \Delta\tilde{X}_{\gamma\gamma}=0,$$

and that

$$\Delta\tilde{X}_{\beta\beta} = V_{|\beta|}(0,\Delta\tilde{X}_{\beta\beta}-\Delta\tilde{Z}_{\beta\beta}),\ \Delta\tilde{X}_{\alpha\gamma}-[\Omega_0(d)]_{\alpha\gamma}\circ\Delta\tilde{X}_{\alpha\gamma} = [\Omega_0(d)]_{\alpha\gamma}\circ\Delta\tilde{Z}_{\alpha\gamma}.$$

By Lemma 5, it holds that

$$\langle\Delta\tilde{X}_{\beta\beta},\ -\Delta\tilde{Z}_{\beta\beta}\rangle = \langle V_{|\beta|}(0,\Delta\tilde{X}_{\beta\beta}-\Delta\tilde{Z}_{\beta\beta}),(\Delta\tilde{X}_{\beta\beta}-\Delta\tilde{Z}_{\beta\beta})-V_{|\beta|}(0,\Delta\tilde{X}_{\beta\beta}-\Delta\tilde{Z}_{\beta\beta})\rangle\ge 0$$

which implies that

$$\langle\Delta X,\Delta Z\rangle = \langle\tilde{\Delta}X,\tilde{\Delta}Z\rangle = \langle\Delta\tilde{X}_{\beta\beta},\Delta\tilde{Z}_{\beta\beta}\rangle + 2\langle\Delta\tilde{X}_{\alpha\gamma},\Delta\tilde{Z}_{\alpha\gamma}\rangle \le 2\langle\Delta\tilde{X}_{\alpha\gamma},\Delta\tilde{Z}_{\alpha\gamma}\rangle.$$

However, simple calculations show that $\Gamma_{\overline{X}}(-\overline{Z},\Delta X) = 2\langle\Delta\tilde{X}_{\alpha\gamma},\Delta\tilde{Z}_{\alpha\gamma}\rangle$. This proves the lemma.

Appendix J. Proof of Proposition 2

It is obvious that part 3 implies part 2. From (17), we see that for any $V_0 \in \partial_B\Pi_{\mathbb{S}^n_+}(W)$, $W\in\mathbb{S}^n$, there exists $V\in\partial_B\Phi(0,W)$ such that

$$V_0(H) = V(0,H),\quad \forall H\in\mathbb{S}^n.$$

Therefore, by Theorem 1, part 2 implies part 1. So, it suffices to show that part 1 implies part 3.

Now, suppose that part 1 holds. Recall that the primal constraint nondegeneracy condition at $\overline{X}$ implies that $\mathcal{M}(\overline{X}) = \{(\overline{y},\overline{Z})\}$; that is, the dual multiplier is unique. Now, by using Lemma 2, we can see that the dual constraint nondegeneracy at $(\overline{y},\overline{Z})$ implies that the strong second-order sufficient condition holds at $\overline{X}$. In particular, it holds that

$$-\Gamma_{\overline{X}}(-\overline{Z},H) > 0,\quad \forall 0\ne H\in \mathrm{app}(\overline{y},\overline{Z}). \tag{J.1}$$

Let $U\in\partial\hat{\mathcal{E}}(\overline{\epsilon},\overline{X},\overline{y},\overline{Z})$ (recall that $\overline{\epsilon}=0$). Then there exists $V\in\partial\Phi(0,\overline{X}-\overline{Z})$ such that

$$U(\Delta\epsilon,\Delta X,\Delta y,\Delta Z) = \begin{pmatrix} \Delta\epsilon \\ \mathcal{A}\Delta X \\ -\mathcal{A}^*\Delta y-\Delta Z \\ \Delta X - V(\Delta\epsilon,\Delta X-\Delta Z)\end{pmatrix},\quad (\Delta\epsilon,\Delta X,\Delta y,\Delta Z)\in\mathbb{R}\times\mathbb{X}.$$

To show that U is nonsingular, it suffices to show that $U(\Delta\epsilon,\Delta X,\Delta y,\Delta Z)=0$ implies that $(\Delta\epsilon,\Delta X,\Delta y,\Delta Z)=0$. So, let us assume that $U(\Delta\epsilon,\Delta X,\Delta y,\Delta Z)=0$, that is,

$$\begin{pmatrix} \Delta\epsilon \\ \mathcal{A}\Delta X \\ -\mathcal{A}^*\Delta y-\Delta Z \\ \Delta X - V(\Delta\epsilon,\Delta X-\Delta Z)\end{pmatrix} = 0. \tag{J.2}$$

We first prove that $\Delta X=0$ by contradiction. To this end, we assume that $\Delta X\ne 0$. By Lemma 4 and (J.2), one can verify that

$$\Delta X\in\mathrm{app}(\overline{y},\overline{Z}). \tag{J.3}$$

Then by (J.1), (J.3) implies that

$$-\Gamma_{\overline{X}}(-\overline{Z},\Delta X) > 0. \tag{J.4}$$

On the other hand, by (J.2), it holds that $\langle\Delta X,\Delta Z\rangle = \langle\Delta X,\ -\mathcal{A}^*\Delta y\rangle = \langle\mathcal{A}\Delta X,\ -\Delta y\rangle = 0$, which together with Lemma 9, yields that

$$\Gamma_{\overline{X}}(-\overline{Z},\Delta X)\ge\langle\Delta X,\Delta Z\rangle = 0. \tag{J.5}$$

However, (J.5) contradicts to (J.4). Thus, $\Delta X=0$ and $V(0,\ -\Delta Z)=0$.

We next prove $\Delta y = 0$ and $\Delta Z = 0$. From Lemma 4, $V(0, -\Delta Z) = 0$ implies that

$$P_\alpha^T \Delta Z P_\alpha = 0, \quad P_\alpha^T \Delta Z P_\beta = 0, \quad P_\alpha^T \Delta Z P_\gamma = 0.$$

On the other hand, from (J.2), we get $\mathcal{A}^*\Delta y + \Delta Z = 0$. Because the primal constraint nondegeneracy condition (9) holds at $\overline{X}$, there exist $X \in \mathbb{S}^n$ and $Z \in \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\overline{X}))$ such that $\mathcal{A}X = \Delta y$, and $X + Z = \Delta Z$. As a consequence, it holds that

$$\begin{aligned}\langle \Delta y, \Delta y\rangle + \langle \Delta Z, \Delta Z\rangle &= \langle \mathcal{A}X, \Delta y\rangle + \langle X + Z, \Delta Z\rangle = \langle \mathcal{A}X, \Delta y\rangle + \langle X, -\mathcal{A}^*(\Delta y)\rangle + \langle Z, \Delta Z\rangle \\ &= \langle Z, \Delta Z\rangle = \langle P^T Z P, P^T \Delta Z P\rangle = 0,\end{aligned}$$

where we have used the fact that $P_\alpha^T \Delta Z P_\alpha = 0$, $P_\alpha^T \Delta Z P_\beta = 0$, $P_\alpha^T \Delta Z P_\gamma = 0$, $Z \in \mathrm{lin}(\mathcal{T}_{\mathbb{S}^n_+}(\overline{X}))$. Thus, $\Delta y = 0$, $\Delta Z = 0$, and hence, U is nonsingular. Thus, the proof is completed.

Appendix K. Proof of Theorem 3

By Theorem 2, we see that $\hat{\mathcal{E}}(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z}) = 0$ and, in particular, $\overline{\epsilon} = 0$. Then by Proposition 2, the primal and dual constraint nondegeneracy conditions imply that every element of $\partial\hat{\mathcal{E}}(0, \overline{X}, \overline{y}, \overline{Z})$ (hence, of $\partial_B\hat{\mathcal{E}}(0, \overline{X}, \overline{y}, \overline{Z})$) is nonsingular. As a consequence (see e.g., proposition 3.1 in Qi and Sun [44]), for all k sufficiently large, it holds that

$$\|\hat{\mathcal{E}}'(\epsilon^k, X^k, y^k, Z^k)^{-1}\| = O(1). \tag{K.1}$$

For simplicity, in this proof, we denote $\overline{w} := (\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})^T$, $w^k := (\epsilon^k, X^k, y^k, Z^k)^T$, $\Delta w^k := (\Delta\epsilon^k, \Delta X^k, \Delta y^k, \Delta Z^k)^T$, $\hat{\mathcal{E}}_k := \hat{\mathcal{E}}(\epsilon^k, X^k, y^k, Z^k)$, $\hat{\mathcal{E}}_* := \hat{\mathcal{E}}(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$ and $\mathcal{J}_k := \hat{\mathcal{E}}'(\epsilon^k, X^k, y^k, Z^k)$ for $k \geq 0$. Then, we can verify that

$$\begin{aligned}\|w^k + \Delta w^k - \overline{w}\| &= \left\| w^k + \mathcal{J}_k^{-1}\left(\begin{pmatrix}\zeta_k\hat{\epsilon}\\ \mathcal{R}_k\end{pmatrix} - \hat{\mathcal{E}}_k\right) - \overline{w}\right\| = \left\| -\mathcal{J}_k^{-1}\left(\hat{\mathcal{E}}_k - \mathcal{J}_k(w^k - \overline{w}) - \begin{pmatrix}\zeta_k\hat{\epsilon}\\ \mathcal{R}_k\end{pmatrix}\right)\right\| \\ &= O(\|\hat{\mathcal{E}}_k - \mathcal{J}_k(w^k - \overline{w})\|) + O(\|\hat{\mathcal{E}}_k\|^{1+\tau}) + O(\|\mathcal{R}_k\|).\end{aligned} \tag{K.2}$$

Because $\hat{\mathcal{E}}$ is locally Lipschitz continuous at $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$, for all k sufficiently large, it holds that

$$\|\hat{\mathcal{E}}_k\| = \|\hat{\mathcal{E}}_k - \hat{\mathcal{E}}_*\| = O(\|w^k - \overline{w}\|). \tag{K.3}$$

Moreover, because $\mathcal{E}'_\epsilon$ is bounded near $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$, it holds that

$$\begin{aligned}\|\mathcal{R}_k\| &\leq \eta_k\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k) + \mathcal{E}'_\epsilon(\epsilon^k, X^k, y^k, Z^k)\Delta\epsilon^k\| \\ &\leq O(\|\hat{\mathcal{E}}_k\|^\tau)(\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k)\| + O(|\Delta\epsilon^k|)) \\ &\leq O(\|\hat{\mathcal{E}}_k\|^\tau)(\|\mathcal{E}(\epsilon^k, X^k, y^k, Z^k)\| + O(|-\epsilon^k + \zeta_k\hat{\epsilon}|)) \\ &\leq O(\|\hat{\mathcal{E}}_k\|^{1+\tau}) = O(\|\hat{\mathcal{E}}_k - \hat{\mathcal{E}}_*\|^{1+\tau}) = O(\|w^k - \overline{w}\|^{1+\tau}).\end{aligned} \tag{K.4}$$

Because Φ is strongly semismooth everywhere (see Proposition 1), $\hat{\mathcal{E}}$ is also strongly semismooth at $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$. Thus, for k sufficiently large, it holds that

$$\|\hat{\mathcal{E}}_k - \mathcal{J}_k(w^k - \overline{w})\| = O(\|w^k - \overline{w}\|^2),$$

Which, together with (K.2)–(K.4), implies that

$$\|w^k + \Delta w^k - \overline{w}\| = O(\|w^k - \overline{w}\|^{1+\tau}). \tag{K.5}$$

Now, by using the strong semi-smoothness of $\hat{\mathcal{E}}$ again and (K.1), we can show that for k sufficiently large,

$$\|w^k - \overline{w}\| = O(\|\hat{\mathcal{E}}_k\|). \tag{K.6}$$

Combining (K.5) and (K.6) and the fact that $\hat{\mathcal{E}}$ (hence, ψ) is locally Lipschitz continuous at $(\overline{\epsilon}, \overline{X}, \overline{y}, \overline{Z})$, we get for k sufficiently large that

$$\begin{aligned}&\psi(\epsilon^k + \Delta\epsilon^k, X^k + \Delta X^k, y^k + \Delta y^k, Z^k + \Delta Z^k) \\ &\quad= \|\hat{\mathcal{E}}(\epsilon^k + \Delta\epsilon^k, X^k + \Delta X^k, y^k + \Delta y^k, W^k + \Delta W^k) - \hat{\mathcal{E}}_*\|^2 \\ &\quad= O(\|w^k + \Delta w^k - \overline{w}\|^2) = O(\|w^k - \overline{w}\|^{2(1+\tau)}) = O(\|\hat{\mathcal{E}}_k\|^{2(1+\tau)}) \\ &\quad= O(\|\psi(\epsilon^k, X^k, y^k, Z^k)\|^{1+\tau}) = o(\|\psi(\epsilon^k, X^k, y^k, Z^k)\|).\end{aligned}$$

This shows that, for k sufficiently large, $w^{k+1} = w^k + \Delta w^k$, that is, the unit step size is eventually accepted. Therefore, the proof is completed.

Endnotes

[1] Note, however, that the strong duality may not hold for (1) and (2) in general (see, e.g., Vandenberghe and Boyd [66]).

[2] See also the benchmark website https://plato.asu.edu/ftp/sparse_sdp.html for state-of-the-art IPM-based SDP solvers.

[3] In the sense that n is several thousands and m is several hundreds of thousands.

[4] Based on our numerical experience, interior point methods such as Mosek become inefficient and/or incapable of handling SDPs with large m, say, $m \geq 10,000$, and first-order methods are typically quite slow to reach our targeted accuracy. Moreover, to the best of our knowledge, there is no publicly available implementation for the CHKS-based smoothing Newton method.

[5] In this case, we use the routine designed for solving general SDPs with arbitrary linear constraints provided by https://github.com/wangjie212/ManiSDP-matlab.

References

[1] Alizadeh F, Haeberly JPA, Overton ML (1997) Complementarity and nondegeneracy in semidefinite programming. *Math. Program.* 77(1):111–128.
[2] Alizadeh F, Haeberly JPA, Overton ML (1998) Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM J. Optim.* 8(3):746–768.
[3] Bonnans JF, Shapiro A (2013) *Perturbation Analysis of Optimization Problems* (Springer Science & Business Media, New York).
[4] Boumal N, Voroninski V, Bandeira A (2016) The non-convex Burer-Monteiro approach works on smooth semidefinite programs. *Adv. Neural Inf. Process. Syst.* 29.
[5] Boumal N, Mishra B, Absil PA, Sepulchre R (2014) Manopt, a Matlab toolbox for optimization on manifolds. *J. Mach. Learn. Res.* 15(42):1455–1459.
[6] Burer S (2009) On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Programming* 120(2):479–495.
[7] Burer S, Monteiro RD (2003) A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Programming* 95(2):329–357.
[8] Burer S, Monteiro RD (2005) Local minima and convergence in low-rank semidefinite programming. *Math. Programming* 103(3):427–444.
[9] Carathéodory C (1911) Über den variabilitätsbereich der Fourier'schen konstanten von positiven harmonischen funktionen. *Rendiconti Del Circolo Matematico di Palermo (1884-1940)* 32(1):193–217.
[10] Chambolle A, Pock T (2011) A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* 40:120–145.
[11] Chan ZX, Sun D (2008) Constraint nondegeneracy, strong regularity, and nonsingularity in semidefinite programming. *SIAM J. Optim.* 19(1):370–396.
[12] Chen B, Harker PT (1993) A non-interior-point continuation method for linear complementarity problems. *SIAM J. Matrix Anal. Appl.* 14(4):1168–1190.
[13] Chen X, Tseng P (2003) Non-interior continuation methods for solving semidefinite complementarity problems. *Math. Programming* 95(3):431–474.
[14] Chen X, Qi L, Sun D (1998) Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities. *Math. Comp.* 67(222):519–540.
[15] Chen X, Qi H, Tseng P (2003) Analysis of nonsmooth symmetric-matrix-valued functions with applications to semidefinite complementarity problems. *SIAM J. Optim.* 13(4):960–985.
[16] Clarke FH (1990) *Optimization and Nonsmooth Analysis* (Society for Industrial and Applied Mathematics, Philadelphia, PA).
[17] Ding L, Yurtsever A, Cevher V, Tropp JA, Udell M (2021) An optimal-storage approach to semidefinite programming using approximate complementarity. *SIAM J. Optim.* 31(4):2695–2725.
[18] Gao Y, Sun D (2010) Calibrating least squares semidefinite programming with equality and inequality constraints. *SIAM J. Matrix Anal. Appl.* 31(3):1432–1457.
[19] Goemans MX, Williamson DP (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM.* 42(6):1115–1145.
[20] Gowda MS, Sznajder R (1999) Weak univalence and connectedness of inverse images of continuous functions. *Math. Oper. Res.* 24(1):255–261.
[21] Helmberg C, Rendl F (2000) A spectral bundle method for semidefinite programming. *SIAM J. Optim.* 10(3):673–696.
[22] Horn RA, Johnson CR (2012) *Matrix Analysis* (Cambridge University Press, New York).
[23] Jarre F, Rendl F (2008) An augmented primal-dual method for linear conic programs. *SIAM J. Optim.* 19(2):808–823.
[24] Jiang X, Vandenberghe L (2022) Bregman primal–dual first-order method and application to sparse semidefinite programming. *Comput. Optim. Appl.* 81(1):127–159.
[25] Jiang K, Sun D, Toh KC (2012) An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP. *SIAM J. Optim.* 22(3):1042–1064.
[26] Kanzow C (1996) Some noninterior continuation methods for linear complementarity problems. *SIAM J. Matrix Anal. Appl.* 17(4):851–868.
[27] Kanzow C, Nagel C (2005) Quadratic convergence of a nonsmooth newton-type method for semidefinite programs without strict complementarity. *SIAM J. Optim.* 15(3):654–672.
[28] Kanzow C, Pieper H (1999) Jacobian smoothing methods for nonlinear complementarity problems. *SIAM J. Optim.* 9(2):342–373.
[29] Kim S, Kojima M, Mevissen M, Yamashita M (2011) Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. *Math. Programming* 129(1):33–68.
[30] Kochenberger G, Hao JK, Glover F, Lewis M, Lü Z, Wang H, Wang Y (2014) The unconstrained binary quadratic programming problem: A survey. *J. Comb. Optim.* 28:58–81.
[31] Kočvara M (2021) Decomposition of arrow type positive semidefinite matrices with application to topology optimization. *Math. Programming* 190(1–2):105–134.
[32] Lovász L (1979) On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory* 25(1):1–7.
[33] Majumdar A, Hall G, Ahmadi AA (2020) Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annu. Rev. Control. Robot. Auton. Syst.* 3:331–360.

[34] Malick J, Povh J, Rendl F, Wiegele A (2009) Regularization methods for semidefinite programming. *SIAM J. Optim.* 20(1):336–356.
[35] Mifflin R (1977) Semismooth and semiconvex functions in constrained optimization. *SIAM J. Control Optim.* 15(6):959–972.
[36] Nie J, Wang L (2014) Semidefinite relaxations for best rank-1 tensor approximations. *SIAM J. Matrix Anal. Appl.* 35(3):1155–1179.
[37] Pataki G (1998) On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.* 23(2):339–358.
[38] Peng J, Wei Y (2007) Approximating k-means-type clustering via semidefinite programming. *SIAM J. Optim.* 18(1):186–205.
[39] Pereyra V, Reinoza A, Robinson SM (1983) Local structure of feasible sets in nonlinear programming, part I: Regularity. *Numerical Methods: Proc. Internat. Workshop Held Caracas, June 14–18, 1982* (Springer, Berlin), 240–251.
[40] Pinar MÇ, Zenios SA (1994) On smoothing exact penalty functions for convex constrained optimization. *SIAM J. Optim.* 4(3):486–511.
[41] Povh J, Rendl F, Wiegele A (2006) A boundary point method to solve semidefinite programs. *Computing* 78:277–286.
[42] Qi L (1993) Convergence analysis of some algorithms for solving nonsmooth equations. *Math. Oper. Res.* 18(1):227–244.
[43] Qi L, Chen X (1995) A globally convergent successive approximation method for severely nonsmooth equations. *SIAM J. Control Optim.* 33(2):402–418.
[44] Qi L, Sun J (1993) A nonsmooth version of Newton's method. *Math. Programming* 58(1–3):353–367.
[45] Qi L, Sun D, Zhou G (2000) A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities. *Math. Programming* 87:1–35.
[46] Ravindran G, Gowda MS (2001) Regularization of P0-functions in box variational inequality problems. *SIAM J. Optim.* 11(3):748–760.
[47] Renegar J (2014) Efficient first-order methods for linear programming and semidefinite programming. Preprint, submitted September 19, https://arxiv.org/abs/1409.5832.
[48] Robinson SM (1984) *Local Structure of Feasible Sets in Nonlinear Programming, Part II: Nondegeneracy* (Springer, Berlin).
[49] Robinson SM (1987) *Local Structure of Feasible Sets in Nonlinear Programming, Part III: Stability and Sensitivity* (Springer, Berlin).
[50] Robinson SM (1992) Normal maps induced by linear transformations. *Math. Oper. Res.* 17(3):691–714.
[51] Robinson SM (2003) Constraint nondegeneracy in variational analysis. *Math. Oper. Res.* 28(2):201–232.
[52] Rockafellar RT (1976) Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.* 1(2):97–116.
[53] Rockafellar RT (1976) Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* 14(5):877–898.
[54] Rockafellar RT, Wets RJB (2009) *Variational Analysis*, vol. 317 (Springer Science & Business Media, Berlin).
[55] Shapiro A (2002) *On Differentiability of Symmetric Matrix Valued Functions* (School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta).
[56] Smale S (2000) *Algorithms for Solving Equations. The Collected Papers of Stephen Smale*, vol. 3 (World Scientific, Singapore), 1263–1286.
[57] Sun D (2006) The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications. *Math. Oper. Res.* 31(4):761–776.
[58] Sun D, Qi L (1999) On NCP-functions. *Comput. Optim. Appl.* 13:201–220.
[59] Sun D, Qi L (2001) Solving variational inequality problems via smoothing-nonsmooth reformulations. *J. Comput. Appl. Math.* 129(1–2):37–62.
[60] Sun D, Sun J (2002) Semismooth matrix-valued functions. *Math. Oper. Res.* 27(1):150–169.
[61] Sun J, Sun D, Qi L (2004) A squared smoothing newton method for nonsmooth matrix equations and its applications in semidefinite optimization problems. *SIAM J. Optim.* 14(3):783–806.
[62] Sun D, Toh KC, Yang L (2015) A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints. *SIAM J. Optim.* 25(2):882–915.
[63] Todd MJ (2001) Semidefinite optimization. *Acta Numer.* 10:515–560.
[64] Toh KC (2004) Solving large scale semidefinite programs via an iterative solver on the augmented systems. *SIAM J. Optim.* 14(3):670–698.
[65] Toh KC, Kojima M (2002) Solving some large scale semidefinite programs via the conjugate residual method. *SIAM J. Optim.* 12(3):669–691.
[66] Vandenberghe L, Boyd S (1996) Semidefinite programming. *SIAM Rev.* 38(1):49–95.
[67] Wang J, Hu L (2023) Solving low-rank semidefinite programs via manifold optimization. Preprint, submitted March 5, https://arxiv.org/abs/2303.01722.
[68] Wang Y, Deng K, Liu H, Wen Z (2023) A decomposition augmented Lagrangian method for low-rank semidefinite programming. *SIAM J. Optim.* 33(3):1361–1390.
[69] Wiegele A (2007) Biq Mac Library—A collection of Max-Cut and quadratic 0-1 programming instances of medium size. *Preprint* 51.
[70] Wolkowicz H, Saigal R, Vandenberghe L (2012) *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, vol. 27 (Springer Science & Business Media, New York).
[71] Yang L, Sun D, Toh KC (2015) SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Program. Comput.* 7(3):331–366.
[72] Yang H, Liang L, Carlone L, Toh KC (2022) An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. *Math. Programming* 201(1):409–472.
[73] Yurtsever A, Tropp JA, Fercoq O, Udell M, Cevher V (2021) Scalable semidefinite programming. *SIAM J. Math. Data Sci.* 3(1):171–200.
[74] Zhao JY (2004) The smoothing function of the nonsmooth matrix valued function. Master thesis, National University of Singapore, Singapore.
[75] Zhao XY, Sun D, Toh KC (2010) A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.* 20(4):1737–1765.
[76] Zheng Y, Fantuzzi G, Papachristodoulou A, Goulart P, Wynn A (2020) Chordal decomposition in operator-splitting methods for sparse semidefinite programs. *Math. Programming* 180(1–2):489–532.