A stabilized sequential quadratic semidefinite programming method for degenerate nonlinear semidefinite programs

Yuya Yamakawa1 • Takayuki Okuno2,3

Received: 19 April 2021 / Accepted: 21 July 2022 / Published online: 16 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
In this paper, we propose a new sequential quadratic semidefinite programming (SQSDP) method for solving degenerate nonlinear semidefinite programs (NSDPs), in which we produce iteration points by solving a sequence of stabilized quadratic semidefinite programming (QSDP) subproblems, which we derive from the min-max problem associated with the NSDP. Unlike the existing SQSDP methods, the proposed one allows us to solve those QSDP subproblems inexactly, and each QSDP is feasible. One more remarkable point of the proposed method is that constraint qualifications or boundedness of Lagrange multiplier sequences are not required in the global convergence analysis. Specifically, without assuming such conditions, we prove the global convergence to a point satisfying any of the following: the stationary conditions for the feasibility problem, the approximate-Karush–Kuhn–Tucker (AKKT) conditions, and the trace-AKKT conditions. Finally, we conduct some numerical experiments to examine the efficiency of the proposed method.

Keywords Nonlinear semidefinite program • Stabilized sequential quadratic semidefinite programming method • Sequential optimality conditions • Global convergence

Yuya Yamakawa
yuya@i.kyoto-u.ac.jp

Takayuki Okuno
takayuki-okuno@st.seikei.ac.jp

1 Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
2 Faculty of Science and Technology, Seikei University, Kichijouji 1-3-1, Musashino, Tokyo 180-8633, Japan
3 Center for Advanced Intelligence Project, RIKEN, Nihonbashi 1-chome Mitsui Building, 15th Floor, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan
1 Introduction

In this paper, we consider the following nonlinear semidefinite program (NSDP):

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g(x) = 0, \quad X(x) \succeq O,
\end{align*}
\]

where \( f : \mathbb{R}^n \to \mathbb{R}, \) \( g : \mathbb{R}^n \to \mathbb{R}^m, \) and \( X : \mathbb{R}^n \to \mathcal{S}_d \) are twice continuously differentiable functions, and \( \mathcal{S}_d \) denotes the set of \( d \times d \) real symmetric matrices. Let \( \mathcal{S}_d^{++} \) denote the set of \( d \times d \) real symmetric positive (semi)definite matrices. For a matrix \( M \in \mathcal{S}_d, \) \( M \succeq O \) and \( M \succ O \) mean \( M \in \mathcal{S}_d^{++} \) and \( M \in \mathcal{S}_d^{++}, \) respectively.

Moreover, let \( g_1, \ldots, g_m \) be the functions such that \( g(x) = [g_1(x) \cdots g_m(x)]^\top \) for all \( x \in \mathbb{R}^n. \) In particular, when the functions \( f, g, \) and \( X \) are affine, NSDP (1) is a linear semidefinite program (LSDP). If the range of \( X \) is limited to the diagonal matrix space, then, it reduces to the standard nonlinear program (NLP).

NSDPs have wide applications in control, finance, eigenvalue problems, structural optimization, and so forth [24, 33, 34]. Motivated by such practical importance, various algorithms have been developed for solving NSDPs so far, for example, sequential quadratic semidefinite programming (SQSDP) methods [5–7, 12, 37–40], interior point methods [19, 22, 23, 30–32, 34], augmented Lagrangian methods [8, 15, 21, 24, 25, 29], and others [20, 35, 36]. However, in comparison with LSDPs and NLPs, studies on algorithms for NSDPs are still much fewer, and there is a lot of room for studying them in more depth. Our aim in this paper is to advance the SQSDP method further.

Let us review existing works about the SQSDP method in more detail. The SQSDP method solves a sequence of quadratic SDP (QSDP) subproblems, which approximate the NSDP, so as to generate a sequence. It can be regarded as an extension of the sequential quadratic programming (SQP) method (e.g. see [4] for a survey) for the NLP. Strengths of the SQSDP method are that it admits both the global convergence and fast local convergence property, and moreover, it does not require a strictly feasible point as a starting point unlike interior point methods. We believe that the first SQSDP method was presented by Fares et al. [6] and the local convergence to a Karush–Kuhn–Tucker (KKT) point was established therein. The global convergence of the SQSDP was studied by Correa and Ramírez [5]. Afterwards, some variants of SQSDP methods were proposed, e.g., a successive linearization method by Kanzow et al. [20], a filter-type method by Zhu and Zhu [40], and a penalty-free method equipped with the second-order correction step by Zhao and Chen [37], who also proposed another SQSDP method that solves a feasible QSDP subproblem at every iteration in [39].

In this paper, we propose a new SQSDP method motivated from the stabilized SQP method for the NLP. The stabilized SQP method was initiated by Wright [28] to devise a superlinear convergent algorithm for solving degenerate NLPs. Since then, the stabilized SQP method have been developed by many researchers, e.g., [9–11, 14, 17, 18]. Our SQSDP method, referred to as a stabilized SQSDP method, can be distinguished from the other SQSDP methods in that the following three points are achieved altogether:
(i) **Consistent subproblem:** The proposed method organizes a consistent QSDP subproblem at each iteration, which always satisfies Slater’s constraint qualification (CQ) and also has a unique optimum.

(ii) **Inexact solution of QSDP:** We are allowed to truncate solving QSDP subproblems when certain criteria are met so as to ensure the global convergence.

(iii) **Stronger result on global convergence:** Without assuming boundedness of dual multiplier sequences or CQs at accumulation points, the global convergence is established successfully. In particular, this property is still valid even when NSDP (1) is infeasible.

We add some explanations regarding the above items. As for (i) and (ii), most the existing SQSDP methods are not well-defined in the sense that their subproblems are possibly infeasible. Moreover, they are impractical because exact optima of their subproblems are required to ensure the global convergence. Item (iii) indicates another advantage of the proposed stabilized SQSDP method. All the existing SQSDP methods assume CQs or boundedness of a dual sequence, as far as we investigated. Those analyses may break down when applied to degenerate NSDPs, e.g., a convex NSDP such that Slater’s CQ fails. In contrast, we establish the global convergence for the stabilized SQSDP method without such assumptions. More specifically, supposing the boundedness of a primal sequence generated by the stabilized SQSDP method, together with a certain controllable assumption, we prove that an accumulation point satisfies one of the following three conditions: a certain stationarity condition related to the feasibility, the AKKT conditions, and the TAKKT conditions. The AKKT and TAKKT conditions are the necessarily optimality conditions that were recently introduced by Andreani et al. [3]. The stabilized SQSDP method is expected to work effectively even when applied to degenerate or infeasible NSDPs.

The proposed stabilized SQSDP method solves a QSDP subproblem that is derived from a regularized quadratic approximation to the minimax Lagrange problem for NSDP (1) to produce a search direction together with Lagrange multiplier estimates. If a point satisfying certain criteria is found, we may terminate solving the QSDP and proceed to the next step. A step size along the obtained search direction is determined with a backtracking line-search using the ordinary augmented Lagrangian. The next dual iterates and each parameter are determined according to the VOMF procedures that we will propose for NSDP (1). The VOMF procedure determines an updating rule of parameters, which is rooted in the one presented by Gill and Robinson [11] for solving NLPs. For the sake of clarity, we first present a prototype algorithm based on the VOMF procedure for solving the NSDP, and also give generic assumptions to obtain the global convergence. We next present the overall stabilized SQSDP method and prove the global convergence using the convergence result of the prototype algorithm.
The remaining part of this paper is organized as follows. In Sect. 2, we introduce several notation and important concepts such as the KKT, AKKT, and TAKKT conditions. Section 3 provides a prototype algorithm for NSDP (1) and its global convergence property. In Sect. 4, we propose a stabilized SQSDP method for finding an AKKT or a TAKKT point of NSDP (1). Section 5 proves the global convergence property of the stabilized SQSDP method. In Sect. 6, we report some numerical results associated with the stabilized SQSDP method. Finally, we make some concluding remarks in Sect. 7.

Throughout this paper, we use the following notation. The set of natural numbers (positive integers) is represented by N. We denote by \( \mathbb{R}^p \) the \( p \)-dimensional real space. The set of real matrices with \( p \)-rows and \( q \)-columns is denoted by \( \mathbb{R}^{p \times q} \). For matrices \( A \) and \( B \) included in \( \mathbb{R}^{p \times q} \), \( \langle A, B \rangle := \text{tr}(A^T B) \), where \( \text{tr}(M) \) denotes the trace of a square matrix \( M \), and the superscript \( T \) means the transposition of a vector or a matrix. Note that if \( q = 1 \), then \( \langle \cdot, \cdot \rangle \) is the inner product of vectors in \( \mathbb{R}^p \). The identity matrix and the all-ones vector are represented by \( I \) and \( e \), respectively, where these dimensions are defined by the context. For a vector \( w \in \mathbb{R}^p \), \( [w]_i \) indicates the \( i \)-th element of \( w \), and \( \|w\| \) denotes the Euclidean norm of \( w \) defined by \( \|w\| := \sqrt{\langle w, w \rangle} \). For a matrix \( W \in \mathbb{R}^{p \times q} \), \( [W]_{ij} \) is the \((i, j)\)-th entry of \( W \), and \( \|W\|_F \) means the Frobenius norm of \( W \) defined by \( \|W\|_F := \sqrt{\langle W, W \rangle} \), and \( \|W\|_2 \) stands for the operator norm of \( W \) defined by \( \|W\|_2 := \max \{ \|WX\| : \|X\| = 1 \} \). For real numbers \( r_1, \ldots, r_d \in \mathbb{R} \), we define

\[
\text{diag}[r_1, \ldots, r_d] := \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & r_d \end{bmatrix}
\]

Let \( U \in \mathbb{S}^d \) be a matrix with an orthogonal diagonalization \( U = PDP^T \), where \( P \) is an orthogonal matrix and \( D \) is a diagonal matrix. We denote by \( \lambda_1^U(U), \ldots, \lambda_d^U(U) \) its eigenvalues satisfying \( D = \text{diag}[\lambda_1^U(U), \ldots, \lambda_d^U(U)] \). The minimum and the maximum eigenvalues of \( U \) are expressed as \( \lambda_{\min}(U) \) and \( \lambda_{\max}(U) \), respectively. Furthermore, we denote by \( [U]_+ \) the projection of \( U \) on \( \mathbb{S}^d_+ \), that is,

\[
[U]_+ := P\text{diag}[[\lambda_1^U(U)]_+, \ldots, [\lambda_d^U(U)]_+]P^T,
\]

where \( [r]_+ := \max\{r, 0\} \) for all \( r \in \mathbb{R} \). Given open sets \( P_1, P_2, P_3 \subset \mathbb{R}^p \), let \( \Phi \) be a mapping from \( P_1 \times P_2 \) to \( P_3 \). We define the Fréchet derivative of \( \Phi \) by \( \nabla \Phi \). Moreover, we denote by \( \nabla_Z \Phi \) the Fréchet derivative of \( \Phi \) with respect to a variable \( Z \in P_1 \). For a closed convex set \( S \), we write \( \Pi_S \) for the metric projector over \( S \). For a set \( T \), the cardinality of \( T \) is expressed as \( \text{card}(T) \). Finally, we will often use the following notation for the functions \( g \) and \( X \) in NSDP (1):
• The matrix $\nabla g(x) \in \mathbb{R}^{n \times m}$ means $\nabla g(x) := [\nabla g_1(x) \cdots \nabla g_m(x)]$;
• the matrix $A_i(x) \in \mathbb{S}^d$ indicates $A_i(x) := \frac{\partial}{\partial x_i} X(x)$ for $i = 1, \ldots, n$;
• the operator $A(x) : \mathbb{R}^n \to \mathbb{S}^d$ is defined by $A(x)u := [u_1 A_1(x) + \cdots + [u_n A_n(x)]$ for all $u \in \mathbb{R}^n$;
• the adjoint operator of $A(x)$ is represented by $A^*(x) : \mathbb{S}^d \to \mathbb{R}^n$, that is, $A^*(x)U = [\langle A_1(x), U \rangle \cdots \langle A_n(x), U \rangle]^\top$ for all $U \in \mathbb{S}^d$.

2 Preliminaries

In this section, we define some notation and terminologies.

2.1 The KKT conditions for NSDP (1)

First, we introduce the Karush–Kuhn–Tucker (KKT) conditions for (1). Define the Lagrange function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{S}^d \to \mathbb{R}$ as

$$L(v) := f(x) - \langle g(x), y \rangle - \langle X(x), Z \rangle,$$

where $v := (x, y, Z)$. Note that $y$ and $Z$ are Lagrange multipliers for $g(x) = 0$ and $X(x) \succeq 0$, respectively. The gradient of $L$ at $v \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{S}^d$ with respect to $x$ is given by

$$\nabla_x L(v) = \nabla f(x) - \nabla g(x)y - A^*(x)Z.$$

The KKT conditions for (1) are represented in terms of the function $L$ as follows:

Definition 1 We say that $v = (x, y, Z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{S}^d$ satisfies the KKT conditions if

$$\nabla_x L(v) = 0, \quad g(x) = 0, \quad \langle X(x), Z \rangle = 0, \quad X(x) \succeq O, \quad Z \succeq O.$$

In particular, we call the above point $x$ a KKT point and also call $(x, y, Z)$ a KKT triplet.

The KKT conditions are the first-order optimality conditions for NSDP (1) under the presence of some CQ. We give a well-known CQ for NSDP (1).

Definition 2 We say that a feasible point $x \in \mathbb{R}^n$ satisfies the Mangasarian–Fromovitz constraint qualification (MFCQ) if
If a point $x \in \mathbb{R}^n$, which is not necessarily a feasible point, satisfies conditions (2) and (3), then we say that $x$ satisfies the extended MFCQ.

A well-known fact is that, given a KKT point $x$, the corresponding Lagrange multipliers set $\{(y, Z) \in \mathbb{R}^m \times S^d : (x, y, Z) \text{ satisfies the KKT conditions}\}$ is non-empty and bounded under the MFCQ. The global convergence properties of many existing methods are shown under the MFCQ or relevant assumptions.

2.2 The AKKT and TAKKT conditions for NSDP (1)

This section provides the definitions of the AKKT and TAKKT conditions for NSDP (1). These concepts have been introduced by Andreani et al. [3]. In the following, we first give the AKKT conditions:

**Definition 3** We say that $x \in \mathbb{R}^n$ satisfies the AKKT conditions if $g(x) = 0$, $X(x) \succeq O$, and there exist sequences $\{x_\ell\} \subset \mathbb{R}^n$, $\{y_\ell\} \subset \mathbb{R}^m$, and $\{Z_\ell\} \subset S^d_+$ such that

$$\lim_{\ell \to \infty} x_\ell = x, \quad \lim_{\ell \to \infty} \nabla_x L(x_\ell, y_\ell, Z_\ell) = 0,$$

$$\lambda^U_j(X(x)) > 0 \implies \exists \ell_j \in \mathbb{N} \text{ s.t. } \lambda^U_j(Z_\ell) = 0 \quad \forall \ell \geq \ell_j,$$

where $j \in \{1, \ldots, d\}$, and $U$ and $U_\ell$ ($\ell \in \mathbb{N}$) are orthogonal matrices such that $U_\ell \to U$ ($\ell \to \infty$),

$$X(x) = U \text{diag}[\lambda^U_1(X(x)), \ldots, \lambda^U_d(X(x))] U^T,$$

$$Z_\ell = U_\ell \text{diag}[\lambda^U_1(Z_\ell), \ldots, \lambda^U_d(Z_\ell)] U_\ell^T.$$

Moreover, we define the TAKKT conditions:

**Definition 4** We say that $x \in \mathbb{R}^n$ satisfies the TAKKT conditions if $g(x) = 0$, $X(x) \succeq O$, and there exist sequences $\{x_\ell\} \subset \mathbb{R}^n$, $\{y_\ell\} \subset \mathbb{R}^m$, and $\{Z_\ell\} \subset S^d_+$ such that

$$\lim_{\ell \to \infty} x_\ell = x, \quad \lim_{\ell \to \infty} \nabla_x L(x_\ell, y_\ell, Z_\ell) = 0, \quad \lim_{\ell \to \infty} \langle X(x_\ell), Z_\ell \rangle = 0.$$
sequence. As well, a TAKKT point and a TAKKT sequence are defined as for the TAKKT conditions.

As was mentioned in the previous section, the KKT conditions are necessary optimality conditions under the MFCQ. In contrast, the AKKT and TAKKT conditions always hold true as necessary optimality conditions in the absence of CQs, as stated in the next theorem.

**Theorem 1** [3, Theorem 2, Theorem 5] Let \( x \) be a local optimum of (1). Then, \( x \) satisfies the AKKT and TAKKT conditions.

The KKT conditions imply both the AKKT and TAKKT conditions. On the other hand, the AKKT conditions were shown not to imply the TAKKT conditions in [3, Example 3]. Moreover, the TAKKT conditions do not imply the AKKT conditions [1, Example 3.1]. Therefore, the AKKT and TAKKT conditions are mutually independent.

The following theorem yields that if an AKKT or a TAKKT point satisfies the MFCQ, it is nothing but a KKT point.

**Theorem 2** [3, Theorem 7, Theorem 8] Suppose that \( x \) satisfies the AKKT (resp., TAKKT) conditions. If the MFCQ holds at \( x \), then \( x \) is a KKT point.

Strictly speaking, the above AKKT and the TAKKT conditions together with Theorems 1 and 2 differ from the original ones presented in [3], in that the equality constraint \( g(x) = 0 \) is not handled there. However, by splitting \( g(x) = 0 \) into \( g(x) \geq 0 \) and \( -g(x) \geq 0 \), we can derive Theorems 1 and 2 from [3, Theorem 7, Theorem 8].

**Remark 1** As shown in Theorem 2, the MFCQ is one of CQs under which AKKT and TAKKT points are KKT points. We remark here that the AKKT-(resp., TAKKT-) regularity is known as the minimal CQ under which AKKT (resp., TAKKT) points are KKT points, and these concepts are studied in [2, Section 7] in detail.

### 3 A prototype algorithm

Before presenting the new SQSDP method, we provide its prototype algorithm and then study its convergence properties under general assumptions. This algorithm makes use of the following function \( F : \mathbb{R}^n \to \mathbb{R} \) as a merit function for NSDP (1):

\[
F(x; y, Z, \sigma) := f(x) + \frac{1}{2\sigma} \| \sigma y - g(x) \|^2 + \frac{1}{2\sigma} \| \sigma Z - X(x) \|_F^2.
\]
where \( y \in \mathbb{R}^m \) and \( Z \in S^d \) are Lagrange multipliers for the equality and semidefinite constraints, respectively, and \( \sigma > 0 \) is a penalty parameter. By [3, Lemma 5], the function \( F \) is continuously differentiable on \( \mathbb{R}^n \), admitting the gradient

\[
\nabla F(x; y, Z, \sigma) = \nabla f(x) - \nabla g(x) \left\{ y - \frac{1}{\sigma} g(x) \right\} - A^* (x) \left[ Z - \frac{1}{\sigma} X(x) \right]_+.
\]

### 3.1 Description of the prototype algorithm

The prototype algorithm generates a sequence of primal iterates \( \{x_k\} \subset \mathbb{R}^n \) together with dual iterates \( \{(y_k, Z_k)\} \). For each \( k \), \( y_k \) and \( Z_k \) are estimates of Lagrange multiplier vectors and matrices corresponding to the constraints \( g(x) = 0 \) and \( X(x) \succeq O \), respectively. Moreover, \( \{(\tilde{y}_k, \tilde{Z}_k)\} \subset \mathbb{R}^m \times S^d \) is produced as a candidate for \( \{(y_k, Z_k)\} \).

For each iteration \( k \in \mathbb{N} \cup \{0\} \), the algorithm attempts to solve

\[
\min_{x \in \mathbb{R}^n} F(x; y_k, Z_k, \sigma_k),
\]

while tuning \((y_k, Z_k)\) together with the penalty parameter \( \sigma_k > 0 \) according to a certain procedure which will be clarified shortly (cf. Algorithm 2). Although there remain some unexplained parts, we first show the overall prototype of algorithm in Algorithm 1 for the sake of understanding. The meaning of each step therein is as below. Hereafter, we denote the current iteration by \( k \in \mathbb{N} \cup \{0\} \) and define \( v_k \in V \) and \( \bar{v}_k \in \bar{V} \) for every iteration \( k \) as follows:

\[
v_k := (x_k, y_k, Z_k) \quad \bar{v}_k := (x_k, \tilde{y}_k, \tilde{Z}_k).
\]

In Step 1, \( x_{k+1} \) is output by approximately solving problem (5), and set \((\tilde{y}_{k+1}, \tilde{Z}_{k+1})\), and define \( \tilde{v}_{k+1} := (x_{k+1}, \tilde{y}_{k+1}, \tilde{Z}_{k+1}) \). We call this phase Mini-F-Phase. For the SQSDP method proposed in the next section, we will clarify how \( \tilde{v}_{k+1} := (x_{k+1}, \tilde{y}_{k+1}, \tilde{Z}_{k+1}) \) is computed there. If \( x_k \) is already a stationary point of (5), we set the Lagrange multiplier estimates \( \tilde{y}_{k+1} \) and \( \tilde{Z}_{k+1} \) in the spirit of the augmented Lagrangian method. Specifically, we put \( \tilde{y}_{k+1} := y_k - \frac{1}{\sigma_k} g(x_{k+1}) \) and \( \tilde{Z}_{k+1} := [Z_k - \frac{1}{\sigma_k} X(x_{k+1})]_+ \) with \( x_{k+1} := x_k \). In Step 2, we update \((y_k, Z_k)\) by performing the procedure, which is called VOMF-ITERATES and is described shortly, with \( \tilde{v}_{k+1} := (x_{k+1}, \tilde{y}_{k+1}, \tilde{Z}_{k+1}) \) as input arguments. In Step 3, we decrease \( \sigma_k \) as necessary to strengthen the penalty for the constraint violation in the function \( F \). In what follows, we explain Steps 2 and 3 in more detail.
3.2 VOMF-ITERATES in Step 2

VOMF-ITERATES is provided \(v^{k+1}, y_k, Z_k, \phi_k, \psi_k, y_k, \) and \(\sigma_k\) as input arguments and generates a new Lagrange multiplier pair \((y_{k+1}, Z_{k+1})\) and a new parameter triplet \((\phi_{k+1}, \psi_{k+1}, y_{k+1})\). It is formally described as in Algorithm 2, wherein the following functions are utilized in order to measure the deviation of a given point \(v = (x, y, Z) \in \mathcal{V}\) from the set of KKT points:

\[
\Phi(v) := r_V(x) + \kappa r_O(v), \quad \Psi(v) := \kappa r_V(x) + r_O(v), \quad r(v) := r_V(x) + r_O(v),
\]

where \(\kappa \in (0, 1)\) is a prefixed weight parameter and the functions \(r_V\) and \(r_O\) are defined by

\[
\begin{align*}
r_V(x) & := \|g(x)\| + \left[\lambda_{\max}(-X(x))\right]_+, \\
r_O(v) & := \|\nabla_x L(v)\| + \|\{X(x), Z}\|.
\end{align*}
\]

Obviously, \(v\) satisfies the KKT conditions if and only if \(\Phi(v) = \Psi(v) = 0\). The procedure consists of the four steps, called the V-, O-, M-, and F-iterates, respectively. These names abbreviate the Violate, Optimality, Minimizing, and False-iterate that were originally given in [11] for nonlinear programming. In the V- and O-iterates, we check the KKT optimality of \(v^{k+1}\) with the value of \(\Phi\) and \(\Psi\). In fact, even if either one of the V- and O-iterates is dropped, we can keep the same convergence properties as those shown later on. Nevertheless, theses two iterates are devised for the sake of practicality. If \(\Phi(v^{k+1})\) and \(\Psi(v^{k+1})\) are not greater than \(\frac{1}{2} \phi_k\) and \(\frac{1}{2} \psi_k\), respectively, we regard \(v^{k+1}\) as a good approximation to KKT point, and set \(y_{k+1} := y_k, Z_{k+1} := Z_k\). We then decrease \(\phi_k\) or \(\psi_k\) to gain a more refined point in subsequent iterations. Otherwise, we proceed to the M-iterate which examines whether the following inequality holds or not:

\[
\|\nabla F(x_{k+1}; y_k, Z_k, \sigma_k)\| \leq y_k.
\]

If it holds true, we regard \(x_{k+1}\) as a good approximate solution of (5). The following updating formulae in the M-iterate are motivated by this fact:

\[
y_{k+1} := \Pi_C(y_k - \frac{1}{\sigma_k} g(x_{k+1})), \quad Z_{k+1} := \Pi_D([Z_k - \frac{1}{\sigma_k} X(x_{k+1})]_+),
\]

where \(C \subset \mathbb{R}^m\) and \(D \subset S^d\) are closed bounded sets defined in Algorithm 2, and further decrease \(\gamma_k\) so as to find a solution satisfying \(\nabla F(x_{k+1}; y_k, Z_k, \sigma_k) = 0\) more accurately through subsequent iterations.

When the if-statement for the M-iterate is false, that is, (7) does not hold, then we proceed to the F-iterate, therein just incrementing \(k\) by one without any other updates. The F-iterate is aimed at gaining a refined point such that one of the criteria for the V-, O-, and M-iterates is satisfied.

Roughly speaking, the V- and O-iterates play a part of generating a TAKKT sequence, while the M-iterate does a part of generating an AKKT sequence.
3.3 Update of the parameter \(\sigma_k\) in Step 3

If the if-statement for the M-iterate is true, that is, condition (7) is fulfilled, then the statements of the V- and O-iterates are false. In other words, the current point does not sufficiently satisfy the feasibility we expect. Then we decrease \(\sigma_k\) to strengthen the penalty for the constraint violation in the function \(F\). Specifically, we update \(\sigma_k\) as follows:

\[
\sigma_{k+1} := \begin{cases} 
\min \left\{ \frac{1}{2} \sigma_k, r(v_{k+1})^{\frac{3}{2}} \right\} & \text{if (7) is satisfied,} \\
\sigma_k & \text{otherwise.}
\end{cases}
\]  

Here, the term \(r(v_{k+1})^{\frac{3}{2}}\) is employed for gaining fast local convergence in the above, other update formulae can be adopted as long as the resulting \(\{\sigma_k\}\) is a monotonically decreasing sequence of positive numbers.

**Algorithm 1** Prototype algorithm

**Require:** Choose \(v_0 := (x_0, y_0, Z_0)\) such that \(Z_0 \succeq O\). Set

\[ k := 0, \quad \overline{y}_0 := y_0, \quad \overline{Z}_0 := Z_0, \quad \phi_0 > 0, \quad \psi_0 > 0, \quad \gamma_0 > 0, \quad \sigma_0 > 0. \]

Set parameters for Mini-\(F\)-Phase and VOMF-ITERATES (see Lines 2 and 7).

1: repeat
2: \[ \text{if } \| \nabla F(x_k; y_k, Z_k, \sigma_k) \| = 0 \text{ then} \quad \triangleright \text{Step 1 (Mini-}F\text{-Phase)} \]
3: \[ x_{k+1} := x_k, \quad \overline{y}_{k+1} := y_k - \frac{1}{\sigma_k} g(x_{k+1}), \quad \overline{Z}_{k+1} := [Z_k - \frac{1}{\sigma_k} X(x_{k+1})]_+. \]
4: \[ \text{else} \]
5: \[ \text{Compute } x_{k+1} \text{ by approximately solving (5) and set } (\overline{y}_{k+1}, \overline{Z}_{k+1}), \]
\[ \text{e.g., Lines 5–7 in Algorithm 4. Set } \overline{v}_{k+1} := (x_{k+1}, \overline{y}_{k+1}, \overline{Z}_{k+1}). \]
6: \[ \text{end if} \]
7: \[ \text{Set} \quad \triangleright \text{Step 2 (VOMF-ITERATES)} \]
\[ (y_{k+1}, Z_{k+1}, \phi_{k+1}, \psi_{k+1}, \gamma_{k+1}) \]
\[ := \text{VOMF-ITERATES}(\overline{v}_{k+1}, y_k, Z_k, \phi_k, \psi_k, \gamma_k, \sigma_k). \]
8: \[ \text{Update } \sigma_k \text{ by} \quad \triangleright \text{Step 3} \]
\[ \sigma_{k+1} := \begin{cases} 
\min \left\{ \frac{1}{2} \sigma_k, r(v_{k+1})^{\frac{3}{2}} \right\} & \text{if } \| \nabla F(x_{k+1}; y_k, Z_k, \sigma_k) \| \leq \gamma_k, \\
\sigma_k & \text{otherwise.}
\end{cases} \]
9: \[ \text{Set } k := k + 1. \quad \triangleright \text{Step 4} \]
10: until \(v_k := (x_k, y_k, Z_k)\) meets a suitable criterion.
A stabilized SQP method for nonlinear semidefinite programs

3.4 Convergence analysis of Algorithm 1

We prove the global convergence property of Algorithm 1. In the subsequent arguments, we use three sets \( \mathcal{I}, \mathcal{J}, \) and \( \mathcal{K} \) defined by

\[
\mathcal{I} := \{ k : y_k, Z_k, \phi_k, \psi_k, \gamma_k \text{ are updated by the V- or O-iterate} \}, \\
\mathcal{J} := \{ k : y_k, Z_k, \phi_k, \psi_k, \gamma_k \text{ are updated by the M-iterate} \}, \\
\mathcal{K} := \{ k : y_k, Z_k, \phi_k, \psi_k, \gamma_k \text{ are updated by the F-iterate} \}.
\]

Notice that \( \mathcal{I}, \mathcal{J}, \) and \( \mathcal{K} \) are mutually disjoint. Moreover, we make the following two sets of assumptions:

---

**Algorithm 2** Procedure for updating \((y_k, Z_k, \phi_k, \psi_k, \gamma_k)\)

1. procedure VOMF-ITERATES(\(\overline{v}_{k+1}, y_k, Z_k, \phi_k, \psi_k, \gamma_k, \sigma_k\))
2. if \(\Phi(\overline{v}_{k+1}) \leq \frac{1}{2} \phi_k\) then
3. \hspace{1em} Set \(\triangleright V\text{-iterate}\)
4. \hspace{1em} \(y_{k+1} := \overline{v}_{k+1}, \quad Z_{k+1} := \overline{Z}_{k+1}, \quad \phi_{k+1} := \frac{1}{2} \phi_k, \quad \psi_{k+1} := \psi_k, \quad \gamma_{k+1} := \gamma_k\).
5. else if \(\Psi(\overline{v}_{k+1}) \leq \frac{1}{2} \psi_k\) then
6. \hspace{1em} Set \(\triangleright O\text{-iterate}\)
7. \hspace{1em} \(y_{k+1} := \overline{v}_{k+1}, \quad Z_{k+1} := \overline{Z}_{k+1}, \quad \phi_{k+1} := \phi_k, \quad \psi_{k+1} := \frac{1}{2} \psi_k, \quad \gamma_{k+1} := \gamma_k\),
8. else if \(\|\nabla F(x_{k+1}; y_k, Z_k, \sigma_k)\| \leq \gamma_k\) then
9. \hspace{1em} Set \(\triangleright M\text{-iterate}\)
10. \hspace{1em} \(y_{k+1} := \Pi_C (y_{k - \frac{1}{\sigma_k} g(x_{k+1})) \), \(Z_{k+1} := \Pi_D ([Z_k - \frac{1}{\sigma_k} X(x_{k+1})]_+)\), \(\phi_{k+1} := \phi_k, \quad \psi_{k+1} := \psi_k, \quad \gamma_{k+1} := \frac{1}{2} \gamma_k\),
11. where the set \(C\) and \(D\) are defined as follows:
12. \(C := \{ y \in \mathbb{R}^m : -y_{\text{max}} \leq y \leq y_{\text{max}} \}, \quad D := \{ Z \in \mathbb{S}^d : O \preceq Z \preceq z_{\text{max}} I \}\).
13. else
14. \hspace{1em} Set \(\triangleright F\text{-iterate}\)
15. \(y_{k+1} := y_k, \quad Z_{k+1} := Z_k, \quad \phi_{k+1} := \phi_k, \quad \psi_{k+1} := \psi_k, \quad \gamma_{k+1} := \gamma_k\).
16. end if
17. return \((y_{k+1}, Z_{k+1}, \phi_{k+1}, \psi_{k+1}, \gamma_{k+1})\)
18. end procedure
Assumption 1

(A1) The functions $f$, $g$, and $X$ are twice continuously differentiable;
(A2) the sequence $\{x_k\}$ is bounded.

Assumption 2 There never occurs a situation with $\text{card}(\mathcal{I}) < \infty$, $\text{card}(\mathcal{J}) < \infty$, and $\text{card}(\mathcal{K}) = \infty$.

Assumption 1 is standard and can be found in many literatures. It is also assumed in the convergence analysis for a stabilized SQSDP method, which is presented in the next section, whereas Assumption 2 will be verified there. In addition to the above, we also implicitly assume that Algorithm 1 generates infinitely many iteration points.

In the following, we give properties associated with $\{\phi_k\}$, $\{\psi_k\}$, $\{\gamma_k\}$, $\{\sigma_k\}$, and $\{Z_k\}$.

Lemma 1 Suppose that Assumption 2 holds. Then, we have the following properties:

(i) If $\text{card}(\mathcal{I}) = \infty$, then $\phi_k \to 0$ or $\psi_k \to 0$ as $k \to \infty$;
(ii) if $\text{card}(\mathcal{I}) < \infty$, then $\{Z_k\}$ is bounded;
(iii) if $\text{card}(\mathcal{I}) < \infty$ and $\text{card}(\mathcal{J}) = \infty$, then $\sigma_k \to 0$ and $\gamma_k \to 0$ as $k \to \infty$.

Proof We show item (i). Note that $\{\phi_k\}$ and $\{\psi_k\}$ are non-increasing, and $\phi_{k+1} = \frac{1}{2}\phi_k$ or $\psi_{k+1} = \frac{1}{2}\psi_k$ for $k \in \mathcal{I}$. Since $\text{card}(\mathcal{I}) = \infty$, it is clear that $\phi_k \to 0$ or $\psi_k \to 0$ as $k \to \infty$.

We next show item (ii). Considering $\text{card}(\mathcal{I}) < \infty$ yields that

$$\exists n_0 \in \mathbb{N} \quad \text{s.t.} \quad k \in \mathcal{J} \cup \mathcal{K} \quad \forall k \geq n_0. \quad (10)$$

Let us define $\tilde{z} := \max\{\lambda_{\max}(Z_0), \lambda_{\max}(Z_1), \ldots, \lambda_{\max}(Z_{n_0}), \tilde{z}_{\max}\}$ and $\tilde{D} := \{Z \in \mathbb{S}^d : O \leq Z \leq \tilde{z}I\}$. We prove $\{Z_k\} \subset \tilde{D}$ by mathematical induction. If $k = 0$, it is clear that $O \leq Z_0 \leq \lambda_{\max}(Z_0)I$, that is, $Z_0 \in \tilde{D}$. Let $k$ be a nonnegative integer, and let $Z_k \in \tilde{D}$. Now, we consider two cases: $k \geq n_0$, $k < n_0$. In the first case, we have from (10) that if $k \in \mathcal{J}$, then $Z_{k+1} = \Pi_D([Z_k - \frac{1}{\sigma_k}\tilde{z}]_+) \in D \subset \tilde{D}$; if $k \in \mathcal{K}$, then $Z_{k+1} = Z_k \in \tilde{D}$. As a result, $Z_{k+1} \in \tilde{D}$. On the other hand, we consider the case where $k < n_0$. Noting $1 \leq k + 1 \leq n_0$ derives that $O \leq Z_{k+1} \leq \max\{\lambda_{\max}(Z_1), \lambda_{\max}(Z_2), \ldots, \lambda_{\max}(Z_{n_0})\}I$, which implies $Z_{k+1} \in \tilde{D}$. Therefore, we obtain that $\{Z_k\} \subset \tilde{D}$. The fact and the boundedness of $\tilde{D}$ yield the desired result.

Finally, we prove item (iii). Note that $\{\gamma_k\}$ and $\{\sigma_k\}$ are non-increasing. It is easily seen that $\gamma_{k+1} = \frac{1}{2}\gamma_k$ for $k \in \mathcal{I}$. Moreover, we have from (8) that $\sigma_{k+1} = \min\{\frac{1}{2}\sigma_k, r(v_{k+1})\tilde{z}\} \leq \frac{1}{2}\sigma_k$ for $k \in \mathcal{J}$. These facts and $\text{card}(\mathcal{J}) = \infty$ imply $\gamma_k \to 0$ and $\sigma_k \to 0$ as $k \to \infty$. The proof is complete. \(\square\)
With the help of the above lemma, we derive the following convergence theorem of Algorithm 1.

**Theorem 3** Suppose that Assumptions 1 and 2 hold. Let \( \{x_k, y_k, Z_k\} \) be a sequence generated by Algorithm 1 and let \( \{\tilde{y}_k\} \) and \( \{\tilde{Z}_k\} \) be defined by \( \tilde{y}_k := y_{k-1} - \frac{1}{\sigma_{k-1}} g(x_k) \), and \( \tilde{Z}_k := [Z_{k-1} - \frac{1}{\sigma_{k-1}} X(x_k)]_+ \), respectively. Then, any accumulation point of \( \{x_k\} \), say \( x^* \), satisfies at least one of the following statements:

1. \( x^* \) is a TAKKT point of (1) and some subsequence of \( \{(x_k, y_k, Z_k)\} \) is a TAKKT sequence corresponding to \( x^* \);
2. \( x^* \) is an AKKT point of (1) and some subsequence of \( \{(x_k, \tilde{y}_k, \tilde{Z}_k)\} \) is an AKKT sequence corresponding to \( x^* \);
3. \( x^* \) is an infeasible point of (1), but a stationary point of the following optimization problem:

\[
\min_{x \in \mathbb{R}^n} h(x) := \frac{1}{2} \|g(x)\|^2 + \frac{1}{2} \|X(x)_+\|^2,
\]

that is to say, \( \nabla h(x^*) = 0 \).

**Proof** We consider the two cases: (a) \( \text{card}(\mathcal{I}) = \infty \); (b) \( \text{card}(\mathcal{I}) < \infty \).

**Case (a):** Let \( \mathcal{P} := \{k \in \mathbb{N} : k - 1 \in \mathcal{I}\} \). Note that \( \text{card}(\mathcal{P}) = \infty \) according to \( \text{card}(\mathcal{I}) = \infty \). Recall that \( v_k := (x_k, y_k, Z_k) \) and \( \tilde{v}_k := (x_k, \tilde{y}_k, \tilde{Z}_k) \). Assumption 1 (A2) implies that \( \{x_k\}_{\mathcal{P}} \) has at least one accumulation point, say \( x^* \). Then, there exists \( \mathcal{L} \subset \mathcal{P} \) such that \( x_k \to x^* \) as \( \mathcal{L} \ni k \to \infty \). It follows from Lemma 1 (i) that \( \phi_k \to 0 \) or \( \psi_k \to 0 \) as \( \mathcal{L} \ni k \to \infty \). Since \( \Phi(v_k) = \Phi(\tilde{v}_k) \leq \frac{1}{2} \phi_{k-1} = \phi_k \) or \( \Psi(v_k) = \Psi(\tilde{v}_k) \leq \frac{1}{2} \psi_{k-1} = \psi_k \) for \( k \in \mathcal{L} \subset \mathcal{P} \), it is clear that \( \Phi(v_k) \to 0 \) or \( \Psi(v_k) \to 0 \) as \( \mathcal{L} \ni k \to \infty \), that is,

\[
\lim_{\mathcal{L} \ni k \to \infty} \nabla J(x_k, y_k, Z_k) = 0, \quad \lim_{\mathcal{L} \ni k \to \infty} \langle Z_k, X(x_k) \rangle = 0,
\]

\[
g(x^*) = \lim_{\mathcal{L} \ni k \to \infty} g(x_k) = 0, \quad [\lambda_{\max}(-X(x^*))]_+ = \lim_{\mathcal{L} \ni k \to \infty} [\lambda_{\max}(-X(x_k))]_+ = 0.
\]

These results and \( \{Z_k\} \subset S_+^d \) derive that \( \{(x_k, y_k, Z_k)\}_{\mathcal{L}} \) is a TAKKT sequence corresponding to \( x^* \). Hence, in this case, we have situation (i).

**Case (b):** We show that, in this case, there occurs (ii) or (iii). Let \( \mathcal{Q} := \{k \in \mathbb{N} : k - 1 \in \mathcal{J}\} \). Assumption 2 implies that \( \text{card}(\mathcal{J}) = \infty \) must hold. Notice that \( \text{card}(\mathcal{Q}) = \infty \) by \( \text{card}(\mathcal{J}) = \infty \). Moreover, Assumption 1 (A2) ensures that \( \{x_k\}_{\mathcal{Q}} \) has at least one accumulation point, say \( x^* \) again. Then, there exists \( \mathcal{M} \subset \mathcal{Q} \) such that \( x_k \to x^* \) as \( \mathcal{M} \ni k \to \infty \). We have from Lemma 1 (iii) that \( \sigma_{k-1} \to 0 \) and \( y_{k-1} \to 0 \) as \( \mathcal{M} \ni k \to \infty \). Note that \( \tilde{y}_k = y_{k-1} - \frac{1}{\sigma_{k-1}} g(x_k), \tilde{Z}_k = [Z_{k-1} - \frac{1}{\sigma_{k-1}} X(x_k)]_+ \), and

\[
\nabla J(x_k, \tilde{y}_k, \tilde{Z}_k) = \nabla f(x_k) - \nabla g(x_k) \tilde{y}_k - A^*(x_k) \tilde{Z}_k = \nabla F(x_k, y_{k-1}, Z_{k-1}, \sigma_{k-1}) \text{ and } \|\nabla F(x_k, y_{k-1}, Z_{k-1}, \sigma_{k-1})\| \leq \gamma_{k-1} \text{ for } k \in \mathcal{M} \subset \mathcal{Q}.
\]
These facts yield that
\[
\lim_{M \ni k \to \infty} \nabla \ell(x_k, \tilde{y}_k, \tilde{Z}_k) = 0, \tag{11}
\]

\[
\nabla h(x^*) = \lim_{M \ni k \to \infty} \sigma_{k-1} \nabla F(x_k, y_{k-1}, Z_{k-1}, \sigma_{k-1}) = 0. \tag{12}
\]

Furthermore, since the boundedness of \( \{Z_k\} \) is ensured by Lemma 1 (ii), we obtain

\[
\lim_{M \ni k \to \infty} \left\{ \sigma_{k-1}Z_{k-1} - X(x_k) \right\} = -X(x^*). \tag{13}
\]

Denote \( X^* := X(x^*), X_k := X(x_k), \) and \( W_k := \sigma_{k-1}Z_{k-1} - X(x_k). \) Take a diagonal decomposition of \( X^* \), i.e., \( X^* = V^* \text{diag}[^1 \lambda_1 V_1 (X^*), \ldots, ^d \lambda_d V_d (X^*)](V^*)^T \), where \( V^* \) is an orthogonal matrix. By (13), there exists \( N \subset M \) such that

\[
^1 \lambda_1 V_1 (W_k) \to -^1 \lambda_1 V_1 (X^*), \ldots, ^d \lambda_d V_d (W_k) \to -^d \lambda_d V_d (X^*) \quad (N \ni k \to \infty),
\]

where \( V_k (k \in L) \) are orthogonal matrices such that \( V_k \to V^* (N \ni k \to \infty) \) and \( W_k = V_k \text{diag}[^1 \lambda_1 V_1 (W_k), \ldots, ^d \lambda_d V_d (W_k)]V_k^T \). Let \( j \in \{1, \ldots, d\} \) be an arbitrary integer. Now, assume that \( ^j \lambda_j V_j (X^*) > 0 \). Combining this assumption and (14) gives

\[
\lim_{N \ni k \to \infty} ^j \lambda_j V_j (W_k) = -^j \lambda_j V_j (X^*) < 0,
\]

and hence there exists \( k_j \in N \) such that \( ^j \lambda_j V_j (W_k) < 0 \) for all \( k \in \{k \in N : k \geq k_j\} \). This fact and \( \tilde{Z}_k = [Z_{k-1} - \frac{1}{\sigma_{k-1}}X_k]_+ = [\frac{1}{\sigma_{k-1}}W_k]_+ \) yield \( ^j \lambda_j V_j (\tilde{Z}_k) = \frac{1}{\sigma_{k-1}}[^j \lambda_j V_j (W_k)]_+ = 0 \) for all \( k \in \{k \in N : k \geq k_j\} \). As a result, we obtain

\[
^j \lambda_j V_j (X^*) > 0 \implies \exists k_j \in N \text{ s.t. } ^j \lambda_j V_j (\tilde{Z}_k) = 0 \quad \forall k \in \{k \in N : k \geq k_j\}. \tag{15}
\]

If \( x^* \) is feasible to NSDP (1), it follows from (11) and (15) that \( \{(x_k, \tilde{y}_k, \tilde{Z}_k)\}_N \) is an AKKT sequence corresponding to \( x^* \), that is, situation (ii) holds. Even if \( x^* \) is not feasible, situation (ii) holds from (12).

Therefore, we conclude that if Case (a) holds, then situation (i) occurs, otherwise, namely, if Case (b) holds, situation (ii) or (iii) is satisfied. \( \square \)

**Remark 2** Theorem 3 indicates that the accumulation point \( x^* \) satisfies the TAKKT, AKKT, or stationary condition of \( \min \{h(x) : x \in \mathbb{R}^n\} \). Note that \( x^* \) does not necessarily satisfy all of these conditions simultaneously. Depending on the circumstances, \( x^* \) satisfies either one of or both the TAKKT and AKKT conditions.

Finally, Theorem 3 entails the following one about global convergence to a KKT point under the presence of the extended MFCQ.

**Theorem 4** Suppose that Assumptions 1 and 2 hold. If any accumulation point of \( \{x_k\} \), say \( x^* \), satisfies the extended MFCQ, then \( x^* \) is a KKT point.

**Proof** Note that AKKT and TAKKT points are KKT points if they satisfy the extended MFCQ. Hence, to prove the desired claim, it is sufficient to show that \( x^* \) is an AKKT or a TAKKT point. We prove this assertion by contradiction, that is,
suppose that $x^*$ is neither an AKKT nor a TAKKT point. By Theorem 3, we see that $x^*$ is infeasible and $\nabla h(x^*) = 0$, yielding
\[
\frac{1}{2}\|g(x^*)\|^2 + \frac{1}{2}\|[-X(x^*)]_+\|_F^2 = h(x^*) \neq 0, \tag{16}
\]
\[
\nabla g(x^*)g(x^*) - A^*(x^*)[-X(x^*)]_+ = \nabla h(x^*) = 0. \tag{17}
\]
Since conditions (2) and (3) are satisfied at $x^*$, the matrix $\nabla g(x^*)$ is of full rank, and there exists $d \in \mathbb{R}^n$ such that $\nabla g(x^*)^Td = 0$ and $X(x^*) + A(x^*)d > O$. It then follows from $[-X(x^*)]_+, O$ that $0 \leq \langle [-X(x^*)]_+, X(x^*) + A(x^*)d \rangle$, which implies
\[
0 \geq \langle [-X(x^*)]_+, -X(x^*) \rangle - \langle A^*(x^*)[-X(x^*)]_+, d \rangle
\]
\[
= \|[-X(x^*)]_+\|_F^2 - \langle \nabla g(x^*)g(x^*), d \rangle
\]
\[
= \|[-X(x^*)]_+\|_F^2 - \langle g(x^*), \nabla g(x^*)^Td \rangle, \tag{18}
\]
where the first equality is obtained by (17). Combining (18) with $\nabla g(x^*)^Td = 0$ yields that $\|[-X(x^*)]_+\|_F = 0$. Then, we have from (17) that $\nabla g(x^*)g(x^*) = 0$. Since $\nabla g(x^*)$ is of full rank, we get $g(x^*) = 0$. Therefore, $h(x^*) = 0$. However, this contrasts (16). The proof is complete. 

\section{An inexact and stabilized SQSDP method}

In this section, we propose a new SQSDP method for NSDP (1) by integrating a sequential quadratic optimization technique into the prototype algorithm, i.e., Algorithm 1. Specifically, we clarify how Mini-F-Phase of the algorithm is performed. At every Mini-F-Phase, a convex quadratic SDP (QSDP) is solved to compute a search direction $p$, and then a step size $\alpha > 0$ is determined along $p$ by means of a line-search technique equipped with the function $F$ as a merit function. We then update $x$ as $x + \alpha p$.

Differently from the existing SQSDP methods, the proposed one may truncate a process of solving the convex QSDP subproblem if certain conditions are satisfied. Another remarkable point is that the QSDP is strictly feasible (Slater’s CQ is fulfilled) and thus the KKT conditions always hold at its optimum. Hereafter, a current iterate is often represented as $(x, y, Z)$ simply.

\subsection{A stabilized quadratic SDP subproblem}

This section specifies how Mini-F-Phase is conducted. We first derive the quadratic SDP subproblem to be solved at $(x, y, Z)$ so as to minimize the merit function $F$. As shown in the subsequent argument, an optimal solution of this problem is indeed the descent direction of the merit function $F$ under some reasonable assumptions.

We shall start with the Lagrange minimax problem regarding NSDP (1):
minimize maximize \( L(x, y, Z) \).

For solving this problem, similarly to Wright [28], we iteratively solve the following problem which approximates the above problem at \((x, y, Z)\):

\[
\begin{aligned}
\min_{\xi \in \mathbb{R}^n} \max_{(\zeta, \Sigma) \in \mathbb{R}^{n \times n}} & \langle \nabla f(x), \xi \rangle + \frac{1}{2} \langle H \xi, \xi \rangle - \langle \zeta, g(x) + \nabla g(x) \rangle - \langle \zeta, x \rangle - \langle \zeta, \Sigma \rangle \\
& - \langle \Sigma, X(x) + A(x) \zeta \rangle - \frac{\sigma}{2} \| \zeta - y \|^2 - \frac{\sigma}{2} \| \Sigma - Z \|^2_F,
\end{aligned}
\tag{19}
\]

where \( \sigma > 0 \) is a parameter shared with the function \( F \) and \( H \in \mathbb{R}^{n \times n} \) can be set to any matrix as long as Assumption 3, which will appear later, is satisfied for the global convergence. Although an arbitrary positive definite matrix is theoretically accepted as \( H \), the Hessian \( \nabla^2_{xx} L(x, y, Z) \) or its approximation is preferable for the sake of rapid convergence. Notice that the maximization part in (19) is a strongly concave problem by virtue of the last two quadratic terms, thus it always has a unique maximum \((\zeta, \Sigma)\) for a prefixed \( \xi \). Notice also that an optimum \( \xi \) of (19) together with the corresponding maximizer \((\zeta, \Sigma)\) solves the following system:

\[
\begin{aligned}
\nabla f(x) + H \xi - \nabla g(x) \zeta - A^*(x) \Sigma &= 0, \\
g(x) + \nabla g(x)^T \xi + \sigma(\zeta - y) &= 0,
\end{aligned}
\tag{20, 21}
\]

\[
S^d_+ \ni X(x) + A(x) \xi + \sigma(\Sigma - Z) \perp \Sigma \in S^d_+,
\tag{22}
\]

where \( \perp \) stands for the perpendicularity in \( S^d_+ \). If \( H \) is positive definite, then (20)–(22) are necessary and sufficient conditions under which \((\xi, \zeta, \Sigma)\) is a unique optimal solution of (19).

A triplet \((\xi, \zeta, \Sigma)\) satisfying (20)–(22) is actually obtained via the KKT conditions of the following quadratic semidefinite program (QSDP):

\[
\begin{aligned}
\min_{(\xi, \zeta, \Sigma) \in \mathcal{W}} & \langle \nabla f(x), \xi \rangle + \frac{1}{2} \langle H \xi, \xi \rangle + \frac{\sigma}{2} \| \zeta \|^2 + \frac{\sigma}{2} \| \Sigma \|^2_F \\
\text{subject to} & \quad g(x) + \nabla g(x)^T \xi + \sigma(\zeta - y) = 0, \\
& \quad X(x) + A(x) \xi + \sigma(\Sigma - Z) \succeq O,
\end{aligned}
\tag{23}
\]

which is further transformed to the following problem by eliminating the variable \( \zeta \) via the relation \( \zeta = y - \frac{1}{\sigma} \langle g(x) + \nabla g(x)^T \xi \rangle \):

\[
\begin{aligned}
\min_{(\xi, \Sigma) \in \mathcal{W}} & \langle \nabla f(x) - \nabla g(x) s, \xi \rangle + \frac{1}{2} \langle M \xi, \xi \rangle + \frac{\sigma}{2} \| \Sigma \|^2_F \\
\text{subject to} & \quad A(x) \xi + \sigma(\Sigma - T) \succeq O,
\end{aligned}
\tag{24}
\]

where

\[
\mathcal{W} := \mathbb{R}^n \times S^d, \quad M := H + \frac{1}{\sigma} \nabla g(x) \nabla g(x)^T, \quad s := y - \frac{1}{\sigma} g(x), \quad T := Z - \frac{1}{\sigma} X(x).
\]
We employ problem (24) as a subproblem of the proposed method to generate a
search direction, although (23) is equivalent to (24) in the sense that both global
optimal values are equal if exist. This is because (24) is expected to be easier to
solve than (23) since the number of variables in (24) is less than that of (23).

Note that (24) is always strictly feasible, that is, Slater’s CQ is satisfied, which
can be ensured by substituting \( (\xi, \Sigma) = (0, I + T) \). Furthermore, if the objective
function of (24) is strongly convex, that is, the matrix \( M \) is positive definite,
problem (24) has a unique optimum. This fact is formally stated in the following
proposition.

**Proposition 1** Suppose that \( M = H + \frac{1}{\sigma} \nabla g(x) \nabla g(x)^T > O \). Then, problem (24) has a
unique optimal solution.

Henceforth, we assume that \( M > O \) and consider solving QSDP (24) at every
iteration. We often call this problem a stabilized QSDP subproblem. It is derived
from the existing subproblem proposed in [28] which was referred to as the stabilized
subproblem therein because of its calming effect on a sequence of Lagrange multi-
plier estimates.

The following proposition connects the merit function \( F \) to QSDP (24).

**Proposition 2** Assume that \( M > O \). Then, a unique solution of QSDP (24), say
\( (\xi^*, \Sigma^*) \in \mathcal{W} \), satisfies \( \langle \nabla F(x_0, Z, \sigma), \xi^* \rangle \leq -\langle M \xi^*, \xi^* \rangle - \sigma \| \Sigma^* - [T]_+ \|_F^2 \). Moreover,
\( \nabla F(x_0, Z, \sigma) = 0 \) if and only if \( (\xi^*, \Sigma^*) = (0, [T]_+) \).

**Proof** Note that Proposition 1 and \( M > O \) imply that problem (24) has a unique optimal solution \( (\xi^*, \Sigma^*) \). First, we prove that
\( \langle \nabla F(x_0, Z, \sigma), \xi^* \rangle \leq -\langle M \xi^*, \xi^* \rangle - \sigma \| \Sigma^* - [T]_+ \|_F^2 \). It follows from (28), (29), and
(4) that
\[
\langle \nabla F(x_0, Z, \sigma), \xi^* \rangle = \langle -M \xi^*, \xi^* \rangle + \langle \Sigma^*, A(x) \xi^* \rangle = \langle [T]_+, A(x) \xi^* \rangle - \langle [T]_+, A(x) \xi^* \rangle.
\]

By (29) and (32), we easily see that \( \langle \Sigma^*, A(x) \xi^* + \sigma (\Sigma^* - T) \rangle = 0 \), that is,
\( \langle \Sigma^*, A(x) \xi^* \rangle = \sigma \langle \Sigma^* - T, \Sigma^* \rangle \). Moreover, we have from (30) and \( [T]_+ \geq O \) that
\( 0 \leq \langle [T]_+, A(x) \xi^* + \sigma (\Sigma^* - T) \rangle = \langle [T]_+, A(x) \xi^* \rangle + \sigma \langle [T]_+, T - \Sigma^* \rangle \), and hence
\[
\langle \Sigma^*, A(x) \xi^* \rangle = \langle [T]_+, A(x) \xi^* \rangle = \sigma \langle \Sigma^* - [T]_+, T - \Sigma^* \rangle
= \sigma \langle \Sigma^* - [T]_+, T - [T]_+ \rangle - \sigma \| \Sigma^* - [T]_+ \|_F^2.
\]

Now note that \( \Sigma^* \geq O \) by (29) and (31). Since \( \cdot \mid_+ \) is the projection onto
\( S_d^{++} \), we obtain \( \langle \Sigma^* - [T]_+, T - [T]_+ \rangle \leq 0 \). This fact and (26) yield that
\( \langle \Sigma^*, A(x) \xi^* \rangle - \langle [T]_+, A(x) \xi^* \rangle \leq -\sigma \| \Sigma^* - [T]_+ \|_F^2 \). It then follows from (25) that
\[
\langle \nabla F(x_0, Z, \sigma), \xi^* \rangle \leq -\langle M \xi^*, \xi^* \rangle - \sigma \| \Sigma^* - [T]_+ \|_F^2.
\]
Now, we assume that $\nabla F(x; y, Z, \sigma) = 0$. We have from (27) that $\xi^* = 0$ and $\Sigma^* = [T]_+$. This means that $(0, [T]_+)$ is a unique solution of problem (24).

Conversely, we suppose that $(0, [T]_+)$ is a unique solution, i.e., $(\xi^*, \Sigma^*) = (0, [T]_+)$. By (29), we obtain $\Lambda^* = \Sigma^* = [T]_+$. It then follows from (28) and (4) that $0 = M\xi^* + \nabla f(x) - \nabla g(x)s - A^*(x)\Lambda^* = \nabla f(x) - \nabla g(x)s - A^*(x)[T]_+ = \nabla F(x; y, Z, \sigma)$. The assertion is proven. \qed

**Remark 3** Proposition 2 is a novelty of this paper and is not seen in the existing researches related to the stabilized SQP-type methods. This result motivates us to solve QSDP (24) in Mini-$F$-Phase.

### 4.2 An inexact solution of the stabilized QSDP subproblem

Although it is often impractical or impossible to solve a QSDP subproblem exactly, the existing SQSDP methods such as [5–7, 12, 37, 40] establish global convergence by using exact optima of QSDP subproblems, as far as we have investigated. In this section, we study when we may truncate a process of solving QSDP (24).

Let $(\xi^*, \Sigma^*) \in \mathcal{W}$ be the unique optimum of (24). Since (24) satisfies Slater’s CQ as mentioned before, the KKT conditions hold at $(\xi^*, \Sigma^*)$, namely, there exists some Lagrange multiplier matrix $\Lambda^*$ such that

$$M\xi^* + \nabla f(x) - \nabla g(x)s - A^*(x)\Lambda^* = 0,$$

$$\sigma(\Sigma^* - \Lambda^*) = 0,$$

$$A(x)\xi^* + \sigma(\Sigma^* - T) \succeq 0,$$

$$\Lambda^* \succeq 0,$$

$$\langle A(x)\xi^* + \sigma(\Sigma^* - T), \Lambda^* \rangle = 0.$$ (33)

Applying a suitable convergent algorithm such as a primal-dual interior point method to QSDP (24) generates a sequence $\{(\xi_j, \Sigma_j, \Lambda_j)\}$ which converges to the above $(\xi^*, \Sigma^*, \Lambda^*)$. In particular, this sequence satisfies the following conditions:

$$\lim_{j \to \infty} \xi_j = \xi^*, \quad \lim_{j \to \infty} \Sigma_j = \Sigma^*, \quad \lim_{j \to \infty} \Lambda_j = \Lambda^*, \quad \lim_{j \to \infty} \eta_j = \eta^* = 0,$$

$$\lim_{j \to \infty} \Theta_j = \Theta^*, \quad \lim_{j \to \infty} \Omega_j = \Omega^*, \quad \lim_{j \to \infty} \langle \Omega_j, \Lambda_j \rangle = \langle \Omega^*, \Lambda^* \rangle = 0.$$ (33)
where \( \{(\eta_j, \Theta_j, \Omega_j)\} \) is a convergent sequence associated with violation error for (28)–(32) and is defined by

\[
\eta_j := M\xi_j + \nabla f(x) - \nabla g(x)s - A^T(x)\Lambda_j, \\
\Theta_j := \sigma(\Sigma_j - \Lambda_j), \\
\Omega_j := A(x)\xi_j + \sigma(\Sigma_j - T)
\]

for each \( j \in \mathbb{N} \cup \{0\} \). As indicated by the convergence analysis in the next section, sufficient decrease of the merit function \( F \) with an appropriate \( \sigma \) leads to a point satisfying optimality conditions for NSDP (1). Hence, if we find a point \( (\xi_j, \Sigma_j, \Lambda_j) \) which decreases \( F \) sufficiently, we may stop solving QSDP (24) or producing the sequence \( \{(\xi_j, \Sigma_j, \Lambda_j)\} \). Specifically, we do it if the following conditions hold for some \( j \in \mathbb{N} \cup \{0\} \):

\[
\langle \nabla F(x; y, Z, \sigma), \xi_j \rangle \leq -c_1 \langle M\xi_j, \xi_j \rangle - c_1 \sigma \|\Lambda_j - [T]_+\|_F^2, \\
\|\eta_j\| \leq c_2 \langle \nabla F(x; y, Z, \sigma), \xi_j \rangle,
\]

where \( c_1 \in (0, 1) \) and \( c_2 > 0 \) are prefixed constants. The above index \( j \in \mathbb{N} \) necessarily exists as is proved in Proposition 3. In the proposed method, we set a search direction as \( p := \xi_j \), and the next candidates, \( \bar{y} \) and \( \bar{Z} \), of Lagrange multipliers \( y \) and \( Z \) as

\[
\bar{y} := \xi_j, \quad \bar{Z} := [\Sigma_j]_+,
\]

respectively, where \( \xi_j := y - \frac{1}{q} \{g(x) + \nabla g(x)^T \xi_j\} \). One may wonder why we do not determine \( y \) and \( Z \) immediately in place of \( \bar{y} \) and \( \bar{Z} \). In the next step, we decide whether or not to set \( \bar{y} \) and \( \bar{Z} \) to be \( y \) and \( Z \) in view of some measures related to the optimality conditions.

Summarizing the above discussion, we can describe the way of solving the QSDP as in Algorithm 3.

**Remark 4** For solving NLPs, some SQPs make use of an approximate solution of a quadratic subproblem to determine the next point. For example, see [16].
4.3 Line-search along $p_k$ and update of $x_k$

From now on, we use a subscript $k \in \mathbb{N} \cup \{0\}$ to denote a current iterate. After computing the search direction $p_k$, we determine a step size along $p_k$ so that the function $F(\cdot; y_k, Z_k, \sigma_k)$ decreases. Notice that (37) with $\xi_j = p_k$ implies that $(\nabla F(x_k; y_k, Z_k, \sigma_k), p_k) \leq 0$. We adopt a backtracking line-search strategy using $F(\cdot; y_k, Z_k, \sigma_k)$ to set a step size $\alpha_k := \beta^{\ell_k}$, where $\beta \in (0, 1)$ is a given parameter and $\ell_k$ is the smallest nonnegative integer such that

$$F(x_k + \beta^{\ell_k} p_k; y_k, Z_k, \sigma_k) \leq F(x_k; y_k, Z_k, \sigma_k) + \tau \beta^{\ell_k} \Delta_k,$$

where $\Delta_k := \max \left\{ \left( \langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle, -\omega \|p_k\|^2 \right) \right\}$, $\omega \in (0, 1)$, and $\tau \in (0, 1)$. If the value of $|\langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle|$ is large, the term $-\omega \|p_k\|^2$ in $\Delta_k$ helps us to adopt $\ell_k$ at an early stage of the above procedure. After computing $\alpha_k = \beta^{\ell_k}$, we update $x_k$ as $x_{k+1} := x_k + \alpha_k p_k$.

4.4 Formal statement of the stabilized SQSDP method

By summarizing the explanation in the previous sections, the proposed stabilized SQSDP method is described as in Algorithm 4.
Finally, it may be worth noting the case where $\nabla F(x_k; y_k, Z_k, \sigma_k) = 0$ is satisfied in Step 1 of Algorithm 4. In this case, the optimal solution of QSDP (24) is calculated by $(0, [Z_k - \frac{1}{\sigma_k} X(x_k+1)]^+)$. Therefore, $v_{k+1} = (x_{k+1}, y_{k+1}, Z_{k+1})$ computed in Line 3 of Algorithm 4 is identical to the output of INEXACT-QSDP $(H_k, x_k, y_k, Z_k, \sigma_k)$. 

\section*{Algorithm 4 Inexact and stabilized SQSDP method (main algorithm)}

\textbf{Require:} Choose $v_0 := (x_0, y_0, Z_0)$ such that $Z_0 \succeq O$. Set

$$
k := 0, \overline{y}_0 := y_0, \overline{Z}_0 := Z_0, \phi_0 > 0, \psi_0 > 0, \gamma_0 > 0, \sigma_0 > 0.
$$

Select $\tau \in (0,1)$, $\omega \in (0,1)$, $c_1 \in (0,1)$, $c_2 > 0$, $\beta \in (0,1)$, $\kappa \in (0,1)$, $y_{\text{max}} > 0$, and $z_{\text{max}} > 0$.

1: repeat

2: \hspace{1em} \textbf{if} $\|\nabla F(x_k; y_k, Z_k, \sigma_k)\| = 0$ \textbf{then} \hspace{1em} \textgreater{} Step 1 (Mini-$F$-Phase)

3: \hspace{1em} $x_{k+1} := x_k, \overline{y}_{k+1} := y_k - \frac{1}{\sigma_k} g(x_{k+1}), \overline{Z}_{k+1} := [Z_k - \frac{1}{\sigma_k} X(x_{k+1})]^+.$

4: \hspace{1em} \textbf{else}

5: \hspace{2em} Choose $H_k \succ O$. Compute $p_k$, $\overline{y}_{k+1}$, and $\overline{Z}_{k+1}$ by

\begin{equation*}
(p_k, \overline{y}_{k+1}, \overline{Z}_{k+1}) := \text{INEXACT-QSDP}(H_k, x_k, y_k, Z_k, \sigma_k).
\end{equation*}

6: \hspace{2em} Compute the smallest nonnegative integer $\ell_k$ such that

\begin{align*}
F(x_k + \beta^{\ell_k} p_k; y_k, Z_k, \sigma_k) &\leq F(x_k; y_k, Z_k, \sigma_k) + \tau \beta^\ell_k \Delta_k, \\
\Delta_k &:= \max\{\|\nabla F(x_k; y_k, Z_k, \sigma_k, p_k), -\omega \|p_k\|^2\}.
\end{align*}

7: \hspace{1em} Set $x_{k+1} := x_k + \beta^{\ell_k} p_k$.

8: \hspace{1em} \textbf{end if}

9: \hspace{1em} Compute $y_{k+1}, Z_{k+1}, \phi_{k+1}, \psi_{k+1}$, and $\gamma_{k+1}$ by \hspace{1em} \textgreater{} Step 2

\begin{equation*}
(y_{k+1}, Z_{k+1}, \phi_{k+1}, \psi_{k+1}, \gamma_{k+1}) := \text{VOMF-ITERATES}(\overline{v}_{k+1}, y_k, Z_k, \phi_k, \psi_k, \gamma_k, \sigma_k).
\end{equation*}

10: \hspace{1em} Update $\sigma_k$ by \hspace{1em} \textgreater{} Step 3

\begin{equation*}
\sigma_{k+1} := \begin{cases} 
\frac{1}{2} \sigma_k, & r(y_{k+1}) \frac{3}{2} \text{ if } \|\nabla F(x_{k+1}; y_k, Z_k, \sigma_k)\| \leq \gamma_k, \\
\sigma_k, & \text{otherwise}.
\end{cases}
\end{equation*}

11: \hspace{1em} Set $k := k + 1$. \hspace{1em} \textgreater{} Step 4

12: \hspace{1em} \textbf{until} $v_k := (x_k, y_k, Z_k)$ meets a suitable criterion.
5 Convergence analysis of the stabilized SQSDP method

We show the global convergence of Algorithm 4.

5.1 Well-definedness of Algorithm 4

In what follows, we assume that \( M > O \) and also that \( \{(\xi_j, \Sigma_j, \Lambda_j, \eta_j, \Theta_j, \Omega_j)\} \) is a sequence produced in INEXACT-QSDP. Recall that this sequence satisfies (33). Moreover, we often use

\[
R_j := \langle \eta_j, \xi_j \rangle + \langle \Omega_j, \Lambda_j \rangle - \langle \Omega_j, [T]_+ \rangle + \sigma \langle \Lambda_j - [T]_+, T - [T]_+ \rangle - \langle \Lambda_j - [T]_+, \Theta_j \rangle
\]

(40)

for each \( j \in \mathbb{N} \cup \{0\} \).

Our goal in Sect. 5.1 is to prove that Algorithm 4 is well-defined in the sense that INEXACT-QSDP is terminated within a finite number of iterations. Specifically, we aim to prove the following proposition:

**Proposition 3** If \( \nabla F(x; y, Z, \sigma) \neq 0 \), then there exists \( j_0 \in \mathbb{N} \) such that (37) and (38) hold for all \( j \geq j_0 \), meaning that INEXACT-QSDP terminates finitely.

In order to prove this proposition, we prepare three lemmas. In the first lemma, we show that \( \xi_j \) is a descent direction of the merit function \( F \) if

\[
R_j \leq (1 - c_1) \langle M \xi_j, \xi_j \rangle + \sigma (1 - c_1) \| \Lambda_j - [T]_+ \|^2_F.
\]

(41)

**Lemma 2** If there exists \( j \in \mathbb{N} \) satisfying (41), then (37) holds.

**Proof** Since \( \nabla F(x; y, Z, \sigma) = \nabla f(x) - \nabla g(x)s - A^*(x)[T]_+ \) by (4), we have

\[
\langle \nabla F(x; y, Z, \sigma), \xi_j \rangle
\]

\[
= -\langle M \xi_j, \xi_j \rangle + \langle \eta_j, \xi_j \rangle + \langle \Lambda_j - [T]_+, A(x) \xi_j \rangle
\]

\[
= -\langle M \xi_j, \xi_j \rangle + \langle \eta_j, \xi_j \rangle + \langle \Lambda_j - [T]_+, \Omega_j \rangle + \sigma \langle \Lambda_j - [T]_+, T - [T]_+ \rangle + \sigma \langle \Lambda_j - [T]_+, [T]_+ - \Sigma_j \rangle
\]

\[
= -\langle M \xi_j, \xi_j \rangle + \langle \eta_j, \xi_j \rangle + \langle \Omega_j, \Lambda_j \rangle - \langle \Omega_j, [T]_+ \rangle + \sigma \langle \Lambda_j - [T]_+, T - [T]_+ \rangle - \sigma \| \Lambda_j - [T]_+ \|^2_F - \langle \Lambda_j - [T]_+, \Theta_j \rangle
\]

\[
= -\langle M \xi_j, \xi_j \rangle - \sigma \| \Lambda_j - [T]_+ \|^2_F + R_j,
\]

where the first, second, and third equalities are derived from (34), (36), and (35), respectively. The combination of (41) and (42) yields that

\[
\langle \nabla F(x; y, Z, \sigma), \xi_j \rangle \leq -c_1 \langle M \xi_j, \xi_j \rangle - c_1 \sigma \| \Lambda_j - [T]_+ \|^2_F.
\]

\( \square \)
In the second lemma, we prove that there exists some index \( j \in \mathbb{N} \) fulfilling condition (41) if the following condition holds:

\[
\exists K > 0 \quad \text{s.t.} \quad K \leq \langle M \xi_j, \xi_j \rangle + \sigma \| \Lambda - [T]_+ \|_F^2 \quad \forall j \in \mathbb{N}. \tag{43}
\]

**Lemma 3** Assume that (43) holds. Then, there exists \( n_0 \in \mathbb{N} \) such that (37) holds for all \( j \geq n_0 \).

**Proof** It is clear that \( \langle \Omega^*, [T]_+ \rangle \geq 0 \) from \( \Omega^* \geq O \). Moreover, we see that \( \langle \Lambda^* - [T]_+, T - [T]_+ \rangle \leq 0 \) by the facts that \( \Lambda^* \geq O \) and \([ \cdot ]_+\) is the projection onto the convex set \( S^d_+ \). We have from these facts, (33), and (40) that

\[
R^* := \lim_{j \to \infty} R_j = -\langle \Omega^*, [T]_+ \rangle + \sigma \langle \Lambda^* - [T]_+, T - [T]_+ \rangle \leq 0. \tag{44}
\]

Since (44) holds, there exists \( n_0 \in \mathbb{N} \) such that \( R_j \leq R_j - R^* \) and \( |R_j - R^*| \leq (1 - c_1)K \) for all \( j \geq n_0 \), that is, \( R_j \leq (1 - c_1)K \) for all \( j \geq n_0 \). Combining this result and (43) yields that (41) holds for all \( j \geq n_0 \). It then follows from Lemma 2 that (37) holds for all \( j \geq n_0 \).

By Lemma 3, we ensure the existence of the index \( j \) such that \( \xi_j \) is the descent direction of \( F \) if (43) holds. In the third lemma, we show that \( \nabla F(x; y, Z, \sigma) \neq 0 \) is a sufficient condition for (43).

**Lemma 4** If \( \nabla F(x; y, Z, \sigma) \neq 0 \), then (43) holds.

**Proof** We prove the assertion by contradiction. To this end, assume that there exists \( \mathcal{L} \subset \mathbb{N} \) such that

\[
\langle M \xi_j, \xi_j \rangle + \sigma \| \Lambda - [T]_+ \|_F^2 \to 0 \quad (\mathcal{L} \ni j \to 0).
\]

Considering \( M > O \) and (33) implies that \( \xi^* = 0 \) and \( \Lambda^* = [T]_+ \). It then follows from (33) and (34) that

\[
0 = \lim_{c_2 j \to \infty} \eta_j = M \xi^* + \nabla f(x) - \nabla g(x)s - \mathcal{A}^\ast(x)\Lambda^* = \nabla f(x) - \nabla g(x)s - \mathcal{A}^\ast(x)[T]_+.
\]

Now, recall that \( s = y - \frac{1}{\sigma} g(x) \) and \( T = Z - \frac{1}{\sigma} X(x) \). As a result, we obtain

\[
\nabla F(x; y, Z, \sigma) = 0 \text{ by (4).}
\]

However, this contradicts \( \nabla F(x; y, Z, \sigma) \neq 0 \).

By using Lemmas 3 and 4, Proposition 3 is proven as follows.

**Proof of Proposition 3** By Lemmas 3 and 4, there exists \( n_0 \in \mathbb{N} \) such that

\[
\langle \nabla F(x; y, Z, \sigma), \xi_j \rangle \leq -c_1 \langle M \xi_j, \xi_j \rangle - c_1 \sigma \| \Lambda - [T]_+ \|_F^2 \quad \forall j \geq n_0. \tag{45}
\]
Lemma 4 and (45) derive that there exists $K > 0$ such that
\[
c_1c_2 K \leq c_1c_2 \langle M \xi_j, \xi_j \rangle + c_1c_2 \sigma \| \Lambda_j - [T]_+ \|^2_F
\leq -c_2 \langle \nabla F(x; y, Z, \sigma), \xi_j \rangle \quad \forall j \geq n_0. \tag{46}
\]
Since (33) indicates that \( \| \eta_j \| \to 0 \) as \( j \to \infty \), there exists \( n_1 \in \mathbb{N} \) such that \( \| \eta_j \| \leq c_1c_2 K \) for all \( j \geq n_1 \). This fact and (46) yield that
\[
\| \eta_j \| \leq c_2 \langle \nabla F(x; y, Z, \sigma), \xi_j \rangle \quad \forall j \geq \max \{ n_0, n_1 \}. \tag{47}
\]
It follows from (45) and (47) that there exists \( j_0 := \max \{ n_0, n_1 \} \) such that (37) and (38) hold for all \( j \geq j_0 \). The proof is complete. \( \square \)

5.2 Global convergence of Algorithm 4

In what follows, we prove the global convergence property of Algorithm 4. Notice that Algorithm 4 is within a scope of the global convergence analysis of Algorithm 1 that we have established in Sect. 3.2, which supposes Assumptions 1 and 2. The aim in this section is to prove that Assumption 2 actually holds as for Algorithm 4 under the presence of Assumption 1 together with the following one:

**Assumption 3** There exist positive constants \( \nu_1 \) and \( \nu_2 \) such that
\[
\nu_1 \leq \lambda_{\min}(H_k + \frac{1}{\sigma_k} \nabla g(x_k) \nabla g(x_k)^\top), \quad \lambda_{\max}(H_k) \leq \nu_2,
\]
for all \( k \in \mathbb{N} \cup \{ 0 \} \).

This assumption is controllable in the sense that it certainly holds if we choose \( H_k \) to be a bounded positive definite matrix for each \( k \in \mathbb{N} \cup \{ 0 \} \). Indeed, for \( a \) and \( b \) with \( 0 < a < b \), we can set \( H_k \) so that \( aI \leq H_k \leq bI \) for each \( k \), e.g., \( H_k = \frac{a+b}{2} I \). Then, the above assumption holds with \( (\nu_1, \nu_2) = (a, b) \).

As a blanket assumption throughout the remaining of the subsection, we suppose that Algorithm 4 generates an infinite set of iterations.

For simplicity, we denote
\[
M_k := H_k + \frac{1}{\sigma_k} \nabla g(x_k) \nabla g(x_k)^\top, \quad \tag{48}
\]
\[
s_k := y_k - \frac{1}{\sigma_k} g(x_k), \quad \tag{49}
\]
\[
T_k := Z_k - \frac{1}{\sigma_k} X(x_k). \quad \tag{50}
\]
Moreover, we use the following notation:
A stabilized SQP method for nonlinear semidefinite programs

\[ \eta_k := M_k p_k + \nabla f(x_k) - \nabla g(x_k) s_k - A^*(x_k) \Lambda_k, \]
\[ \Theta_k := \sigma_k (\Sigma_k - \Lambda_k), \]
\[ \Omega_k := A(x_k) p_k + \sigma_k (\Sigma_k - T_k), \]

where \( \Sigma_k \) and \( \Lambda_k \) are final iteration points of the finite sequences \( \{ \Sigma_j \} \) and \( \{ \Lambda_j \} \), respectively, generated in INEXACT-QSDP. From Line 7 in INEXACT-QSDP, it is clear that

\[ \langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle \leq -c_1 \langle M_k p_k, p_k \rangle - c_1 \sigma_k \| \Lambda_k - [T_k]_+ \|^2 \leq 0, \tag{52} \]
\[ \| \eta_k \| \leq c_2 \langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle. \tag{53} \]

From now, we will show that there never occurs a situation that \( \text{card}(\mathcal{I}) < \infty \), \( \text{card}(\mathcal{J}) < \infty \), and \( \text{card}(\mathcal{K}) = \infty \), where recall that the sets \( \mathcal{I}, \mathcal{J}, \) and \( \mathcal{K} \) are related to VOMF-ITERATES presented as Algorithm 2 and are defined by (9). To this end, we prepare the following lemma.

**Lemma 5** Suppose that Assumptions 1 and 3 hold. If there occurs a situation that \( \text{card}(\mathcal{I}) < \infty \), \( \text{card}(\mathcal{J}) < \infty \), and \( \text{card}(\mathcal{K}) = \infty \), then

(i) there exist \( k_0 \in \mathbb{N}, \hat{\gamma} \in \mathbb{R}^m, \hat{Z} \in \mathbb{S}^d, \hat{\sigma} \in \mathbb{R} \) such that \( k \in \mathcal{K}, y_k = \hat{\gamma}, Z_k = \hat{Z}, \gamma_k = \hat{\gamma} \) for all \( k \geq k_0 \);
(ii) \( \{ p_k \}_{k \geq k_0} \) is bounded;
(iii) \( \liminf_{k \to \infty} | \Delta_k | > 0. \)

**Proof** First, we prove item (i). Since \( \text{card}(\mathcal{I}) < \infty \), \( \text{card}(\mathcal{J}) < \infty \), and \( \text{card}(\mathcal{K}) = \infty \), there exists \( k_0 \in \mathbb{N} \) such that \( k \in \mathcal{K}, y_k = \hat{\gamma}, Z_k = \hat{Z}, \) and \( \gamma_k = \hat{\gamma} \) for all \( k \geq k_0 \), where \( \hat{\gamma} := y_{k_0}, \hat{Z} := Z_{k_0}, \) and \( \hat{\gamma} := y_{k_0} \). Moreover (8) yields that \( \sigma_k = \hat{\sigma} \) for all \( k \geq k_0 \), where \( \hat{\sigma} := \sigma_{k_0} \).

Next, we show item (ii). By Assumption 1 (A2), there exists a compact set \( \Gamma \) such that \( \{ x_k \} \subset \Gamma \). From now on, assume that \( k \geq k_0 \), that is, item (i) holds. The combination of (49), (50), and Assumption 1 (A1) and (A2) implies that

\[ \| s_k \| \leq \| \hat{\gamma} \| + \frac{1}{\varepsilon} \sup_{x \in \Gamma} \| g(x) \| =: s_{\max} < \infty \]
and
\[ \| T_k \|_F \leq \| \hat{Z} \|_F \leq \| \hat{Z} \|_F + \frac{1}{\varepsilon} \sup_{x \in \Gamma} \| X(x) \|_F =: T_{\max} < \infty. \]
Assumption 3 and (52) derive that

\[ \frac{c_1}{c_2} \| p_k \|^2 \leq -\langle \nabla F(x_k; \hat{\gamma}, \hat{Z}, \hat{\sigma}), p_k \rangle \leq \| \nabla F(x_k, \hat{\gamma}, \hat{Z}, \hat{\sigma}) \| \| p_k \|. \]

These results and (4) yield that
\[
\begin{align*}
\frac{c_1}{\nu_1} \|p_k\| &\leq \left\| \nabla f(x_k) \| + \| \nabla g(x_k) \|_2 \| s_k \| + \sqrt{\sum_{j=1}^{n} \| A_j(x_k) \|_{F}^{2} \| [T]_+ \|_{F}^{2}} \\
&\leq \sup_{x \in \Gamma} \left\{ \| \nabla f(x) \| + s_{\text{max}} \| \nabla g(x) \|_2 + T_{\text{max}} \sqrt{\sum_{j=1}^{n} \| A_j(x) \|_{F}^{2}} \right\},
\end{align*}
\]

Therefore, the boundedness of \( \{p_k\}_{k \geq k_0} \) is derived from Assumption 1.

Finally, we prove item (iii). In order to show this, we first verify that
\[
\liminf_{k \to \infty} |\langle \nabla F(x_k, y_k, Z_k, \sigma_k), p_k \rangle| > 0.
\]
Now, we assume to the contrary that
\[
\liminf_{k \to \infty} |\langle \nabla F(x_k, y_k, Z_k, \sigma_k), p_k \rangle| = 0,
\]
that is, there exists \( \mathcal{L} \subset \mathbb{N} \) such that
\[
|\langle \nabla F(x_k, y_k, Z_k, \sigma_k), p_k \rangle| \to 0 \quad (\mathcal{L} \ni k \to \infty). \tag{54}
\]
Let \( k \) be a positive integer satisfying \( k \geq k_0 \). Note that item (i) yields \( k \in \mathcal{K}, y_k = \hat{y}, Z_k = \hat{Z}, \gamma_k = \hat{\gamma}, \) and \( \sigma_k = \hat{\sigma} \). Now, we can easily see that the if-statement of the M-iterate in Algorithm 2 is false for \( k \geq k_0 \), namely,
\[
\| \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}) \| > \hat{\gamma} > 0 \quad \forall k \geq k_0 + 1. \tag{55}
\]
Meanwhile, it follows from (52), (53), and Assumption 3 that \( 0 \leq c_1 \nu_1 \|p_k\|^2 + c_1 \hat{\sigma} \| \Lambda_k - [T_k]_+ \|_{F}^2 \leq -\langle \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}), p_k \rangle \) and \( \| \eta_k \| \leq c_2 |\langle \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}), p_k \rangle| \). By considering
\[
|\langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle| \to 0 \quad \text{as } \mathcal{L} \ni k \to \infty,
\]
we obtain \( \|p_k\| \to 0, \| \Lambda_k - [T_k]_+ \|_{F} \to 0, \) and \( \| \eta_k \| \to 0 \). Moreover, we have from (4), (48), (51), and Assumptions 1 and 3 that
\[
\| \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}) \| \leq \| \eta_k \| + \left\{ \nu_2 + \frac{1}{\hat{\sigma}} \sup_{x \in \Gamma} \| \nabla g(x) \|_2 \right\} \|p_k\| + \left\{ \sup_{x \in \Gamma} \sqrt{\sum_{j=1}^{n} \| A_j(x) \|_{F}^2} \right\} \| \Lambda_k - [T_k]_+ \|_{F} \to 0
\]
as \( \mathcal{L} \ni k \to \infty \). However, this fact contradicts (55). As a result, we conclude that
\[
\liminf_{k \to \infty} |\langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle| > 0.
\]
Secondly, we verify that \( \liminf_{k \to \infty} \|p_k\| > 0 \) by contradiction. Suppose to the contrary that there exists \( \mathcal{M} \subset \mathbb{N} \) such that \( \|p_k\| \to 0 \) as \( \mathcal{M} \ni k \to \infty \). Then, it is clear that \( \langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle \to 0 \) as \( \mathcal{M} \ni k \to \infty \). Since this result corresponds to (54), the remainder of the proof can be given in a way similar to that of \( \liminf_{k \to \infty} |\langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle| > 0 \). Therefore, we obtain \( \liminf_{k \to \infty} \|\langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle| > 0 \) and \( \liminf_{k \to \infty} \|p_k\| > 0 \). It follows from \( \Delta_k = \max \{ |\langle \nabla F(x_k; y_k, Z_k, \sigma_k), p_k \rangle|, -\omega \|p_k\|^2 \} \) that \( \liminf_{k \to \infty} |\Delta_k| > 0 \). The proof is complete. \( \square \)
Using Lemma 5, we show that Algorithm 4 does not generate an infinite set of iterations satisfying $\text{card}(I) < \infty$, $\text{card}(J) < \infty$, and $\text{card}(K) = \infty$.

**Theorem 5** Suppose that Assumptions 1 and 3 hold. Then, there never occurs a situation that $\text{card}(I) < \infty$, $\text{card}(J) < \infty$, and $\text{card}(K) = \infty$, meaning that Assumption 2 holds.

**Proof** We prove this theorem by contradiction. Suppose that Algorithm 4 generates an infinite set of iterations satisfying card($I$) < $\infty$, card($J$) < $\infty$, and card($K$) = $\infty$. By Lemma 5 (i), there exists $k_0 \in \mathbb{N}$ such that $k \in K$, $y_k = \hat{y}$, $Z_k = \hat{Z}$, $\gamma_k = \hat{\gamma}$, and $\sigma_k = \hat{\sigma}$ for all $k \geq k_0$. From now on, we assume that $k \geq k_0$. Notice that the if-statement of Step 1 is false because if there exists an iteration $k_1$ such that it is true, then $k_1$ is included in $J$ at least, that is, it contradicts $k_1 \in K$. In what follows, we derive a contradiction with Lemma 5 (iii). Note that $\Delta_k \leq 0$ due to (52). We have from (39) that $F(x_{k+1}; \hat{y}, \hat{Z}, \hat{\sigma}) \leq F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}) + \tau \beta_k \Delta_k$, that is,

$$0 \leq -\tau \beta_k \Delta_k \leq F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}) - F(x_{k+1}; \hat{y}, \hat{Z}, \hat{\sigma}).$$

By Assumption 1, it is clear that $\{F(x_k; \hat{y}, \hat{Z}, \hat{\sigma})\}_{k \geq k_0}$ is bounded below. Furthermore, it is a non-increasing sequence. These facts and (56) yield $\beta_k \Delta_k \to 0 (k \to \infty)$. Then, there are the two cases: (a) $\liminf_{k \to \infty} \beta_k > 0$; (b) $\liminf_{k \to \infty} \beta_k = 0$.

**Case (a):** We readily have

$$\Delta_k \to 0 \quad (k \to \infty).$$

**Case (b):** In this case, there exists $L \subset \mathbb{N}$ such that $\ell_{k} \to \infty$ as $L \ni k \to \infty$. Moreover, Assumption 1 (A2) implies that there exist $L_2 \subset \mathbb{N}$ and $x^* \in \mathbb{R}^n$ such that $x_k \to x^*$ as $L \ni k \to \infty$. Let $L := L_1 \cap L_2$, $k \in L$, and $\delta_k := \beta_k^{-1}(> 0)$. Without loss of generality, we assume $\ell_k \geq 1$ for all $k \in L$ because $\ell_k \to \infty$ as $L \ni k \to \infty$. Recall that $\ell_k$ is the smallest nonnegative integer such that $F(x_k + \beta_k \delta_k p_k; \hat{y}, \hat{Z}, \hat{\sigma}) \leq F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}) + \tau \delta_k \Delta_k$. Since $\ell_k - 1$ does not satisfy the inequality, we get $F(x_k + \delta_k p_k; \hat{y}, \hat{Z}, \hat{\sigma}) > F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}) + \tau \delta_k \Delta_k$. From $\langle \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}), p_k \rangle \leq \max\{\langle \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}), p_k \rangle, -\omega\|p_k\|^2\} = \Delta_k \leq 0$, we have

$$(\tau - 1)\Delta_k < \frac{F(x_k + \delta_k p_k; \hat{y}, \hat{Z}, \hat{\sigma}) - F(x_k; \hat{y}, \hat{Z}, \hat{\sigma})}{\delta_k} - \langle \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}), p_k \rangle.$$

Applying the mean value theorem derives that there exists $\theta_k \in (0, 1)$ satisfying

$$0 \leq (\tau - 1)\Delta_k \leq \langle \nabla F(x_k + \theta_k \delta_k p_k; \hat{y}, \hat{Z}, \hat{\sigma}) - \nabla F(x_k; \hat{y}, \hat{Z}, \hat{\sigma}), p_k \rangle,$$

where note that $\tau \in (0, 1)$. Notice that $\{p_k\}_{k \geq k_0}$ is bounded from Lemma 5 (ii). Notice also that $\delta_k = \beta_k^{-1} \to 0$ as $L \ni k \to \infty$. As a result, we see that $x^* = \lim_{L \ni k \to \infty} x_k = \lim_{L \ni k \to \infty} (x_k + \theta_k \delta_k p_k)$. Thus, the continuity of $\nabla F$ and (58) yield that

$$\Delta_k \to 0 \quad (L \ni k \to \infty).$$

\(\square\) Springer
We have from (57) and (59) that \( \liminf_{k \to \infty} |\Delta_k| = 0 \). Therefore, this contradicts Lemma 5 (iii).

Combining Theorems 3, 4, and 5 readily yields the following theorem.

**Theorem 6** Suppose that Assumptions 1 and 3 hold. Let \( \{(x_k, y_k, Z_k)\} \) be a sequence generated by Algorithm 1 and let \( \{\tilde{y}_k\} \) and \( \{\tilde{Z}_k\} \) be defined by \( \tilde{y}_k := y_{k-1} - \frac{1}{\sigma_{k-1}} g(x_k) \), and \( \tilde{Z}_k := [Z_{k-1} - \frac{1}{\sigma_{k-1}} X(x_k)]_+ \), respectively. Then, any accumulation point of \( \{x_k\} \), say \( x^* \), satisfies at least one of the following statements:

(i) \( x^* \) is a TAKKT point of (1) and some subsequence of \( \{(x_k, y_k, Z_k)\} \) is a TAKKT sequence corresponding to \( x^* \);

(ii) \( x^* \) is an AKKT point of (1) and some subsequence of \( \{(x_k, \tilde{y}_k, \tilde{Z}_k)\} \) is an AKKT sequence corresponding to \( x^* \);

(iii) \( x^* \) is an infeasible point of (1), but a stationary point of the following optimization problem:

\[
\min_{x \in \mathbb{R}^n} h(x) := \frac{1}{2} \|g(x)\|^2 + \frac{1}{2} \|[-X(x)]_+\|^2_F,
\]

that is to say, \( \nabla h(x^*) = 0 \).

Moreover, if any accumulation point of \( \{x_k\} \) satisfies the extended MFCQ, then \( x^* \) is a KKT point.

**Remark 5** One may ask whether the analysis of the current paper is word-by-word extension of that for the stabilized SQP-type methods such as [11, 28] in NLP-literature. However, it is not the case. Indeed, [11, 28] consider convergence to a KKT point, whereas ours to TAKKT and AKKT points. We emphasize that the TAKKT and AKKT conditions are concepts quite peculiar to the NSDP, and thus the analysis conducted so far to establish convergence to such AKKT and TAKKT points is non-trivial.

**6 Numerical experiments**

We conduct some numerical experiments to examine the efficiency of Algorithm 4. The experiments consist of two parts. In the first part, we solve two degenerate problems such that Slater’s CQ does not hold. In the second one, we solve non-degenerate problems satisfying Slater’s CQ. For the sake of comparison, we also implemented the augmented Lagrangian (AL) method proposed in [3]. The program was implemented with MATLAB R2020b and ran on a machine with an Intel Core i9-9900K 3.60 GHz CPU and 128 GB RAM.

In what follows, we describe the concrete setting of Algorithm 4 and the AL method. We first explain the setting of Algorithm 4. We adopted the following as stopping conditions:

\[ r(v_k) \leq 10^{-6}, \quad \gamma_k \leq 10^{-6}, \text{ or } k_{\text{max}} = 100, \]
where we notice that the function \( r \) is defined by (6). In Step 2, we relaxed the condition \( \| \nabla F(x_k; y_k, Z_k, \sigma_k) \| = 0 \) to

\[ \| \nabla F(x_k; y_k, Z_k, \sigma_k) \| \leq 10^{-6}. \]

We utilized SDPT3 version 4.0 \cite{26, 27} for solving subproblem (24), where we set its stopping criterion to \( \| \text{gaptol} \| = 10^{-10} \). To ensure the positive definiteness of \( M_k \) defined by (48) with \( H_k := \nabla^2_{xx} L(v_k) \), we modified it by

\[ M_k \leftarrow \begin{cases} M_k & \text{if the Cholesky decomposition of } M_k \text{ is successful,} \\ M_k + \mu_k I & \text{otherwise,} \end{cases} \]

where \( \mu_k := |\lambda_{\min}(M_k)| + 10^{-5} \). The parameters in Algorithm 4 were set as follows:

\[ \begin{align*} 
\tau & := 10^{-4}, \quad \omega := 10^{-4}, \quad \beta := 0.5, \quad \kappa := 10^{-5}, \quad \gamma_{\max} := 10^6, \quad \zeta_{\max} := 10^6, \\
\phi_0 & := 10^3, \quad \psi_0 := 10^3, \quad \gamma_0 := 10^{-1}, \quad \sigma_0 := 10^{-1}.
\end{align*} \]

Moreover, its initial point was selected as \( (x_0, y_0, Z_0) := (0, 0, O) \).

Next, we explain the setting of the AL method. The overall AL method is described in Appendix A. We used the MATLAB unconstrained optimizer \textit{fminunc} to find an approximate minimizer \( x_{k+1} \) of the unconstrained optimization subproblem in Step 2 which satisfies

\[ \| \frac{\nabla L(x_{k+1}, y_{k+1}) - \rho_k g(x_{k+1}), [Z_k - \rho_k X(x_{k+1})]_+]}{\| \nabla F(x_k; y_k, Z_k, \sigma_k) \|} \| \leq \epsilon_k, \]

where \( \{ \epsilon_k \} \) is defined as \( \epsilon_{k+1} := \frac{1}{2} \min \{ \epsilon_k, r(v_k) \} \) with \( \epsilon_0 = 1 \). The parameters were set as follows:

\[ \begin{align*} 
\epsilon & := 10^{-6}, \quad \tau := 0.5, \quad \gamma := 2, \quad k_{\max} := 100, \quad \gamma_{\max} := 10^6, \quad \zeta_{\max} := 10^6, \quad \rho_0 := 10.
\end{align*} \]

Its initial point \( (x_0, y_0, Z_0) \) was selected as the same one used in Algorithm 4.

### 6.1 Solving degenerate problems

This section deals with two degenerate problems described in \cite[Section 3]{13}. The first one is of the following form:

\[ \begin{align*} 
\min_{X \in \mathbb{S}^N} & \quad \langle C, X \rangle \\
\text{subject to} & \quad [X]_{jj} = 1, \ j = 1, \ldots, N, \\
& \quad \langle J, X \rangle = 0, \ X \succeq O,
\end{align*} \]

(60)

where \( C \in \mathbb{S}^N \) is a constant matrix whose elements are chosen randomly from \([-1, 1]\), and \( J := ee^\top \). Due to \( e^\top X e = \langle J, X \rangle = 0 \), problem (60) has no strictly feasible point. Thus, Slater’s CQ fails for this problem.

The second problem is as follows:
where \( M, N \in \mathbb{N} \) are constants with \( 0 < M \leq N \), and \( b := [0, 1, 1, \ldots, 1]^T \in \mathbb{R}^M \), and \( \alpha \in \mathbb{R}^N \) is a constant vector whose \( n \)-th element is set to zero and the others are chosen randomly from \([0, 1]\), and \( A_j := v_j v_j^T \in \mathbb{S}^N \) for each \( j = 1, \ldots, N \), where \( v_1, \ldots, v_N \in \mathbb{R}^N \) are arbitrary orthonormal basis vectors. Problem (61) also has no strictly feasible point because \( v_1^T X v_1 = \langle A_1, X \rangle = [b]_1 = 0 \). Hence, Slater’s CQ does not hold.

We generated 10 instances for each of problems (60) and (61) with \( N = 5, 10, 15, \) and 20 in the above manner and applied the proposed SQSDP methods and the AL method to them. Tables 1, 2, 3, and 4 summarize the obtained results, where “Averaged \# iteration”, “Averaged time”, “Averaged \( r(v^*) \)”, and “Averaged max\{\|y^*\|, \|Z^*\|_F\}” represent the averaged results for 10 instances for each \( N \), and herein \( v^* \), \( y^* \), and \( Z^* \) express the final iteration point of the sequence \( \{v_k\}, \{y_k\}, \) and \( \{Z_k\} \) generated by each algorithm, respectively.
A stabilized SQP method for nonlinear semidefinite programs

CPU time (s) stand for the number of iterations and spent computational time in seconds, respectively. In addition, the maximum and minimum of \( r(v^*) \) and \( \max(\|y^*\|, \|Z^*\|_F) \) among the 10 runs are shown in the tables. Tables 1 and 2 show the results of problem (60), while Tables 3 and 4 do those of problem (61). Figures 1 and 2 monitor change of \( r(v_k) \) in Algorithm 4 and the AL method for one of the 10 instances of problems (60) and (61), respectively.

From the tables, we have the following observations: According to Tables 1 and 2, Algorithm 4 and the AL method stopped by reaching the maximum number of iterations 100 for all the instances, but the results of the two algorithms are crucially different. Indeed, while the AL method diverged in each \( N \), all the final iteration points found by Algorithm 4 satisfied \( r(v^*) \leq 10^{-3} \), which means that Algorithm 4 found points satisfying the KKT conditions approximately for all \( N \).

Next, according to Tables 3 and 4 concerning problem (61), Algorithm 4 successfully found KKT points for all the instances, whereas the AL method failed in most cases. Furthermore, from the values of \( \max(\|y^*\|, \|Z^*\|_F) \) in Tables 2 and 4, the sequences of Lagrange multipliers computed by the AL method were likely to diverge. As results unseen from the tables, the V-, O-, M-, and F-iterate are performed for about 33.3%, 20.8%, 0.3%, and 45.6% of all the iterations per a single instance.

| Table 3 | Performance of Algorithm 4 on problem (61) |
|----------|------------------------------------------|
| \((N, M)\) | (15,5) | (15,10) | (15,15) | (20,7) | (20,14) | (20,20) |
| Averaged \# ite | 3.4 | 4.3 | 3.0 | 4.0 | 4.4 | 3.6 |
| Averaged cputime(s) | 4.1 | 5.1 | 2.9 | 20.7 | 22.8 | 15.8 |
| Averaged \( r(v^*) \) | 4.1e–07 | 3.9e–09 | 1.5e–07 | 2.4e–07 | 8.1e–09 | 1.0e–07 |
| Max of \( r(v^*) \) | 7.4e–07 | 1.1e–08 | 8.6e–07 | 9.1e–07 | 3.5e–08 | 5.1e–07 |
| Min of \( r(v^*) \) | 2.6e–10 | 7.3e–10 | 5.2e–09 | 3.4e–10 | 5.7e–10 | 7.8e–12 |
| Averaged \( \max(\|y^*\|, \|Z^*\|_F) \) | 2.3e+00 | 2.9e+00 | 2.1e+00 | 2.7e+00 | 2.7e+00 | 2.5e+00 |
| Max of \( \max(\|y^*\|, \|Z^*\|_F) \) | 3.5e+00 | 3.7e+00 | 2.3e+00 | 3.5e+00 | 3.0e+00 | 3.3e+00 |
| Min of \( \max(\|y^*\|, \|Z^*\|_F) \) | 1.8e+00 | 2.4e+00 | 1.4e+00 | 2.1e+00 | 2.4e+00 | 2.3e+00 |

| Table 4 | Performance of the AL method on problem (61) |
|----------|------------------------------------------|
| \((N, M)\) | (15,5) | (15,10) | (15,15) | (20,7) | (20,14) | (20,20) |
| Averaged \# ite | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Averaged cputime(s) | 107.0 | 108.2 | 109.3 | 342.8 | 343.7 | 332.3 |
| Averaged \( r(v^*) \) | 2.9e+31 | 3.4e+31 | 3.7e+31 | 3.4e+31 | 4.3e+31 | 4.2e+31 |
| Max of \( r(v^*) \) | 3.2e+31 | 4.3e+31 | 4.1e+31 | 3.7e+31 | 4.7e+31 | 4.8e+31 |
| Min of \( r(v^*) \) | 2.4e+31 | 2.9e+31 | 3.4e+31 | 3.0e+31 | 3.9e+31 | 3.8e+31 |
| Averaged \( \max(\|y^*\|, \|Z^*\|_F) \) | 1.1e+31 | 1.2e+31 | 1.2e+31 | 1.2e+31 | 1.3e+31 | 1.3e+31 |
| Max of \( \max(\|y^*\|, \|Z^*\|_F) \) | 1.1e+31 | 1.3e+31 | 1.3e+31 | 1.2e+31 | 1.4e+31 | 1.4e+31 |
| Min of \( \max(\|y^*\|, \|Z^*\|_F) \) | 9.4e+30 | 1.1e+31 | 1.2e+31 | 1.1e+31 | 1.3e+31 | 1.2e+31 |
run of Algorithm 4 for solving problem (60). On the other hand, for problem (61), we observed that the V-iterate was employed in almost all the iterations.

6.2 Solving non-degenerate problems

Next, we examine the efficiency of Algorithm 4 for two kinds of non-degenerate problems. Those problems are also solved in [31, 34]. The first one is the Gaussian channel capacity problem:

maximize \[ \frac{1}{2} \sum_{j=1}^{N} \log(1 + t_j) \]
subject to \[ \frac{1}{N} \sum_{j=1}^{N} x_j \leq 1, \quad x \geq 0, \quad t \geq 0, \]
\[ \left[ 1 - a_j t_j \sqrt{r_j} \quad a_j x_j + r_j \right] \succeq O, \quad j = 1, \ldots, N, \tag{62} \]

where the constants \( a_1, \ldots, a_N, r_1, \ldots, r_N \in \mathbb{R} \) are generated randomly from \([0, 1]\).

The second problem is the nearest correlation matrix problem:

minimize \[ \frac{1}{2} \| X - A \|_F^2 \]
subject to \[ [X]_{jj} = 1, \quad j = 1, \ldots, N, \]
\[ X - \eta I \succeq O, \tag{63} \]

where \( A \in \mathbb{S}^N \) is a constant matrix, and \( \eta \in \mathbb{R} \) is a positive constant. In the experiments, the elements of \( A \) are generated randomly from \([-1, 1]\) with \([A]_{jj} = 1\) for \( j = 1, \ldots, N \), and we set \( \eta := 10^{-3} \). We generate 10 instances of each problem in this manner and apply the algorithms to them. The obtained results are summarized in Tables 5, 6, 7 and 8 and Figs. 3 and 4, where each column in the tables represents the same as in Tables 1, 2, 3, and 4 and the figures also indicate the same in Figs. 1 and 2.

| \( N \) | 5     | 10   | 15   | 20   |
|--------|-------|------|------|------|
| Averaged \#ite | 13.7  | 11.0 | 12.7 | 11.8 |
| Averaged cputime(s) | 6.6   | 65.7 | 656.7 | 3051.7 |
| Averaged \( r(v^*) \) | 4.0e-07 | 4.2e-07 | 2.7e-07 | 4.1e-07 |
| Max of \( r(v^*) \) | 9.9e-07 | 9.7e-07 | 1.0e-06 | 9.6e-07 |
| Min of \( r(v^*) \) | 3.1e-08 | 1.5e-07 | 1.7e-08 | 2.3e-08 |
| Averaged max\( \|y^*\|, \|Z^*\|_F \) | 1.1e+00 | 1.6e+00 | 1.9e+00 | 2.2e+00 |
| Max of max\( \|y^*\|, \|Z^*\|_F \) | 1.1e+00 | 1.6e+00 | 1.9e+00 | 2.2e+00 |
| Min of max\( \|y^*\|, \|Z^*\|_F \) | 1.1e+00 | 1.6e+00 | 1.9e+00 | 2.2e+00 |
Algorithm 4 used the V-iterates in all the iterations for each problem-instance. From Tables 5, 6, 7 and 8, we see that both the algorithms successfully found KKT points for all the problem instances. However, there exist substantial difference between the computational times the algorithms spent. Indeed, from Averaged cputimes(s) of Tables 5 and 6 on problem (62), we see that the AL method was likely to find the KKT points faster than Algorithm 4. In contrast, according to Tables 7 and 8 on problem (63), Algorithm 4 is much faster than the AL method.

| Table 6 | Performance of the AL method on problem (62) |
|---------|---------------------------------------------|
| $N$     | 5   | 10  | 15  | 20  |
| Averaged #ite  | 6.0 | 6.0 | 7.3 | 6.2 |
| Averaged cputime(s) | 0.2 | 0.2 | 0.6 | 0.9 |
| Averaged $r(v^*)$ | 1.2e−07 | 2.4e−07 | 2.2e−07 | 2.7e−07 |
| Max of $r(v^*)$    | 1.2e−07 | 2.4e−07 | 9.4e−07 | 4.0e−07 |
| Min of $r(v^*)$    | 1.2e−07 | 2.4e−07 | 6.5e−08 | 4.6e−08 |
| Averaged max{$\|y^*\|,\|Z^*\|_F$} | 1.1e+00 | 1.6e+00 | 1.9e+00 | 2.2e+00 |
| Max of max{$\|y^*\|,\|Z^*\|_F$} | 1.1e+00 | 1.6e+00 | 1.9e+00 | 2.2e+00 |
| Min of max{$\|y^*\|,\|Z^*\|_F$} | 1.1e+00 | 1.6e+00 | 1.9e+00 | 2.2e+00 |

| Table 7 | Performance of Algorithm 4 on problem (63) |
|---------|---------------------------------------------|
| $N$     | 5   | 10  | 15  | 20  |
| Averaged #ite  | 4.2 | 5.6 | 5.0 | 5.8 |
| Averaged cputime(s) | 0.3 | 1.1 | 2.5 | 10.1 |
| Averaged $r(v^*)$ | 3.8e−07 | 1.6e−07 | 7.0e−08 | 5.2e−08 |
| Max of $r(v^*)$    | 5.5e−07 | 6.8e−07 | 4.1e−07 | 4.0e−07 |
| Min of $r(v^*)$    | 8.6e−08 | 3.5e−09 | 3.2e−09 | 6.5e−12 |
| Averaged max{$\|y^*\|,\|Z^*\|_F$} | 6.6e−01 | 2.9e+00 | 5.7e+00 | 8.5e+00 |
| Max of max{$\|y^*\|,\|Z^*\|_F$} | 1.5e+00 | 3.8e+00 | 6.5e+00 | 9.5e+00 |
| Min of max{$\|y^*\|,\|Z^*\|_F$} | 2.2e−01 | 2.1e+00 | 4.9e+00 | 7.6e+00 |

| Table 8 | Performance of the AL method on problem (63) |
|---------|---------------------------------------------|
| $N$     | 5   | 10  | 15  | 20  |
| Averaged #ite  | 9.4 | 12.5 | 14.7 | 16.5 |
| Averaged cputime(s) | 0.3 | 5.3 | 60.8 | 401.7 |
| Averaged $r(v^*)$ | 4.7e−07 | 4.9e−07 | 6.8e−07 | 6.6e−07 |
| Max of $r(v^*)$    | 9.3e−07 | 1.0e−06 | 9.9e−07 | 9.3e−07 |
| Min of $r(v^*)$    | 1.4e−07 | 2.4e−07 | 1.8e−07 | 4.6e−07 |
| Averaged max{$\|y^*\|,\|Z^*\|_F$} | 6.6e−01 | 2.9e+00 | 5.7e+00 | 8.5e+00 |
| Max of max{$\|y^*\|,\|Z^*\|_F$} | 1.5e+00 | 3.8e+00 | 6.5e+00 | 9.5e+00 |
| Min of max{$\|y^*\|,\|Z^*\|_F$} | 2.2e−01 | 2.1e+00 | 4.9e+00 | 7.6e+00 |
7 Concluding remarks

In this paper, we have proposed a stabilized SQSDP method (Algorithm 4) for NSDP (1). In the algorithm, we approximately solve stabilized QSDP subproblems, which are derived from the Lagrange minimax problem of the NSDP, to generate search directions. Those subproblems are always feasible and have global optima.
We integrated the V-, O-, M-, and F-iterates, which originate from the stabilized SQP method [11] for NLPs, into Algorithm 4. We showed that, under some mild assumptions, it globally converges to one of the following points: a stationary point concerning the feasibility, an AKKT point, and a TAKKT point. One remarkable point of that analysis is that we need no CQs or boundedness of the produced dual sequence. In the numerical experiments, we observed that Algorithm 4 could successfully solve

![Fig. 3 Histories of $r(v_k)$ generated by each method applied to problem (62)](image1)

![Fig. 4 Histories of $r(v_k)$ generated by each method applied to problem (63)](image2)
many test problems which the AL method failed to solve. From this fact, Algorithm 4 is promising in comparison with the AL method [3].

As a future work, it is an interesting direction to study superlinear convergence of Algorithm 4.

A Supplementary for the numerical experiments

In Appendix A, we give the existing augmented Lagrangian (AL) method [3].

Algorithm 1 The AL method

1. Select \( \varepsilon > 0, \tau \in (0,1), \gamma > 1, k_{\text{max}} \in \mathbb{N}, y_{\text{max}} > 0, \) and \( z_{\text{max}} > 0 \). Choose \((x_0, y_0, Z_0) \in \mathbb{V}\). Set \( \rho_0 > 0, \quad y_0 := y_0, \quad Z_0 := Z_0, \) and \( k := 0 \). \( \blacktriangleright \) Step 0
2. If \( k = k_{\text{max}}, \ v_k \leq \varepsilon, \) or the following conditions are satisfied, then stop.
   \[ \|\nabla_x L(v_k)\| \leq \varepsilon, \quad \|g(x_k)\| + \|[-X(x_k)]_+\| \leq \varepsilon, \quad \|U_k - S_k\|_F \leq \varepsilon, \quad \|\nabla f(x_k)\| \leq \varepsilon, \quad \lambda_j^{U_k} (\nabla f(x_k)) = 0, \quad \lambda_j^{S_k} (Z_k) \leq \varepsilon, \]
   where \( U_k \) and \( S_k \) are orthogonal matrices such that they diagonalize \( X(x_k) \) and \( Z_k \), respectively, that is, \( X(x_k) = U_k \text{diag}[\lambda_{j_1}^{U_k}(X(x_k)), \ldots, \lambda_{n}^{U_k}(X(x_k))]U_k^T \) and \( Z_k = S_k \text{diag}[\lambda_{j_1}^{S_k}(Z_k), \ldots, \lambda_{m}^{S_k}(Z_k)]S_k^T \). \( \blacktriangleright \) Step 1
3. Solve the following unconstrained optimization problem
   \[ \min_{x \in \mathbb{R}} f(x) + \frac{1}{2\rho_k} \|y_k - \rho_k g(x)\|^2 + \frac{1}{2\rho_k} \|Z_k - \rho_k X(x)\|_F^2, \]
   and find its approximate minimizer \( x_{k+1} \). \( \blacktriangleright \) Step 2
4. Update \( \rho_k \) by
   \[ \rho_{k+1} = \begin{cases} \rho_k & \text{if} \ u_{k+1} \leq \tau u_k, \\
\gamma \rho_k & \text{otherwise}, \end{cases} \]
   where \( u_k := \max \{\|g(x_k)\|, \|V_k\|_F\} \). \( \blacktriangleright \) Step 3
5. Set
   \[ y_{k+1} := \frac{1}{\rho_k} Z_k - X(x_k) , \quad Z_{k+1} := \frac{1}{\rho_k} \Pi_{D}(Z_{k+1}) , \]
   \[ y_{k+1} := \Pi_{C}(y_{k+1}) , \quad Z_{k+1} := \Pi_{D}(Z_{k+1}) , \]
   where \( C = \{y \in \mathbb{R}^m : -y_{\text{max}} \leq y \leq y_{\text{max}} \} \) and \( D = \{Z \in \mathbb{S}^l : O \preceq Z \preceq Z_{\text{max}} I\} \). \( \blacktriangleright \) Step 4
6. Set \( k := k + 1, \) and go back to Step 1. \( \blacktriangleright \) Step 5

Acknowledgements The authors would like to thank Professor Ellen Hidemi Fukuda for her helpful comments. We would also like to express our gratitude to the three reviewers for their valuable suggestions. This research was in part supported by the Japan Society for the Promotion of Science KAKENHI for 20K19748, 20H04145, and 21K17709.

Data availability Not applicable.

References

1. Andreani, R., Fukuda, E.H., Haeser, G., Santos, D.O., Secchin, L.D.: Optimality conditions for nonlinear second-order cone programming and symmetric cone programming. Optimization (2019). http://www.optimization-online.org/DB_HTML/2019/10/7436.html
2. Andreani, R., Gómez, W., Haeser, G., Mito, L.M., Ramos, A.: On optimality conditions for nonlinear conic programming. Technical report. http://www.optimization-online.org/DB_HTML/2020/03/7660.html (2020)
3. Andreani, R., Haeser, G., Viana, D.S.: Optimality conditions and global convergence for nonlinear semidefinite programming. Math. Program. 180, 203–235 (2020)
4. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. Acta Numer. 4, 1–51 (1995)
5. Correa, R., Ramírez, H.: A global algorithm for nonlinear semidefinite programming. SIAM J. Optim. 15(1), 303–318 (2004)
6. Fares, B., Noll, D., Apkarian, P.: Robust control via sequential semidefinite programming. SIAM J. Control. Optim. 40(6), 1791–1820 (2002)
7. Freund, R.W., Jarre, F., Vogelbusch, C.H.: Nonlinear semidefinite programming: sensitivity, convergence, and an application in passive reduced-order modeling. Math. Program. Ser. B 109(2–3), 581–611 (2007)
8. Fukuda, E.H., Lourenço, B.F.: Exact augmented Lagrangian functions for nonlinear semidefinite programming. Comput. Optim. Appl. 71(2), 457–482 (2018)
9. Gill, P.E., Kungurtsev, V., Robinson, D.P.: A stabilized SQP method: global convergence. IMA J. Numer. Anal. 37(1), 407–443 (2017)
10. Gill, P.E., Kungurtsev, V., Robinson, D.P.: A stabilized SQP method: superlinear convergence. Math. Program. 163, 369–410 (2017)
11. Gill, P.E., Robinson, D.P.: A globally convergent stabilized SQP method. SIAM J. Optim. 23(4), 1983–2010 (2013)
12. Gómez, W., Ramírez, H.: A filter algorithm for nonlinear semidefinite programming. Comput. Appl. Math. 29(2), 297–328 (2010)
13. Gruber, G., Rendl, F.: Computational experience with ill-posed problems in semidefinite programming. Comput. Optim. Appl. 21, 201–212 (2002)
14. Hager, W.W.: Stabilized sequential quadratic programming. Comput. Optim. Appl. 12(1–3), 253–273 (1999)
15. Huang, X.X., Teo, K.L., Yang, X.Q.: Approximate augmented Lagrangian functions and nonlinear semidefinite programs. Acta Math. Sin. Engl. Ser. 22(5), 1283–1296 (2006)
16. Izmailov, A.F., Solodov, M.V.: A truncated SQP method based on inexact interior-point solutions of subproblems. SIAM J. Optim. 20(3), 2584–2613 (2010)
17. Izmailov, A.F., Solodov, M.V., Uskov, E.I.: Combining stabilized SQP with the augmented Lagrangian algorithm. Comput. Optim. Appl. 62(2), 405–429 (2015)
18. Izmailov, A.F., Uskov, E.I.: Subspace-stabilized sequential quadratic programming. Comput. Optim. Appl. 67, 129–154 (2017)
19. Jarre, F.: An interior method for nonconvex semidefinite programs. Optim. Eng. 1(4), 347–372 (2000)
20. Kanzow, C., Nagel, C., Kato, H., Fukushima, M.: Successive linearization methods for nonlinear semidefinite programs. Comput. Optim. Appl. 31(3), 251–273 (2005)
21. Kočvara, M., Stingl, M.: PENNON: a code for convex nonlinear and semidefinite programming. Optim. Methods Softw. 18(3), 317–333 (2003)
22. Leibfritz, F., Mostafa, E.M.E.: An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems. SIAM J. Optim. 12(4), 1048–1074 (2002)
23. Okuno, T.: Local convergence of primal-dual interior point methods for nonlinear semi-definite optimization using the family of Monteiro–Tsuchiya directions. https://arxiv.org/pdf/2009.03020.pdf (2020)
24. Stingl, M.: On the Solution of Nonlinear Semidefinite Programs by Augmented Lagrangian Methods. Ph.D. Thesis, Institute of Applied Mathematics II, Friedrich-Alexander University of Erlangen-Nuremberg (2005) (submitted)
25. Sun, D., Sun, J., Zhang, L.: The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming. Math. Program. Ser. A 114(2), 349–391 (2008)
26. Toh, K.C., Todd, M.J., Tütüncü, R.H.: SDPT3—a Matlab software package for semidefinite programming, version 1.3. Optim. Methods Softw. 11(1–4), 545–581 (1999)
27. Tütüncü, R.H., Toh, K.C., Todd, M.J.: Solving semidefinite-quadratic-linear programs using SDPT3. Math. Program. Ser. B 95(2), 189–217 (2003)
28. Wright, S.J.: Superlinear convergence of a stabilized SQP method to a degenerate solution. Comput. Optim. Appl. 11(3), 253–275 (1998)
29. Wu, H., Luo, H., Ding, X., Chen, G.: Global convergence of modified augmented Lagrangian methods for nonlinear semidefinite programming. Comput. Optim. Appl. 56(3), 531–558 (2013)
30. Yamakawa, Y., Yamashita, N.: A two-step primal-dual interior point method for nonlinear semidefinite programming problems and its superlinear convergence. J. Oper. Res. Soc. Jpn. 57(3–4), 105–127 (2014)
31. Yamakawa, Y., Yamashita, N.: A differentiable merit function for shifted perturbed Karush–Kuhn–Tucker conditions of the nonlinear semidefinite programming. Pac. J. Optim. 11(3), 557–579 (2015)
32. Yamashita, H., Yabe, H.: Local and superlinear convergence of a primal-dual interior point method for nonlinear semidefinite programming. Math. Program. Ser. A 132(1–2), 1–30 (2012)
33. Yamashita, H., H, Yabe: A survey of numerical methods for nonlinear semidefinite programming. J. Oper. Res. Soc. Jpn. 58(1), 24–60 (2015)
34. Yamashita, H., Yabe, H., Harada, K.: A primal-dual interior point method for nonlinear semidefinite programming. Math. Program. Ser. A 11(3), 557–579 (2015)
35. Yang, L., Yu., B.: A homotopy method for nonlinear semidefinite programming. Comput. Optim. Appl. 56, 81–96 (2013)
36. Zhang, S., Ang, J., Sun, J.: An alternating direction method for solving convex nonlinear semidefinite programming problem. Optimization 62(4), 527–543 (2013)
37. Zhao, Q., Chen, Z.: On the superlinear local convergence of a penalty-free method for nonlinear semidefinite programming. J. Comput. App. Math. 308(15), 1–19 (2016)
38. Zhao, Q., Chen, Z.W.: An SQP-type method with superlinear convergence for nonlinear semidefinite programming. Asia-Pac. J. Oper. Res. 35, 1850009 (2018)
39. Zhao, Q., Chen, Z.W.: A line search exact penalty method for nonlinear semidefinite programming. Comput. Optim. Appl. 75, 467–491 (2020)
40. Zhu, Z.B., Zhu, H.L.: A filter method for nonlinear semidefinite programming with global convergence. Acta Math. Sin. 30(10), 1810–1826 (2014)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.