Distributed Optimization for Rank-Constrained Semidefinite Programs

Chaoying Pei, Sixiong You, Chuangchuang Sun, and Ran Dai, *Member, IEEE*

***Abstract*—This letter develops a distributed optimization framework for solving the rank-constrained semidefinite programs (RCSPs). Since the rank constraint is non-convex and discontinuous, solving an optimization problem with rank constraints is NP-hard and notoriously time-consuming, especially for large-scale RCSPs. In the proposed approach, by decomposing an unknown matrix into a set of submatrices with much smaller sizes, the rank constraint on the original matrix is equivalently transformed into a set of constraints on the decomposed submatrices. The distributed framework allows parallel computation of subproblems while requiring coordination among them to satisfy the coupled constraints. As the scale of every subproblem solved independently is significantly reduced, the decomposition scheme and the distributed framework can be applied to large-scale RCSPs. Moreover, optimality conditions of the proposed distributed optimization algorithm for RCSPs at the converged point are analyzed. Finally, the efficiency and effectiveness of the proposed method are demonstrated via simulation examples for solving the image denoising problem.**

***Index Terms*—Distributed optimization, rank-constrained optimization.**

I. Introduction

A RANK-CONSTRAINED semidefinite program (RCSP) is to minimize a convex objective function of positive semidefinite matrices subject to a set of convex constraints and rank constraints [1]. In recent years, RCSP has attracted increasing attention due to its extensive applications, such as signal processing, system identification, and image noise reduction [2], just to name a few. Moreover, many nonconvex optimization problems can be equivalently converted to RCSPs. For example, a polynomial programming problem can be formulated as a rank-one constrained semidefinite programming problem [3]. A rank minimization problem (RMP) can also be equivalently transformed into a RCSP [2]. Due to the wide range of applications of RMPs in the field of machine learning, such as low-rank kernel learning, multi-stage multi-task learning, and recommender systems, it further implies the extensive applications of RCSPs [4].

However, the non-linearity and non-convexity of the rank constraint lead to challenges in developing a scalable RCSP algorithm with robust convergence. Most of the existing approaches for RCSP focus on solving a special type of problem with certain types of constraints or objective functions. For example, alternating projection methods have been developed for RCSP problems with linear matrix inequalities (LMI) [5]. Methods based on linearization and factorization have been applied to linear regression problems and problems with bilinear matrix inequalities [6]. A good initial guess is usually required for these methods to find a convergent solution. Besides, an efficient greedy algorithm has been designed for large-scale unconstrained low-rank problems [7]. Although these algorithms have demonstrated high computing performance when solving a specific type of problem, they are not applicable to general RCSPs. Our prior work in [2] developed an iterative algorithm to solve general RCSPs. However, for large-scale RCSPs, the iterative algorithm involving semidefinite constraints at each iteration is time-consuming.

Meanwhile, approximation and relaxation techniques have been introduced to solve RCSPs or RMPs to reduce the computational complexity. For instance, the nuclear norm has been used as a surrogate for the non-convex rank function [8]. In Eckart and Young's theorem [9], an approximated low-rank matrix is obtained by using the truncated singular value decomposition (SVD), which has been applied as an efficient approach when searching for a low-rank matrix [10]. However, these approximation approaches cannot guarantee an optimal solution to the original problem. Work in [11] provides the conditions when the relaxed problem is equivalent to the original low rank optimization problem. We aim to develop a scalable RCSP algorithm that considers the exact rank constraint without approximation or relaxation.

Distributed optimization approaches have been developed to improve the scalability and efficiency in solving large-scale optimization problems [12]. In [13], an iterative algorithm based on proximal minimization has been developed and applied to solve convex optimization problems. In addition to solving convex optimization problems, distributed approaches have been applied to solve some of the nonconvex optimization

Manuscript received 21 March 2022; revised 30 May 2022; accepted 19 June 2022. Date of publication 28 June 2022; date of current version 11 July 2022. Recommended by Senior Editor F. Dabbene. *(Corresponding author: Ran Dai.)*

Chaoying Pei, Sixiong You, and Ran Dai are with the School of Aeronautic and Astronautics, Purdue University, West Lafayette, IN 47907 USA (e-mail: peic@purdue.edu; you84@purdue.edu; randai@purdue.edu).

Chuangchuang Sun is with the Aerospace Engineering Department, Mississippi State University, Starkville, MS 39762 USA (e-mail: csun@ae.msstate.edu).

Digital Object Identifier 10.1109/LCSYS.2022.3186939

problems. For example, a distributed augmented Lagrangian method has been developed, which requires the objective function to be separable and the constraints to be convex or linear [14]. However, due to the rank constraint, there are limited studies on developing a distributed algorithm for RCSPs.

To improve the scalability and efficiency for solving RCSPs, a distributed optimization framework based on matrix decomposition and proximal minimization is proposed to decompose a large-scale RCSP into a set of small-scale RCSPs. Each subproblem is then solved via an iterative rank minimization (IRM) method developed in our previous work [2]. The distributed framework allows parallel computation of the decomposed RCSPs while applying a coordination scheme among them to satisfy the coupled constraints. Furthermore, the optimality conditions of the proposed distributed optimization framework at the converged point are analyzed. In the end, the image noise reduction problem is used as a simulation example with comparative results obtained from centralized methods [2], [15], [16] to demonstrate the efficiency and effectiveness of the distributed optimization algorithm.

In summary, the contribution of this letter includes two manifolds. The first one is the matrix decomposition scheme that decomposes a matrix rank constraint into a set of constraints on its submatrices in smaller sizes. The second is a distributed framework that solves the decomposed subproblems in parallel to cooperatively satisfy the rank constraint, semidefinite constraint, and affine functions of the original problem. The overall goal is to improve the scalability and computational efficiency when solving large-scale RCSPs. Thus, compared with existing RCSP algorithms, this letter represents the first attempt to decompose the rank constraint. Moreover, the exact rank constraint is satisfied without relaxation or approximation.

The remainder of this letter is organized as follows. Section II introduces the problem formulation of RCSP and its equivalent conversion. The distributed optimization framework and the proof of convergence are described in Section III. The simulation example is presented in Section IV. We conclude this letter in Section V.

II. Problem Formulation

A RCSP problem is formulated as

$$\begin{aligned} \min_{\mathbf{X}} \ & J = f(\mathbf{X}) \\ \text{s.t.} \ & g_k(\mathbf{X}) \le 0,\ k = 1, \ldots, c, \\ & \operatorname{rank}(\mathbf{X}) \le r,\ \ \mathbf{X} \in \mathbb{S}_+^n, \end{aligned} \tag{1}$$

where $X \in \mathbb{S}_+^n$ is a positive semidefinite matrix to be determined, f is a convex objective function, $g_k(\mathbf{X}) \le 0,\ k = 1, \ldots, c$, is a convex set of constraints, and r is the upper bound of the rank of $\mathbf{X}$.

By ignoring the rank constraint, problem (1) can be relaxed as a semidefinite programming (SDP) problem. Although solving the relaxed problem yields a lower bound of the objective value, the solution in general cannot guarantee to satisfy the rank constraint in the original problem (1). Thus, an equivalent conversion is introduced to replace both the rank constraint and the semidefinite constraint.

Lemma 1 [17, Lemma 2.1]: For a rectangular $\mathbf{Z} \in \mathbb{R}^{m\times n}$, the rank constraint on $\mathbf{Z}$, $\operatorname{rank}(\mathbf{Z}) \le r$, can be equivalently transformed into $\mathbf{Z} = \mathbf{N_1}\mathbf{N_2}$, where $\mathbf{N_1} \in \mathbb{R}^{m\times r}$ and $\mathbf{N_2} \in \mathbb{R}^{r\times n}$. In addition, for a semidefinite matrix $\mathbf{X} \in \mathbb{S}_+^n$, the rank constraint $\operatorname{rank}(\mathbf{X}) \le r$, together with the semidefinite constraint $\mathbf{X} \succeq 0$, are equivalent to that there exists a matrix $\mathbf{N} \in \mathbb{R}^{n\times r}$ such that $\mathbf{X} = \mathbf{N}\mathbf{N}^T$.

Given Lemma 1, problem (1) can be reformulated as

$$\begin{aligned} \min_{\mathbf{X},\mathbf{N}} \ & J = f(\mathbf{X}) \\ \text{s.t.} \ & g_k(\mathbf{X}) \le 0,\ k = 1, \ldots, c, \\ & \mathbf{X} = \mathbf{N}\mathbf{N}^T, \end{aligned} \tag{2}$$

where $\mathbf{N} \in \mathbb{R}^{n\times r}$. By introducing matrix $\mathbf{N}$, the rank constraint and the semidefinite constraint on the original matrix can be decomposed into a set of constraints on its submatrices, which will be explained in the next section.

III. Distributed Optimization Framework

A. Problem Decomposition

To solve RCSP in (1) in a distributed manner, a decomposition method is introduced below to decompose the original rank constraint into a set of small-scale rank constraints. When the unknown matrix $\mathbf{X} \in \mathbb{S}_+^n$ in (2) is written as $\mathbf{X} = \mathbf{N}\mathbf{N}^T$, the rank constraint $\operatorname{rank}(\mathbf{X}) \le r$ is satisfied when there exists $\mathbf{N} \in \mathbb{R}^{n\times r}$. Matrix $\mathbf{N}$ can be written as the composition of row elements in the form of $\mathbf{N} = [\mathbf{N_1}, \ldots, \mathbf{N_k}]^T$, where $\mathbf{N_i} \in \mathbb{R}^{p_i\times r}$, $\sum_{i=1}^k p_i = n$. Thus, based on $\mathbf{X} = \mathbf{N}\mathbf{N}^T$, $\mathbf{X}$ can be expressed as

$$\mathbf{X} = \begin{bmatrix} \mathbf{N_1}\mathbf{N_1}^T & \mathbf{N_1}\mathbf{N_2}^T & \cdots & \mathbf{N_1}\mathbf{N_k}^T \\ \mathbf{N_2}\mathbf{N_1}^T & \mathbf{N_2}\mathbf{N_2}^T & \cdots & \mathbf{N_2}\mathbf{N_k}^T \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{N_k}\mathbf{N_1}^T & \mathbf{N_k}\mathbf{N_2}^T & \cdots & \mathbf{N_k}\mathbf{N_k}^T \end{bmatrix}. \tag{3}$$

Here we denote the submatrices of $\mathbf{X}$ as $\mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T$.

In general, there are multiple ways to partition a matrix. Therefore, the decomposition scheme in (3) is not unique. After decomposition, the non-zero entries are required to be included in at least one of the decomposed submatrices. When special structures of the unknown matrix are considered, the decomposition scheme can be adapted to facilitate the distributed computation.

For the RCSP in (2), submatrices $\mathbf{N_1}, \mathbf{N_2}, \ldots, \mathbf{N}_k$ are introduced after the matrix decomposition. We denote the pair set $(i, j) \in \mathcal{D}$ for all decomposed submatrices $\mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T$, where $\mathcal{D}$ denotes the set of the row and column indices pair for all decomposed submatrices. Then constraint $\mathbf{X} = \mathbf{N}\mathbf{N}^T$ can be equivalently decomposed into a series of small-scale rank constraints $\mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T$, $(i, j) \in \mathcal{D}$. For every decomposed submatrix $\mathbf{X}_{ij}$, $(i, j) \in \mathcal{D}$, by handling the other submatrices as given constants, the objective and constraints can be expressed as $J = f_{ij}(\mathbf{X}_{ij})$ and $g_{k,ij}(\mathbf{X}_{ij}) \le 0,\ j = 1, \ldots, c_{ij}$, respectively, where $\mathbf{X}_{ij} \in \mathbb{R}^{p_i\times p_j}$, f_{ij} and $g_{k,ij}$ are the objective function and constraints involving $\mathbf{X}_{ij}$ in the subproblem, and c_{ij} denotes the number of constraints involving $\mathbf{X}_{ij}$. Then for each submatrix

$\mathbf{X}_{ij}$, $(i,j) \in \mathcal{D}$, a subproblem is formulated as

$$\begin{aligned} \min_{\mathbf{X}_{ij},\mathbf{N}_i,\mathbf{N}_j} \quad & J = f_{ij}(\mathbf{X}_{ij}) \\ \text{s.t.} \quad & g_{k,ij}(\mathbf{X}_{ij}) \le 0,\ k = 1,\ldots,c_{ij}, \\ & \mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T. \end{aligned} \tag{4}$$

Through the decomposition, the original RCSP problem in (1) is decomposed into a set of small-scale subproblems when each subproblem involves blocks of $\mathbf{X}$. To solve each subproblem of (4) using the IRM algorithm developed in [2], the following lemmas are introduced to reformulate the rank constraint in each subproblem.

Lemma 2 [2, Proposition 2]: $\mathbf{X} = \mathbf{N}\mathbf{N}^T$ can be equivalently transformed into $\text{rank}(\begin{bmatrix} \mathbf{I}_r & \mathbf{N}^T \\ \mathbf{N} & \mathbf{X} \end{bmatrix}) \le r$ and $\begin{bmatrix} \mathbf{I}_r & \mathbf{N}^T \\ \mathbf{N} & \mathbf{X} \end{bmatrix} \succeq 0$, where $\mathbf{X} \in \mathbb{S}^n$ and $\mathbf{N} \in \mathbb{R}^{n\times r}$.

However, lemma 2 is not applicable to the non-square matrices. As a result, another lemma for non-square matrices is introduced, where $\mathbf{X}_{ij} \in \mathbb{R}^{p_i\times p_j}$ with $p_i \ne p_j$.

Lemma 3 [2, Lemma 1]: For a given matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\text{rank}(\mathbf{X}) \le r$ if and only if there exist symmetric matrices $\mathbf{Y} \in \mathbb{R}^{m\times m}$ and $\mathbf{Z} \in \mathbb{R}^{n\times n}$ such that

$$\text{rank}(\mathbf{Y}) + \text{rank}(\mathbf{Z}) \le 2r, \begin{bmatrix} \mathbf{Y} & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Z} \end{bmatrix} \succeq 0. \tag{5}$$

According to Lemmas 2 and 3, two conversions for $\mathbf{X}_{ii} = \mathbf{N}_i\mathbf{N}_i^T$ and $\mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T$, $i \ne j$, are listed below, respectively.

(1) When $i = j$, we have $\mathbf{X}_{ii} = \mathbf{N}_i\mathbf{N}_i^T$, where $\mathbf{X}_{ii} \in \mathbb{R}^{p_i\times p_i}$, $\mathbf{N}_i \in \mathbb{R}^{p_i\times r}$. By introducing an extended matrix $\mathbf{X}_{ii}^{ex} = \begin{bmatrix} \mathbf{I}_r & \mathbf{N}_i^T \\ \mathbf{N}_i & \mathbf{X}_{ii} \end{bmatrix} \in \mathbb{R}^{(p_i+r)\times(p_i+r)}$, the constraint $\mathbf{X}_{ii} = \mathbf{N}_i\mathbf{N}_i^T$ can be reformulated as $\text{rank}(\mathbf{X}_{ii}^{ex}) \le r$ and $\mathbf{X}_{ii}^{ex} \succeq 0$ according to Lemma 2

(2) When $i \ne j$, we have $\mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T$, where $\mathbf{X}_{ij} \in \mathbb{R}^{p_i\times p_j}$, $\mathbf{N}_i \in \mathbb{R}^{p_i\times r}$, $\mathbf{N}_j \in \mathbb{R}^{p_j\times r}$. By introducing an extended matrix $\mathbf{X}_{ij}^{ex} = \begin{bmatrix} \mathbf{X}_{ij} & \mathbf{N}_i \\ \mathbf{N}_j^T & \mathbf{I}_r \end{bmatrix} \in \mathbb{R}^{(p_i+r)\times(p_j+r)}$, and extra two matrices $\mathbf{Y} \in \mathbb{R}^{(p_i+r)\times(p_i+r)}$ and $\mathbf{Z} \in \mathbb{R}^{(p_j+r)\times(p_j+r)}$, the constraint $\mathbf{X}_{ij} = \mathbf{N}_i\mathbf{N}_j^T$ can be reformulated as $\begin{bmatrix} \mathbf{Y} & \mathbf{X}_{ij}^{ex} \\ (\mathbf{X}_{ij}^{ex})^T & \mathbf{Z} \end{bmatrix} \succeq 0$ and $\text{rank}(\mathbf{Y}) + \text{rank}(\mathbf{Z}) \le 2r$, according to Lemma 3. In addition, the rank constraint on $\mathbf{Y}$ and $\mathbf{Z}$ can be reformulated as $\text{rank}\left(\begin{bmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z} \end{bmatrix}\right) \le 2r$.

Consequently, based on the equivalent conversions introduced above, when $i = j$, problem (4) can be written as

$$\begin{aligned} \min_{\mathbf{X}_{ii},\mathbf{N}_i} \quad & J = f_{ii}(\mathbf{X}_{ii}) && \text{(6a)} \\ \text{s.t.} \quad & g_{k,ii}(\mathbf{X}_{ii}) \le 0,\ k = 1,\ldots,c_{ii}, && \text{(6b)} \\ & \mathbf{X}_{ii}^{ex} = \begin{bmatrix} \mathbf{I}_r & \mathbf{N}_i^T \\ \mathbf{N}_i & \mathbf{X}_{ii} \end{bmatrix}, && \text{(6c)} \\ & \text{rank}(\mathbf{X}_{ii}^{ex}) \le r,\ \mathbf{X}_{ii}^{ex} \succeq 0, && \text{(6d)} \end{aligned}$$

where $\mathbf{X}_{ii} \in \mathbb{R}^{p_i\times p_i}$. When $i \ne j$, problem (4) is written as

$$\begin{aligned} \min_{\mathbf{X}_{ij},\mathbf{N}_i,\mathbf{N}_j,\mathbf{Y},\mathbf{Z}} \quad & J = f_{ij}(\mathbf{X}_{ij}) && \text{(7a)} \\ \text{s.t.} \quad & g_{k,ij}(\mathbf{X}_{ij}) \le 0,\ k = 1,\ldots,c_{ij}, && \text{(7b)} \\ & \mathbf{X}_{ij}^{ex} = \begin{bmatrix} \mathbf{X}_{ij} & \mathbf{N}_i \\ \mathbf{N}_j^T & \mathbf{I}_r \end{bmatrix}, && \text{(7c)} \\ & \text{rank}(\begin{bmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z} \end{bmatrix}) \le 2r, && \text{(7d)} \\ & \begin{bmatrix} \mathbf{Y} & \mathbf{X}_{ij}^{ex} \\ (\mathbf{X}_{ij}^{ex})^T & \mathbf{Z} \end{bmatrix} \succeq 0, && \text{(7e)} \end{aligned}$$

where $\mathbf{X}_{ij} \in \mathbb{R}^{p_i\times p_j}$.

Therefore, through decomposition and reformulations expressed in (6) and (7), the dimensions of the rank constraint and positive semidefinite constraint can be reduced from $n\times n$ to $(r+p_i)\times(r+p_i)$ and $(2r+p_i+p_j)\times(2r+p_i+p_j)$, respectively, in each subproblem. For large-scale RCSPs with $n \gg 2r$, the scale of the subproblem can be significantly reduced via the decomposition scheme.

B. A Local Solver via Iterative Rank Minimization Algorithm

The next step applies a local solver, named IRM, developed in [2] to solve all subproblems. The IRM algorithm can solve general RCSPs with a guaranteed local convergence rate and is thus applied here to solve all decomposed subproblems in the distributed framework. Based on the fact that for $\mathbf{X} \in \mathbb{S}^n$, if its rank is smaller than r, then it has at least $n - r$ zero eigenvalues.

Therefore, by sorting the eigenvalues of $\mathbf{X}$, it holds that $\mathbf{V}^T\mathbf{X}\mathbf{V} = 0$, where $\mathbf{V} \in \mathbb{R}^{n\times(n-r)}$ represents the corresponding eigenvectors of the $n - r$ smallest eigenvalues of $\mathbf{X}$. IRM is to gradually reduce the $n - r$ smallest eigenvalues to zero through iterations to satisfy the rank constraint while minimizing the cost function. More details of the IRM algorithm can be referred to [2]. When applying the IRM method to the RCSP in (1), each iteration is formulated as

$$\begin{aligned} \min_{\mathbf{X}^l} \quad & J = f(\mathbf{X}^l) + \omega^l e^l && \text{(8a)} \\ \text{s.t.} \quad & g_k(\mathbf{X}^l) \le 0,\ k = 1,\ldots,c, && \text{(8b)} \\ & e^l\mathbf{I}_{n-r} - (\mathbf{V}^{l-1})^T\mathbf{X}^l\mathbf{V}^{l-1} \succeq 0, && \text{(8c)} \\ & e^l \le e^{l-1},\ \mathbf{X}^l \in \mathbb{S}_+^n, && \text{(8d)} \end{aligned}$$

where $\mathbf{X}^l$ is the matrix to be optimized at the lth iteration of the IRM method. Constraint (8c) provides an upper bound for $\mathbf{V}^T\mathbf{X}\mathbf{V}$. ω^l is a weighting factor for e^l, which is increasing along with the iterations. By iteratively solving the convex optimization problem (8) and minimizing e^l as a penalty term of the objective function, the rank constraint can be gradually approached. Through the IRM approach, each subproblem of RCSP in the distributed framework can be solved independently with guaranteed local convergence.

C. Coordination Scheme in the Distributed Framework

The decomposition scheme may lead to coupled submatrices in the decomposed subproblems. As each subproblem is solved independently, we cannot reach a consensus on the coupled submatrices for the local solutions from different subproblems. Therefore, a coordination scheme based on proximal minimization is designed to reach a consensus on the coupled elements. At each step h, subproblems of RCSPs are solved

independently (two categories need to be considered for $i = j$ and $i \neq j$) with penalty terms to drive the coupled submatrices to reach the consensus.

When $i = j$, the RCSP subproblem at the hth step is formulated as

$$\min_{\mathbf{X}_{ii}^h, \mathbf{N}_i^h} \quad J = f_{ii}(\mathbf{X}_{ii}^h) + \eta_1^h \|\mathbf{N}_i^h - \bar{\mathbf{N}}_i^{h-1}\| \tag{9a}$$

$$\text{s.t.} \quad \text{(6b), (6c), (6d)} \tag{9b}$$

where $\bar{\mathbf{N}}_i^{h-1}$ is the average solution of $\mathbf{N}_i$ obtained from all coupled subproblems at the last step $h-1$, η_1^h is a weighting factor at the hth step, and $\{\eta_1^h\}$ is an increasing sequence. For the remaining submatrices involved in functions f_{ii} and $g_{k,ii}$, $k = 1, \ldots, c$, they are set as given constants with values obtained from the last step $h-1$.

Similarly, when $i \neq j$, the RCSP subproblem at the hth step is formulated as

$$\min_{\mathbf{X}_{ij}^h, \mathbf{N}_i^h, \mathbf{N}_j^h, \mathbf{Y}, \mathbf{Z}} \quad J = f_{ij}(\mathbf{X}_{ij}^h) + \eta_1^h \|\mathbf{N}_i^h - \bar{\mathbf{N}}_i^{h-1}\| + \eta_2^h \|\mathbf{N}_j^h - \bar{\mathbf{N}}_j^{h-1}\| \tag{10a}$$

$$\text{s.t.} \quad \text{(7b), (7c), (7d), (7e)} \tag{10b}$$

where η_1^h and η_2^h are weighting factors at the hth step and they are increasing over iterations. Specifically, they are set as exponentially increasing functions in this letter. By considering the norm of $\mathbf{N}_i^h - \bar{\mathbf{N}}_i^{h-1}$ and $\mathbf{N}_j^h - \bar{\mathbf{N}}_j^{h-1}$ as penalty terms in the objective function, the goal is to gradually reduce the gap between $\mathbf{N}^h$ and $\bar{\mathbf{N}}^{h-1}$ iteratively.

At each step h, after solving all subproblems (9) and (10) in parallel via the local IRM solver, the submatrices of $\mathbf{X}_{ij}^h$ and $\mathbf{N}_i^h$ can be obtained independently. For every submatrix that is involved in multiple subproblems, the average of each coupled submatrix will be computed and used in the penalty term in the next step. The penalty terms in (10) only require communication with its neighboring subproblems involving X_{ij} to obtain $\bar{\mathbf{N}}_i^{h-1}$ and $\bar{\mathbf{N}}_j^{h-1}$. In other words, the rank constraint can be satisfied in a distributed manner without global communication. However, considering general RCSPs with fully coupled convex constraints, i.e., $g_k(\mathbf{X}) \leq 0$, $k = 1, \ldots, c$, that are not separable, aggregation among all subproblems is required to obtain $\bar{\mathbf{N}}^h$ for coordinating the coupled convex constraints in the next step $h+1$, After obtaining $\bar{\mathbf{N}}^h$, $\bar{\mathbf{X}}^h$ is updated as $\bar{\mathbf{X}}^h = \bar{\mathbf{N}}^h(\bar{\mathbf{N}}^h)^T$. Assume that the algorithm converges at hth iteration, then the optimized $\bar{\mathbf{N}}^*$ and $\bar{\mathbf{X}}^*$ are determined by $\bar{\mathbf{N}}^h$ and $\bar{\mathbf{X}}^h$.

The distributed optimization algorithm for RCSPs is summarized as below in Algorithm 1.

Algorithm 1 Distributed Optimization Algorithm for RCSPs

Require: Problem parameters g_k, r, f, initial guess $\mathbf{N}^0$, and algorithm parameter ϵ and h_{max}

Ensure: Matrix $\mathbf{X}^*$

1: Decompose $\mathbf{X}$ to formulate subproblems (6) and (7);
2: While $\|\bar{\mathbf{X}}^h - \bar{\mathbf{X}}^{h-1}\| \geq \epsilon$ && $h \leq h_{max}$;
3: Solve subproblems (9) and (10) via IRM to obtain $\{\mathbf{X}_{ii}^h, \mathbf{N}_i^h\}$ and $\{\mathbf{X}_{ij}^h, \mathbf{N}_i^h, \mathbf{N}_j^h\}$, respectively;
4: Aggregating all the obtained $\mathbf{N}_i^h$ by averaging the coupled elements and update $\bar{\mathbf{N}}^h$, $\bar{\mathbf{X}}^h$;
5: $h = h + 1$, update $\eta_1^{h+1} > \eta_1^h$ and $\eta_2^{h+1} > \eta_2^h$
6: end while
7: **return** $\bar{\mathbf{N}}^* = \bar{\mathbf{N}}^h$, $\bar{\mathbf{X}}^* = \bar{\mathbf{X}}^h$

D. Optimality Analysis of Distributed Solution

The proximal minimization algorithm has been developed to solve centralized/distributed convex optimization problems [12], [13]. With an appropriate selection of the weighting factors, e.g., $\{\eta^h\}$, of the penalty terms, the proof of convergence for the proximal minimization algorithm has been provided under some assumptions, e.g., the objective function is Lipschitz continuous on the constrained set [12], [13]. Therefore, within a local compact neighborhood of RCSP's stationary point, the local convergence can also be obtained for Algorithm 1 based on the proximal minimization. However, considering the rank constraint and matrix decomposition, the optimality conditions for subproblems at the converging point are not directly applicable to the original RCSP. Therefore, in the following we provide an analysis of optimality conditions at the converging point.

Proposition 1: Assume the distributed algorithm (Algorithm 1) converges to $\{\bar{\mathbf{N}}^*, \bar{\mathbf{X}}^*\}$ at the hth iteration, which satisfies $\bar{\mathbf{N}}^h = \bar{\mathbf{N}}^*$ and $\bar{\mathbf{X}}^h = \bar{\mathbf{X}}^*$, then the converged solution $\{\bar{\mathbf{N}}^*, \bar{\mathbf{X}}^*\}$ is a stationary point of problem (2).

Proof: To complete this proof, we will first analyze the properties of solutions for the distributed subproblems (9) and (10). Note that in subproblem (9), the constraints $\text{rank}((\mathbf{X}_{ij}^{ex})^h) \leq r$ and $(\mathbf{X}_{ij}^{ex})^h \succeq 0$ is equivalent to $\mathbf{X}_{ii}^h = \mathbf{N}_i^h(\mathbf{N}_i^h)^T$. Additionally, as $\mathbf{N}_i^h = \bar{\mathbf{N}}_i^*$, problem (9) can be equivalently rewritten as

$$\min_{\mathbf{X}_{ii}^h \in \mathbb{R}^{p_i \times p_i}} \quad J = f_{ii}(\mathbf{X}_{ii}^h)$$

$$\text{s.t.} \quad \mathbf{g}_{k,ii}(\mathbf{X}_{ii}^h) \leq 0, \; k = 1, 2, \ldots, c_{ii}, \quad \mathbf{X}_{ii}^h = \bar{\mathbf{N}}_i^*(\bar{\mathbf{N}}_i^*)^T. \tag{11}$$

Then, the Lagrangian of problem (11) can be written as

$$\mathcal{L} = f(\mathbf{X}_{ii}^h) + \sum_{k=1}^{c_{ii}} \mu_k(\mathbf{g}_{k,ii}(\mathbf{X}_{ii}^h)) - \mathbf{Tr}(\Phi_{ii}, \mathbf{X}_{ii}^h - \bar{\mathbf{N}}_i^*(\bar{\mathbf{N}}_i^*)^T)$$

where $\mu_k \geq 0$, $k = 1, 2, \ldots, c_{ii}$, and $\Phi_{ii} \in \mathbb{S}^{p_i}$ are the Lagrange multipliers for constraints $\mathbf{g}_{k,ii}(\mathbf{X}_{ii}^h) \leq 0$ and $\mathbf{X}_{ii}^h = \bar{\mathbf{N}}_i^*(\bar{\mathbf{N}}_i^*)^T$, respectively. Then, the Karush-Kuhn-Tucker (KKT) conditions of (11) can be expressed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}_{ii}^h} = f'(\mathbf{X}_{ii}^h) + \sum_{k=1}^{c_{ii}} \mu_k(\mathbf{g}'_{k,ii}(\mathbf{X}_{ii}^h)) - \Phi_{ii} = \mathbf{0}, \tag{12a}$$

$$\mu_k(\mathbf{g}_{k,ii}(\mathbf{X}_{ii}^h)) = 0, \; k = 1, \ldots, c_{ij}, \tag{12b}$$

$$\mathbf{Tr}(\Phi_{ii}, \mathbf{X}_{ii}^h - \bar{\mathbf{N}}_i^*(\bar{\mathbf{N}}_i^*)^T) = 0. \tag{12c}$$

As for problem (6), the Lagrangian can be expressed as

$$\mathcal{L} = f_{ii}(\mathbf{X}_{ii}^h) + \sum_{k=1}^{c_{ii}} \lambda_k(\mathbf{g}_{k,ii}(\mathbf{X}_{ii}^h)) - \mathbf{Tr}(\Psi_{ii}, \mathbf{X}_{ii}^h - \mathbf{N}_i^h(\mathbf{N}_i^h)^T)$$

where $\lambda_k \geq 0 \in \mathbb{R}, k = 1, 2, \ldots, c$, and $\Psi_{ii} \in \mathbb{S}^{p_i}$ are the Lagrange multipliers. Then its KKT conditions can be

expressed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}_{ii}^h} = f'(\mathbf{X}_{ii}^h) + \sum_{k=1}^{c_{ii}} \lambda_k (\mathbf{g}'_{k,ii}(\mathbf{X}_{ii}^h)) - \Psi_{ii} = \mathbf{0}, \quad (13a)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{N}_i^h} = 2\Psi_{ii}\mathbf{N}_i^h = \mathbf{0}, \quad (13b)$$

$$\lambda_k(\mathbf{g}_{k,ii}(\mathbf{X}_{ii}^h)) = 0,\ k = 1, \ldots, c, \quad (13c)$$

$$\mathbf{Tr}(\Psi_{ii}, \mathbf{X}_{ii}^h - \mathbf{N}_i^h(\mathbf{N}_i^h)^T) = 0. \quad (13d)$$

Let $\Psi_{ii} = \Phi_{ii}$, $\lambda_k = \mu_k$, $\mathbf{N}_i^h = \bar{\mathbf{N}}_i^*$, then the KKT condition of (6) and (11) are exactly the same. Therefore, the solution $\{\bar{\mathbf{N}}_i^h, \bar{\mathbf{X}}_{ii}^h\}$ is a stationary point of problem (6).

Similarly, problem (10) can be equivalently rewritten as

$$\min_{\mathbf{X}_{ij}^h \in \mathbb{R}^{p_i \times p_j}} \quad J = f_{ij}(\mathbf{X}_{ij}^h) \quad (14a)$$

$$\text{s.t.} \quad \mathbf{g}_{k,ij}(\mathbf{X}_{ij}^h) \le 0,\ k = 1, 2, \ldots, c_{ij}, \quad (14b)$$

$$\mathbf{X}_{ij}^h = \bar{\mathbf{N}}_i^* (\bar{\mathbf{N}}_j^*)^T, \quad (14c)$$

Then, the Lagrangian of problem (14) can be written as

$$\mathcal{L} = f(\mathbf{X}_{ij}^h) + \sum_{k=1}^{c_{ij}} \mu_k(\mathbf{g}_{k,ij}(\mathbf{X}_{ij}^h)) - \mathbf{Tr}(\Phi_{ij}, \mathbf{X}_{ij}^h - \bar{\mathbf{N}}_i^*(\bar{\mathbf{N}}_j^*)^T)$$

where $\mu_k \ge 0$, $k = 1, 2, \ldots, c$, and $\Phi_{ij} \in \mathbb{R}^{p_i \times p_j}$ are the Lagrange multipliers. Then, the KKT conditions of (14) can be expressed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}_{ij}^h} = f'(\mathbf{X}_{ij}^h) + \sum_{k=1}^{c_{ij}} \mu_k (\mathbf{g}'_{k,ii}(\mathbf{X}_{ij}^h)) - \Phi_{ij} = \mathbf{0}, \quad (15a)$$

$$\mu_k(\mathbf{g}_{k,ij}(\mathbf{X}_{ij}^h)) = 0,\ k = 1, \ldots, c_{ij}, \quad (15b)$$

$$\mathbf{Tr}(\Phi_{ij}, \mathbf{X}_{ij}^h - \bar{\mathbf{N}}_i^*(\bar{\mathbf{N}}_j^*)^T) = 0. \quad (15c)$$

As for problem (7), the Lagrangian can be expressed

$$\mathcal{L} = f(\mathbf{X}_{ij}^h) + \sum_{k=1}^{c_{ij}} \lambda_k(\mathbf{g}_{k,ij}(\mathbf{X}_{ij}^h)) - \mathbf{Tr}(\Psi_{ij}, \mathbf{X}_{ij}^h - \mathbf{N}_i^h(\mathbf{N}_j^h)^T)$$

where $\lambda_k \ge 0$, $k = 1, 2, \ldots, c$, and $\Psi_{ij} \in \mathbb{R}^{p_i \times p_j}$ are the Lagrange multipliers. Let $\Psi_{ij} = \Phi_{ij}$, $\lambda_k = \mu_k$, $\mathbf{N}_i^h = \bar{\mathbf{N}}_i^*$, $\mathbf{N}_j^h = \bar{\mathbf{N}}_j^*$, then the KKT condition of (7) and (14) are exactly the same. Therefore, we can conclude that $\{\bar{\mathbf{N}}_i^*, \bar{\mathbf{N}}_j^*, \bar{\mathbf{X}}_{ij}^*\}$ is also a stationary point of problem (7).

The Lagrangian of problem (2) is written as

$$\mathcal{L} = f(\mathbf{X}) + \sum_{k=1}^{c} \delta_k(\mathbf{g}_k(\mathbf{X})) - \mathbf{Tr}(\Phi, \mathbf{X} - \mathbf{N}\mathbf{N}^T), \quad (16)$$

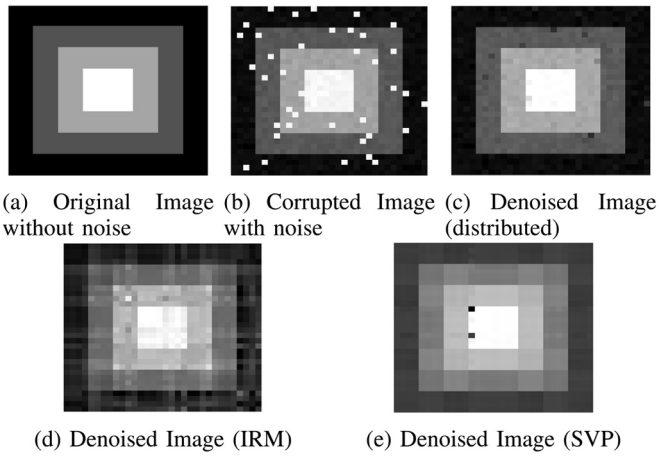
where $\delta_k \ge 0$, and $\Phi \in \mathbb{S}^n$. Then the KKT conditions of (2) can be expressed as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = f'(\mathbf{X}) + \sum_{k=1}^{c} \delta_k(\mathbf{g}'_k(\mathbf{X})) - \Phi = \mathbf{0}, \quad (17a)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{N}} = 2\Phi\mathbf{N} = \mathbf{0}, \quad (17b)$$

$$\mu_k(\mathbf{g}_k(\mathbf{X})) = 0,\ k = 1, \ldots, c, \quad (17c)$$

$$\mathbf{Tr}(\Phi, \mathbf{X} - \mathbf{N}\mathbf{N}^T) = 0. \quad (17d)$$



(a) Original Image without noise (b) Corrupted Image with noise (c) Denoised Image (distributed)

(d) Denoised Image (IRM) (e) Denoised Image (SVP)

Fig. 1. Image noise reduction results.

Let $\delta_k = \mu_k$ and $\Phi = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1k} \\ \vdots & \vdots & \vdots \\ \Phi_{k1} & \cdots & \Phi_{kk} \end{bmatrix}$, we have that the KKT conditions in (12) and (15) are the exact KKT conditions of problem (2). Therefore, we can conclude that the corresponding solution $(\bar{\mathbf{N}}^*, \bar{\mathbf{X}}^*)$ is a stationary point of problem (2). ■

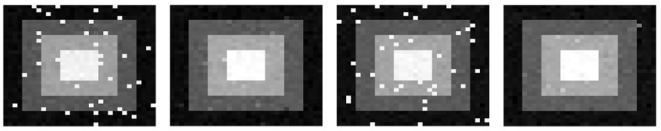
IV. Simulation Results

This section presents the simulation results for the image noise reduction problem to validate the effectiveness and efficiency of the distributed optimization framework for RCSPs. The simulation is run on a laptop with a 2.7 GHz processor and 16 GB memory. The image noise reduction problem is to restore corrupted images with noise and/or partial pixels missing. We consider an 8-byte gray-scale picture, where the color in each pixel can be represented by an integer in the range $[0, 255]$. Thus, the picture can be represented as a low-rank matrix. Then, recovering a corrupted image is classified as a low-rank matrix completion problem, which can be formulated as a RCSP:

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times n}} \quad J = \frac{1}{2}\|\mathbf{M}_\Omega(\mathbf{X} - \mathbf{X}_0)\|$$
$$\text{s.t.} \quad \text{rank}(\mathbf{X}) \le r, \quad (18)$$

where $\mathbf{X}_0 \in \mathbb{R}^{n \times n}$ is the corrupted picture with noised and missing pixels, $\Omega \in \mathbb{R}^{n \times n}$ is a matrix with the available indices of the noised or missing pixels. Specifically, the function $\mathbf{M}_\Omega(\mathbf{Z})$ is defined as $\mathbf{M}_\Omega(\mathbf{Z}) = 0$ for $(i, j) \in \Omega$, otherwise $\mathbf{M}_\Omega(\mathbf{Z}) = \mathbf{Z}(i, j)$. The upper bound of rank on the unknown matrix $\mathbf{X}$ is denoted by r. The value of r is estimated by calculating the number of the eigenvalues of $\mathbf{X}_0$ greater than a threshold $\alpha\lambda_{max}$, where λ_{max} is the largest eigenvalue (absolute value) of matrix $\mathbf{X}_0$, and α is a parameter to control the effect of noise reduction, which is set as 0.2 in this case.

The distributed optimization algorithm is applied to the noise reduction problem, where Fig. 1a presents the original image without any noise, and Fig. 1b is the corrupted image with both random noise and fixed pattern noise (FPN). We set the Gaussian distribution of random noise as $N \sim \mathcal{N}(0, 9)$, and the FPN ratio $\beta = 4\%$. In the presented case, the matrix



(a) Corrupted (b) Denoised (c) Corrupted (d) Denoised

Fig. 2. Image noise reduction of two cases.

to be optimized is $\mathbf{X} \in \mathbb{R}^{32\times 32}$, which is decomposed into 16×16 independent subproblems that can be calculated in parallel. The noise reduction result using the distributed algorithm is shown in Fig. 1c, which demonstrates the effect of the noise reduction.

In this case, starting with the initial guess of $\mathbf{X}$ as an all-ones matrix, it only requires $h = 3$ steps for the distributed algorithm to converge. When using a single computing unit to carry all subproblem calculations in sequence, it requires 122 mins to find a converged solution. It indicates that when implementing the distributed framework with parallel calculations for all 196 subproblems, the computation time is expected to be 196 times faster compared to the sequential computation of all subproblems using a single computation unit, which leads to an estimated computation time of around 37 seconds for the distributed algorithm.

For comparison, the same noise reduction problem is solved by IRM without decomposition in a centralized manner, which takes 290 seconds to find a converged solution. The denoised result is provided in Fig. 1d. Specifically, the objective value of the centralized and distributed algorithms are 123.25 and 12.28, respectively. Compared with the centralized algorithm, the distributed optimization algorithm demonstrates much improved denoising effects. Moreover, by implementing the distributed framework with parallel computation of all decomposed subproblems, the computation time has an order of magnitude reduction.

Furthermore, to benchmark the performance of the proposed distributed algorithm, we also implemented the singular value projection (SVP) algorithm [15], [16] to solve the same noise reduction problem, which yields an objective value of 110.74, as shown in Fig. 1e. It takes 0.25 seconds for the SVP to converge, which is faster than the proposed algorithm. However, when solving each subproblem in (9) or (10) via the interior point method, the computational time complexity is $O(n_d^6)$ [2], where n_d denotes the dimension of submatrix $\mathbf{X}_{ii}^{ex}$ or $\mathbf{X}_{ij}^{ex}$. On the other hand, the computational time complexity of SVP is $O(n^3)$ due to the SVD operation in each iteration, where n is the dimension of the original unknown matrix. Therefore, when $n >> n_d$, the distributed framework will demonstrate the computational advantage in terms of scalability.

In addition, to verify the robustness of the proposed approach, extensive simulation cases are generated with random noise distributions. Due to the space limitation, two randomly selected cases are shown in Fig. 2. In the conducted 50 cases, all of them yield a converged solution, which indicates the robustness of the proposed distributed algorithm. The mean objective values of the centralized and distributed method are 119.12 and 13.07, respectively. Besides, the average computation time per computing unit with overall steps of the centralized and distributed method are 249.17 seconds with 5.2 steps and 39 seconds with 2.3 steps, respectively. In summary, the simulation examples verify the fast convergence, scalability, and robustness of the proposed distributed optimization algorithm.

V. Conclusion

This letter develops a distributed optimization algorithm based on matrix decomposition and proximal minimization for solving rank-constrained semidefinite programs (RCSPs). By decomposing the RCSP into a set of subproblems, the dimension of each subproblem is greatly reduced and the subproblems can be solved in parallel via a local solver. The computational efficiency, robustness, and scalability are verified by applying the distributed algorithm to the image noise reduction problem. Thus we argue that the proposed distributed algorithm can serve as a promising solver for large-scale optimization problems with rank constraints.

References

[1] V. S. Mai, D. Maity, B. Ramasubramanian, and M. C. Rotkowitz, "Convex methods for rank-constrained optimization problems," in *Proc. Conf. Control Appl.*, 2015, pp. 123–130.

[2] C. Sun and R. Dai, "Rank-constrained optimization and its applications," *Automatica*, vol. 82, pp. 128–136, Aug. 2017.

[3] S. Burer and H. Dong, "Representing quadratically constrained quadratic programs as generalized copositive programs," *Oper. Res. Lett.*, vol. 40, no. 3, pp. 203–206, 2012.

[4] R. Meka, P. Jain, C. Caramanis, and I. S. Dhillon, "Rank minimization via online learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 656–663.

[5] Y. Ebihara, K. Tokuyama, and T. Hagiwara, "Structured controller synthesis using LMI and alternating projection method," *Int. J. Control*, vol. 77, no. 12, pp. 1137–1147, 2004.

[6] G. Meyer, "Geometric optimization algorithms for linear regression on fixed-rank matrices," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Liège, Liège, Belgium, 2011.

[7] S. Shalev-Shwartz, A. Gonen, and O. Shamir, "Large-scale convex minimization with a low-rank constraint," 2011, *arXiv:1106.1622*.

[8] A. Sagan and J. E. Mitchell, "Low-rank factorization for rank minimization with nonconvex regularizers," *Comput. Optim. Appl.*, vol. 79, no. 2, pp. 273–300, 2021.

[9] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[10] G. H. Golub, A. Hoffman, and G. W. Stewart, "A generalization of the Eckart-Young-Mirsky matrix approximation theorem," *Linear Algebra Appl.*, vols. 88–89, pp. 317–327, Apr. 1987.

[11] R. Madani, S. Sojoudi, G. Fazelnia, and J. Lavaei, "Finding low-rank solutions of sparse linear matrix inequalities using convex optimization," *SIAM J. Optim.*, vol. 27, no. 2, pp. 725–758, 2017.

[12] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Sci., 2015.

[13] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, "Distributed constrained optimization and consensus in uncertain networks via proximal minimization," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1372–1387, May 2018.

[14] N. Chatzipanagiotis and M. M. Zavlanos, "On the convergence of a distributed augmented lagrangian method for nonconvex optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4405–4420, Sep. 2017.

[15] P. Jain, R. Meka, and I. Dhillon, "Guaranteed rank minimization via singular value projection," in *Advances in Neural Information Processing Systems*, vol. 23. Red Hook, NY, USA: Curran Assoc., 2010.

[16] H. Zhang, Y. Bi, and J. Lavaei, "General low-rank matrix optimization: Geometric analysis and sharper bounds," *Advances in Neural Information Processing Systems*, vol. 34. Red Hook, NY, USA: Curran Assoc., 2021.

[17] C. Sun and R. Dai, "A customized ADMM for rank-constrained optimization problems with approximate formulations," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, 2017, pp. 3769–3774.