An Inexact Proximal DC Algorithm with Sieving Strategy for Rank Constrained Least Squares Semidefinite Programming

Mingcai Ding 1 · Xiaoliang Song 1 · Bo Yu 1

Received: 30 September 2021 / Revised: 25 March 2022 / Accepted: 1 April 2022 / Published online: 30 April 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
In this paper, the optimization problem of supervised distance preserving projection (SDPP) for data dimensionality reduction is considered, which is equivalent to a rank constrained least squares semidefinite programming (RCLSSDP). Due to the combinatorial nature of rank function, the rank constrained optimization problems are NP-hard in most cases. In order to overcome the difficulties caused by rank constraint, a difference-of-convex (DC) regularization strategy is employed, then RCLSSDP is transferred into a DC programming. For solving the corresponding DC problem, an inexact proximal DC algorithm with sieving strategy (s-iPDCA) is proposed, whose subproblems are solved by an accelerated block coordinate descent method. The global convergence of the sequence generated by s-iPDCA is proved. To illustrate the efficiency of the proposed algorithm for solving RCLSSDP, s-iPDCA is compared with classical proximal DC algorithm, proximal gradient method, proximal gradient-DC algorithm and proximal DC algorithm with extrapolation by performing dimensionality reduction experiment on COIL-20 database. From the computation time and the quality of solution, the numerical results demonstrate that s-iPDCA outperforms other methods. Moreover, dimensionality reduction experiments for face recognition on ORL and YaleB databases demonstrate that rank constrained kernel SDPP is efficient and competitive when comparing with kernel semidefinite SDPP and kernel principal component analysis in terms of recognition accuracy.

Keywords Supervised distance preserving projection · Rank constraint · Inexact proximal difference-of-convex algorithm · Sieving strategy · Accelerated block coordinate descent · Data dimensionality reduction
1 Introduction

Dimensionality reduction plays a crucial role in handling high-dimensional data. The supervised distance preserving projection (SDPP) was proposed by Zhu et al. [1] for data dimensionality reduction, which minimizes the difference of local structure between projected input covariates and their corresponding responses. Given sample points \( \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X} \subset \mathbb{R}^d \) and their corresponding responses \( \{y_1, y_2, \ldots, y_n\} \subset \mathcal{Y} \subset \mathbb{R}^m \), the form of SDPP is as follows:

\[
\min_{P} \frac{1}{n} \sum_{i,j=1}^{n} G_{i,j} \left( \|P^T x_i - P^T x_j\|^2 - \|y_i - y_j\|^2 \right)^2,
\]

where \( G \) denotes a graph matrix, whose element \( G_{i,j} \) equals to 1 if \( x_j \in \mathcal{N}(x_i) \), otherwise, equals to 0. \( \mathcal{N}(x_i) \) is a neighborhood of \( x_i \). \( P \in \mathbb{R}^{d \times r} (r \ll d) \) denotes the protection matrix. \( \|P^T x_i - P^T x_j\| \) and \( \|y_i - y_j\| \) are the pairwise distances among the projected input covariates and distances among the responses, respectively. The SDPP model can be widely used in machine learning, including visualization, regression and classification of high dimensional data [1].

In order to solve problem (1), the semidefinite programming (SDP) relaxation of SDPP was obtained by introducing a new variable \( U = PP^\top \in S_+^d \) [1, 2], which can be formulated into a least squares SDP:

\[
\min_{U \in S_+^d} J(U) = \frac{1}{n} \|A(U) - b\|^2,
\]

where \( A : S_+^d \rightarrow \mathbb{R}^p \) is a linear operator that can be explicitly represented as

\[
A(U) = \left[ (A_1, U), (A_2, U), \ldots, (A_p, U) \right]^\top, \quad A_i \in S_+^d, i = 1, 2, \ldots, p
\]

and \( p = \sum_{i,j=1}^{n} G_{i,j} \). In [2], to preserve the global structure of input covariates in the projected space, the total variance of projected covariates \( \sum_{i=1}^{n} \|P^T x_i\|^2 \) was incorporated to the objective function of (1), and the SDP relaxation of the resulting problem was formulated as

\[
\min_{U \in S_+^d} \frac{\lambda}{n} \|A(U) - b\|^2 - \langle \Psi, U \rangle,
\]

where \( \Psi = \sum_{i=1}^{n} x_i x_i^\top \) and \( \lambda \) is a penalty parameter. In fact, the hidden rank constraint, i.e., \( \text{rank}(U) \leq r \), has been ignored in the SDP relaxation problems [1, 2]. Let \( \hat{U}^* \in S_+^d \) be the optimal solution of (2), generally, \( \text{rank}(\hat{U}^*) > r \) holds, which means that \( \hat{U}^* \) does not satisfy the rank constraint. Then the projection matrix \( \hat{P}^* \) obtained by performing square root decomposition, or spectral decomposition on \( \hat{U}^* \) is not the optimal solution of (1), which leads to poor dimensionality reduction performance. To address this issue, an intuitive idea is to consider the rank constraint in the model. By adding the rank constraint to (2), a new rank constrained least squares semidefinite programming (RCLSSDP) is obtained as follows:

\[
\min_{U \in S_+^d} J(U) = \frac{1}{n} \|A(U) - b\|^2
\]

\[
\text{s.t. } \text{rank}(U) \leq r.
\]

\[1\] The process to obtain (2) is shown in Appendix 1.
It is noted that problem (3) is equivalent to the original SDPP (1).

The RCLSSDP covers many important applications such as nearest correlation matrix [3–5], sensor network localization [6, 7] and classical multidimensional scaling [8]. Due to the combinatorial nature of rank function, the rank constrained matrix optimization problems are NP-hard in most cases [9]. Most of the existing methods solved the relaxed or simplified rank constrained optimization problems by introducing a convex approximate function and minimized the relaxed convex problem instead of the exact rank function over a convex set. One of the most commonly used convex relaxation methods for rank constrained optimization is the nuclear norm minimization [10, 11]:

\[
\min_{U} f(U) + \lambda \|U\|_* \quad \text{s.t. } U \in \mathbb{R}^{p_1 \times p_2},
\]

where \( \|U\|_* = \sum_{i=1}^{\min(p_1, p_2)} \sigma_i(U) \) is the nuclear norm of \( U \) and \( \lambda \) is a penalty parameter. Some efficient algorithms have been proposed to solve (4), such as proximal gradient descent method [12] and proximal point method [13]. Another widely considered convex relaxation method for the rank constrained optimization problem is called the max norm minimization [14, 15], which uses max norm as regularizer. Although these convex relaxation methods can generate a low rank solution, even a minimum rank solution, a convex relaxation function cannot represent the exact rank function and the rank constrained solution cannot be guaranteed. Furthermore, the penalty parameter needed to be carefully tuned to ensure that the optimal solution of penalty problem satisfies low rank constraint and its related fitting term is small enough.

In addition to the convex relaxation methods, a nonconvex method to deal with the rank constraint has been investigated [16–18]. In this method, the rank constraint \( \text{rank}(U) \leq r \) was equivalently converted into an equality constraint \( \|U\|_* - \|U\|_{(r)} = 0 \) with nuclear norm function \( \|U\|_* \) and Ky-Fan \( r \)-norm function \( \|U\|_{(r)} \). Then an exact penalty approach was used to penalize the equality constraint into the objective function with a chosen penalty parameter. Since the nuclear norm function \( \|U\|_* \) and Ky-Fan \( r \)-norm function \( \|U\|_{(r)} \) are both convex, the rank constrained optimization problem was converted into a difference-of-convex (DC) programming [3, 4], which can be solved under the DC approach framework [19]. In recent decades, many algorithms for solving DC programming were proposed, including classical DC algorithm (DCA) [19–22], proximal DCA [23], DCA with proximal bundle [24, 25] and proximal DCA with extrapolation [26, 27]. In [18], a semi-proximal DCA was proposed to solve the rank constrained SDP, and the subproblems of their algorithm were solved by an efficient majorized semismooth Newton-CG augmented Lagrangian method based on the software package SDPNAL+ [28]. This technique performed well in some nonconvex problems, e.g. quadratic assignment problem, standard quadratic programming and minimum-cut graph tri-partitioning problem. However, in their algorithms, subproblems needed to be solved accurately or to high accuracy, which affects the efficiency of the algorithm.

For large-scale DC programming \( \min_{x} f_1(x) - f_2(x) \), it may be impossible or unnecessary to solve the convex subproblems of the DCA exactly. Generally, a large amount of computation is required to obtain a high-accuracy solution, so it is time-consuming to solve the subproblems to high accuracy. In particular, it is unnecessary to solve the subproblem to high accuracy at the initial iteration of the DCA. To overcome this issue, a popular approach is to consider the inexactness of solution when solving the subproblem of DCA. The subproblem of proximal DCA in step \( k \) can be expressed as

\[
\min_{x} G(x) := f_1(x) - \langle w^k, x \rangle + \frac{\alpha}{2} \|x - x^k\|^2,
\]
where \( w^k \in \partial f_2(x^k) \). Let \( x^{k+1} \) be an inexact solution of \( \min_x G(x) \), then there exists an inexact term \( \Delta^{k+1} \in \partial G(x^{k+1}) \). In general inexact proximal DCA, the inexact solution \( x^{k+1} \) and the corresponding inexact term \( \Delta^{k+1} \) were required to satisfy \( \| \Delta^{k+1} \| \leq \eta_k \| x^{k+1} - x^k \| \) with a given constant \( \alpha > \eta_k > 0 \). Since \( \Delta^{k+1} \) is related to \( x^{k+1} \) implicitly, the algorithm for solving the subproblem may be unable to terminate. Thus this inexact strategy is difficult to implement in practical application, especially when the problem is nonsmooth. Therefore, it remains a challenging task to design an inexact proximal DCA that can guarantee the theoretical convergence and has good numerical performance for solving the large-scale DC programming.

Our main contributions in this paper are as follows.

(1) To efficiently solve the RCLSSDP, we propose an algorithm named inexact proximal DCA with sieving strategy (s-iPDCA). A simple and numerically implementable inexact strategy is proposed to inexactly solve the subproblem of s-iPDCA. In s-iPDCA, \( \| \Delta^{k+1} \|_F \leq \epsilon_{k+1} \) is used as the termination condition for solving subproblems. The non-negative sequence \( \{ \epsilon_k \} \) is only required to satisfy \( \lim_{k \to \infty} \epsilon_k = 0 \). This inexact condition ensures that the algorithm for solving the subproblem is always terminable. Moreover, we employ a sieving strategy to ensure the convergence of s-iPDCA. The sequence generated by s-iPDCA globally converges to a stationary point of the corresponding DC problem.

(2) Based on the special structure of the dual subproblem of s-iPDCA, we use an effective accelerated block coordinate descent (ABCD) method to solve the subproblem of s-iPDCA from dual.

(3) In ABCD and s-iPDCA, low rank structure of the solution is fully utilized to reduce the storage and computation cost and improve the efficiency of the whole algorithm framework.

1.1 Notations

Below are some common notations to be used in this paper. We use \( S_q \) to denote the linear subspace of all \( q \times q \) real symmetric matrices and use \( S_q^+ (S_q^-) \) to denote the positive (negative) semidefinite matrix cone. Let \( \| \cdot \|_F \) denote the Frobenius norm of matrices. \( \| \cdot \| \) is used to represent the \( l_2 \) norm of vectors and matrices, respectively. For a given index set \( \mathcal{L} \subset \{ 1, \cdots, q \} \), \( |\mathcal{L}| \) denotes the size of \( \mathcal{L} \). The identity matrix is expressed as \( I \) and the identity operator is expressed as \( \mathcal{I} \). vec(\( \cdot \)) is used to denote the vectorization of matrices. dist(\( \cdot, \cdot \)) is used to denote the distance between of two sets (points). We denote the Ky-Fan \( k \)-norm of matrix \( U \) as \( \| U \|_{(k)} = \sum_{i=1}^{k} \sigma_i(U) \), where \( \sigma_i(U) \) is the \( i \)-th largest singular value of \( U \). \( \langle U, V \rangle = \sum_{i,j=1}^{q} U_{ij}V_{ij} \) is used to denote the inner product between square matrix \( V \) and square matrix \( V' \).

Let \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \) be the eigenvalues of \( U \in S_q \) being arranged in nonincreasing order and \( q_i \) be the eigenvector of \( U \) corresponding to \( \lambda_i \). We denote \( \mu := \{ i | \lambda_i > 0 \} \) and \( \nu := \{ i | \lambda_i < 0 \} \) as the index sets of positive eigenvalues and negative eigenvalues, respectively. Then the spectral decomposition of \( U \) can be given as

\[
U = QAQ^T, \quad A = \begin{bmatrix} A_{\mu} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & A_{\nu} \end{bmatrix}.
\]
where $\Lambda$ is the diagonal matrix whose $i$-th diagonal entry is $\lambda_i$, and $Q$ is the orthogonal matrix whose $i$-th column vector is $q_i$. Evidently, the positive (negative) semidefinite matrix cone projection of $U$ is represented as

$$\Pi_{S^+}(U) = Q_{\mu} \Lambda_{\mu} Q_{\mu}^T \quad \left( \Pi_{S^-}(U) = Q_{\nu} \Lambda_{\nu} Q_{\nu}^T \right),$$

where $Q_{\mu} \in \mathbb{R}^{q \times |\mu|}$ ($Q_{\nu} \in \mathbb{R}^{q \times |\nu|}$) is the sub-matrix of $Q$ indexed by $\mu$ ($\nu$).

Due to $A_i = \tau_i \tau_i^T$, $\tau_i \in \mathbb{R}^d$, $i = 1, \cdots, p$ (see Appendix 1), then the matrix representation of the linear operator $A$ can be formulated as

$$A = \left[ \text{vec} \left( \tau_1 \tau_1^T \right), \text{vec} \left( \tau_2 \tau_2^T \right), \cdots, \text{vec} \left( \tau_p \tau_p^T \right) \right]^T \in \mathbb{R}^{p \times d^2},$$

then we get that for each $U \in S^d$, $A(U) = \text{Vec}(U)$. For any $z \in \mathbb{R}^p$, $A^*(z) = \sum_{i=1}^p z_i A_i = \sum_{i=1}^p z_i \tau_i \tau_i^T$, where $z_i$ denotes the $i$-th entry of $z$, then the computation cost of $A^*(z)$ is $O(d^2 + d^2p)$. For any $z \in \mathbb{R}^p$, $A A^*(z)$ can be computed as

$$\begin{aligned}
A A^*(z) &= A \left( \sum_{i=1}^p z_i \tau_i \tau_i^T \right) = \sum_{i=1}^p z_i A \left( \tau_i \tau_i^T \right) \\
&= \sum_{i=1}^p z_i \left[ \langle \tau_1 \tau_1^T, \tau_i \tau_i^T \rangle, \langle \tau_1 \tau_2^T, \tau_i \tau_i^T \rangle, \cdots, \langle \tau_1 \tau_p^T, \tau_i \tau_i^T \rangle \right]^T \\
&= \sum_{i=1}^p \left[ \left( \tau_1 \tau_1^T \right)^2, \left( \tau_2 \tau_2^T \right)^2, \cdots, \left( \tau_p \tau_p^T \right)^2 \right] \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_p \end{bmatrix}.
\end{aligned}$$

Hence, the matrix representation of $A A^*$ can be formulated as

$$C := \begin{bmatrix}
\left( \tau_1 \tau_1^T \right)^2, \left( \tau_2 \tau_2^T \right)^2, \cdots, \left( \tau_p \tau_p^T \right)^2 \\
\left( \tau_1 \tau_2^T \right)^2, \left( \tau_2 \tau_2^T \right)^2, \cdots, \left( \tau_p \tau_p^T \right)^2 \\
\vdots & \vdots & \ddots & \vdots \\
\left( \tau_1 \tau_p^T \right)^2, \left( \tau_2 \tau_p^T \right)^2, \cdots, \left( \tau_p \tau_p^T \right)^2
\end{bmatrix} \in \mathbb{R}^{p \times p}. \quad (5)$$

### 2 Rank Constrained Least Squares Semidefinite Programming

As discussed above, the rank constraint, i.e., $\text{rank}(U) \leq r$, is equivalent to a DC constraint $\|U\|_* - \|U\|_{(r)} = 0$. Then the RCLSSDP in (3) can be reformulated as the equivalent form:

$$\begin{aligned}
\min_{U \in S^d_r} & \frac{1}{n} \|A(U) - b\|^2 \\
\text{s.t.} & \langle U, I \rangle - \|U\|_{(r)} = 0,
\end{aligned} \quad (6)$$
where \( \|U\|_r = \langle U, I \rangle \) is due to \( U \in S^d_+ \). Even if the problem in (3) is converted to the above form, the difficulty caused by rank constraint is still not eliminated. To address this problem, we employ an exact penalty approach. For a given penalty parameter \( c \), by penalizing the equality constraint in (6) into the objective function, we obtain a least squares SDP with DC regularization term as follows:

\[
\min_{U \in S^d_+} J_c(U) = \frac{1}{n} \|A(U) - b\|^2 + c \left( \langle U, I \rangle - \|U\|_r \right).
\] (7)

For the penalized problem in (7), we have the following conclusions.

Let \( U_c^* \) be a global optimal solution of (7), \( U^* \) be a global optimal solution of (6) and \( \hat{U}^* \) be a global optimal solution of (2).

**Proposition 1** If \( \text{rank}(U_c^*) \leq r \), then \( U_c^* \) is a global optimal solution of (6).

**Proof** For the details of proof, one can refer to [4, Proposition 3.1] and [29]. \( \square \)

**Proposition 2** Let \( \epsilon > 0 \) be a given constant and \( U_r \) be a feasible solution of (6). Assume that \( c > 0 \) is a chosen constant such that \( (J(U_r) - J(\hat{U}^*)) / c \leq \epsilon \). Then

\[
\langle U_c^*, I \rangle - \|U_c^*\|_{(r)} \leq \epsilon \quad \text{and} \quad J_c(U_c^*) \leq J_c(U^*) \leq J(U^*) = \overline{J}.
\] (8)

**Proof** For the details of proof, one can refer to [4, Proposition 3.2] and [29]. \( \square \)

From Proposition 2, it is easy to see that an \( \epsilon \)-optimal solution to (6) in the sense of (8) is guaranteed by solving (7) with a chosen penalty parameter \( c \). This provides the rationale to replace the rank constraint in (6) by the penalty function \( c \left( \langle U, I \rangle - \|U\|_r \right) \).

By assuming that the objective function is global Lipschitz continuous, Bi et al. [16] and Liu et al. [30] have studied the sufficient condition of penalty term to be exact for some rank constrained optimization problems. Hence, their inexact penalty parameter is based on the global Lipschitz continuous constant. However, in this paper, the objective function \( J(U) = \frac{1}{n} \|A(U) - b\|^2 \) is not global Lipschitz continuous on \( S^d_+ \). Thus the exact penalty results in [16, 30] cannot be directly applied to (7). Therefore, we give the exact penalty property of (7) with respect to (6) as below.

Set \( \mathcal{R} = \{U \in S^d : \|U\|_r = \|U\|_r = 0 \} \), then the feasible set of (6) can be denoted as \( \mathcal{F} = \mathcal{R} \cap S^d_+ \). For any \( U \in S^d_+ \), by assuming that \( U \) has the spectral decomposition of the form \( \sum_{i=1}^d \lambda_i q_i q_i^T \), it holds that \( \Pi_{\mathcal{R}}(U) = \sum_{i=1}^r \lambda_i q_i q_i^T \in \mathcal{F} \). For a given \( \epsilon > 0 \), let \( \overline{c} \) be the chosen constant such that

\[
(J(\Pi_{\mathcal{R}}(\hat{U}^*))) - J(\hat{U}^*)) / \overline{c} \leq \epsilon.
\]

Then we give an exact penalty parameter as follows:

\[
\hat{c} = \max \left( \overline{c}, \frac{1}{n} \|A\|_F^2 \epsilon + \frac{2 \sqrt{n} T}{n} \|A\|_F \right)
\] (9)

where \( A \) is the matrix representation of the linear operator \( A \) and \( \overline{J} = J(U^*) \). Based the exact penalty parameter \( \hat{c} \) defined in (9), we have the following conclusions.

**Lemma 1** Let \( \hat{c} \) be the constant defined in (9). Then for any \( c > \hat{c} \), it holds that

\[
J(\Pi_{\mathcal{R}}(U_c^*)) \leq J(U_c^*) + \hat{c} \left( \langle U_c^*, I \rangle - \|U_c^*\|_{(r)} \right)
\] (10)
\textbf{Proof} For any $U \in S_+^d$, we have
\begin{align*}
\|U - \Pi_{\mathcal{R}}(U)\|_F \leq \|U - \Pi_{\mathcal{R}}(U)\|_\star = \sum_{i=r+1}^d \lambda_i \tag{11}
= \|U\|_\star - \|U\|_{(r)} = \langle U, I \rangle - \|U\|_{(r)},
\end{align*}
where the last equality is due to $U \in S_+^d$. From Proposition 2 and the definition of $\hat{c}$ in (9), we get that for any $c > \hat{c} \geq \bar{c}$,
\begin{align*}
\langle U_c^*, I \rangle - \|U_c^*\|_{(r)} \leq \epsilon \text{ and } J(U_c^*) = \frac{1}{n} \|A(U_c^*) - b\|^2 \leq J.
\end{align*}
Then we have
\begin{align*}
\|A(U_c^*) - b\| \leq \sqrt{nJ}.
\end{align*}
Evidently, it holds that
\begin{align*}
\|A(\Pi_{\mathcal{R}}(U_c^*)) - U_c^*\| = \|A(\Pi_{\mathcal{R}}(U_c^*)) - U_c^*\|_F \leq \|A\|_F \|\Pi_{\mathcal{R}}(U_c^*) - U_c^*\|_F. \tag{12}
\end{align*}
From (11), we have
\begin{align*}
\|\Pi_{\mathcal{R}}(U_c^*) - U_c^*\|_F \leq \langle U_c^*, I \rangle - \|U_c^*\|_{(r)}.
\end{align*}
Combining this with (12), $\langle U_c^*, I \rangle - \|U_c^*\|_{(r)} \leq \epsilon \text{ and } \|A(U_c^*) - b\| \leq \sqrt{nJ}$, it holds that
\begin{align*}
J(\Pi_{\mathcal{R}}(U_c^*)) = \frac{1}{n} \|A(\Pi_{\mathcal{R}}(U_c^*)) - b\|^2
= \frac{1}{n} \|A(U_c^*) - b + A(\Pi_{\mathcal{R}}(U_c^*) - U_c^*)\|^2
\leq J(U_c^*) + \frac{1}{n} \|A(\Pi_{\mathcal{R}}(U_c^*) - U_c^*)\|^2
+ \frac{2}{n} \|A(U_c^*) - b\| \|A(\Pi_{\mathcal{R}}(U_c^*) - U_c^*)\|
\leq J(U_c^*) + \frac{1}{n} \|A\|^2 \epsilon + \frac{2\sqrt{nJ}}{n} \|A\|_F \langle U_c^*, I \rangle - \|U_c^*\|_{(r)}
\leq J(U_c^*) + \hat{c} \langle U_c^*, I \rangle - \|U_c^*\|_{(r)}.
\end{align*}
where the last inequality is due to the definition of $\hat{c}$ in (9). This completes the proof. \hfill \Box

Based on Lemma 1, we have the following conclusion about the exact penalty property of (7) with respect to (6).

\textbf{Proposition 3} Let $\hat{c}$ be the constant defined in (9). Then for any $c > \hat{c}$, $U_c \in S_+^d$ is a global optimal solution of (7) if and only if $U_c$ is a global optimal solution of (6).

\textbf{Proof} For sufficiency, if $U_c$ is a global solution of (6), then $\langle U_c, I \rangle - \|U_c\|_{(r)} = 0$. From (10), we have
\begin{align*}
J_c(U_c^*) = J(U_c^*) + c \langle U_c^*, I \rangle - \|U_c^*\|_{(r)}
\geq J(U_c^*) + \hat{c} \langle U_c^*, I \rangle - \|U_c^*\|_{(r)}
\geq J(\Pi_{\mathcal{R}}(U_c^*)) \geq J(U_c) = J_c(U_c),
\end{align*}
where the first inequality is due to $c > \hat{c}$ and $\langle U^*_c, I \rangle - \|U^*_c\|_r \geq 0$, the third inequality follows from the optimality of $U_c$ for (6), the last equality is due to $\langle U_c, I \rangle - \|U_c\|_r = 0$. These inequalities implies that $J_c(U_c) \leq J_c(U^*_c)$. From the global optimality of $U^*_c$ for (7), we obtain that $U_c$ is a global optimal solution of (7) with penalty parameter $c$.

For necessity, if $U_c$ is a global optimal solution of (7) with penalty parameter $c$, from Proposition 1, we have that it suffices to prove rank$(U_c) \leq r$, i.e., $\langle U_c, I \rangle - \|U_c\|_r = 0$. Suppose for contradiction that $\langle U_c, I \rangle - \|U_c\|_r > 0$. From Lemma 1 and the global optimality of $U_c$, we can obtain that

$$J(\PiR(U_c)) \leq J(U_c) + \hat{c} \left( \langle U_c, I \rangle - \|U_c\|_r \right).$$

Hence,

$$J(\PiR(U_c)) \leq J(U_c) + \hat{c} \left( \langle U_c, I \rangle - \|U_c\|_r \right)$$

$$< J(U_c) + c \left( \langle U_c, I \rangle - \|U_c\|_r \right)$$

$$= J_c(U_c) \leq J_c(\PiR(U_c)) = J(\PiR(U_c)),$$

where the second inequality is due to $c > \hat{c}$ and $\langle U_c, I \rangle - \|U_c\|_r > 0$, the third inequality is due to optimality of $U_c$ for (7), the last equality is due to $\langle \PiR(U_c), I \rangle - \|\PiR(U_c)\|_r = 0$. These inequalities immediately lead to a contradiction $J(\PiR(U_c)) < J(\PiR(U_c))$. Thus, $\langle U_c, I \rangle - \|U_c\|_r = 0$ and $U_c$ is a global optimal solution of (7). This completes the proof.

Remark 1 Since it is difficult to obtain $U^*$ and $\hat{U}^*$, then it is not practical to directly compute $\hat{c}$ for general RCLSSDP. Based on Propositions 1 and 3, we employ a strategy of gradually increasing the penalty parameter $c$. When $c$ is large enough, i.e., $c > \hat{c}$, the global optimal solution of (7) with penalty parameter $c$ is also a global optimal solution of (6).

As discussed above, the objective function $J(U) = \frac{1}{n} \|A(U) - b\|^2$ is not global Lipschitz continuous on $S^d_+$, but it is local Lipschitz continuous, i.e., for any $U, \hat{U} \in B(\bar{U}, \varepsilon)$ with $\varepsilon > 0$, it holds that

$$|J(U) - J(\hat{U})| \leq L^F(U)\|U - \hat{U}\|_F,$$

where $L^F(U) = \max_{U \in B(\bar{U}, \varepsilon)} \frac{2}{n} \|A^*(A(U) - b)\|_F$. Based the local Lipschitz continuity of $J(U)$, we give the following local exact penalty property of (7).

Proposition 4 For any local minimizer $\bar{U} \in F$ of (6), let $L^F(U)$ be the local Lipschitz continuous constant of $J(U)$ on the neighbourhood $B(U, \varepsilon)$ with $\varepsilon > 0$. Then for any $c > L^F(\bar{U})$, $\bar{U}$ is also a local minimizer of (7) with penalty parameter $c$. Conversely, if $\bar{U} \in S^d_+$ is a local minimizer of (7) with penalty parameter $c > 0$ and rank$(\bar{U}) \leq r$, then $\bar{U}$ is also a local minimizer of (6).

Proof For the first part, if $\bar{U} \in F$ is a local minimizer of (6), then

$$J(\bar{U}) \leq J(U) \quad \text{for any} \quad U \in B(\bar{U}, \varepsilon) \cap F. \quad (13)$$

Notice that $\sum_{i=r+1}^d \lambda_i(U)$ is continuous and $\sum_{i=r+1}^d \lambda_i(U) = \langle \bar{U}, I \rangle - \|\bar{U}\|_r = 0$, where $\lambda_i(U)$ denotes the $i$-th largest eigenvalue of $U$. Hence, there must exist $\tilde{\varepsilon} > 0$ such that $\sum_{i=r+1}^d \lambda_i(U) < \frac{\varepsilon}{2}$ for any $U \in B(\bar{U}, \tilde{\varepsilon})$. Let $0 < \tilde{\varepsilon} \leq \min\left(\frac{\varepsilon}{2}, \varepsilon\right)$, then for any $U \in B(\bar{U}, \tilde{\varepsilon}) \cap S^d_+$, it holds that $\PiR(U) \in F$ and

$$\|U - \PiR(U)\|_F \leq \|U - \PiR(U)\|_2 = \sum_{i=r+1}^d \lambda_i(U) < \frac{\varepsilon}{2}.$$
Consequently, for any $U \in B(\tilde{U}, \bar{c}) \cap \mathcal{S}_+^d$,

$$\|U - \Pi_R(U)\|_F \leq \|U - \Pi_R(U)\|_F + \|\tilde{U} - U\|_F < \frac{\bar{c}}{2} + \bar{c} \leq \varepsilon,$$

where the last inequality is due to $0 < \bar{c} \leq \min\left(\frac{\bar{c}}{2}, \bar{c}\right)$. This implies that for any $U \in B(\tilde{U}, \bar{c}) \cap \mathcal{S}_+^d$, $\Pi_R(U) \in B(\tilde{U}, \bar{c}) \cap \mathcal{F}$. Combining this with (13), we obtain that for any $U \in B(\tilde{U}, \bar{c}) \cap \mathcal{S}_+^d$,

$$J_c(U) = J(\tilde{U}) \leq J(\Pi_R(U)) \leq J(U) + L_f^{\varepsilon}(\tilde{U})\|U - \Pi_R(U)\|_F \leq J(U) + c\left(\|U, I\|_I - \|U\|(r)\right) = J_c(U),$$

where the first equality follows from $\langle \tilde{U}, I \rangle - \|\tilde{U}\|(r) = 0$, the second inequality is due to the local Lipschitz continuity of $J(U)$ on $B(\tilde{U}, \bar{c})$, the last inequality is due to $c > L_f^{\varepsilon}(\tilde{U})$.

These inequalities imply that $\tilde{U}$ is a local minimizer of (7) with penalty parameter $c$.

For the second part, if $\tilde{U}$ is a local minimizer of (7) and rank$(\tilde{U}) \leq r$, then for any $U \in B(\tilde{U}, \bar{c}) \cap \mathcal{F}$, it holds that

$$J(\tilde{U}) = J_c(\tilde{U}) \leq J_c(U) = J(U),$$

where the first equality is due to $\langle \tilde{U}, I \rangle - \|\tilde{U}\|(r) = 0$, the second equality follows from $\langle U, I \rangle - \|U\|(r) = 0$. This implies that $\tilde{U}$ is a local minimizer of (6). This completes the proof. \hfill \square

### 3 Inexact Proximal DC Algorithm with Sieving Strategy for (7)

For a fixed penalty parameter $c$, the penalized problem in (7) can be formulated into the following standard DC form:

$$\min_{U \in \mathcal{S}_+^d} J_c(U) = \frac{1}{n} \sum_{i=1}^{n} f_c(U) - \|A(U) - b\|^2 + c\langle U, I \rangle - c\|U\|(r). \quad (14)$$

Let us first briefly review the classical proximal DC algorithm (PDCA) for solving the DC problem in (14). The details are shown in Algorithm 1.

**Algorithm 1 Proximal DC algorithm for (14)**

**Step 0.** Give $c > 0$, tolerance error $\varepsilon \geq 0$ and proximal parameter $\alpha > 0$. Initialize $U^0 \in \mathcal{S}_+^d$. Choose $W^0 \in \partial g_c(U^0)$. Set $k = 0$.

**Step 1.** Compute $U^{k+1}$ by solving the following subproblem:

$$\min_{U} G_c^{k}(U) := f_c(U) - \langle U, W^k \rangle + \delta_{\mathcal{S}_+^d}(U) + \frac{\alpha}{2} \|U - U^k\|_F^2. \quad (15)$$

**Step 2.** If $\|U^{k+1} - U^k\|_F \leq \varepsilon$, stop and return $U^{k+1}$.

**Step 3.** Choose $W^{k+1} \in \partial g_c(U^{k+1})$, set $k \leftarrow k + 1$, go to Step 1.
Remark 2 Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ be the eigenvalues of $U^{k+1}$ being arranged in non-increasing order, then $U^{k+1}$ has spectral decomposition $U^{k+1} = \sum_{i=1}^{d} \lambda_i q_i q_i^\top$, where $q_i$ is the eigenvector of $U^{k+1}$ corresponding to the eigenvalue $\lambda_i$. Then the subgradient $W^{k+1} \in \partial g_c(U^{k+1})$ is chosen as $W^{k+1} = c \sum_{i=1}^{r} q_i q_i^\top$.

Since the convex subproblem in (15) has no closed-form solution, it is impossible and unnecessary to solve this subproblem exactly in practice. Especially in the initial stage of PDCA, we only need to solve the subproblem to a low or medium precision. Hence, it is necessary to take the inexactness of the solution of the subproblems into account. Thus, we inexactly solve (15), i.e.,

$$U^{k+1} \approx \min_U G_c^k(U).$$

Then there exists an inexact term $\Delta^{k+1}$ such that $\Delta^{k+1} \in \partial G_c^k(U^{k+1})$.

Remark 3 Although the inexact term $\Delta^{k+1}$ is unknown before computing the inexact solution $U^{k+1}$, we can say that equivalently, $U^{k+1}$ is the solution of the following minimization problem:

$$\min_U G_c^k(U) - \langle \Delta^{k+1}, U \rangle. \tag{16}$$

In order to design an efficient inexact PDCA, we need to deal with the following three issues: (1) finding an inexact strategy to ensure the theoretical convergence of the corresponding inexact PDCA; (2) giving an efficient algorithm to solve the subproblems of the inexact PDCA; (3) finding a strategy to compute or estimate the inexact term $\Delta^{k+1}$ so that the inexact strategy can be numerically implemented.

### 3.1 An Effective Inexact Strategy and Algorithm Framework

For a given $\epsilon_{k+1} > 0$, if

$$\text{dist}\left(0, \partial G_c^k(U^{k+1})\right) \leq \|\Delta^{k+1}\|_F \leq \epsilon_{k+1},$$

then we say that $U^{k+1}$ is an $\epsilon_{k+1}$-inexact solution of (15). In traditional inexact algorithm [31, 32], the sequence $\{\epsilon_k\}$ was assumed to satisfy the condition $\sum_{k=0}^{\infty} \epsilon_k < \infty$. Although this inexact strategy is numerical implementable, it cannot guarantee the convergence of the corresponding inexact PDCA. To ensure the convergence of inexact PDCA for solving (14), an inexact condition similar to that of [23, 33] can be given as follows:

$$\text{dist}\left(0, \partial G_c^k(U^{k+1})\right) \leq \|\Delta^{k+1}\|_F \leq \eta_k \|U^{k+1} - U^k\|_F \tag{17}$$

with $0 \leq \eta_k \leq \alpha$. The inexact PDCA for solving (14) is shown in Algorithm 2. However, since the inexact term $\Delta^{k+1}$ is related to $U^{k+1}$ implicitly, then the condition (17) may not hold even if the subproblem of the inexact PDCA is solved to the higher accuracy. As a result, the algorithm for solving the subproblem may be unable to terminate. Thus, the inexact condition in (17) is not a good termination criterion for solving subproblem.

To address the issues caused by above two inexact strategies, we first use $\|\Delta^{k+1}\|_F \leq \epsilon_{k+1}$ as the termination criterion for solving the subproblem in (15). The sequence $\{\epsilon_k\}$ is required to satisfy $\lim_{k \to \infty} \epsilon_k = 0$. Clearly, this termination criterion is simple, which can ensure that the algorithm for solving the subproblem is terminable. However, by only using this inexact strategy, the obtained inexact solution sequence $\{U^k\}$ may not converge...
only the iteration counter $k$

a post-processing on the inexact solution satisfying inexact condition $\parallel PDCA$ with sieving strategy (s-iPDCA), see Algorithm 3 for more details.

the sieving condition as the initial point of the next iteration. Therefore, we give an inexact inexact solution satisfying the sieving condition and take the inexact solution not satisfying criterion for solving subproblems. By using this post-processing technique, we retain the 'sieve' to perform a post-processing on the inexact solution rather than the termination
descent property of objective function. Thus we use a sieving condition similar to (17) as the ‘sieve’ to perform a post-processing on the inexact solution rather than the termination criterion for solving subproblems. By using this post-processing technique, we retain the inexact solution satisfying the sieving condition and take the inexact solution not satisfying the sieving condition as the initial point of the next iteration. Therefore, we give an inexact PDCA with sieving strategy (s-iPDCA), see Algorithm 3 for more details.

| Algorithm 2 Inexact PDCA for (14) |
|-----------------------------------|
| **Step 0.** Give $c > 0$, tolerance error $\varepsilon \geq 0$, proximal parameter $\alpha > 0$ and non-negative sequence $\{\eta_k\}$ with $0 \leq \eta_k \leq \alpha$. Initialize $U^0 \in S_c^d$. Choose $W^0 \in \partial g_c(U^0)$. Set $k = 0$. |
| **Step 1.** Compute $U^{k+1}$ by inexactly solve (15), $U^{k+1} \approx \arg \min_U G_c^k(U)$, such that the inexact condition in (17) holds. |
| **Step 2.** If $\parallel U^{k+1} - U^k \parallel F \leq \varepsilon$, stop and return $U^{k+1}$. |
| **Step 3.** Choose $W^{k+1} \in \partial g_c(U^{k+1})$, set $k \leftarrow k + 1$, go to **Step 1**. |

and $\{ J_c(U^k) \}$ may not decrease. To deal with this problem, we design a ‘sieve’ to perform a post-processing on the inexact solution satisfying inexact condition $\parallel \Delta^{k+1} \parallel F \leq \epsilon_{k+1}$. Notice that choosing a suitable ‘sieve’ is crucial to the convergence of the inexact PDCA. As mentioned above, the condition in (17) is not a good termination criterion for solving subproblems, but it can guarantee the convergence of the corresponding inexact PDCA and the descent property of objective function. Thus we use a sieving condition similar to (17) as the ‘sieve’ to perform a post-processing on the inexact solution rather than the termination criterion for solving subproblems. By using this post-processing technique, we retain the inexact solution satisfying the sieving condition and take the inexact solution not satisfying the sieving condition as the initial point of the next iteration. Therefore, we give an inexact PDCA with sieving strategy (s-iPDCA), see Algorithm 3 for more details.

| Algorithm 3 Inexact PDCA with sieving strategy for (14) |
|-----------------------------------|
| **Step 0.** Give $c > 0$, tolerance error $\varepsilon \geq 0$, non-negative monotone descent sequence $\{\epsilon_k\}$, proximal parameter $\alpha > 0$ and sieving parameter $\kappa \in (0, 1)$. Initialize $U^0 = V^0 \in S_c^d$. Choose $W^0 \in \partial g_c(U^0)$. Set $k = 0$. |
| **Step 1.** Compute $V^{k+1}$ by inexactly solve (15), $V^{k+1} \approx \arg \min_U G_c^k(U)$, such that the inexact condition $\parallel \Delta^{k+1} \parallel F \leq \epsilon_{k+1}$ holds. |
| **Step 2.** If $\parallel V^{k+1} - U^k \parallel F \leq \varepsilon$, stop and return $V^{k+1}$. |
| **Step 3.** If the sieving condition $\parallel \Delta^{k+1} \parallel F \leq (1 - \kappa)\alpha\parallel V^{k+1} - U^{k} \parallel F$ (18) holds, set $U^{k+1} : = V^{k+1}$ and choose $W^{k+1} \in \partial g_c(U^{k+1})$; otherwise, set $U^{k+1} : = U^{k}$ and $W^{k+1} : = W^{k}$. Set $k \leftarrow k + 1$, go to **Step 1**. |

**Remark 4** Let $V^{k+1}$ be an inexact solution satisfying the inexact condition $\parallel \Delta^{k+1} \parallel F \leq \epsilon_{k+1}$. If $V^{k+1}$ satisfies the sieving condition in (18), we say that a serious step is performed in s-iPDCA, otherwise, we say that a null step is performed. We call the inexact solution satisfying sieving condition the stability center. It should be noted that, as shown in Algorithm 3, when (18) doesn’t hold true, the stability center and the subgradient of $g_c$ remain unchanged, while only the iteration counter $k$ and inexact error bound $\epsilon_{k+1}$ are changed. The sieving parameter $\kappa \in (0, 1)$ is used to balance the efficiency of s-iPDCA and the inexactness of the solutions for (15).
Remark 5 When $\varepsilon$ is set to 0, the following two situations would occur: (1) only finite serious steps are performed, then infinite null steps are performed or the s-iPDCA terminates in finite steps; (2) infinite serious steps are performed. When $\varepsilon > 0$, to ensure that as many serious steps as possible are performed in Algorithm 3, one can adjust the descent speed of $\{\varepsilon_k\}$.

3.2 A Dual-Based Accelerated Block Coordinate Descent Method for (15)

The main computation of s-iPDCA is in solving subproblem (15). Hence, it is essential to employ an efficient method to solve (15) for the effectiveness of the whole algorithm. As shown in Appendix 1, the dual problem of (15) can be equivalently formulated as the following minimization problem:

$$\min_{z,Y} \frac{n}{4} \|z\|^2 + b^Tz + \delta_{S^d_+}(Y) + \frac{1}{2\alpha} \|A^*(z) - Y - \Phi^k_c\|_F^2,$$

where $\Phi^k_c = W^k - cI + \alpha U^k$. The KKT conditions for solving (19) are given as

$$\Pi_{S^d_+} \left( Y + \frac{1}{\alpha} \left( A^*(z) - Y - \Phi^k_c \right) \right) = Y \quad (20)$$

$$\frac{n}{2} z + b + \frac{1}{\alpha} A \left( A^*(z) - Y - \Phi^k_c \right) = 0 \quad (21)$$

Problem (19) belongs to a general class of unconstrained, multi-block convex optimization problems with coupled objective function. Thus, this problem can be solved efficiently by a two-block accelerated block coordinate descent (ABCD) method. We give an efficient ABCD method to inexactly solve (19), see Algorithm 4 for more details.

**Algorithm 4** Accelerated block coordinate descent method for (19)

**Step 0.** Initialize $\tilde{Y}^1 \in S^d_+$, acceleration factor $t_1 = 1$ and $z^0 \in \mathbb{R}^p$. Give inexact error bound $\zeta_k$ expressed in (36). Set $j = 1$.

**Step 1.** Update $z$:

$$z^j = \arg \min_z \frac{n}{4} \|z\|^2 + z^Tb + \frac{1}{2\alpha} \|A^*(z) - \tilde{Y}^j - \Phi^k_c\|_F^2. \quad (22)$$

**Step 2.** Update $Y$:

$$Y^j = \arg \min_Y \frac{1}{2\alpha} \|A^*(z^j) - Y - \Phi^k_c\|_F^2 + \delta_{S^d_+}(Y). \quad (23)$$

**Step 3.** If $\|\frac{n}{2} z^j + b + \frac{1}{\alpha} A(A^*(z^j) - Y^j - \Phi^k_c)\| \leq \zeta_k$, stop.

**Step 4.** Compute

$$t_{j+1} = \frac{1 + \sqrt{1 + 4t^2_j}}{2}, \quad \beta_j = \frac{t_{j-1}}{t_{j+1}}, \quad \tilde{Y}^{j+1} = Y^j + \beta_j(Y^j - Y^{j-1}).$$

Set $j \leftarrow j + 1$, go to Step 1.

For the $z$-subproblem of Algorithm 4 in $j$-th iteration, it is obvious that solving the subproblem is equivalent to solving the following system:

$$\frac{n}{2} z + b + \frac{1}{\alpha} A \left( A^*(z) - \tilde{Y}^j - \Phi^k_c \right) = 0. \quad (24)$$
By using preconditioned conjugate gradient (PCG) method, we can solve the above linear system efficiently, especially when its scale is large. For the $Y$-subproblem of Algorithm 4 in $j$-th iteration, it has a closed-form solution which can be computed by

$$Y^j = \arg \min_{Y} \delta_{S^d_+}(Y) + \frac{1}{2\alpha} \| A^*(z^j) - Y - \Phi^k_c \|_F^2$$

$$= \Pi_{S^d_+} \left( A^*(z^j) - \Phi^k_c \right),$$

where $\Pi_{S^d_+}$ denotes the orthogonal projection on $S^d_+$. From the relationship between primal variable and dual variable, we can obtain a feasible solution

$$U^{(j)} = \frac{1}{\alpha} \left( Y^j + \Phi^k_c - A^*(z^j) \right) = \frac{1}{\alpha} \Pi_{S^d_+} \left( \Phi^k_c - A^*(z^j) \right)$$

of (15) in each iteration of Algorithm 4. From (25), we have

$$\Pi_{S^d_+} \left( Y^j + \frac{1}{\alpha} \left( A^*(z^j) - Y^j - \Phi^k_c \right) \right) = \Pi_{S^d_+} \left( \Pi_{S^d_+} \left( A^*(z^j) - \Phi^k_c \right) + \frac{1}{\alpha} \Pi_{S^d_+} \left( A^*(z^j) - \Phi^k_c \right) \right) = Y^j.$$

Thus,

$$Y^j = \Pi_{S^d_+} \left( Y^j + \frac{1}{\alpha} \left( A^*(z^j) - Y^j - \Phi^k_c \right) \right)$$

holds in each iteration of Algorithm 4.

We measure the accuracy of an approximate optimal solution $(z^j, Y^j)$ for (19) by using the following residuals of KKT equations:

$$j := \Pi_{S^d_+} \left( Y^j + \frac{1}{\alpha} \left( A^*(z^j) - Y^j - \Phi^k_c \right) \right) - Y^j$$

and

$$\gamma^j := \frac{n}{2} \varepsilon^j + b + \frac{1}{\alpha} A \left( A^*(z^j) - Y^j - \Phi^k_c \right).$$

Let $\zeta_k > 0$ be a given accuracy tolerance, then we terminate Algorithm 4 when $\max(\|A^j\|, \|j\|_F) \leq \zeta_k$. From (27), we have $j = 0$, which means that the KKT condition in (20) exactly holds in each iteration of Algorithm 4. Then we do not need to compute the positive semidefinite matrix cone projection in (20) any more, which saves a lot of computation when the matrix dimension $d$ is large. Consequently, we use $\|\gamma^j\| \leq \zeta_k$ as the termination condition of Algorithm 4.

### 3.3 An Efficient Strategy for Estimating the Inexact Term

Since Algorithm 3 is a type of inexact method, then we need to check the inexact condition $\|\Delta^{k+1}\|_F \leq \epsilon_{k+1}$ and the sieving condition in (18) in each iteration, thus it is important to give an efficient strategy to compute or estimate $\Delta^{k+1}$ for the whole s-iPDCA.

Let

$$F^k_c(U) = f_c(U) - \langle U, W^k \rangle + \frac{\alpha}{2} \| U - U^k \|_F^2,$$
then we have $G_c^k(U) = F_c^k(U) + \delta S_+(U)$, and $F_c^k(U)$ is the smooth part of $G_c^k(U)$ with gradient

$$\nabla F_c^k(U) = \frac{2}{n} A^T (A(U) - b) - \Phi_c^k + \alpha U,$$

where $\Phi_c^k = W^k - cI + \alpha U^k$. If we obtain an inexact solution $U_f \in S^d_+$ by solving $\min_U G_c^k(U)$, then there exists an inexact term $\Delta_f$ such that

$$\Delta_f \in \nabla F_c^k(U_f) + \partial \delta S_+(U_f).$$

(29)

According to the Second Prox Theorem [34,Theorem 6.39], (29) is equivalent to

$$U_f = \Pi_{S^d_+} \left( U_f - \nabla F_c^k(U_f) + \Delta_f \right).$$

(30)

Since $\delta S_+$ is nonsmooth, it is impossible to directly obtain the inexact term $\Delta_f$ from (29) and (30). To address this issue, we introduce an auxiliary variable $\tilde{U}_f$ defined as

$$\tilde{U}_f := \Pi_{S^d_+} \left( U_f - \nabla F_c^k(U_f) \right).$$

(31)

Then (31) can be rewritten as

$$\tilde{U}_f = \Pi_{S^d_+} \left( \tilde{U}_f - \nabla F_c^k(\tilde{U}_f) + \tilde{\Delta}_f \right),$$

(32)

where

$$\tilde{\Delta}_f = U_f - \tilde{U}_f - \nabla F_c^k(U_f) + \nabla F_c^k(\tilde{U}_f).$$

(33)

Similarly, (32) is equivalent to

$$\tilde{\Delta}_f \in \nabla F_c^k(\tilde{U}_f) + \partial \delta S_+(\tilde{U}_f),$$

which implies that $\tilde{U}_f$ is an inexact solution of (15) with inexact term $\tilde{\Delta}_f$. Then we have

$$\tilde{U}_f = \arg \min_U F_c^k(U) + \delta S_+(U) - \langle U, \tilde{\Delta}_f \rangle.$$

Therefore, for a feasible solution $U_f$, the inexact solution $\tilde{U}_f$ and inexact term $\tilde{\Delta}_f$ can be obtained by using the expressions in (31) and (33), respectively. By using this technique, we give the following two strategies to compute or estimate $\Delta^{k+1}$.

1. At the $j$-th iteration of Algorithm 4, we can obtain a feasible solution $U^{(j)}$ defined in (26). By using the expressions in (31) and (33), we can compute the inexact solution $\tilde{U}^{(j)}$ and inexact term $\tilde{\Delta}^{(j)} \in \partial G_c^k(\tilde{U}^{(j)})$. If $\|\tilde{\Delta}^{(j)}\| \leq \epsilon_{k+1}$, we can let $V^{k+1} := \tilde{U}^{(j)}$ and $\Delta^{k+1} := \tilde{\Delta}^{(j)}$, which means that $V^{k+1}$ is an inexact solution to subproblem $\min_U G_c^k(U)$ satisfying inexact condition $\|\Delta^{k+1}\|_F \leq \epsilon_{k+1}$. Evidently, by using this strategy to compute $V^{k+1}$ and $\Delta^{k+1}$, we need to compute $U^{(j)}$ and $\tilde{\Delta}^{(j)}$ in each iteration of Algorithm 4, which means that we need to compute the positive semidefinite matrix cone projection defined in (31) one more time. Thus this strategy is not very efficient in practical numerical experiments.

2. It is noted that Algorithm 4 is a dual algorithm. We use the KKT residual $y^j$ of the dual problem to terminate Algorithm 4 in practical numerical experiments. If we can give an upper bound estimate of $\|\tilde{\Delta}^{(j)}\|_F$ from $\|y^j\|$ according to the relationship between $\tilde{\Delta}^{(j)}$ and $y^j$, we only need to check whether this upper bound is smaller than the given inexact bound $\epsilon_{k+1}$, instead of directly computing $\|\tilde{\Delta}^{(j)}\|_F$ in each iteration of Algorithm 4. This will further improve the efficiency of the whole algorithm framework.
From $U^{(j)} = \frac{1}{\alpha} (Y^j + \Phi_e^k - A^*(z^j))$ and the definition of $\gamma^j$ in (28), we have
\[
- \frac{2}{n} (A(U^{(j)}) - b) = \frac{2}{n} \gamma^j - z^j.
\]

Since $\tilde{U}^{(j)}$ is computed by using the expression in (31), then we have
\[
\begin{align*}
\tilde{U}^{(j)} &= \Pi_{S^d^+} (U^{(j)} - \nabla F^k_e(U^{(j)})) \\
&= \Pi_{S^d^+} (U^{(j)} - \frac{2}{n} A^* (A(U^{(j)}) - b) + \Phi_e^k - \alpha U^{(j)}) \\
&= \Pi_{S^d^+} (U^{(j)} + A^* (\frac{2}{n} \gamma^j - z^j) + \Phi_e^k - (Y^j + \Phi_e^k - A^*(z^j))) \\
&= \Pi_{S^d^+} (U^{(j)} + \frac{2}{n} A^*(\gamma^j - Y^j)),
\end{align*}
\]

where the third equality follows from (34). Hence, the inexact term $\tilde{\Delta}^{(j)}$ can be formulated as
\[
\tilde{\Delta}^{(j)} = U^{(j)} - \tilde{U}^{(j)} - \nabla F^k_e(U^{(j)}) + \nabla F^k_e(\tilde{U}^{(j)})
\]
\[
= (\frac{2}{n} A^* A + (\alpha - 1) I) \left( \tilde{U}^{(j)} - U^{(j)} \right),
\]

where $I$ denotes the identity operator. From the non-expansiveness of the projection operator, we have
\[
\|\tilde{U}^{(j)} - U^{(j)}\|_F = \|\Pi_{S^d^+} (U^{(j)} + \frac{2}{n} A^* (\gamma^j - Y^j) - U^{(j)})\|_F
\]
\[
= \|\Pi_{S^d^+} (U^{(j)} + \frac{2}{n} A^* (\gamma^j - Y^j) - U^{(j)})\|_F
\]
\[
= \|\Pi_{S^d^+} (U^{(j)} + \frac{2}{n} A^* (\gamma^j - Y^j) - U^{(j)})\|_F
\]
\[
\leq \frac{2}{n} \|A^* (\gamma^j)\|_F,
\]

where the third equality follows from (25) and (26). Thus, we have
\[
\|\tilde{\Delta}^{(j)}\|_F = \| (\frac{2}{n} A^* A + (\alpha - 1) I) \left( \tilde{U}^{(j)} - U^{(j)} \right)\|_F
\]
\[
\leq \frac{2}{n} \|A^* A + (\alpha - 1) I\|_F \|A^* (\gamma^j)\|_F
\]
\[
\leq \frac{2}{n} \|A\|_F \frac{2}{n} A^T A + (\alpha - 1) I \|F\| \|\gamma^j\| \frac{\|\gamma^j\|_{\ell_{k+1}^1}}{\zeta_k},
\]

where $\zeta_k$ is defined as
\[
\zeta_k := \frac{\alpha}{2} \|A\|_F^{-1} \frac{2}{n} A^T A + (\alpha - 1) I \|F\|^{-1} \|\gamma^j\|_{\ell_{k+1}^1}.
\]

Consequently, in each iteration of Algorithm 4, we only need to check if the condition $\|\gamma^j\| \leq \zeta_k$ holds. When $\|\gamma^j\| \leq \zeta_k$ holds, that implies $\|\tilde{\Delta}^{(j)}\|_F \leq \epsilon_{k+1}$, then we let $V^{k+1} := \tilde{U}^{(j)}$ and $\Delta^{k+1} := \tilde{\Delta}^{(j)}$. In this strategy, the calculation of (31) and (32) need to be performed only once, respectively, which saves a lot of computation. It is more efficient than the first strategy in practical numerical experiments, although it may overestimate the inexactness of the solution $\tilde{U}^{(j)}$.

It should be noted that the cost of computing the norm $\frac{2}{n} A^T A + (\alpha - 1) I \|F\|$ directly is expensive when $d^2 >> p$. Thus, we compute this norm as
\[ \|2nA^T A + (\alpha - 1) I\|_F = \sqrt{\|2nAA^T\|_F^2 + \|(1 - \alpha)I\|_F^2} - \langle (1 - \alpha)AA^T, \frac{1}{n}I \rangle, \]
which reduces the computing cost from \( O(d^k) \) to \( O(p^2) \).

### 3.4 Reducing Computation and Storage Cost by Using Low Rank Structure of Solution

It is noted that the main computation of Algorithm 4 is in solving the \( z \)-subproblem and \( Y \)-subproblem, the previous one is solved by equivalently solving the system defined in (24) by using PCG method and the latter one has a closed-form solution. Hence, it is important to reduce the computation cost of solving (24) for the efficiency of whole algorithm framework. In PCG method for solving (24), the major computation is in computing the term \( A (A^*(z) - \tilde{Y}^j - \Phi^k) \).

Although the linear operator \( A \) can be transformed into the form of matrix-vector product, i.e., \( A(U) = A \text{vec}(U) \), it is impractical to store the matrix \( A \in \mathbb{R}^{p \times d^2} \) when matrix dimension \( d \) is large. Thus, we can directly specify the action of the linear operator \( A \) and its conjugate \( A^* \) in our algorithms. Since \( A \) is a linear operator and \( \Phi^k = W^k - cI + \alpha U^k \), then

\[
A \left( A^*(z) - \tilde{Y}^j - \Phi^k \right) = cA(I) - A(W^k) - A(\tilde{Y}^j) - \alpha A(U^k) + A A^*(z).
\]

Next, we will compute each term on the right side of the above equation according to its special structure.

Firstly, for \( A(I) \), since \( \tau_i \tau_i^T, \tau_i \in \mathbb{R}^d, i = 1, \ldots, p \) (see Appendix 1), then \( A(I) \) can be computed by

\[
A(I) = \left[ \langle \tau_1 \tau_1^T, I \rangle, \langle \tau_2 \tau_2^T, I \rangle, \ldots, \langle \tau_p \tau_p^T, I \rangle \right] = \left[ \tau_1^T \tau_1, \tau_2^T \tau_2, \ldots, \tau_p^T \tau_p \right]^T,
\]
and its computation cost is \( O(pd) \).

Secondly, for \( A(W^k) \), \( A(U^k) \) and \( A(\tilde{Y}^j) \), we will make full use of the low structure of the matrices to reduce their computation cost. For a low rank matrix, we have the following property.

**Proposition 5** Let \( T \in S_{d^2}^d \) be a rank-\( \bar{r} \) (\( \bar{r} \ll d \)) matrix with factor form \( T = VV^T \), where \( V \in \mathbb{R}^{d \times \bar{r}} \). Then \( A(T) \) can be computed as

\[
A(T) = A(VV^T) = \left[ \langle A_1, VV^T \rangle, \langle A_2, VV^T \rangle, \ldots, \langle A_p, VV^T \rangle \right]^T, \tag{37}
\]

where \( \langle A_i, VV^T \rangle, i = 1, \ldots, p \) can be computed by

\[
\langle A_i, VV^T \rangle = \langle \tau_i \tau_i^T, VV^T \rangle = \tau_i^T VV^T \tau_i = (V^T \tau_i)^T (V^T \tau_i).
\]

Thus, the amount of computation for specifying the operator \( A \) on a rank-\( \bar{r} \) matrix is reduced from \( O(p(d^2 + d)) \) to \( O((\bar{r}d + \bar{r})p) \). This strategy can significantly reduce the computation cost of \( A(T) \) since \( \bar{r} \ll d \).

The rank-\( r \) matrix \( W^k = c \sum_{j=1}^{r} q_j q_j^T \) can be rewritten as \( W^k = c\tilde{Q}\tilde{Q}^T \) with \( \tilde{Q} = [q_1, \ldots, q_r] \in \mathbb{R}^{d \times r} \). Then we can compute \( A(W^k) \) by using the expression in (37) with \( V = \sqrt{c} \tilde{Q} \). Consequently, the amount of computation of \( A(W^k) \) is reduced to \( O((rd + r)p) \).

Since \( \tilde{Y}^{j+1} = Y^j + \beta_j (Y^j - Y^{j-1}) \), then we have

\[
A(\tilde{Y}^{j+1}) = (1 + \beta_j)A(Y^j) - \beta_j A(Y^{j-1}).
\]
From (25), we have $Y^j = \Pi_{S^k_c}(A^*(z^j) - \Phi^k)$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ be the eigenvalues of $(A^*(z^j) - \Phi^k)$ being arranged in nonincreasing order. Denote $\mu := \{i|\lambda_i > 0\}$ and $\nu := \{i|\lambda_i < 0\}$. Then $(A^*(z^j) - \Phi^k)$ has the following spectral decomposition

$$A^*(z^j) - \Phi^k = QAQ^\top, \quad \Lambda = \begin{bmatrix} \Lambda_\mu & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Lambda_\nu \end{bmatrix},$$

where $\Lambda$ is the diagonal matrix whose $i$-th diagonal entry is $\lambda_i$, and $Q$ is the orthogonal matrix whose $i$-th column vector is the eigenvector of $(A^*(z^j) - \Phi^k)$ corresponding to the eigenvalue $\lambda_i$. From the formulation of $Y^j$ and $U^{(j)}$ defined in (25) and (26), respectively, we have

$$\text{rank}(Y^j) + \text{rank}(U^{(j)}) = |\mu| + |\nu| \leq d.$$

Hence, if $|\mu| < \frac{d}{2}$ holds, we can compute $Y^j$ and $U^{(j)}$ by

$$Y^j = Q_\mu \Lambda_\mu Q_\mu^\top, \quad U^{(j)} = -\frac{1}{\alpha}(A^*(z^j) - \Phi^k - Q_\mu \Lambda_\mu Q_\mu^\top),$$

otherwise, we can compute $Y^j$ and $U^{(j)}$ by

$$Y^j = A^*(z^j) - \Phi^k - Q_\nu \Lambda_\nu Q_\nu^\top, \quad U^{(j)} = -\frac{1}{\alpha}(Q_\nu \Lambda_\nu Q_\nu^\top).$$

Therefore, if $|\mu| < \frac{d}{2}$, $A(Y^j)$ can be computed by using the expression in (37) with $V = Q_\mu \sqrt{\Lambda_\mu}$, its computation cost is $O((d|\mu| + |\mu|)p)$, otherwise, $A(Y^j)$ can be computed by

$$A(Y^j) = A(A^*(z^j) - \Phi^k - Q_\nu \Lambda_\nu Q_\nu^\top)$$

$$= AA^*(z^j) - A(W^k) + cA(I) - \alpha A(U^k) - A(Q_\nu \Lambda_\nu Q_\nu^\top),$$

where $-A(Q_\nu \Lambda_\nu Q_\nu^\top)$ can be computed by using the expression in (37) with $V = Q_\nu \sqrt{\Lambda_\nu}$ and the computation cost of $-A(Q_\nu \Lambda_\nu Q_\nu^\top)$ is $O((d|v| + |v|)p)$.

**Remark 6** For the practical numerical experiments of low rank optimization problems, $|\mu| \geq \frac{d}{2}$ holds in most cases. 

As mentioned in Sect. 3.3, $U^{k+1}$ can be computed by using the expression in (35), which implies that there exists a $\tilde{f}$, such that $|y^{\tilde{f}}| \leq \zeta_k$,

$$U^{k+1} = \Pi_{S^k_c}(U^{(j)} + \frac{\alpha}{n} A^*(\gamma^{\tilde{f}}) - Y^{\tilde{f}}).$$

Thus, similar to $Y^{j}$, $U^{k+1}$ can be formulated into a factor form $U^{k+1} = V V^\top$ with $V \in \mathbb{R}^{d \times \tilde{r}}$ ($\tilde{r} \ll d$), then $A(U^{k+1})$ can be computed by using the expression in (37) and its computation cost is $O((d\tilde{r} + \tilde{r})p)$. 

Finally, for $A(A^*(z))$, it should be noted that the matrix $A^*(z)$ may have no low rank structure, then the computation cost of $A(A^*(z))$ is $O(2p(d^2 + d))$. To reduce the computation cost of $A(A^*(z))$, by storing the matrix $C$ defined in (5), we can compute $A(A^*(z))$ by matrix-vector product $A(A^*(z)) = Cz$. Then the computation cost of $A(A^*(z))$ is reduced from $O(2p(d^2 + d))$ to $O(p^2)$ ($p \ll d^2$).

Furthermore, similar to $A(A^*(z) - \tilde{Y}^j - \Phi^k)$, we need to compute the term $A(A^*(z) - Y^j - \Phi^k)$ of the KKT residual $y^j$ defined in (28), then we can reduce the computation cost of $y^j$ by using the above strategy.
4 Convergence Analysis of s-iPDCA

In this section, we present the convergence of s-iPDCA for solving (14). A feasible point \( U \in S^d_+ \) is said to be a stationary point of the DC problem in (14) if

\[
(\nabla f_c(U) + \partial \delta_{S^d_+}(U)) \cap \partial g_c(U) \neq \emptyset.
\]

The following results on the convergence of s-iPDCA for solving (14) follows from the basic convergence theorem of DCA [19].

To ensure the objective function in (14) is level-bounded, we assume \( \mathcal{A} \) satisfies the Restricted Isometry Property (RIP) condition [35], which is also used as one of the most standard assumptions in the low-rank matrix recovery literatures [14, 36, 37].

Definition 1 (Restricted Isometry Property) Let \( \mathcal{A} : S^d_+ \to \mathbb{R}^p \) be a linear map. For every integer \( s \) with \( 1 \leq s \leq d \), define the s-restricted isometry constant to be the smallest number \( R_s \) such that

\[
(1 - R_s)\|U\|_F^2 \leq \|\mathcal{A}(U)\|^2 \leq (1 + R_s)\|U\|_F^2
\]

holds for all \( U \) of rank at most \( s \). And \( \mathcal{A} \) is said to satisfy the s-restricted isometry property (s-RIP) if \( 0 \leq R_s < 1 \).

Proposition 6 Assume that \( \mathcal{A} \) satisfies d-restricted isometry property (d-RIP). Let \( \{U^k\} \) be the sequence of stability center generated by s-iPDCA for solving (14), then the following statements hold.

1. For any \( U \in S^d_+ \), \( J_c(U) \) in (14) is lower bounded and level-bounded.
2. The sequence \( \{J_c(U^k)\} \) is nonincreasing.
3. The sequence \( \{U^k\} \) is bounded.

Proof For statement (1), as shown in (14), the \( J_c(U) \) is the sum of two non-negative functions:

\[
\frac{1}{n}\|\mathcal{A}(U) - b\|^2 + c((U, I) - \|U\|_F),
\]

thus \( J_c(U) \) is lower bounded. Combining this with the d-RIP of \( \mathcal{A} \), it holds that for any \( U \in S^d_+ \),

\[
J_c(U) = \frac{1}{n}\|\mathcal{A}(U) - b\|^2 + c((U, I) - \|U\|_F) \geq \frac{1}{n}\|\mathcal{A}(U) - b\|^2
\]

\[
\geq \frac{1}{n}\|\mathcal{A}(U)\|^2 - \frac{2}{n}\|\mathcal{A}(U)\|\|b\| + \frac{2}{n}\|b\|^2
\]

\[
\geq \frac{1}{n}(1 - R_d)^2\|U\|_F^2 - \frac{2}{n}(1 + R_d)\|U\|_F\|b\| + \frac{1}{n}\|b\|^2.
\]

According to the above inequality, we have that for any \( \rho \in \mathbb{R}, \{U \in S^d_+ | J_c(U) \leq \rho \} \) is bounded, i.e., \( J_c(U) \) is level-bounded.

For statement (2), if \( U^{k+1} \) is generated in null step, namely, \( U^{k+1} = U^k \), then the \( J_c(U^{k+1}) = J_c(U^k) \) holds immediately. In another situation, when \( U^{k+1} = V^{k+1} \), since \( V^{k+1} \) is an inexact solution of (15) with inexact term \( \Delta^{k+1} \), we have

\[
U^{k+1} = \arg\min_U G_c^k(U) - \langle \Delta^{k+1}, U \rangle.
\]

Since \( G_c^k(U) \) is strongly convex, then the following inequality holds:

\[
f_c(U^{k+1}) - \langle U^{k+1}, W^k \rangle + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 - \langle U^{k+1}, \Delta^{k+1} \rangle
\]

\[
\leq f_c(U^k) - \langle U^k, W^k \rangle - \langle U^k, \Delta^{k+1} \rangle - \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2.
\]

\( \Box \) Springer
Thanks to the convexity of \( g_c(U) \), we have
\[
g_c(U^{k+1}) \geq g_c(U^k) + \langle U^{k+1} - U^k, W^k \rangle.
\]
Combining this with (39), we get
\[
\alpha \|U^{k+1} - U^k\|^2_F - \langle U^{k+1} - U^k, \Delta^{k+1} \rangle \leq \left[ f_c(U^k) - g_c(U^k) \right] - \left[ f_c(U^{k+1}) - g_c(U^{k+1}) \right].
\]
(40)

Since \( U^{k+1} = V^{k+1} \) is generated in serious step of s-iPDCA, so the condition in (18) holds:
\[
\|\Delta^{k+1}\|_F \leq (1 - \kappa)\alpha\|\Delta^k\|_F.
\]

Then we have
\[
\alpha \|U^{k+1} - U^k\|^2_F - \langle U^{k+1} - U^k, \Delta^{k+1} \rangle \geq \alpha \|U^{k+1} - U^k\|^2_F - \|U^{k+1} - U^k\|_F \|\Delta^{k+1}\|_F \geq \kappa \alpha \|U^{k+1} - U^k\|^2_F.
\]

Applying this to (40), we get
\[
\kappa \alpha \|U^{k+1} - U^k\|^2_F \leq \left[ f_c(U^k) - g_c(U^k) \right] - \left[ f_c(U^{k+1}) - g_c(U^{k+1}) \right].
\]
(41)

Thus the sequence \( \{ f_c(U^k) - g_c(U^k) \} \) is non-increasing.

For statement (3), according to statement (2), we have \( J_c(U^k) \leq J_c(U^0) \), i.e., \( \{ J_c(U^k) \} \) is bounded. From statement (1), we get that \( J_c \) is level-bounded. Combining this with the boundedness of \( \{ J_c(U^k) \} \), we have \( \{ U^k \} \) is bounded. This completes the proof. \( \square \)

By setting the tolerance error \( \varepsilon = 0 \), we divide the convergence analysis of s-iPDCA into the following two parts. In the first part, we consider the situation that only finite serious steps are performed in s-iPDCA for solving (14). In the second part, we consider the situation that infinite serious steps are performed in s-iPDCA for solving (14).

### 4.1 Finite Serious Steps in s-iPDCA

For the situation that only finite serious steps are performed in s-iPDCA for solving (14), we have the following convergence results.

**Theorem 1**  Set the tolerance error \( \varepsilon = 0 \). Suppose that only finite serious steps are performed in s-iPDCA for solving (14). Then the following statements hold.

1. If s-iPDCA terminates in finite steps, i.e., there exists a \( \hat{k} > 0 \) such that \( V^{\hat{k}+1} = U^{\hat{k}} \), then the stability center \( U^{\hat{k}} \) is an \( \varepsilon_1 \)-stationary point of (14).

2. If after \( \hat{k} \)-th iteration of s-iPDCA, only null step is performed, i.e., for any \( k > \hat{k} \), \( U^k = U^{\hat{k}} \) and \( W^k = W^{\hat{k}} \), then
\[
\lim_{k \to \infty} V^{k+1} = U^{\hat{k}}.
\]
(42)
and the stability center \( U^{\hat{k}} \) generated in the last serious step is a stationary point of (14).
Proof} For statement (1), since $V^{k+1}$ is an $\epsilon_{k+1}$-inexact solution of (15) with inexact term $\Delta^{k+1}$, then we have $\Delta^{k+1} \in \partial G_c^{\ast}(U^{\hat{k}})$, i.e.,
\[
\Delta^{k+1} \in \nabla f_c(U^{\hat{k}}) - W^{\hat{k}} + \partial \Omega (U^{\hat{k}}).
\]
Thus, there exists a $\xi^{\hat{k}} \in \partial \Omega (U^{\hat{k}})$ such that
\[
\nabla f_c(U^{\hat{k}}) - W^{\hat{k}} + \xi^{\hat{k}} - \Delta^{k+1} = 0.
\]
Then we have
\[
\|\nabla f_c(U^{\hat{k}}) - W^{\hat{k}} + \xi^{\hat{k}}\|_F = \|\Delta^{k+1}\|_F \leq \epsilon_{k+1} < \epsilon_1,
\]
and
\[
0 \in \partial \Omega \left[ \delta \Omega (U^{\hat{k}}) - g_c(U^{\hat{k}}) \right] + \nabla f_c(U^{\hat{k}}),
\]
which means that 0 belongs to the $\epsilon_1$-inexact subdifferential of $f_c(U) - g_c(U) + \partial \Omega (U)$ at point $U^{\hat{k}}$. Then $U^{\hat{k}}$ is an $\epsilon_1$-inexact point of (14).

For statement (2), we first prove $\lim_{k \to \infty} V^{k+1} = \hat{U}$. Since the sieving condition in (18) does not hold for any $k > \hat{k}$, i.e.,
\[
(1 - \kappa)\alpha\|V^{k+1} - \hat{U}\|_F < \|\Delta^{k+1}\|_F \leq \epsilon_{k+1}
\]
holds for any $k > \hat{k}$. Thanks to the monotonic descent property of $\{\epsilon_k\}$ and $\lim_{k \to \infty} \epsilon_k = 0$, taking limit on both sides of inequality (43), we have
\[
\lim_{k \to \infty} (1 - \kappa)\alpha\|V^{k+1} - \hat{U}\|_F = 0,
\]
then $\lim_{k \to \infty} V^{k+1} = \hat{U}$. Next, we will prove that $\hat{U}$ is a stationary point of (14). Since $V^{k+1}$ is an inexact solution of (15) with inexact term $\Delta^{k+1}$, then it holds that for any $k > \hat{k}$,
\[
\Delta^{k+1} \in \nabla f_c(V^{k+1}) - \hat{W} + \partial \Omega (V^{k+1}) + \alpha(V^{k+1} - \hat{U}).
\]
Then there exists a $\zeta^{k+1} \in \partial \Omega (V^{k+1})$ such that
\[
\nabla f_c(V^{k+1}) - \hat{W} + \zeta^{k+1} + \alpha(V^{k+1} - \hat{U}) - \Delta^{k+1} = 0.
\]
Then we have
\[
\|\nabla f_c(V^{k+1}) - \hat{W} + \zeta^{k+1}\|_F \leq \alpha\|V^{k+1} - \hat{U}\|_F + \|\Delta^{k+1}\|_F
\]
\[
\leq \alpha\|V^{k+1} - \hat{U}\|_F + \epsilon_{k+1}.
\]
From $\lim_{k \to \infty} V^{k+1} = \hat{U}$, we get that $\{V^k\}$ is bounded. Then the boundedness of $\{\zeta^k\}$ can be obtained from (46). As a consequence of [38,Proposition 4.1.1], there exists a subset $\mathcal{K}' \subset \mathcal{K} = \{0, 1, 2, \cdots\}$ such that $\lim_{k \in \mathcal{K}'} \zeta^{k+1} = \hat{\xi} \in \partial \Omega (\hat{U})$. Since $\lim_{k \in \mathcal{K}'} \epsilon_k = 0$, we have
\[
\lim_{k \in \mathcal{K}'} \|\nabla f_c(V^{k+1}) - \hat{W} + \zeta^{k+1}\|_F = 0.
\]
which implies that
\[ 0 \in \nabla f_c(U^\hat{k}) - \partial g_c(U^\hat{k}) + \partial \delta S_{\hat{z}}(U^\hat{k}), \]
and \( U^\hat{k} \) is a stationary point of (14). This completes the proof. \( \square \)

4.2 Infinite Serious Steps are Performed in s-iPDCA

In this subsection, we consider the situation that infinite serious steps are performed in s-iPDCA for solving (14) when the tolerance error \( \varepsilon \) is set to 0. The sequence \( \{U^k\} \) is shown as
\[
\left\{ \ldots, U^{k-N_l}, U^{k-N_l+1}, \ldots, U^k, U^{k+1}, \ldots, U^{k+N_{l+1}+2}, \ldots \right\},
\]
where \( U^{kl} \) denotes the stability center generated in \( l \)-th serious step. Since the subsequence \( \{U^{k-N_l+1}, \ldots, U^k\} \) is the collection of the stability centers in null steps between \( l \)-th serious step and \( (l+1) \)-st serious step, then \( U^{kl} = U^{k-N_l} = U^{k-N_l+1} = \cdots = U^k \) hold. Based on the assumption that infinite serious steps are performed in s-iPDCA for solving (14), we have that only finite stability centers generated in null steps between two adjacent serious steps and the stability centers in null steps are the finite repetition of that in serious steps. By removing the \( U^{kl} \) generated in null steps from \( \{U^k\} \), we obtain a subsequence \( \{U^{kl}\} \). In addition, let \( \{W^{kl}\} \) denote the set of chosen subgradient of \( g_c \) at \( U^{kl} \). Let \( \{\Delta^{kl}\} \) be the set of inexact term related to \( U^{kl} \), then \( \|\Delta^{kl}\|_F \leq (1-\kappa)\alpha\|U^{kl} - U^{kl-1}\|_F \) holds.

For \( \{U^{kl}\} \), we have the following global subsequential convergence results.

**Theorem 2** (Global subsequential convergence of s-iPDCA) Set the tolerance error \( \varepsilon = 0 \). Let \( \{U^{kl}\} \) be the stability center sequence generated in serious step of s-iPDCA for solving (14). Then the following statements hold.

(1) \( \lim_{l \to \infty} \|U^{kl} - U^{kl+1}\|_F = 0 \).

(2) Any accumulation point \( \hat{U} \in \{U^{kl}\} \) is a stationary point of (14).

**Proof** For statement (1), since \( U^{kl} \) and \( U^{kl+1} \) are the stability centers generated in two adjacent serious steps, by replacing the \( U^k \) and \( U^{kl} \) in (41) with \( U^{kl} \) and \( U^{kl+1} \), respectively, we obtain
\[
\kappa\alpha \|U^{kl+1} - U^{kl}\|_F^2 \leq \left[ f_c(U^{kl}) - g_c(U^{kl}) \right] - \left[ f_c(U^{kl+1}) - g_c(U^{kl+1}) \right].
\]

From statement (1) of Proposition 6, we get that \( f_c(U) - g_c(U) \) is lower bounded and the sequence \( \{f_c(U^{kl}) - g_c(U^{kl})\} \) is nonincreasing and lower bounded. Then, we have
\[
\liminf_{l \to \infty} \left[ f_c(U^{kl}) - g_c(U^{kl}) \right] < \infty.
\]

By summing both sides of (49) from \( l = 0 \) to \( \infty \), we obtain
\[
\sum_{l=0}^{\infty} \kappa\alpha \|U^{kl+1} - U^{kl}\|_F^2 \leq \left[ f_c(U^0) - g_c(U^0) \right] - \liminf_{l \to \infty} \left[ f_c(U^{kl+1}) - g_c(U^{kl+1}) \right] < \infty.
\]
Hence, there exists a $\zeta^{k_{i+1}} \in \partial\delta_{\mathcal{S}^d}(U^{k_{i+1}})$ such that
\[
\nabla f_c(U^{k_{i+1}}) - W^{k_{i}} + \zeta^{k_{i+1}} + \alpha(U^{k_{i+1}} - U^{k_{i}}) - \Delta^{k_{i+1}} = 0.
\]
Since $\|\Delta^{k_{i+1}}\|_F \leq \epsilon_{k_{i+1}}$, then we get
\[
\|\nabla f_c(U^{k_{i+1}}) - W^{k_{i}} + \zeta^{k_{i+1}}\|_F \leq \alpha\|U^{k_{i+1}} - U^{k_{i}}\|_F + \epsilon_{k_{i+1}}.
\] (51)

From the boundedness of $\{U^k\}$, we have that $\{U^k\}$ is also bounded, then there exists a subset $\mathcal{L}' \subset \mathcal{L} = \{0, 1, 2, \ldots\}$ such that $\{U^k\}_{\mathcal{L}'}$ converges to an accumulation point $\overline{U} \in \{U^k\}_{\mathcal{L}'}$. Combining the boundedness of $\{U^k\}_{\mathcal{L}'}$ with the finite-valued property and convexity of $g_c$, we have that the subsequence $\{W^k\}_{\mathcal{L}'}$ is also bounded. Thus, from (51), we can obtain the boundedness of $\{\zeta^k\}_{\mathcal{L}'}$. Let $\overline{W}$ and $\overline{\zeta}$ be an accumulation point of $\{W^k\}_{\mathcal{L}'}$ and $\{\zeta^k\}_{\mathcal{L}'}$, respectively. As a consequence of [38, Proposition 4.1.1], we may assume without loss of generality that there exists a subset $\mathcal{L}'' \subset \mathcal{L}'$ such that $\lim_{l \in \mathcal{L}''} W^k = \overline{W} \in \partial g_c(\overline{U})$, $\lim_{l \in \mathcal{L}''} \zeta^k = \overline{\zeta} \in \partial\delta_{\mathcal{S}^d}(\overline{U})$. From $\lim_{k \to \infty} \epsilon_k = 0$, we have $\lim_{l \in \mathcal{L}''} \epsilon_k = 0$. Taking limit on the two sides of inequality in (51) with $l \in \mathcal{L}''$, we have
\[
\|\nabla f_c(\overline{U}) - \overline{W} + \overline{\zeta}\|_F = \lim_{l \in \mathcal{L}''} \|\nabla f_c(U^{k_{l+1}}) - W^{k_{l}} + \zeta^{k_{l+1}}\|_F \\
\leq \lim_{l \in \mathcal{L}''} \alpha\|U^{k_{l+1}} - U^{k_{l}}\|_F + \lim_{l \in \mathcal{L}''} \epsilon_{k_{l+1}} = 0.
\] (52)

which implies that $\|\nabla f_c(\overline{U}) - \overline{W} + \overline{\zeta}\|_F = 0$. Therefore, any accumulation point $\overline{U} \in \{U^k\}$ satisfies the following optimality condition:
\[
0 \in \nabla f_c(\overline{U}) - \partial g_c(\overline{U}) + \partial\delta_{\mathcal{S}^d}(\overline{U}).
\]
This implies that any accumulation point of $\{U^k\}$ is a stationary point of (14). This completes the proof. \qed

In order to display that the sequence $\{U^k\}$ actually converges to a stationary point of (14) when infinite serious steps are performed in Algorithm 3, we construct the following auxiliary function:
\[
E(U, W, V, T) = f_c(U) - \langle U, W \rangle + g^*_c(W) + \alpha\|U - V\|^2_F - \langle T, U - V \rangle.
\] (53)

where $g^*_c$ is the convex conjugate of $g_c$, given as
\[
g^*_c(W) = \sup_{U \in \mathcal{S}^d} \{\langle W, U \rangle - g_c(U)\}.
\]

Then $f_c(U) - g_c(U) \leq f_c(U) - \langle U, W \rangle + g^*_c(W)$ holds. Thanks to the fact that $g_c(U) = c\|U\|_{c, 1}$ is a proper closed convex function, we have that $g^*_c(W)$ is also a proper closed convex function and the Young’s inequality holds
\[
g^*_c(W) + g_c(U) \geq \langle U, W \rangle,
\]
\section*{Springer}
and the equality holds if and only if $W \in \partial g_c(U)$. Moreover, for any $U, W \in S^d$, $W \in \partial g_c(U)$ if and only if $U \in \partial g_c^2(W)$.

From the Tarski-Seidenberg Theorem [39, Theorem 8.6.6], we have that the semidefinite programming representable sets are all semialgebraic [40, 41]. Hence, the function $E$ defined in (53) belongs to the class of semialgebraic functions. According to [42–44], we get that semialgebraic functions satisfy Kurdyka-Łojaziewicz (KL) property.

Next, we display that $E(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}})$ is lower bounded, nonincreasing and convergent, and for any $l \geq 1$ the subdifferential of function $E$ at point $(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}})$ is bounded.

**Proposition 7** Let $E$ be defined in (53). Suppose that infinite serious steps are performed in s-iPDCA for solving (14). Let $\{U^{k_l}\}$, $\{\Delta^{k_l}\}$ and $\{W^{k_l}\}$ be the subsequences generated in serious steps of s-iPDCA for solving (14), then the following statements hold.

1. For any $l \geq 1$,
   
   \[ J_c(U^{k_{l+1}}) \leq E(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}}). \]  

2. For any $l \geq 1$,
   
   \[
   E(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}}) - E(U^{k_{l}}, W^{k_{l-1}}, U^{k_{l-1}}, \Delta^{k_{l}}) \leq -\kappa \alpha \|U^{k_{l}} - U^{k_{l-1}}\|^2_{F}.
   \]  

3. The set of accumulation points of the sequence $\{(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}})\}$, denoted by $\Omega$, is a nonempty compact set.

4. The limit $\Upsilon = \lim_{l \to \infty} E(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}})$ exists and $E \equiv \Upsilon$ on $\Omega$.

5. There exists a constant $\rho > 0$ such that for any $l \geq 1$, we have
   
   \[
   \text{dist}(0, \partial E(U^{k_{l+1}}, W^{k_{l}}, U^{k_{l}}, \Delta^{k_{l+1}})) \leq \rho \|U^{k_{l+1}} - U^{k_{l}}\|_{F}.
   \]  

**Proof** For space limitations, we present the proof of statements (1)-(5) in Appendix 1. □

For simplicity of notation, we set $E^{k_{l}} = E(U^{k_{l}}, W^{k_{l-1}}, U^{k_{l-1}}, \Delta^{k_{l}})$ for each $l > 0$. Based on the KL property of function $E$, for $\{U^{k_{l}}\}$, we have the following global sequential convergence results.

**Theorem 3** Set the tolerance error $\epsilon = 0$. Suppose that infinite serious steps are performed in s-iPDCA for solving (14). Let $\{U^{k_{l}}\}$ be the stability center sequence generated in serious steps of s-iPDCA for solving (14). Then $\{U^{k_{l}}\}$ converges to a stationary point of (14). Moreover, \[ \sum_{l=0}^{\infty} \|U^{k_{l+1}} - U^{k_{l}}\|_{F} < \infty. \]

**Proof** From Proposition 7, we have that $\{E^{k_{l}}\}$ is nonincreasing and its limit $\Upsilon$ exists. Then, we get $E^{k_{l}} \geq \Upsilon$ for any $l > 0$. Next, we will prove that for any $l > 0$, $E^{k_{l}} \geq \Upsilon$. To this end, we suppose that $\exists L > 0$ such that $E^{k_{L}} = \Upsilon$, then $E^{k_{l}} = \Upsilon$ holds for all $l > L$. From (55), we have $U^{k_{L}} = U^{k_{L}}$ for each $l \geq L$. This implies that only finite serious steps are performed in s-iPDCA, which is contrary to the assumption.

Since $E$ satisfies the KL property at each point in the compact set $\Omega \subset \text{dom} \partial E$ and $E \equiv \Upsilon$ on $\Omega$, thus it satisfies the uniform KL property [44]. Then there exist $\epsilon > 0$ and a continuous concave function $\varphi : (0, a) \rightarrow \mathbb{R}_{+}$ being continuously differentiable and monotonically increasing on $(0, a)$ and satisfying $\varphi(0) = 0$ with $a > 0$ such that for any $(U, W, V, T) \in \Theta$,

\[
\varphi'(E(U, W, V, T) - \Upsilon) \cdot \text{dist}(0, \partial E(U, W, V, T)) \geq 1,
\]
According to the above inequality, we get

\[ \text{Then by applying the arithmetic mean–geometric mean inequality, we obtain} \]

\[ \text{where the last inequality follows from (57). Let} \]

\[ \text{Thus, there exists a constant } \bar{L} > 0 \text{ such that for any } l > \bar{L} - 2, \]

\[ \text{From Proposition 7, we have that the sequence } \{E^k\} \text{ converges to } \gamma, \text{ then exists a constant} \]

\[ \text{Let } \tilde{L} = \max \{L, \bar{L}\}, \text{ then for any } l > \tilde{L}, \text{ we have } (U^{k-1}, W^{k-1}, U^{k-2}, \Delta^{k-1}) \in \Theta \text{ and} \]

\[ \varphi'(E^{k-1} - \gamma) \cdot \text{dist}(0, \partial E^{k-1}) \geq 1. \]  

(57)

By using the concavity of \( \varphi \), it holds that for each \( l > \tilde{L} \),

\[ \left[ \varphi(E^{k-1} - \gamma) - \varphi(E^{k+1} - \gamma) \right] \cdot \text{dist}(0, \partial E^{k-1}) \]

\[ \geq \varphi'(E^{k-1} - \gamma) \cdot \text{dist}(0, \partial E^{k-1}) \cdot (E^{k-1} - E^{k+1}) \]

\[ \geq E^{k-1} - E^{k+1}, \]

where the last inequality follows from (57). Let \( B^k = \varphi(E^k - \gamma) \) for each \( l > 0 \). Since \( \varphi \) is monotone increasing on \( (0, a) \) and \( \{E^k\} \) is nonincreasing, then we get that \( \{B^k\} \) is nonincreasing. Combining the results in (55), (56) and (58), we have

\[ \|U^k - U^{k-1}\|_F^2 + \|U^{k-1} - U^{k-2}\|_F^2 \]

\[ \leq \frac{\rho}{\kappa \alpha} (B^{k-1} - B^{k+1}) \|U^{k-1} - U^{k-2}\|_F. \]  

(59)

According to the above inequality, we get

\[ \frac{\rho}{\kappa \alpha} (B^{k-1} - B^{k+1}) - \|U^{k-1} - U^{k-2}\|_F \geq 0. \]

Then by applying the arithmetic mean–geometric mean inequality, we obtain

\[ \|U^k - U^{k-1}\|_F \]

\[ \leq \sqrt{\frac{\rho}{2\kappa \alpha} (B^{k-1} - B^{k+1}) - \frac{1}{2} \|U^{k-1} - U^{k-2}\|_F} + \sqrt{\frac{3}{4} \|U^{k-1} - U^{k-2}\|_F + \|U^{k-1} - U^{k-2}\|_F}. \]

Consequently, we have

\[ \frac{1}{4} \|U^k - U^{k-1}\|_F \]

\[ \leq \frac{\rho}{4\kappa \alpha} (B^{k-1} - B^{k+1}) + \frac{3}{4} \left( \|U^{k-1} - U^{k-2}\|_F - \|U^k - U^{k-1}\|_F \right) \]  

(60)
Summing both sides of (60) from \( l = \tilde{L} \) to \( \infty \), we have

\[
\frac{1}{4} \sum_{l=\tilde{L}}^{\infty} \| U^{kl} - U^{kl-1} \|_F \\
\leq \frac{\rho}{4\kappa\alpha} (B^{k\tilde{L}-1} + B^{k\tilde{L}}) - \lim_{l \to \infty} \frac{\rho}{4\kappa\alpha} (B^{kl} + B^{kl+1}) \\
+ \frac{3}{4} \left( \| U^{k\tilde{L}-1} - U^{k\tilde{L}-2} \|_F - \lim_{l \to \infty} \| U^{kl} - U^{kl-1} \|_F \right).
\]

From \( \lim_{l \to \infty} \frac{\rho}{4\kappa\alpha} (B^{kl} + B^{kl+1}) = 0 \) and \( \lim_{l \to \infty} \| U^{kl} - U^{kl-1} \|_F = 0 \), we obtain

\[
\frac{1}{4} \sum_{l=\tilde{L}}^{\infty} \| U^{kl} - U^{kl-1} \|_F \\
\leq \frac{\rho}{4\kappa\alpha} (B^{k\tilde{L}-1} + B^{k\tilde{L}}) + \frac{3}{4} \| U^{k\tilde{L}-1} - U^{k\tilde{L}-2} \|_F < \infty.
\]

Thus the subsequence \( \{ U^{kl} \} \) is convergent and \( \sum_{i=0}^{\infty} \| U^{k+1} - U^k \|_F < \infty \). From Theorem 2, we have that the sequence \( \{ U^k \} \) generated by s-iPDCA converges to a stationary point of (14). This completes the proof.

Finally, we give the global sequential convergence of \( \{ U^k \} \).

**Theorem 4** (Global sequential convergence of s-iPDCA) Set the tolerance error \( \varepsilon = 0 \). Suppose that infinite serious steps are performed in s-iPDCA for solving (14). Let \( \{ U^k \} \) be the stability center sequence generated by s-iPDCA for solving (14). Then \( \{ U^k \} \) converges to a stationary point of (14). Moreover, \( \sum_{k=0}^{\infty} \| U^{k+1} - U^k \|_F < \infty \).

**Proof** From the assumption that infinite serious steps are performed in s-iPDCA for solving (14), we obtain that \( \{ U^{kl} \} \) is just the subsequence of \( \{ U^k \} \) removing the finite repeated points. Thus, from Theorem 3, we get that \( \{ U^k \} \) is also convergent to a stationary point of (14). Moreover, we have

\[
\sum_{k=0}^{\infty} \| U^{k+1} - U^k \|_F = \sum_{l=0}^{\infty} \| U^{kl+1} - U^{kl} \|_F < \infty.
\]

This completes the proof.

5 Numerical Experiments

In this section, we perform numerical experiments to demonstrate the efficiency of our s-iPDCA for solving the RCLSSDP and the effectiveness of our rank constraint SDPP model in face recognition.

**Computational environment** All experiments are performed in Matlab 2020a on a 64-bit PC with an Intel(R) Xeon(R) CPU E5-2609 v2 (2.50GHz)(2 processor) and 56GB of RAM.

**Introduction of experimental data and model** The numerical experiments are performed on three standard data sets, including COIL-20 database, ORL face images database and Extended Yale Face Database B (YaleB) [45]. The dimension of the image in these databases is much larger than the number of data points, e.g. the ORL database contains 400 face images with \( d = 112 \times 92 \), the cropped YaleB database contains 2414 face images with \( d = 168 \times 192 \).
and the cropped COIL-20 database contains 1440 images with $d = 128 \times 128$. Based on this observation, we use a kernel technique to reduce the size of positive semidefinite matrix in (3) from $d \times d$ to $n \times n$. Similar to [1], the kernel SDPP with Gaussian kernel function

$$\exp\left(-\frac{\|x-y\|^2}{\zeta t^2}\right)$$

(61)

where $K$ is the kernel matrix and $K_i$ is the $i$-th column vector of $K$. In Gaussian kernel function, $t$ and $\zeta > 0$ are the kernel parameters. The parameter $t$ is calculated by Silverman rule of thumb, $t = 1.06n^{-0.2} \sqrt{\frac{\sum_{i=1}^{n} \|x_i-x\|^2}{n-1}}$. The parameter $\zeta$ can be adjusted to achieve higher recognition accuracy. Similar to (2), the SDP relaxation of (61) can be formulated as

$$\min_{U \in S^+_n} J(U) = \frac{1}{n} \|A(U) - b\|^2,$$

(62)

where $A : S^+_n \rightarrow \mathbb{R}^p$. Similarly, we add a rank constraint to (62), shown as

$$\min_{U \in S^+_n} J(U) = \frac{1}{n} \|A(U) - b\|^2$$

s.t. rank$(U) \leq r$.

(63)

We call the dimensionality reduction models in (62) and (63) kernel SDPP (KSDPP) and rank constraint kernel SDPP (RCKSDPP), respectively. Similar to (3), for a given penalty parameter $c$, (63) can also be transferred into a DC programming:

$$\min_{U \in S^+_n} J_c(U) = \frac{1}{n} \|A(U) - b\|^2 + c\langle U, I \rangle - c\|U\|_{(r)}.$$

(64)

In our experiments, the neighbourhoods of RCKSDPP and KSDPP are set as $k$-nearest neighbor ($k$-nn). We divide our numerical experiments into the following two parts. (1) We compare the numerical performance of s-iPDCA with classical PDCA, PDCA with extrapolation (PDCAe), proximal gradient-DCA (PG-DCA) and proximal gradient (PG) method for solving (63). (2) We apply RCKSDPP to perform dimensionality reduction for face recognition on ORL and YaleB databases. The results are compared with KSDPP and kernel principal component analysis (KPCA).

**Scaling of experimental data** Before starting the experiments, we scale the models in (62) and (63) as follows. Let $\tilde{A} = \frac{A}{\|A\|}$, $\tilde{b} = \frac{b}{\|b\|}$ and $\tilde{U} = \frac{A^*U}{\|b\|}$. After the scaled problem is solved, the optimal (suboptimal) solution and optimal (suboptimal) value can be obtained by rescaling. For convenience, we omit the ‘$\sim$’ for the remainder of this paper.

**Initialization** For our s-iPDCA, PDCA, PG-DCA, PDCAe, we use a low precision solution of the convex problem in (62) as their initial solution and a boundary point method [46] is used to solve (62) to a low precision $\varepsilon = 2 \times 10^{-4}$. When a sufficiently small penalty parameter is chosen, (64) can be solved easily with the initial solution from (62). By this observation, we start these algorithms with a small penalty parameter $c_0$, then we gradually increase $c_i$ until the rank constraint is satisfied, i.e., rank$(U_{c_i}) \leq r$. 

Springer
5.1 Comparing s-iPDCA with Other Methods for (63)

To illustrate the effectiveness of s-iPDCA for solving (63), we compare the numerical performance of s-iPDCA with that of other methods.

**Experimental data** The COIL-20 database is used in this experiment. The COIL-20 database is a collection of gray images of 1440 images from 20 objects, where each object has 72 different images. The images of each object, with uniformly distributed rotation angles \([0^\circ, 5^\circ, \cdots, 355^\circ]\), have been cropped into size of \(128 \times 128\). By sampling from the COIL-20, we obtain 8 data sets with \(n = 100, 200, 400, 600, 800, 1000, 1200, 1440\). In each data set, the number of the sampling images of each object is the same, e.g. 5 images of each object is chosen when \(n = 100\) and 72 images of each object is chosen when \(n = 1440\).

**Comparing algorithms** Based on the above 8 data sets, we compare the performance of s-iPDCA with classical PDCA (Algorithm 1), PG method (Algorithm 5), PG-DCA (Algorithm 6) and PDCAe (Algorithm 7) for solving (63).

**Algorithm 5 PG for (63)**

**Step 0.** Give tolerance error \(\varepsilon \geq 0\) and proximal parameter \(L > 0\). Initialize \(U^0 \in \mathcal{F} = \mathcal{S}_n^+ \cap \{U : \text{rank}(U) \leq r\}\). Set \(k = 0\).

**Step 1.** Compute

\[
U^{k+1} = \arg \min_U (\nabla J(U^k), U) + \delta_{\mathcal{F}}(U) + \frac{L}{2} \|U - U^k\|^2_F. \tag{65}
\]

**Step 2.** If \(\|U^{k+1} - U^k\|_F \leq \varepsilon\), stop and return \(U^{k+1}\), otherwise, set \(k \rightarrow k + 1\) and go to **Step 1**.

**Remark 7** From [47, Proposition 3.1] and [48, Proposition 2.6], we have that projection onto the set \(\mathcal{F}\) has closed-form solution. Let \(U = Q \Lambda Q^T\) be the spectral decomposition of \(U \in \mathcal{S}_n^+\), then a projection of \(U\) onto \(\mathcal{F}\) can be computed as

\[
\Pi_{\mathcal{F}}(U) = \begin{cases}
Q_{\mu} \Lambda_{\mu} Q_{\mu}^T, & |\mu| \leq r \\
Q_{\mu_r} \Lambda_{\mu_r} Q_{\mu_r}^T, & |\mu| > r
\end{cases}
\tag{66}
\]

where \(\mu\) is the index set of the positive eigenvalues of \(U\) and \(\mu_r\) is the set of the first \(r\) element of \(\mu\). Evidently, (65) has a closed-form solution \(U^{k+1} = \Pi_{\mathcal{F}}(U^{k} - \frac{1}{L} \nabla J(U^{k}))\).

**Algorithm 6 PG-DCA for (64)**

**Step 0.** Give \(c > 0\), tolerance error \(\varepsilon \geq 0\) and proximal parameter \(L > 0\). Initialize \(U^0 \in \mathcal{S}_n^+.\) Choose \(W^0 \in \partial g_c(U^0)\). Set \(k = 0\).

**Step 1.** Compute

\[
U^{k+1} = \arg \min_U (\nabla f_c(U^k) - W^k, U) + \delta_{\mathcal{S}_n^+}(U) + \frac{L}{2} \|U - U^k\|^2_F. \tag{67}
\]

**Step 2.** If \(\|U^{k+1} - U^k\|_F \leq \varepsilon\), stop and return \(U^{k+1}\).

**Step 3.** Choose \(W^{k+1} \in \partial g_c(U^{k+1})\). Set \(k \rightarrow k + 1\) and go to **Step 1**.
Algorithm 7 PDCAe [26, 27] for (64)

Step 0. Give \( c > 0 \), tolerance error \( \epsilon \geq 0 \) and proximal parameter \( L > 0 \). Initialize \( u^0 = \tilde{u}^0 \in S^n_+ \) and \( t_0 = 1 \). Choose \( w^0 \in \partial g_c(u^0) \). Set \( k = 0 \).

Step 1. Compute

\[
\begin{align*}
u^{k+1} &= \arg \min_U (\nabla f_c(u^k) - w^k, u) + \delta_{S^n_+}(u) + \frac{L}{2} \| u - \tilde{u}^k \|^2_F. \tag{68}
\end{align*}
\]

Step 2. If \( \| u^{k+1} - u^k \|_F \leq \epsilon \), stop and return \( u^{k+1} \).

Step 3. Compute

\[
\begin{align*}
t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \quad \beta_k = \frac{t_k - 1}{t_k + 1}, \quad u^{k+1} = u^{k+1} + \beta_k (u^{k+1} - u^k).
\end{align*}
\]

Choose \( w^{k+1} \in \partial g_c(u^{k+1}) \). Set \( k \rightarrow k + 1 \) and go to Step 1.

Remark 8 The strategy of gradually increasing the penalty parameter \( c \) is also used in PG-DCA and PDCAe for solving (63). The subproblems in (67) and (68) also have closed-form solutions, \( \Pi S^n_+ = \Pi S^n_+ (u^k - \frac{1}{L} (\nabla f_c(u^k) - W^k) \) and \( u^{k+1} = \Pi S^n_+ (u^{k+1} - \frac{1}{L} (\nabla f_c(u^{k+1}) - W^k) \), respectively.

Parameters setup In this experiment, the projection dimension is set as \( r = 5 \). The kernel parameter is set as \( \xi = 2 \). The neighborhood size of the k-nn is set as \( k = \text{round}(\log(n)) \). For PDCA, s-iPDCA, PG-DCA and PDCAe, the initial penalty parameter \( c_0 \) is set as \( c_0 = 10^{-5} \). The other parameters of the algorithms are set as below.

Parameters of s-iPDCA For the s-iPDCA, choosing the appropriate and effective proximal parameter is an essential thing from the perspective of both theory analysis and numerical implementation. Note that for numerical efficiency, the general principle is that the proximal parameter chosen as small as possible such that the corresponding subproblem could take larger step-length while the subproblem still could be solved relatively easily. Through numerical experiments, we observe that within a certain range, the smaller the value of \( \alpha \) is, the more efficient the s-iPDCA could be. Thus the values of \( \alpha \) is chosen through a 5-fold cross-validation procedure from the set of candidates

\[
\{ 2 \times 10^{-7}, 4 \times 10^{-7}, 6 \times 10^{-7}, 8 \times 10^{-7}, 10^{-6}, 2 \times 10^{-6}, 4 \times 10^{-6}, 6 \times 10^{-6} \}.
\]

The sieving parameter \( \kappa \in (0, 1) \) is used to balance the efficiency of s-iPDCA and the inexactness of solution for proximal subproblem. Large \( \kappa \) requires the new stability center to be more accurate, which results in fewer serious steps being performed. In order to get more serious steps, the selected \( \kappa \) should be as small as possible.

The sequence \( \{ \epsilon_k \} \) is set as follows: if \( v^k \) satisfies the sieving condition in (18), set

\[
\epsilon_{k+1} = \max \left( \frac{k}{1 + \kappa}, \rho_1 \right) \epsilon_k,
\]

otherwise, set \( \epsilon_{k+1} = \max \left( \frac{k}{1 + \kappa}, \rho_2 \right) \epsilon_k \). The constants, \( \rho_1, \rho_2 \in (0, 1) \), are used to ensure that \( \{ \epsilon_k \} \) drops slowly enough and that as many serious steps as possible could occur.

Therefore, the parameters of s-iPDCA are set as: \( \alpha = 2 \times 10^{-6} (n = 100, n = 200), \alpha = 10^{-6} (n = 400), \alpha = 9 \times 10^{-7} (n = 600), \alpha = 8 \times 10^{-7} (n = 800), \alpha = 6 \times 10^{-7} (n = 1000), \alpha = 5 \times 10^{-7} (n = 1200, n = 1440), \kappa = 0.1, \epsilon_1 = 0.1, \rho_1 = 0.99 \) and \( \rho_2 = 0.9 \).

Parameters of PDCA The proximal parameter of PDCA is set to be the same as that of s-iPDCA. It should be noted that the subproblems of PDCA should be exactly solved. However, in order to improve the efficiency of PDCA in practice, we solve the dual subproblems of
PDCA by Algorithm 4 to a fixed precision $\zeta_k = 10^{-3}$, which is mostly benefit to PDCA in practice.

**Parameters of PG, PG-DCA and PDCAe** The gradient Lipschitz smooth constant of $J(U) = \frac{1}{n} \| A(U) - b \|^2$ can be computed as $\frac{1}{n} \| A^T A \|$. For PG method, PG-DCA and PDCAe, we know that $L \geq \frac{2}{n} \| A^T A \|$ is a sufficient condition so that these algorithms is convergent. Thus the proximal parameter $L$ is required to be greater than $\frac{2}{n} \| A^T A \|$. Through numerical experiments, we observe that when $L = \frac{2}{n} \| A^T A \|$, Algorithms 5, 6 and 7 can achieve almost the best numerical performance. Thus we set $L = \frac{2}{n} \| A^T A \|$ for these three algorithms.

**Initialization** We set the initial solution for all these five algorithms as $U^0 = SS^T$ with a randomly generated matrix $S \in \mathbb{R}^{n \times r}$.

**Termination criterion** The termination conditions of PDCA, PG-DCA and PDCAe are set as $|J_c(U^{k+1}) - J_c(U^k)| / |1 + J_c(U^k)| \leq 10^{-7}$. The termination condition of s-iPDCA is set as $|J_c(U^{k+1}) - J_c(U^k)| / |1 + J_c(U^k)| \leq 10^{-7}$. The termination condition of PG is set as $|J(U^{k+1}) - J(U^k)| / |1 + J(U^k)| \leq 10^{-7}$. When rank($U$) $\leq r$, $J_c(U) = J(U)$, then these termination conditions are consistent.

As a comparison, numerical results for the total solving time (t/s), outer iterations (Iter) and optimal value of the least squares function $J(U) = \frac{1}{n} \| A(U) - b \|^2$ ($J$) obtained by PDCA, s-iPDCA, PG method, PG-DCA and PDCAe are presented in Table 1. These results are averaged over 20 runs of tests of these five algorithms. As a result from Table 1, one can see that our s-iPDCA outperforms the PDCA, PG method, PG-DCA and PDCAe for solving (63) from computation time and optimal objective function value. The optimal value of s-iPDCA is smaller than that of other methods. Although the number of total iteration of s-iPDCA is greater than that of PDCA in most cases, its total computation time is much less than that of PDCA, which verifies that our inexact strategy is efficient. Furthermore, although the subproblems of the PG, PG-DCA and PDCAe have closed-form solutions, they take more computation time and iterations than s-iPDCA to solve (63) for all data sets.

### 5.2 Dimensionality Reduction for Face Recognition

In this subsection, two numerical experiments for face recognition are performed to display the advantage of our model in practical application. The dimensionality reduction performance of RCKSDPP is compared with KSDPP and KPCA. The convex problem in (62) is solved by the boundary point method [46]. The RCLSSDP in (63) is solved by s-iPDCA. We divide the face images database into training set and testing set randomly. Then we proceed the following steps for face recognition. Firstly, a projected matrix is obtained from RCKSDPP, KSDPP or KPCA. Secondly, we reduce the dimension of face images in both training set and testing set by applying the above projected matrix. Thirdly, we use the nearest neighbor method as classifier to identify whether the projected covariate belongs to an individual.

**Details of experimental data** Two face databases, ORL and YaleB, are used in our experiments. The ORL database contains 400 images from 40 individuals, where each individual has 10 different images. The size of each image is $92 \times 112$. For each individual, the face in the images are rotated, scaled or tilted to a mild degree. In addition, we only extract the subset of YaleB database that containing 2,414 frontal pose images of 38 individuals under different illuminations for each individual. We crop all images from YaleB database into $168 \times 192$ pixels. The image set is partitioned into different training set and testing set. We use the $G_p/T_q$ to indicate that $p$ face images per individual are randomly selected for training and $q$ face images from the remaining are used for testing.
| n   | PDCA   | s-iPDCA | PG    | PG-DCA  | PDCAe  |
|-----|--------|---------|-------|---------|--------|
|     | J      | Iter    | t/s   | J       | Iter    | t/s   | J      | Iter    | t/s   | J      | Iter    | t/s   |
| 100 | 1.25e-4| 79      | 3.89  | 1.04e-4 | 88      | 1.80  | 5.40e-3 | 513     | 1.92  | 5.79e-3 | 2976    | 17.22 | 1.14e-3 | 447     | 3.04  |
| 200 | 8.32e-5| 105     | 13.24 | 8.05e-5 | 100     | 4.94  | 1.71e-3 | 265     | 4.96  | 1.63e-3 | 465     | 11.17 | 3.09e-4 | 185     | 5.30  |
| 400 | 1.38e-3| 157     | 80.85 | 1.37e-3 | 155     | 41.96 | 4.52e-2 | 1488    | 110.67| 5.04e-2 | 2604    | 249.79| 4.30e-3 | 511     | 58.03 |
| 600 | 3.00e-3| 119     | 133.40| 3.01e-3 | 119     | 71.10 | 7.37e-2 | 1366    | 245.89| 6.95e-2 | 1738    | 386.52| 6.27e-3 | 450     | 106.79|
| 800 | 5.88e-3| 130     | 329.23| 5.88e-3 | 131     | 159.17| 1.09e-1 | 1184    | 460.44| 1.15e-2 | 1156    | 546.29| 9.32e-2 | 468     | 226.75|
| 1000| 7.91e-3| 146     | 499.55| 7.91e-3 | 146     | 239.34| 1.04e-1 | 1499    | 1007.48| 1.10e-2 | 1487    | 1215.03| 1.23e-2 | 435     | 362.83|
| 1200| 1.25e-2| 187     | 893.41| 1.25e-2 | 201     | 419.87| 1.28e-1 | 1367    | 1539.96| 1.28e-1 | 1387    | 1850.46| 1.77e-2 | 409     | 593.16|
| 1440| 1.40e-2| 208     | 1043.01| 1.40e-2 | 177     | 554.37| 8.49e-2 | 961     | 1969.41| 8.37e-2 | 967     | 2385.17| 1.44e-2 | 329     | 815.78|

Bold values indicate the best among the results in the same data group.
Parameters setup for ORL. In order to achieve higher recognition accuracy, we perform tests to find a optimum projected dimension \( r \). In this experiment, we set \( r = 40 \). To ensure numerical efficiency and recognition accuracy, the suitable neighborhood size of \( k \)-nn is set as \( k = \text{round}(\log(n)) \) for both RCKSDPP and KSDPP. It is noted that a suitable kernel parameter \( \varsigma \) is important for these three models to obtain high recognition accuracy. In this experiment, we set \( \varsigma = 25 \) for all of these three models. The parameters of s-iPDCA are set as: \( c_0 = 10^{-6}, \kappa = 0.1, \alpha = 3 \times 10^{-5}, \epsilon_0 = 10^{-1} \). We adjust the sequence \( \{\epsilon_k\} \) as Sect. 5.1 and set \( \rho_1 = 0.99, \rho_2 = 0.9 \). The termination condition of s-iPDCA is set as \( \frac{\|V^{k+1} - U^k\|_F}{1 + \|U^k\|_F} \leq 5 \times 10^{-4} \). The tolerance error of the boundary point method for KSDPP is set as \( \epsilon = 10^{-5} \).

Parameters setup for YaleB. Compared with ORL database, YaleB database has different illuminations, which makes it difficulty to perform recognition task on the YaleB database. To achieve higher recognition rate, we set the projected dimension \( r = 45 \). In order to get recognition results as quickly as possible, 20 individuals are randomly selected for face recognition in each run of the RCKSDPP, KSDPP and KPCA. The neighborhood size of the \( k \)-nn for RCKSDPP and KSDPP are all set as \( k = 3 \). The most suitable value of \( \varsigma \) is set for each model: \( \varsigma = 7 \) for KSDPP and RCKSDPP, while \( \varsigma = 2000 \) for KPCA. The parameters of s-iPDCA are set as \( c_0 = 10^{-6}, \epsilon_0 = 10^{-1}, \alpha = 5 \times 10^{-6} \) and termination condition \( \frac{\|V^{k+1} - U^k\|_F}{1 + \|U^k\|_F} \leq 5 \times 10^{-4} \). The tolerance error of the boundary point method for KSDPP is set as \( \epsilon = 10^{-5} \).

The average recognition accuracy and the standard deviation across 50 runs of tests of the KSDPP, KPCA and RCKSDPP are computed for these two databases. As a comparison, recognition results on ORL and YaleB for average recognition accuracy \( \pm \) standard deviation \((Re \pm std)\) and average solving time (t/s) obtained by these three models are presented in Tables 2 and 3, respectively. In addition, the results for average optimal value of the least squares function \( J(U) = \frac{1}{n} \|A(U) - b\|^2 \) \((J)\) and average outer iterations (Iter) obtained by s-iPDCA and the boundary point method are displayed in Tables 2 and 3. As a result, from Tables 2 and 3, one can see that RCKSDPP outperforms KSDPP and KPCA for all situations. Furthermore, for each model, the larger the training set is, the higher the recognition accuracy is obtained. The comparison results on YaleB illustrates that the unsupervised method, KPCA, has much lower recognition accuracy than RCKSDPP and KSDPP. Moreover, RCKSDPP is more powerful to reduce the dimension of complex data than KSDPP. This results demonstrate that the RCKSDPP is effective to reduce the dimension of image data. Compared with KSDPP, the solution of RCKSDPP satisfies the rank constraint \( \text{rank}(U^*) \leq r \). Then the projection matrix \( P^* \) obtained by performing square root decomposition, or spectral decomposition on \( U^* \) is one of the optimal solutions of (61). That’s the reason that RCKSDPP has higher recognition accuracy than KSDPP.

6 Conclusion

In this paper, we reformulated the optimization problem of supervised distance preserving projection (SDPP) for data dimensionality reduction into a rank constrained least squares semidefinite programming (RCLSSDP). To address the difficulty brought by the rank constraint, we employed the exact penalty approach to transfer RCLSSDP into a difference-of-convex (DC) programming. To solving the corresponding DC programming, an inexact proximal DC algorithm with sieving strategy (s-iPDCA) was proposed, whose subproblems...
Table 2  Dimensionality reduction results on the ORL database

| Partitions | KSDPP | RCKSDPP | KPCA |
|------------|-------|---------|------|
|            | Re ± std | J | Iter | t/s | Re ± std | J | Iter | t/s | Re ± std | t/s |
| $G_2/T_3$  | 80.15 ± 3.26 | 6.53e-2 | 1523 | 19.47 | **84.97 ± 1.96** | 1.12e-4 | 186 | 6.72 | 79.75 ± 2.41 | 0.95 |
| $G_3/T_7$  | 88.11 ± 2.56 | 9.88e-2 | 1242 | 21.67 | **92.14 ± 1.87** | 8.25e-4 | 289 | 20.97 | 84.67 ± 12.08 | 0.98 |
| $G_4/T_6$  | 92.32 ± 1.80 | 1.33e-1 | 1366 | 38.33 | **93.58 ± 1.49** | 1.76e-3 | 330 | 30.20 | 88.59 ± 12.58 | 1.00 |
| $G_5/T_5$  | 95.20 ± 0.76 | 1.245e-1 | 1840 | 72.43 | **95.90 ± 1.74** | 2.87e-3 | 336 | 42.77 | 93.39 ± 2.16 | 1.01 |
| $G_6/T_4$  | 96.37 ± 1.49 | 1.47e-1 | 1751 | 92.93 | **97.19 ± 1.29** | 4.34e-3 | 352 | 47.92 | 95.01 ± 1.63 | 1.03 |
| $G_7/T_3$  | 97.00 ± 0.75 | 2.14e-1 | 1710 | 158.48 | **98.33 ± 1.18** | 8.31e-3 | 634 | 70.24 | 96.53 ± 1.64 | 1.04 |
| $G_8/T_2$  | 98.02 ± 1.62 | 2.11e-1 | 1276 | 160.97 | **98.75 ± 0.59** | 1.01e-2 | 356 | 76.87 | 97.15 ± 2.02 | 1.05 |

Bold values indicate the best among the results in the same data group.
| Partitions | KSDPP | | | | RCKSDPP | | | | KPCA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Re ± std | J | Iter | t/s | Re ± std | J | Iter | t/s | Re ± std | t/s |
| $G_{10}/T_{50}$ | 68.24 ± 1.67 | 7.13e-2 | 1918 | 42.05 | 75.80 ± 2.44 | 3.77e-3 | 345 | 35.88 | 55.43 ± 2.57 | 1.52 |
| $G_{20}/T_{40}$ | 79.98 ± 2.53 | 1.30e-1 | 2093 | 169.73 | 91.10 ± 1.90 | 8.87e-3 | 337 | 84.31 | 67.13 ± 2.63 | 1.81 |
| $G_{30}/T_{30}$ | 85.70 ± 1.76 | 1.52e-1 | 2212 | 379.76 | 93.97 ± 1.29 | 1.22e-2 | 294 | 123.09 | 71.35 ± 2.65 | 2.15 |
| $G_{40}/T_{20}$ | 87.10 ± 1.76 | 1.50e-1 | 2389 | 702.58 | 95.20 ± 0.92 | 1.86e-2 | 252 | 148.66 | 74.33 ± 3.00 | 2.46 |
| $G_{50}/T_{10}$ | 87.50 ± 2.89 | 1.91e-1 | 2476 | 1494.51 | 95.65 ± 1.60 | 2.33e-2 | 232 | 189.96 | 74.57 ± 3.74 | 2.80 |

Bold values indicate the best among the results in the same data group.
were solved by an efficient accelerated block coordinate descent (ABCD) method from their dual. In addition, the low rank structure of solution was fully utilized to reduce the storage and computation cost of ABCD and s-iPDCA. Moreover, the global convergence of the sequence generated by s-iPDCA was proven.

In numerical experiments, we compared our s-iPDCA with the classical proximal DC algorithm, proximal gradient method, proximal gradient-DC algorithm and proximal DC algorithm with extrapolation for solving RCLSSDP. The results demonstrate that s-iPDCA outperforms the other methods from computation time and quality of solution. Moreover, dimensionality reduction experiments for face recognition on the ORL and YaleB databases demonstrate that rank constrained kernel SDPP is efficient and competitive when comparing with kernel semidefinite SDPP and kernel principal component analysis in terms of recognition accuracy.

Acknowledgements The authors would like to thank the Associate Editor and anonymous referees for their helpful suggestions. The Bo Yu’s and Mingcai Ding’s work was supported by the National Natural Science Foundation of China (Grand No. 11971092). Xiaoliang Song’s work was supported by the Fundamental Research Funds for the Central Universities (Grand No. DUT20RC(3)079).

Funding The authors have not disclosed any funding.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interest.

Appendix A Derivations of (2) and (62)

Firstly, we give the derivation of (2). Let $\mathbf{U} = \mathbf{PP}^\top$, then the SDP relaxation of (1) can be formulated as

$$
\min J(\mathbf{U}) = \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{G}_{i,j} \left( \langle \mathbf{U}, (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \rangle - \| \mathbf{y}_i - \mathbf{y}_j \|^2 \right)^2. \tag{A1}
$$

Suppose the number of non-zero elements in $i$-th row of the graph matrix $\mathbf{G}$ is $n_i$ and $\mathbf{G}_{i,jk} = 1$, $k = 1, \cdots, n_i$, then (A1) can be simplified to

$$
\min_{\mathbf{U} \in \mathbb{S}^d_+} J(\mathbf{U}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{n_i} \left( \langle \mathbf{U}, (\mathbf{x}_i - \mathbf{x}_{jk})(\mathbf{x}_i - \mathbf{x}_{jk})^\top \rangle - \| \mathbf{y}_i - \mathbf{y}_{jk} \|^2 \right)^2. \tag{A2}
$$

Let $l := \sum_{s=1}^{i-1} n_s + k$ be the index of the $k$-th non-zero element in $i$-th row of $\mathbf{G}$. Let $\mathbf{\tau}_l = \mathbf{x}_i - \mathbf{x}_{jk}$ and $\mathbf{b}_l = \| \mathbf{y}_i - \mathbf{y}_{jk} \|^2$. Then (A1) can be formulated as

$$
\min_{\mathbf{U} \in \mathbb{S}^d_+} J(\mathbf{U}) = \frac{1}{n} \sum_{l=1}^{p} \left( \langle \mathbf{U}, \mathbf{\tau}_l \mathbf{\tau}_l^\top \rangle - \| \mathbf{b}_l \|^2 \right)^2, \tag{A3}
$$

where $p = \sum_{i,j=1}^{n} \mathbf{G}_{i,j}$. Clearly, (A3) can be formulated into a least squares form:

$$
\min_{\mathbf{U} \in \mathbb{S}^d_+} J(\mathbf{U}) = \frac{1}{n} \| \mathbf{A}(\mathbf{U}) - \mathbf{b} \|^2, \tag{A4}
$$
where $\mathcal{A} : S_+^d \rightarrow \mathbb{R}^p$ is a linear operator that can be explicitly represented as

$$
\mathcal{A}(U) = [\langle A_1, U \rangle, \langle A_2, U \rangle, \ldots, \langle A_p, U \rangle]^T, \quad \mathcal{A}_l = \tau_l \tau_l^T, \quad l = 1, 2, \ldots, p.
$$

Then $\mathcal{A}(U)$ can be computed as

$$
\mathcal{A}(U) = [\tau_1^T U \tau_1, \tau_2^T U \tau_2, \ldots, \tau_p^T U \tau_p]^T,
$$

its computation cost is $O((d^2 + d)p)$.

It is noted that the least squares SDP in (62) can be obtained from (61) by the same way, then we omit its derivation.

**Appendix B Derivation of (19)**

The subproblem of Algorithm 3 can be formulated as

$$
\min_U \frac{1}{n} \| \mathcal{A}(U) - b \|^2 + c\langle U, I \rangle - \langle U, W^k \rangle + \delta_{S_+^d}(U) + \frac{\alpha}{2} \| U - U^k \|^2_F. \quad (B5)
$$

By ignoring the constant term, (B5) can be simplified as

$$
\min_U \frac{1}{n} \| \mathcal{A}(U) - b \|^2 - \langle U, \Phi_c^k \rangle + \delta_{S_+^d}(U) + \frac{\alpha}{2} \| U \|^2_F, \quad (B6)
$$

where $\Phi_c^k = W^k + \alpha U^k - cI$. To obtain the dual problem, we introduce two auxiliary variables: $s$ and $T$ such that (B6) is equivalent to

$$
\min_U \frac{1}{n} \| s \|^2 - \langle U, \Phi_c^k \rangle + \delta_{S_+^d}(T) + \frac{\alpha}{2} \| U \|^2_F
$$

s.t. $\mathcal{A}(U) - b = s$, $U = T$. \quad (B7)

The Lagrange function of (B7) can be expressed as

$$
\mathcal{L}(U, T, s; Y, z) = \frac{1}{n} \| s \|^2 - \langle U, \Phi_c^k \rangle + \delta_{S_+^d}(T) + \frac{\alpha}{2} \| U \|^2_F + z^T (\mathcal{A}(U) - b - s) + \langle Y, T - U \rangle. \quad (B8)
$$

Then the objective function of the dual problem of (B5) can be obtained by

$$
\inf_{U, T, s} \mathcal{L}(U, T, s; Y, z)
$$

$$
= -\frac{n}{4} \| z \|^2 - z^T b - \frac{1}{2\alpha} \| A^*(z) - \Phi_c^k - Y \|^2_F - \delta_{S_+^d}^*(\mathcal{A}^*(z) - \Phi_c^k - Y), \quad (B9)
$$

where the last equality uses the fact that $\delta_{S_+^d}^*(\mathcal{-Y}) = \delta_{S_+^d}(-Y) = \delta_{S_+^d}^*(Y)$. Then the dual problem of (B6) can be expressed as

$$
\max_{Y, z} -\frac{n}{4} \| z \|^2 - z^T b - \frac{1}{2\alpha} \| A^*(z) - \Phi_c^k - Y \|^2_F - \delta_{S_+^d}(Y). \quad (B10)
$$
Thus the dual problem of (B5) can be equivalently formulated as the following minimization problem:

$$\min_{Y, z} \frac{n}{4} \|z\|^2 + z^{T} b + \frac{1}{2\alpha} \|A^*(z) - \Phi_{c}^{k} - Y\|^2_{F} + \delta_{S_{+}^{d}} (Y).$$  \hfill (B11)

### Appendix C Proof of the Statements (1)–(5) in Proposition 7

For statement (1), since $$U_{k+1}^{k+1} = V^{k+1}$$ is the stability center generated in serious step, then the condition (18) holds, shown as

$$\|\Delta_{k+1}\|_{F} \leq (1 - \kappa)\|U_{k+1}^{k} - U_{k}^{k}\|_{F}.$$  \hfill (B12)

Then we have

$$\kappa \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} \leq \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} - \langle \Delta_{k+1}, U_{k+1}^{k} - U_{k}^{k}\rangle.$$  \hfill (B13)

Consequently,

$$E(U_{k+1}^{k}, W^{k}, U_{k}^{k}, \Delta_{k+1}^{k})$$

$$= f_{c}(U_{k+1}^{k}) - \langle U_{k+1}^{k}, W^{k}\rangle + g_{c}^{*}(W^{k})$$

$$+ \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} - \langle \Delta_{k+1}^{k}, U_{k+1}^{k} - U_{k}^{k}\rangle$$

$$= f_{c}(U_{k+1}^{k}) - \langle U_{k+1}^{k} - U_{k}^{k}, W^{k}\rangle - g_{c}(U_{k}^{k})$$

$$+ \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} - \langle \Delta_{k+1}^{k}, U_{k+1}^{k} - U_{k}^{k}\rangle$$

$$\geq f_{c}(U_{k+1}^{k}) - g_{c}(U_{k+1}^{k}) + \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} - \langle \Delta_{k+1}^{k}, U_{k+1}^{k} - U_{k}^{k}\rangle$$

$$\geq f_{c}(U_{k+1}^{k}) - g_{c}(U_{k+1}^{k}) + \kappa \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} \geq J_{c}(U_{k+1}^{k}),$$

where the second equality follows from the convexity of $$g_{c}$$ and the fact that $$W^{k} \in \partial g_{c}(U^{k})$$, and the first inequality follows from the convexity of $$g_{c}$$.

For statement (2), since $$U_{k+1}^{k+1} \in S_{+}^{d}$$ is an inexact solution of (15) with inexact term $$\Delta_{k+1}^{k}$$, we have

$$U_{k+1}^{k} = \arg \min_{U} G_{c}^{k}(U) - \langle \Delta_{k+1}^{k}, U\rangle.$$  

Since $$G_{c}^{k}(U) - \langle \Delta_{k+1}^{k}, U\rangle$$ is strongly convex, then the following inequality holds:

$$f_{c}(U_{k+1}^{k}) - \langle U_{k+1}^{k}, W^{k}\rangle + \frac{\alpha}{2} \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} - \langle \Delta_{k+1}^{k}, U_{k+1}^{k}\rangle$$

$$\leq f_{c}(U^{k}) - \langle U^{k}, W^{k}\rangle - \langle \Delta_{k+1}^{k}, U^{k}\rangle - \frac{\alpha}{2} \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2}.$$  \hfill (B14)

Thus, we have

$$E(U_{k+1}^{k}, W^{k}, U_{k}^{k}, \Delta_{k+1}^{k})$$

$$= f_{c}(U_{k+1}^{k}) - \langle U_{k+1}^{k}, W^{k}\rangle + g_{c}^{*}(W^{k})$$

$$+ \alpha \|U_{k+1}^{k} - U_{k}^{k}\|_{F}^{2} - \langle \Delta_{k+1}^{k}, U_{k+1}^{k} - U_{k}^{k}\rangle$$

$$\leq f_{c}(U^{k}) - \langle U^{k}, W^{k}\rangle + g_{c}^{*}(W^{k}) = f_{c}(U^{k}) - g_{c}(U^{k}),$$

where the last equality follows from the convexity of $$g_{c}$$ and the fact that $$W^{k} \in \partial g_{c}(U^{k})$$.

Similar to (B13), the following inequality holds:

$$\kappa \alpha \|U_{k}^{k} - U_{k-1}^{k}\|_{F}^{2} \leq \alpha \|U_{k}^{k} - U_{k-1}^{k}\|_{F}^{2} - \langle \Delta_{k}^{k}, U_{k}^{k} - U_{k-1}^{k}\rangle.$$  \hfill (B15)
Consequently, we have
\[ E(U^{k+1}, W^k, U^k, \Delta^{k+1}) \leq f_c(U^{k+1}) - g_c(U^k) \]
\[ \leq f_c(U^k) - (U^k, W^{k-1}) + g_c^*(W^{k-1}) \]
\[ = E(U^k, W^{k-1}, U^{k-1}, \Delta^{k-1}) - \alpha\|U^k - U^{k-1}\|_F^2 + \alpha \Delta^{k-1} \]
\[ \leq E(U^k, W^{k-1}, U^{k-1}, \Delta^{k-1}) - \kappa\alpha\|U^k - U^{k-1}\|_F^2, \]
where the second inequality follows from the convexity of \( g_c \) and the Young’s inequality applied to \( g_c \). The last inequality is due to (B15).

For statement (3), we first note from Proposition 6 that \{\( U^k \)\} is bounded. The boundedness of \{\( W^k \)\} follows immediately from the finite-valued property and the convexity of \( g_c \) and the fact that \( W^k \in \partial g_c(U^k) \). The boundedness of \{\( \Delta^k \)\} is followed by the fact that \( \lim_{l \to \infty} \epsilon_{k+l} = 0 \). Then, the bounded sequence \{\( (U^{k+1}, W^k, U^k, \Delta^{k+1}) \)\} has nonempty accumulation point set \( \Omega \).

For statement (4), since \( J_c \) is bounded below, from (54) and (55), we have that \( E(U^{k+1}, W^k, U^k, \Delta^{k+1}) \) is nonincreasing and bounded below. Thus, the limit \( \gamma = \lim_{l \to \infty} E(U^{k+1}, W^k, U^k, \Delta^{k+1}) \) exists. Next, we will prove that \( E = \gamma \) on \( \Omega \). Take any \((\hat{U}, \hat{W}, \hat{U}, \hat{\Delta}) \in \Omega \). Since the above limit exists, there exists a subset \( L' \subset L \) such that
\[ \lim_{l \in L'} (U^{k+1}, W^k, U^k, \Delta^{k+1}) = (\hat{U}, \hat{W}, \hat{U}, \hat{\Delta}). \]

From the optimality of \( U^{k+1} \) and the feasibility of \( \hat{U} \) for solving \( \min U G_c^*(U) \), we have
\[ f_c(U^{k+1}) - (U^{k+1}, W^{k+1}) + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 \leq f_c(U^k) - (U^k, W^{k+1}) + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 \]
\[ \leq f_c(U^k) - (U^k, W^{k+1}) + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 - (U^k, \Delta^{k+1}). \]

Rearranging terms in the above inequality, we obtain that
\[ f_c(U^{k+1}) - (U^{k+1} - \hat{U}, W^k + \Delta^{k+1}) + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 \leq f_c(U^k) + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2. \]

From the boundedness of \{\( U^k \)\}, \{\( W^k \)\} and \{\( \Delta^k \)\}, we have
\[ \lim_{l \in L'} (U^{k+1} - \hat{U}, W^k) = 0, \lim_{l \in L'} (U^{k+1} - \hat{U}, \Delta^{k+1}) = 0. \]

Then, we have
\[ \gamma = \lim_{l \in L'} E(U^{k+1}, W^k, U^k, \Delta^{k+1}) \]
\[ = \lim_{l \in L'} f_c(U^{k+1}) - (U^{k+1}, W^k) + g_c^*(W^k) \]
\[ + \alpha\|U^{k+1} - U^k\|_F^2 - (\Delta^{k+1}, U^{k+1} - U^k) \]
\[ = \lim_{l \in L'} f_c(U^{k+1}) - (U^{k+1} - \hat{U}, W^k + \Delta^{k+1}) + \alpha\|U^{k+1} - U^k\|_F^2 \]
\[ - (U^{k+1}, W^k) + g_c^*(W^k) - (\Delta^{k+1}, U^{k+1} - U^k) \]
\[ \leq \lim_{l \in L'} f_c(U^k) + \frac{\alpha}{2}\|\hat{U} - U^k\|_F^2 + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 - (U^{k+1}, W^k) \]
\[ + g_c^*(W^k) - (\Delta^{k+1}, U^{k+1} - U^k) \]
\[ \leq f_c(U^k) + \frac{\alpha}{2}\|\hat{U} - U^k\|_F^2 + \frac{\alpha}{2}\|U^{k+1} - U^k\|_F^2 - (U^{k+1}, W^k) \]
\[ + g_c^*(W^k) - (\Delta^{k+1}, U^{k+1} - U^k) \]
$$\begin{align*}
&= \limsup_{l \in \mathcal{L}'} f_c(\tilde{U}) + \frac{\alpha}{2}\|\tilde{U} - U^{k_l}\|^2_F + \frac{\alpha}{2}\|U^{k_{l+1}} - U^{k_l}\|^2_F
&\quad - \langle U^{k_{l+1}} - U^{k_l}, W^{k_l}\rangle - g_c(U^{k_l}) - \langle \Delta^{k_{l+1}}, U^{k_{l+1}} - U^{k_l}\rangle
&= f_c(\tilde{U}) - g_c(\tilde{U}) = J_c(\tilde{U}) \leq E(\tilde{U}, \tilde{W}, \tilde{U}, \tilde{\Delta}),
\end{align*}$$

where the fourth equality follows from the convexity of $g_c$ and $W^{k_l} \in \partial g_c(U^{k_l})$, and the last inequality holds from (54) with $l$ trending to infinity. Since $E$ is lower semicontinuous, we also have

$$E(\tilde{U}, \tilde{W}, \tilde{U}, \tilde{\Delta}) = \liminf_{l \in \mathcal{L}'} E(U^{k_{l+1}}, W^{k_l}, U^{k_l}, \Delta^{k_{l+1}}) = \gamma.$$

Thus, $E \equiv \gamma$ on $\Omega$.

For statement (5), since the subdifferential of the function $E$ at the point $(U^{k_{l+1}}, W^{k_l}, U^{k_l}, \Delta^{k_{l+1}})$ is

$$\partial E(U^{k_{l+1}}, W^{k_l}, U^{k_l}, \Delta^{k_{l+1}}) = \left[ \nabla f_c(U^{k_{l+1}}) - W^{k_l} + 2\alpha(U^{k_{l+1}} - U^{k_l}) - \Delta^{k_{l+1}} + \partial \delta_S^c(U^{k_{l+1}}) \right],$$

Since $U^{k_{l+1}}$ is the optimal solution of (16), we have

$$0 \in \nabla f_c(U^{k_{l+1}}) - W^{k_l} + \alpha(U^{k_{l+1}} - U^{k_l}) - \Delta^{k_{l+1}} + \partial \delta_S^c(U^{k_{l+1}}).$$

Since $U^{k_l} \in \partial g_c^*(W^{k_l})$, then

$$\begin{bmatrix}
\alpha(U^{k_{l+1}} - U^{k_l}) \\
\Delta^{k_{l+1}} \\
-\alpha(U^{k_{l+1}} - U^{k_l}) + \Delta^{k_{l+1}} \\
U^{k_l} - U^{k_{l+1}}
\end{bmatrix} \in \partial E(U^{k_{l+1}}, W^{k_l}, U^{k_l}, \Delta^{k_{l+1}})$$

(B18)

Since $U^{k_{l+1}}$ is the stability center of serious step, so it satisfies the test (18), i.e., $\|\Delta^{k_{l+1}}\|_F \leq (1 - \kappa)\alpha\|U^{k_{l+1}} - U^{k_l}\|_F$. Thus there exists a constant $\rho$ such that the following inequality holds:

$$\text{dist}(0, \partial E(U^{k_{l+1}}, W^{k_l}, U^{k_l}, \Delta^{k_{l+1}})) \leq \rho\|U^{k_{l+1}} - U^{k_l}\|_F$$

(B19)

This completes the proof.

\[ \square \]

References

1. Zhu, Z.-X., Similä, T., Corona, F.: Supervised distance preserving projections. Neural Process. Lett. 38(3), 445–463 (2013)
2. Jahan, S.: On dimension reduction using supervised distance preserving projection for face recognition. Univ. J. Appl. Math. 6(3), 94–105 (2018)
3. Gao, Y.: Structured low rank matrix optimization problems: a penalty approach. PhD thesis, National University of Singapore (2010)
4. Gao, Y., Sun, D.-F.: A majorized penalty approach for calibrating rank constrained correlation matrix problems. http://www.math.nus.edu.sg/~matsundf/MajorPen_May5.pdf (2010)
5. Qi, H.-D., Yuan, X.-M.: Computing the nearest euclidean distance matrix with low embedding dimensions. Math. Program. 147(1), 351–389 (2014)
6. Singer, A.: A remark on global positioning from local distances. Proc. Natl. Acad. Sci. **105**(28), 9507–9511 (2008)
7. Shang, Y., Rumi, W., Zhang, Y., Fromherz, M.: Localization from connectivity in sensor networks. IEEE Trans. Parallel Distrib. Syst. **15**(11), 961–974 (2004)
8. Torgerson, W.S.: Multidimensional scaling: I. theory and method. Psychometrika **17**(4), 401–419 (1952)
9. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The computational complexity of some problems of linear algebra. J. Comput. Syst. Sci. **58**(3), 572–596 (1999)
10. Candès, E.J., Plan, Y.: Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. IEEE Trans. Inf. Theory **57**(4), 2342–2359 (2011)
11. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
12. Toh, K.-C., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific J. Opt. **6**(615–640), 15 (2010)
13. Jiang, K.-F., Sun, D.-F., Toh, K.-C.: A partial proximal point algorithm for nuclear norm regularized matrix least squares problems. Math. Program. Comput. **6**(3), 281–325 (2014)
14. Cai, T.T., Zhang, A.: Sparse representation of a polytope and recovery of sparse signals and low-rank matrices. IEEE Trans. Inf. Theory **60**(1), 122–132 (2014)
15. Lee, J., Recht, B., Salakhutdinov, R.R., Srebro, N., Tropp, J.A.: Practical large-scale optimization for max-norm regularization. Neural Inform. Process. Syst. **23**, 1297–1305 (2010)
16. Bi, S.-J., Pan, S.-H.: Error bounds for rank constrained optimization problems and applications. Oper. Res. Lett. **44**(3), 336–341 (2016)
17. Gotoh, J.Y., Takeda, A., Tono, K.: Dc formulations and algorithms for sparse optimization problems. Math. Program. **169**(1), 141–176 (2018)
18. Jiang, Z.-X., Zhao, X.-Y., Ding, C.: A proximal dc approach for quadratic assignment problem. Comput. Optim. Appl. **67**(2), 825–851 (2021)
19. Tao, P.D., An, L.T.H.: Convex analysis approach to dc programming: theory, algorithms and applications. Acta Math. Vietnam **22**(1), 289–355 (1997)
20. An, L.T.H., Tao, P.D.: The dc (difference of convex functions) programming and DCA revisited with dc models of real world nonconvex optimization problems. Ann. Oper. Res. **133**(1–4), 23–46 (2005)
21. Le Thi, H.A., Pham Dinh, T., Muu, L.D.: Exact penalty in d.c. programming. Vietnam J. Math. **27**(2), 169–178 (1999)
22. Le Thi, H.A., Pham Dinh, T., Van Ngai, H.: Exact penalty and error bounds in dc programming. J. Global Optim. **52**(3), 509–535 (2012)
23. Souza, J.C.O., Oliveira, P.R., Soubeyran, A.: Global convergence of a proximal linearized algorithm for difference of convex functions. Optim. Lett. **10**(7), 1529–1539 (2016)
24. Gaudioso, M., Giallombardo, G., Miglionico, G., Bagirov, A.M.: Minimizing nonsmooth dc functions via successive dc piecewise-affine approximations. J. Global Optim. **71**(1), 37–55 (2018)
25. de Oliveira, W.: Proximal bundle methods for nonsmooth dc programming. J. Global Optim. **75**(2), 523–563 (2019)
26. Liu, T.-X., Pang, T.-K., Takeda, A.: A refined convergence analysis of pdcae with applications to simultaneous sparse recovery and outlier detection. Comput. Optim. Appl. **73**(1), 69–100 (2019)
27. Wen, B., Chen, X.-J., Pang, T.-K.: A proximal difference-of-convex algorithm with extrapolation. Comput. Optim. Appl. **69**(2), 297–324 (2018)
28. Sun, D.-F., Toh, K.-C., Yuan, Y.-C., Zhao, X.-Y.: Sdmap+: a matlab software for semidefinite programming with bound constraints (version 1.0). Optim. Methods Softw. **35**(1), 87–115 (2020)
29. Bai, S., Qi, H.-D., Xiu, N.: Constrained best euclidean distance embedding on a sphere: a matrix optimization approach. SIAM J. Optim. **25**(1), 439–467 (2015)
30. Liu, T., Lu, Z., Chen, X., Dai, Y.-H.: An exact penalty method for semidefinite-box-constrained low-rank matrix optimization problems. IMA J. Numer. Anal. **40**(1), 563–586 (2020)
31. Jiang, K.-F., Sun, D.-F., Toh, K.-C.: An inexact accelerated proximal gradient method for large scale linearly constrained convex sdp. SIAM J. Optim. **22**(3), 1042–1064 (2012)
32. Sun, D.-F., Toh, K.-C., Yang, L.-Q.: An efficient inexact abc method for least squares semidefinite programming. Mathematics **26**(2), 1072–1100 (2015)
33. Wang, Y.-Y., Liu, R.-S., Ma, L., Song, X.-L.: Task embedded coordinate update: a realizable framework for multivariate non-convex optimization. Proc. AAAI Conf. Art. Intell. **33**, 277–286 (2019)
34. Beck, A.: First-order Methods in Optimization. SIAM, Philadelphia (2017)
35. Candès, E.J.: The restricted isometry property and its implications for compressed sensing. C.R. Math. **346**(9–10), 589–592 (2008)
36. Luo, Y.-T., Huang, W., Li, X.-D., Anru R, Z.: Recursive importance sketching for rank constrained least squares: algorithms and high-order convergence. arXiv preprint arXiv:2011.08360 (2020)
37. Bhojanapalli, S., Neyshabur, B., Srebro, N.: Global optimality of local search for low rank matrix recovery. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 3880–3888 (2016)
38. Hiriart, J.B.: Convex Analysis and Minimization Algorithms II. Springer, New York (1993)
39. Mishra, B.: Algorithmic algebra. Springer, New York (1993)
40. Bental, A., Nemirovski, A., Overton, M.: Lectures on modern convex optimization: analysis, algorithms, and engineering applications. SIAM, New York (2001)
41. Ioffe, A.: An invitation to tame optimization. SIAM J. Optim. 19(4), 1894–1917 (2009)
42. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Math. Program. 116(1), 5–16 (2009)
43. Bolte, J., Pauwels, E.: Majorization-minimization procedures and convergence of sqp methods for semi-algebraic and tame programs. Math. Oper. Res. 41(2), 442–465 (2016)
44. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. 146(1), 459–494 (2014)
45. Georgiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. 23(6), 643–660 (2001)
46. Povh, J., Rendl, F., Wiegele, A.: A boundary point method to solve semidefinite programs. Computing 78(3), 277–286 (2006)
47. Lu, Z., Zhang, Y.: Sparse approximation via penalty decomposition methods. SIAM J. Optim. 23(4), 2448–2478 (2012)
48. Lu, Z., Zhang, Y., Li, X.: Penalty decomposition methods for rank minimization. Optim. Methods Softw. 30(3), 531–558 (2014)

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