Analysis and Application of Distributed Event-triggered Algorithm for Matrix-variable Convex Optimization

Yiyuan Chai[a], Tianyi Xia[b], Sitian Qin[b], Jiqiang Feng[c,*], Chen Xu[c]

[a] *School of Electronics and Information Engineering, Shenzhen University, Shenzhen, 518060, China*
[b] *Department of Mathematics, Harbin Institute of Technology, Weihai, 264209, China*
[c] *Department of Mathematics and Statistics, Shenzhen Institute of Computing Sciences, Shenzhen University, Shenzhen 518060, China,*

Abstract

Matrix-variable optimization is an extension of vector-variable optimization that has shown potential in numerous significant applications. This paper presents a distributed matrix-formed algorithm for solving matrix-variable convex optimization problems with general equality, inequalities and set constraints. To alleviate the impact of frequent communication, a Zeno-free event-triggered mechanism is incorporated in the algorithm, which allows the agent to commence in unconstrained initial states. Theoretical analysis shows that the algorithm can be implemented in parallel on matrix space and the states of which finally globally converge to the optimal solution of the matrix-variable optimization problem. Additionally, the algorithm's efficacy in solving linear matrix equations and blind image restoration is showcased. The experimental results demonstrate that the distributed event-triggered algorithm herein outperforms the existing vector-formed algorithm and the traditional centralized matrix-formed algorithm in terms of computational speed and accuracy.

Keywords: Matrix-variable convex optimization; distributed algorithm; event-triggered mechanism; globally converge; linear matrix equation; image restoration.

1. Introduction

In the past decades, the size of data and system scale have significantly increased, which has increased the complexity of optimization problems within model control, intelligent computing, and wireless communication etc. [1, 2, 3]. Large-scale optimization problems are often beyond the capabilities of standard centralized algorithms due to limits in the hardware level of central processing unit and communication capacity. Therefore, numerous researchers have developed distributed algorithms based on multi-agent systems and demonstrated notable benefits in communication cost reduction, computational strain alleviation, and privacy preservation. The utilization of distributed algorithms has found extensive application in machine learning, interdisciplinary decision-making, distributed mobile robot networks, etc.[4, 5].

Currently, in widely focused optimization problems, it is a common practice for decision variables to be in vector format [6, 7]. However, within the context of big data era, a large amount of data is commonly in matrix format. In some practical engineering challenges, such as image restoration [8], personalized recommendation [9], and sparse encoding [10], optimization problems involving matrix variables are more prevalent as compared to those involving vector variables. Incorporating matrix variables directly into optimization problems can effectively preserve the structure and characteristics of the original problem, therefore enhancing the efficiency and precision of the optimization process. As a distinctive category of matrix-variable optimization problems, solving linear matrix equations (eg. continuous-time algebraic Riccati equation) holds significant importance in the field of stable system [11] and robust control [12].

[*] Corresponding author
Email addresses: `chaiyiyuan2018@email.szu.edu.cn` (Yiyuan Chai), `skylar.xty@gmail.com` (Tianyi Xia), `qinsitian@hitwh.edu.cn` (Sitian Qin), `fengjq@szu.edu.cn` (Jiqiang Feng), `xuchen@szu.edu.cn` (Chen Xu)

[1] This research is supported by the National Science Foundation of China under Grant 62176073 and Grant 11871178; Natural Science Foundation of Shenzhen under Grant 20200821143547001.

Currently, distributed optimization algorithms have demonstrated their efficacy in addressing linear matrix problems. In [13], it was the first proposition for employing a distributed algorithm to solve the linear equation $AX = b$, which operates on the assumption that each agent possesses the capability to access multiple rows of matrices A and b. By considering set constraints, Chen et al. [14] proposed a distributed algorithm based on projection method to find the least squares solution of the Stein equation. Cheng et al. [15] employed two distributed optimization models and related fractional order continuous-time algorithms to solve Sylvester equations. Additionally, in [16] authors presented various decomposition techniques for finding distributed solutions to linear matrix equations and suggested introduced alternative variables as a means to develop a distributed continuous time algorithm specifically designed for solving the equation $AXB = F$.

Despite the fact that distributed computing has made significant progress in solving linear matrix equations[15, 14, 17], there is still a lack of widely used distributed algorithm designed for general matrix-variable optimization problems where the objective function is typically convex or linear. That is, the distributed algorithms proposed up to this point still excessively rely on the specific form of the objective function. In addition, due to concerns about manufacturing efficiency and safety, many practical applications of matrix-variable optimization issues require for some specific constraints. Non-negative matrix optimization is a popular group of problems that calls for all components of the matrix decomposition factor to be non-negative[18, 19, 20]. This is due to the fact that negative aspects are frequently meaningless when dealing with practical issues like satellite-transmitted photos, real-time films that robots receive, and large-scale text that is stored in databases. In [21], the authors developed two matrix-form recurrent neural networks to solve matrix-variable optimization problems with linear constraints for effective application in blind image restoration, but in a centralized algorithm. Therefore, considering computational burden alleviation and privacy preservation, developing a continuous-time algorithm in a distributed way is essential for matrix-variable optimization problems with general constraints.

However, when dealing with matrix-variable optimization problems, the process of detailing local information within the matrix typically leads to a significant escalation in the complexity of interactions, thus results in an increase of data flow. For example, the distributed algorithm proposed by the authors in [17] utilized a block partition matrix decomposition method in which the number of agents employed scales with the square of the matrix dimension. Furthermore, the limited network transmission capacity and external disruptions impede the uninterrupted exchange of information among agents [22, 23], hence impairing the efficacy of distributed optimization algorithms. The findings demonstrated that the implementation of an event-triggered mechanism can significantly mitigate communication load and enhance communication stability [24, 25, 26]. For instance, a novel distributed optimization algorithm with event-triggered mechanisms was proposed in [27], where the event detection is decentralized and uses sampled data without continuous interactions to determine the threshold. Without monitoring of the measurement error continuously, in [28], a self-triggered mechanism is proposed, in which the next triggering time is precomputed based on the information available at the current time. Motivated by the studies above, it is crucial to create a distributed event-triggered optimization algorithm without Zeno behaviour for matrix-variable optimization problems with general constraints.

On the basis of previous research, this paper focuses on developing a general distributed optimization algorithm with event-triggered mechanism for solving matrix-variable optimization problem with equality and inequality constraints. Further, applying it to specific linear matrix equations and image restoration problems. The main contributions of this paper are listed as follows:

- Compared with the previous distributed algorithms for vector-variable optimization problems in [6, 7, 12], this paper studies the distributed algorithms for general matrix form optimization problems. The proposed algorithm is helpful to protect the original structure of the problem and to carry out parallel processing of large-scale optimization problems. Further, by considering multiple types of geometric constraints on matrix variables, this study takes into account practical scenario factors to ensure that the resulting solutions have significant practicability in specific applications.

- In order to reduce the communication loss of agents and ensure the stability of information interaction in comparison with [29, 30], we design a provably correct distributed event-triggered communication scheme without Zeno behavior to ensure the stability and reliability of the system. Using modified Lagrange function and derivative feedback, a distributed continuous-time algorithm is designed. The global convergence of the proposed algorithm under any initial conditions is strictly proved.

- We apply the proposed algorithm to solving linear matrix equations, which shows that our work extends the research content in [14, 30, 17], and the algorithm is effective in cases where the objective function is generally convex. On the other hand, in the application of blind image restoration, compared with the general vector and centralized matrix methods [21], the distributed event triggering algorithm proposed in this paper has obvious advantages in computing speed and accuracy.

The subsequent sections of this paper are organized as follows: In Section II, we provide essential preliminaries for readers. Section III outlines a matrix-variable optimization problem with linear constraints and introduces a distributed matrix-formed event-triggered algorithm. Section IV presents the theoretical analysis of the distributed event-triggered algorithm. In Section V, we showcase its practical applications in solving linear matrix equations and blind image restoration, supported by experimental results. Finally, our conclusions are summarized in Section VI.

2. Preliminaries

2.1. Matrices

Denote $\mathbb{R}^{m\times n}$ as the set of $m \times n$-dimensional real matrices. For a matrix $Q \in \mathbb{R}^{m\times n}$, we denote Q^T, $\text{rank}(Q)$, $\ker(Q)$, $\text{tr}(Q)$, $\text{int}(Q)$ as the transpose, rank, range, kernel, trace, relative interior of Q respectively. Denote $A \otimes B$ as the Kronecker product of matrices A and B. Denote $\|\cdot\|_F$ as the Frobenius norm of real matrices defined by $\|Q\|_F = \sqrt{tr(Q^TQ)} = \sqrt{\sum_{i,j} Q_{i,j}^2}$. And the Frobenius inner product of real matrices $P, Q \in \mathbb{R}^{m\times n}$ is written as $\langle P, Q\rangle_F = tr(P^{\mathrm{T}}Q) = \sum_{i,j}(P)_{i,j}(Q)_{i,j}$. Let $\text{col}\{x_1, \dots, x_n\} = \left[x_1^T, \dots, x_n^T\right]^T$. Denote 1_n as the n-dimensional vector composed of 1 and 0_n as the n-dimensional vector composed of 0. $A \succ 0$ ($A \succeq 0$) means that matrix A is positive definite (positive semi-definite). Next, a definition of embedded operator is given.

Definition 2.1. *[16] Let $\{l_j\}_{j=1}^N$ and $\{v_j\}_{j=1}^N$ be two sequences of n positive integers satisfying $\sum_{j=1}^N l_j = l$ and $\sum_{j=1}^N v_j = v$, and let $A_i \in \mathbb{R}^{v_j\times l_j}$ for $j \in \{1, \cdots, N\}$.*

The column embedded operator $\mathscr{J}_i^C$ on sub-matrix A_i is defined as

$$\mathscr{J}_i^C(A_i) = \left[0_{v_i\times l_1}, \dots, 0_{v_i\times l_{i-1}}, A_i, 0_{v_i\times l_{i+1}}, \dots, 0_{v_i\times l_N}\right] \in \mathbb{R}^{v_i\times l}.$$

The row embedded operator $\mathscr{J}_i^R$ on sub-matrix A_i is defined as

$$\mathscr{J}_i^R(A_i) = \left[0_{l_i\times v_1}, \dots, 0_{l_i\times v_{i-1}}, A_i^{\mathrm{T}}, 0_{l_i\times v_{i+1}}, \dots, 0_{l_i\times v_N}\right]^{\mathrm{T}} \in \mathbb{R}^{v\times l_i}.$$

2.2. Graph Theory

Some preliminaries about graph theory refers to [14]. For an n-agent system, the information exchange topology between the agents herein is described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges and $\mathcal{A} = [a_{i,j}] \in \mathbb{R}^{n\times n}$ represents the adjacency matrix of $\mathcal{G}$ where the element $a_{i,j}$ is the weight of the edge (j, i), $a_{i,j} = a_{j,i} > 0$ if $\{j, i\} \in \mathcal{E}$ and $a_{i,j} = 0$ otherwise. The Laplacian matrix of graph $\mathcal{G}$ is defined as $L = \mathcal{D} - \mathcal{A}$, where $\mathcal{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $\mathcal{D}_{i,i} = \sum_{j=1}^n a_{i,j}$, $i \in \{1, \dots, n\}$. If the undirected graph $\mathcal{G}$ is connected, then $L = L^T \succeq 0$, $\text{rank}(L) = n - 1$, and $\ker(L) = \{k1_n : k \in \mathbb{R}\}$.

2.3. Convex Analysis and Stability

Definition 2.2. *[14]*

1. *A set $\Omega \subseteq \mathbb{R}^{m\times n}$ is convex, if $\lambda X_1 + (1 - \lambda)X_2 \in \Omega$ holds for any X_1, $X_2 \in \Omega$ and $\lambda \in [0, 1]$.*
2. *If a function $f : \Omega \to \mathbb{R}$ is said to be convex, then Ω is a convex set and*

$$f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2), for\ \forall X_1, X_2 \in \Omega, \lambda \in (0, 1).$$

3. *Let $f : \Omega \to \mathbb{R}$ be a lower semi-continuous convex function. The subdifferential ∂f of f at $X \in \Omega$ is*

$$\partial f(X) := \{U \in \mathbb{R}^{m\times n} : \langle U, Y - X\rangle_F \leqslant f(Y) - f(X),\ \forall Y \in \Omega\}. \tag{1}$$

and the elements of $\partial f(X)$ are subgradients of f at point X.

Definition 2.3. *Let $\mathbf{\Omega} \subset \mathbb{R}^{m\times n}$ be a convex set.*

1. *Define $P_{\mathbf{\Omega}}(\cdot)$ as the projection operator of matrix $U \in \mathbb{R}^{m\times n}$ on $\mathbf{\Omega}$, given by*

$$P_{\mathbf{\Omega}}(U) = \operatorname{argmin}_{V\in\mathbf{\Omega}} \|U - V\|_{\mathrm{F}}^2 \tag{2}$$

2. *$\mathcal{N}_{\mathbf{\Omega}}(U)$ is the normal cone of U on $\mathbf{\Omega}$ if*

$$\mathcal{N}_{\mathbf{\Omega}}(U) = \left\{W \in \mathbb{R}^{m\times n} : \langle W, V - U\rangle_{\mathbf{F}} \leq 0, \quad \forall V \in \mathbf{\Omega}\right\} \tag{3}$$

On the basis above, we introduce following lemmas:

Lemma 2.1. *If $\mathbf{\Omega}$ is a closed convex set, then*

1. *For $\forall U \in \mathbb{R}^{m\times n}$, $V \in \mathbf{\Omega} \subset \mathbb{R}^{m\times n}$, there is*

$$\|P_\Omega(V) - P_\Omega(U)\|_F \leq \|V - U\|_F, \tag{4a}$$

$$\langle U - P_{\mathbf{\Omega}}(U), V - P_{\mathbf{\Omega}}(U)\rangle_F \leq 0 \tag{4b}$$

2. *V satisfies that*

$$tr((U - V)^T \nabla f(V) \geq 0, \forall U \in \mathbb{R}^{m\times n} \tag{5}$$

if and only if $V = P_\Omega(U - \alpha\nabla f(U))$, where α is a positive constant.

3. Problem description and algorithm design

3.1. Problem Description

In this paper, we consider following matrix-variable optimization problem with general constraints, the global cost function of which can be reformulated as the sum of N local cost functions:

$$\begin{aligned}
\text{minimize} \quad & f(X) = \sum_{i=1}^{N} f^i(X) \\
\text{subject to} \quad & g^i(X) \leq 0, \\
& A_i X = b_i, \\
& X \in \bar{\Omega} \triangleq \bigcap_{i=1}^{N} \Omega^i,\ i = 1, 2, ..., N,
\end{aligned} \tag{6}$$

where $X \in \mathbb{R}^{m\times n}$ is a matrix variable, $A_i \in \mathbb{R}^{p_i\times m}$ is a constant matrix, $b_i \in R^{p_i\times n}$, $f_i : \mathbb{R}^{m\times n} \to \mathbb{R}$ is local objective function, which are continuous convex functions but may be non-smooth, $g^i = \left(g^{i1}, g^{i2}, \ldots, g^{ir_i}\right)^{\mathrm{T}} : \mathbb{R}^{m\times n} \to \mathbb{R}^{r_i}$ represents local inequality constraints and of which all components g^{ir} $(i \in \{1, 2, \cdots N\},\ r = 1, 2, \ldots, r_i)$ are convex but may be non-smooth. Each Ω^i is a closed convex set. Without loss of generality, the following assumption is set up throughout this paper.

Assumption 1. *There exists an interior point $\hat{X} \in \bar{\Omega}$ such that $g^i(\hat{X}) < 0$, and $A_i\hat{X} = b_i$, where $i \in \{1, 2, ..., N\}$.*

Assumption 2. *The topology graph $\mathcal{G}$ of N-agents is undirected and connected.*

Remark 1. *The matrix-variable optimization problems (6) in this article have significant theoretical and practical value in certain application scenarios compared to those with variables in vector form. First of all, there are intrinsic benefits to using matrix variables to describe multidimensional data structures. For instance, matrices offer an understandable representation for pictures, films, and other high-dimensional data [31] that better satisfies the particular structural requirements of particular fields. Secondly, a larger mathematical expression space for optimization problems is provided by the basic matrix operations, such as matrix multiplication [32] and matrix decomposition [33], which not only enhances the flexibility of the algorithm, but also enables more accurate capture of multiple dependencies of underlying data. In addition, utilizing the characteristics of matrices, such as rank, eigenvalues, and singular values, can help us better investigate and comprehend the underlying structure of the original data. These properties enable the matrix-variable optimization (6) to be applied more effectively and profoundly in a variety of domains, including machine learning, signal processing, image recognition, etc.*

Actually, to further reduce the computational cost for large-scale optimization problems, it requires to seek distributed solver. That is, in considered multi-agent network $\mathcal{G}$, every agent only knows one particular objective function f^i and the feasible constraint g^i for $i = 1, 2, \cdots, N$. Under Assumption 2, the optimization problem (6) can be reformulated as following distributed matrix-variable optimization problem according to [34]:

$$\begin{aligned} \text{minimize} \quad & f(X) = \sum_{i=1}^{N} f^i(X_i) \\ \text{subject to} \quad & X_i = X_j, \\ & g^i(X_i) \leq 0, \\ & A_i X_i = b_i, \\ & X \in \bar{\Omega} = \prod_{i=1}^{N} \Omega^i, \ i, j \in \{1, 2, \cdots, N\}, \end{aligned} \tag{7}$$

where $X_i \in \mathbb{R}^{m\times n}$ is the feasible solution on agent i to problem (7), $\boldsymbol{X} = \text{col}\{X_1, X_2, ..., X_N\} \in \mathbb{R}^{Nm\times n}$. After this transformation, it can be seen that each agent i only knows local cost function and local constraints information. Thus, we only need to investigate the problem (7) in the following content.

Remark 2. *This study translates the original matrix-variable optimization problem (6) to a distributed optimization framework (7), which aims to fully utilize the parallel characteristic benefits of distributed computing to enable the system to achieve efficient computation. Meanwhile, consensus constraints are used to guarantee the robustness and consistency of global solutions by coordinating decision-making among agents in a distributed setting. Furthermore, our work has produced more precise and trustworthy predictions in complicated circuit design and structural analysis, as well as profoundly extended research on approaches for addressing matrix equation problems in [14, 30, 17]. Furthermore, our study considers multiple types of geometric constraints on matrix variables, taking practical scenario factors into account, such as physical constraints, resource allocation, and real operating conditions. This ensures noticeable practicality in specific applications such as supply chain optimization and network flow control.*

3.2. Distributed Event-triggered Algorithm Design

In this subsection, an event-triggered law for determining the triggering instants is presented to overcome the influence of large data flow brought by real-time interactive communication.

Let $\tilde{X}_i(t) = X_i\left(t_k^i\right)$, $\tilde{\lambda}_i(t) = \lambda_i\left(t_k^i\right)$, $\forall t \in \left[t_k^i, t_{k+1}^i\right)$, where t_k^i is the k-th triggering instant of i-th agent $(i = 1, 2, \cdots, N)$ defined by

$$t_{k+1}^i = \inf\left\{t > t_k^i \mid \psi_i(t) \geq 0\right\}, \tag{8}$$

where $t_0^i = 0$ and $\psi_i(t)$ is the triggering function which will be designed later. The i-th agent updates its states according to the information received at the latest event time instant. For i-th agent and $t \geq 0$, define a measurement error $\mathbf{e}_i(t) = \|\mathbf{e}_i^X(t)\|_F + \|\mathbf{e}_i^\lambda(t)\|_F$, where $\mathbf{e}_i^X(t) = \tilde{X}_i(t) - X_i(t)$ and $\mathbf{e}_i^\lambda(t) = \tilde{\lambda}_i(t) - \lambda_i(t)$, respectively. Then, we design the triggering function ψ_i to give an explicit description of how to trigger an action based on specific conditions as follows:

$$\psi_i(t) = (\alpha + \sum_{j\in\mathcal{N}_i} a_{ij}^2)\|\mathbf{e}_i(t)\|_F - \omega e^{-\varsigma t} \tag{9}$$

where $\omega, \varsigma \in \mathbb{R}_+$, $\alpha = c/4$. At the triggered time, the i-th agent broadcasts its current state to neighbors. Meanwhile, the measurement error $\mathbf{e}_i$ is reset to zero.

Remark 3. *The event triggering rule (8) introduced in this article makes the algorithm more flexible and efficient in automatic control and communication systems. Its core advantage is that it does not require operations based on fixed time intervals, but only activates when specific conditions or events occur, thereby saving resources and reducing overall system overhead. The structure of the trigger function $\psi_i(t)$ showcases a dynamic boundary related to the current deviation of the system and the weight of adjacent nodes, which means that the triggering mechanism can adaptively adjust based on the current system state and the influence brought by adjacent nodes. In addition, by combining the time decay term* $\mathrm{e}^{-\varsigma t}$ *with the error norm, it ensures that triggering only occurs when it's actually needed, thereby reducing frequent and unnecessary information interaction and enabling the real system to operate in a more intelligent, efficient, and energy-saving manner.*

To cooperatively seek the optimal solution of problem (7) over N-agent network, we present a continuous-time algorithm for distributed optimization problem (7):

$$\begin{cases} \dot{X}_i \in 2\left[-X_i + P_{\Omega_i}\left(X_i - \partial f^i\left(X_i\right) - \sum_{j\in\mathcal{N}_i} a_{ij}\left(\tilde{X}_i - \tilde{X}_j\right) - \sum_{j\in\mathcal{N}_i} a_{ij}\left(\tilde{\lambda}_i - \tilde{\lambda}_j\right)\right.\right. \\ \qquad \left.\left. + A_i^{\mathrm{T}}\mu_i - A_i^{\mathrm{T}}\left(A_iX_i - b_i\right) - \partial^{\mathrm{T}} g^i(X_i)[(\gamma_i + g^i)^+ \otimes I_n]\right)\right] \\ \dot{\lambda}_i = X_i \\ \dot{\mu}_i = -A_iX_i + b_i \\ \dot{\gamma}_i = -\gamma_i + \left(\gamma_i + g^i\right)^+ \end{cases} \tag{10}$$

where $\lambda_i \in \mathbb{R}^{m\times n}$, $\gamma_i = (\gamma_{i1}, \gamma_{i2}, \dots \gamma_{ir_i})^{\mathrm{T}} \in \mathbb{R}^{r_i}$ and $\mu_i \in \mathbb{R}^{p_i\times n}$ are auxiliary variables. $\left(\gamma_i + g^i\right)^+ = \max\{0, \gamma_i + g^i\}$.

Remark 4. *In the delineated distributed continuous-time algorithm (10), the state dynamics of each agent or node are predominantly driven by a series of interconnected components. Specifically, the term $-X_i$ introduces a damping effect, compelling the states to regress towards equilibrium. The projection function, $P_{\Omega_i}(\cdot)$, ensures that the evolving states remain within a specific set Ω_i, thereby maintaining the feasibility of solutions. Through the gradient term, $\partial f^i(X_i)$, the system is induced to evolve in a manner that minimizes the objective function, as it directs the system's trajectory along the steepest descent direction of the function. Furthermore, the auxiliary variables λ_i operates encoding constraint-related information,which acts as a Lagrange multiplier, while μ_i and γ_i aid in encoding system constraints and adjustments respectively. Given the structure of the system, particularly the gradient terms, the convexity of the functions involved ensures the existence and computability of the required subdifferentials.*

4. Main Results

Theorem 4.1. *Under Assumptions 1-2, if X_i^* is the optimal solution of distributed matrix-variable optimization problem (10), then there exist $\lambda_i^* \in \mathbb{R}^{m\times n}$, $\mu_i^* \in \mathbb{R}^{p_i\times n}$, $\gamma_i^* \in \mathbb{R}^{r_i}$, such that $(X_i^*, \lambda_i^*, \mu_i^*, \gamma_i^*)$*

satisfies

$$\begin{cases} X_i^* = P_{\Omega_i}\Bigg(X_i^* - \partial f^i\left(X_i^*\right) - \sum\limits_{j\in\mathcal{N}_i} a_{ij}\left(X_i^* - X_j^*\right) - \sum\limits_{j\in\mathcal{N}_i} a_{ij}\left(\lambda_i^* - \lambda_j^*\right) \\ \qquad +A_i^{\mathrm{T}}\mu_i^* - A_i^{\mathrm{T}}\left(A_iX_i^* - b_i\right) - \partial^T g^i(X_i^*)[(\gamma_i^* + g^i(X_i^*))^+ \otimes I_n]\Big) & \text{(11a)} \\ \gamma_i^* = \left(\gamma_i^* + g^i(X_i^*)\right)^+ & \text{(11b)} \\ A_iX_i^* = b_i, & \text{(11c)} \\ \sum\limits_{i=1}^{N}\sum\limits_{j\in\mathcal{N}_i} a_{ij}(X_i^* - X_j^*) = 0,\ i,j\in\{1,2,...,N\}, & \text{(11d)} \end{cases}$$

and vice versa.

Proof. Necessity: According to the KKT condition, if X_i^* is the optimal solution of distributed matrix-variable optimization problem (7), then for $i = 1, 2, \ldots, N$, there exist $\lambda_i^* \in \mathbb{R}^{m\times n}$, $\mu_i^* \in \mathbb{R}^{p_i\times n}$, $\gamma_i^* \in \mathbb{R}^{r_i}$ such that the following conditions is satisfied:

$$\begin{cases} 0 \in \sum\limits_{i=1}^{N} \partial f^i(X_i^*) + \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} a_{ij}(\lambda_i^* - \lambda_j^*) - \sum\limits_{i=1}^{N} A_i^{\mathrm{T}}\mu_i^* \\ \qquad + \sum\limits_{i=1}^{N} \partial g_i\left(X_i^*\right)\left(\gamma_i^* \otimes I_n\right) + N_\Omega(\boldsymbol{X^*}), & \text{(12a)} \\ \gamma_{ij}^* \geq 0, g_{ij}\left(x_i^*\right) \leq 0, \gamma_{ij}^* g_{ij}\left(x_i^*\right) = 0,\ i \in I,\ j = 1, 2, \ldots, r_i, & \text{(12b)} \\ A_iX_i^* = b_i, & \text{(12c)} \\ \sum\limits_{i=1}^{N}\sum\limits_{j\in\mathcal{N}_i} a_{ij}(X_i^* - X_j^*) = 0,\ i,j\in\{1,2,...,N\}, & \text{(12d)} \end{cases}$$

By the definition of the normal cone in (3), for any $X_i,\ X_i^*$ within the feasible set Ω_i, there exist measurable function $\eta_i \in \partial f^i(X_i)$, $\eta_i^* \in \partial f^i(X_i^*)$, as well as $\xi_i \in \partial g^i\left(X_i\right)$ and $\xi_i^* \in \partial g^i\left(X_i^*\right)$ such that

$$\langle X_i - X_i^*, \eta_i^* + \sum_{j\in\mathcal{N}} a_{ij}(\lambda_i^* - \lambda_j^*) - A_i^{\mathrm{T}}\mu_i^* + \xi_i^*(\gamma_i^* \otimes I_n)\rangle_F \geq 0 \tag{13}$$

Here, we define $\varLambda_i^* := \eta_i^* + \sum_{j\in\mathcal{N}} a_{ij}(\lambda_i^* - \lambda_j^*) - A_i^{\mathrm{T}}\mu_i^* + \xi_i^*(\gamma_i^* \otimes I_n)$. Now, considering the projection property in (4b), for any $X_i \in \Omega_i$, we have

$$\langle X_i - P_{\Omega_i}(X_i^* - \varLambda_i^*), P_{\Omega_i}(X_i^* - \varLambda_i^*) - (X_i^* - \varLambda_i^*)\rangle \geq 0 \tag{14}$$

Since X_i^* is the optimal solution of matrix-variable optimization problem (7), it follows that $X_i^* \in \Omega_i$. Replacing X_i with X_i^* in the equation above, it has

$$\langle X_i^* - P_{\Omega_i}(X_i^* - \varLambda_i^*), P_{\Omega_i}(X_i^* - \varLambda_i^*) - (X_i^* - \varLambda_i^*)\rangle \geq 0$$

Furthermore, when combined with the previous equation (13), it leads to:

$$\|X_i^* - P_{\Omega_i}(X_i^* - \varLambda_i^*)\|_F^2 \leq \langle X_i^* - P_{\Omega_i}(X_i^* - \varLambda_i^*), \varLambda_i^*\rangle \leq 0 \tag{15}$$

The last inequality arises from the fact that $P_{\Omega_i}(X_i^* - \varLambda_i^*) \in \Omega_i$, ultimately leading to:

$$X_i^* = P_{\Omega_i}(X_i^* - \varLambda_i^*) \tag{16}$$

Now, let's define sets: $I_0 := \{i|\ \gamma_i^* = 0\}$, $I_+ := \{i|\ \gamma_i^* > 0\}$, $K_0 := \{i|\ g^i(X_i^*) = 0\}$ and $K_- = \{i|\ g^i(X_i^*) < 0\}$. If X_i^* and λ_i^* satisfy (12), then we can conclude that i belongs to either $I_0 \bigcap K_0$, $I_+ \bigcap K_0$, or $I_+ \bigcap K_0$.

In these three cases, it is evident from (12) and the definition of $(\gamma_i^* + g^i(X_i^*)^+$ that $(\gamma_i^* + g^i(X_i^*)^+ = \gamma_i^*$ holds for any i. Therefore, with (12b), (12c) and (12d), it can be deduced that (12a) is also established.

Sufficiency: According to (11b), $\gamma_i^* g^i(X_i^*) = 0$ is established. Moreover, as $g^i(X_i^*) \leq 0$, it indicates that (12b) is satisfied. Substituting (11b), (11c), (11d) into (11a), we can infer the existence of $\eta_i^* \in \partial f^i(X_i^*)$ and $\xi_i^* \in \partial g^i(X_i^*)$, such that

$$X_i^* = P_{\Omega_i}\left(X_i^* - \eta_i^* - \sum_{j\in\mathcal{N}_i} a_{ij}\left(\lambda_i^* - \lambda_j^*\right) + A_i^{\mathrm{T}}\mu_i^* - \xi_i^*(\gamma_i^* \otimes I_n)\right).$$

Similarity, by denoting $\eta_i^* + \sum_{j\in\mathcal{N}} a_{ij}(\lambda_i^* - \lambda_j^*) - A_i^{\mathrm{T}}\mu_i^* + \xi_i^*(\gamma_i^* \otimes I_n)$ as Λ_i^*, we can conclude that:

$$\langle X_i - X_i^*, \Lambda_i^* \rangle = \langle X_i - P_{\Omega_i}(X_i^* - \Lambda_i^*), \Lambda_i^* \rangle = \langle X_i - P_{\Omega_i}(X_i^* - \Lambda_i^*), P_{\Omega_i}(X_i^* - \Lambda_i^*) - (X_i^* - \Lambda_i^*) \rangle.$$

By considering the projection property (4b), for any X_i in Ω_i, it can be deduced that (12a) is established by considering (13)., thereby completing the sufficiency proof.

□

It should be pointed that the right hand of algorithm (10) is a nonempty set-valued map with compact convex values, then there exists a local state solution of algorithm (10) starting from arbitrary initial point. By the convexity of cost function, the optimal solution X^* of (7) exits. Recalling Theorem 4.1, the point satisfying (11) coincides with the optimal solution of matrix-variable optimization problem (7). In the following part, we are ready to analyze the convergence of (10).

Theorem 4.2. *Assume that Assumptions 1 and 2 are valid, for any given initial condition $X(0), \lambda(0), \mu(0), \gamma(0)$, the trajectory of distributed event-triggered algorithm (10) with the triggering function (9) converges to the optimal solution of distributed matrix-variable optimization problem (7).*

Proof. See Appendix. □

Remark 5. *To illustrate the performance of our proposed distributed event-triggered algorithm, we compare the proposed matrix-formed algorithm in (10) with existing vector-formed continuous-time distributed sub-gradient algorithm in [35]. First, the proposed matrix-formed distributed algorithm model generalizes the vector-formed. Second, in terms of model complexity defined as the state space and the total number of multiplications/divisions per iteration, we see that the proposed matrix-formed distributed algorithm have the state space being $N*(m\times n+n\times n)$, where N represents the number of agents and the computational complexity being $O(mn^2)$, respectively. In contrast, we define a function $\varphi(x)$, where $x = \left[x_1^T, \cdots, x_n^T\right] \in \mathbb{R}^{m\times n}$ and $x_i \in \mathbb{R}^m (i = 1, \cdots, n)$. Then (6) can be*

$$\min \varphi(\mathrm{x}) = \min \sum_{i=1}^{N} \varphi^i(x_i) \;\; s.t. \quad \Lambda\mathrm{x} = \mathbf{b}, \quad g(x) \leq 0, \quad \mathrm{x} \in \Omega \tag{17}$$

where $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$, $\Omega_i = \{\mathrm{x_i} \geq 0\}\,(i = 1, \ldots N)$, $\mathbf{b} = \left[\mathbf{b}_1^T, \ldots, \mathbf{b_n}^T\right] \in \mathbb{R}^{n^2}$, and $\Lambda = \mathrm{diag}(A, A, \ldots, A) \in \mathbb{R}^{n^2\times mn}$.

To solve (17), authors proposed a vector-type continuous-time distributed sub-gradient algorithm in [35] described as:

$$\begin{cases} \dot{x}_i \in & 2\Big[-x_i + P_i\Big(x_i - \partial\varphi_i(x_i) - \sum_{j,j\in N_i} a_{ij}(x_i - x_j) \\ & - \sum_{j,j\in N_i} a_{ij}(\lambda_i - \lambda_j) + A_i^T\mu_i - A_i^T(A_i x_i - b_i) \\ & -\partial g_i(x_i)(\gamma_i + g_i)^+\Big)\Big] \\ \dot{\lambda}_i = & x_i \\ \dot{\mu}_i = & -A_i x_i + b_i \\ \dot{\gamma}_i = & -\gamma_i + (\gamma_i + g_i)^+. \end{cases} \tag{18}$$

where $x_i \in \mathbb{R}^m, \lambda_i \in \mathbb{R}^m, \mu_i \in \mathbb{R}^m$ and $\gamma_i \in \mathbb{R}^1$. It is seen that the vector-type continuous-time distributed sub-gradient algorithm in [35] has the state space being $(m+m+m+1)*n*N$ and algorithm complexity being $O(mn^3)$ We see that the proposed matrix-formed algorithm have the lower model complexity than the vector-formed one.

Next, we will show that the distributed event-triggered algorithm (10) does not exhibit the Zeno behavior.

Theorem 4.3. *The distributed event-triggered algorithm (10) does not exhibits Zeno behaviors and for any agent, the interval between two consecutive triggering instants in finite time is strictly positive.*

Proof. It follows from (10) that the upper right-hand Dini derivative of e_i over the interval $t \in \left[\tau_k^i, \tau_{k+1}^i\right)$ can be written as

$$\begin{aligned}
&\mathrm{D}^+ e_i(t) = \|\dot{X}_i\|_F + \|\dot{\lambda}_i\| F \\
&\dot{X}_i = 2\left[-X_i + P_{\Omega_i}\left(X_i - \eta_i - \sum_{j\in\mathcal{N}_i} a_{ij}\left(\tilde{X}_i - \tilde{X}_j\right) - \sum_{j\in\mathcal{N}_i} a_{ij}\left(\tilde{\lambda}_i - \tilde{\lambda}_j\right)\right.\right. \\
&\qquad \left.\left. + A_i^{\mathrm{T}}\mu_i - A_i^{\mathrm{T}}\left(A_i X_i - b_i\right) - \xi_i[(\gamma_i + g^i)^+ \otimes I_n]\right)\right] \\
&\dot{\lambda}_i = X_i
\end{aligned}$$

where $\eta_i \in \partial f^i(X_i)$ and $\xi_i \in \partial g^i(X_i)$. Considering that $e_i\left(t_k^i\right) = 0$, the solution of $e_i(t)$ is obtained as follows:

$$\begin{aligned}
e_i(t) = &\int_{\tau_{k'}}^{t} \|P_{\Omega_i}\left(X_i(\tau) - \eta_i(\tau) - \sum_{j\in\mathcal{N}_i} a_{ij}(X_i(\tau_k^i) - X_j(\tau_m^j)) - \sum_{j\in\mathcal{N}_i} a_{ij}(\lambda_i(\tau_{k'}^i) - \lambda_j(\tau_m^j))\right. \\
&\qquad \left. + A_i^{\mathrm{T}}\mu_i(\tau) - A_i^{\mathrm{T}}(A_i X_i(\tau) - b_i) - \xi_i(\tau)[(\gamma_i(\tau) + g^i(\tau))^+ \otimes I_n]\right)\|_F \ \ \mathrm{d}\tau \\
&+ 3\int_{\tau_k^i}^{t} \|X_i(\tau)\|_F \ \ \mathrm{d}\tau
\end{aligned}$$

Here, τ_k^i and τ_m^j are the latest triggering instants of agent i and agent j, respectively. Since X_i, λ_i, μ_i and γ_i are bounded by above analysis. Further, Ω_i is bounded closed convex set. Thus, we define c_1 and c_2 as positive constants such that $\|X_i\| \leq c_1$ and

$$\|P_{\Omega_i}(X_i - \eta_i - \sum_{j\in\mathcal{N}_i} a_{ij}(\tilde{X}_i - \tilde{X}_j) - \sum_{j\in\mathcal{N}_i} a_{ij}(\tilde{\lambda}_i - \tilde{\lambda}_j) + A_i^{\mathrm{T}}\mu_i - A_i^{\mathrm{T}}(A_i X_i - b_i) - \xi_i^{\mathrm{T}}[(\gamma_i + g^i)^+ \otimes I_n])\|_F \leq c_2$$

This inequality holds for all $i, j \in \mathcal{I}$ and $t > 0$. Let $q = 3c_1 + c_2$. According to the triggering function (9), the next event will not be triggered until $\|\mathbf{e}_i(t)\|^2 = \omega e^{-\varsigma t} = \vartheta, t \in \left[\tau_k^i, \tau_{k+1}^i\right)$. Thus, it follows that $\left\|\mathbf{e}_i\left(\tau_{k+1}^{i-}\right)\right\|_F = \vartheta$ before the next triggering time instant τ_{k+1}^i, where τ_{k+1}^{i-} is the left limit of τ_{k+1}^i. Let $\tilde{\tau}_i = \tau_{k+1}^{i-} - \tau_k^i$, take account of inequality above, we have $\vartheta \leq q\tilde{\tau}_i$. Thus, it is evident that $\tilde{\tau}_i \geq \vartheta/q > 0$, which indicates a lower bound on the inter-event intervals is strictly positive. □

Remark 6. *The event-triggered mechanism (9) plays a crucial role in our study by not only reducing communication overhead but, more importantly, by eliminating Zeno behavior. In practical applications, the avoidance of Zeno behavior is of paramount importance. Zeno behavior refers to a scenario where a system triggers events infinitely rapidly within infinitesimally short time intervals, leading to an infinite consumption of computational and communication resources and an inability to terminate. By introducing the event-triggered mechanism, we can ensure system stability and bounded computational resource utilization, thus ensuring the feasibility and practicality of the algorithm. This innovative approach holds significant practical implications in addressing distributed matrix-variable optimization problems, allowing the algorithm (10) to handle complex real-world scenarios with enhanced feasibility and efficiency.*

5. Applications for matrix-variable optimization

As the applications of matrix-variable optimization problem (7), we apply algorithm (10) in solving linear matrix equation and in the application of blind image restoration.

5.1. Application in solving linear matrix equation

Consider following linear matrix equation:

$$AX = C \tag{19}$$

where $A = [1,6,3;3,7,6;4,3,0;1,6,3]$, $C = [1,6,3;3,7,6;5,2,9;1,6,3]$ and $X \in \mathbb{R}^{3\times3}$ is the unknown matrix which satisfies the set constraints $X \in \Omega = \{X| -10 \leqslant X_{ij} \leqslant 10\}$; the inequality constraints

$$-\left(X_{11}^2 + X_{11}\right) + 5 \leqslant 0 \qquad X_{22} + \ln X_{22} - 2 \leqslant 0$$
$$e^{(-X_{31})} + X_{31} - 2 \leqslant 0 \qquad \frac{1}{2}X_{32}^2 - 1.5 \leqslant 0$$

and equality constraints $DX = b$, where $D = [1,6,3;3,7,6;4,3,0]$ and $b = [3,3,2;8,5,9;6,6,6]$.

Note that the linear matrix equation (19) may not have a solution X under the constraints. Nevertheless, it always has a least squares solution, which is the solution of following optimization problem:

$$\min_X \|AX - C\|_F^2 \tag{20}$$

To solve (19), we consider the distributed computation of a least squares solution to (19) over a 4-agent network as shown in Fig. Moreover, the interaction of the four agents under the communication topology is in Fig.**??**. In the complicated problem, each agent i $(i = 1,2,3,4)$ knows a part of distributed information of A and C. Then solving linear matrix equation (19) can be reformulated as following matrix-variable optimization problem in a distributed way:

$$\begin{aligned}
&\min_X \|AX - C\|_F^2 = \min_{X_i} \sum_{i=1}^{4} \|A_iX_i - C_i\|^2 \\
&\text{subject to } \sum_{i=1}^{4}\sum_{j\in N_i} a_{ij}\left(X_i - X_j\right) = 0 \\
&\qquad g^i\left(X_i\right) \leqslant 0 \\
&\qquad D_iX_i = b_i \\
&\qquad X \in \Omega = \prod_{i=1}^{4} \Omega_i,\ i \in \{1,2,3,4\}
\end{aligned} \tag{21}$$

where $X_i \in \mathbb{R}^{3\times3}$ is the unknown matrix needs to be calculated, $A_i \in \mathbb{R}^{1\times3}$ and $C_i \in \mathbb{R}^{1\times3}$ are only known by the i-th agent, the value of which comes from $A = [A_1;A_2;A_3;A_4]$ and $C = [C_1;C_2;C_3;C_4]$. The other coefficients are same as described above.

Next, we illustrate the effectiveness of our proposed event-triggered continuous-time algorithm (10) in solving linear matrix equation (19) with constraints by the distributed way.

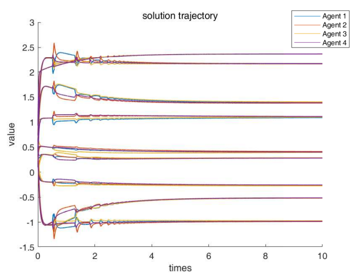

Figure 1: The trajectories of X_{ij} with algorithm (10)

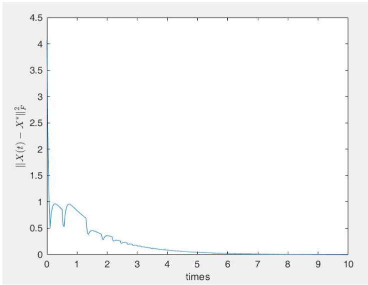

Figure 2: The trajectory of $\|AX - C\|_F^2$ with algorithm (10)

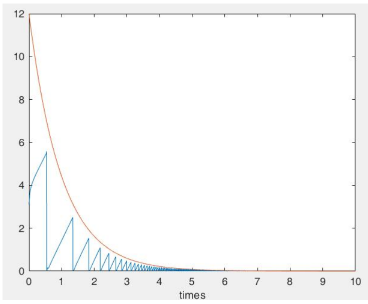

Figure 3: Behaviors of $_i^2(t)$ and $\omega e^{-\varsigma t}$ in distributed event-triggered algorithm (10)

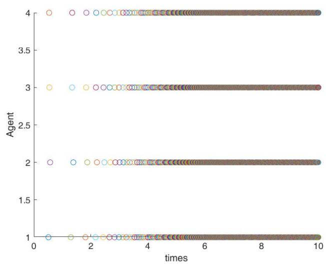

Figure 4: Triggering instants of distributed event-triggered algorithm (10)

From the simulation results shown in Fig.1, from arbitrary initial points, the trajectory behaviors of event-triggered algorithm (10) globally converge to the least squares solution $X^* = [2.3717, 0.4127, 1.090; -0.2690, 1.3972, -0.9839; 0.2856, -0.5196, 2.1716]$ to linear matrix equation (19). Fig. 2 depicts the behaviors of the objection function $\|AX - C\|_F$, which shows that $f(X)$ reaches the minimum in the end.

Fig. 3-4 show the behaviors of algorithm brought by the event-triggered mechanism. Firstly, we test several solutions by parameter sensitivity in order to set the best parameters in trigger function (9). The main purpose of introducing the trigger mechanism is to reduce the information interaction between the agent and the neighbor, so as to effectively reduce the communication cost, without affecting the convergence characteristics of the algorithm. Therefore, the sensitivity analysis of algorithm parameters ω and ς is carried out by measuring the trigger times. Finally, we choose $\omega = 12$ and $\varsigma = 1.0$ in triggering function (9), the results show that the triggering times are the least, the and the solution of the algorithm (10) can converge to the optimal value.

Fig. 3 shows the trajectories of ${}_i^2(t)$ and $\omega e^{-\varsigma t}$ for $i = 1, 2, 3, 4$. It indicates that the algorithm (10) with event-triggered mechanism can address the linear matrix equation (19) and lower the communication burden. Fig. 4 shows the triggering instants of all agents, it indicates that the communication is always discrete without Zeno behavior contributing to stable distributed system.

5.2. Application in image restoration

In this subsection, we focus on applying our presented matrix-valued distributed event-triggered algorithm in image restoration problem. The goal of image restoration is to handle image degradation that occurs during image acquisition and processing, and to restore the original image as much as possible. The image restoration algorithm has been widely applied in various fields, such as video denoising tasks [36], biological images reconstruction[37], underwater image enhancement[38]. Image blur kernel describes the way in which image blur is generated. According to whether the blur kernel is known or not, image deblurring algorithms can be divided into two types. The first type is non blind image restoration, where the blur kernel information is known. When the blur kernel is unknown, the image restoration problem is called blind image restoration [39, 40].

In the previous studies, usually degraded image is converted into the following vector-variable optimization model [21]:

$$g = \tilde{H} f + \text{Noise} \tag{22}$$

Some vector-variable optimization algorithms are applied in solving restoration problems. In [41], the authors proposed numerical methods. While over decades, recurrent neural networks (RNN) are developed for image restoration [21], with potential advantages of addressing nonlinear and non-smooth optimization problems, but considered time-consuming and requiring large storage capacity. Moreover, the vector-variable optimization algorithms has limit advantage in preserving the structure and characteristics of the original problem, therefore it motivates us to design the matrix-variable optimization algorithm to enhance the efficiency and precision of the optimization process.

In order to reduce the storage space of (22), one better way is based on the matrix observation model [42]:

$$\begin{aligned} G &= H_1 F H_2 + \text{Noise} \\ \text{where} \quad \tilde{H} &= H_2 \odot H_1 \end{aligned} \tag{23}$$

where the blur matrix $\tilde{H}$ can be decomposed into $H_1 \in \mathcal{R}^{m \times m}$ and $H_2 \in \mathcal{R}^{n \times n}$ by the blur kernel decomposition algorithm [43].

To reduce computation time, recent works on matrix-variable optimization model are popular for blind image deblurring method. With the proof in [42] that the blur matrices H_1 and H_2 satisfy the constraints, the constrained matrix degradation model is proposed as:

$$\begin{aligned} G =& H_1 X H_2 + W \\ \text{s.t.} \quad & E_1 H_1 = E_1, H_1 \geqslant 0 \\ & E_2 H_2 = E_2, H_2 \geqslant 0 \\ & X \in \Omega \end{aligned} \tag{24}$$

where $H_1 \in \mathbb{R}^{m\times m}$ and $H_2 \in \mathbb{R}^{n\times n}$ is the blur circulant matrix satisfying $E_iH_i = E_i$, $E_1 \in R^{m\times m}$ and $E_2 \in R^{n\times n}$ are matrix with elements being 1. $X \in R^{m\times m}$ denotes the matrix form of the original image, $G \in \mathbb{R}^{m\times n}$ is the matrix form of the degraded image, $W \in \mathbb{R}^{m\times n}$ represents the additional Gaussian Noise. $\Omega = \{X \in R^{m\times n} \mid L \leqslant X \leqslant U\}$, L is a zero matrix , U is an matrix with elements being 255. Then, the application to image restoration can be reformulated to solving following matrix-variable optimization problem:

$$
\begin{aligned}
\min_{X,H_1,H_2} & \frac{\beta}{2}\|H_1XH_2 - G\|_F^2 + \Phi\left(D_xX, D_yX\right) + \frac{\beta_1}{2}\|H_1\|_F^2 + \frac{\beta_2}{2}\|H_2\|_F^2 \\
\text{s.t. } & E_1H_1 = E_1, \quad H_1 \geqslant 0 \\
& E_2H_2 = E_2, \quad H_2 \geqslant 0 \\
& X \in \Omega
\end{aligned}
\tag{25}
$$

where $\beta, \beta_1 \in \mathbb{R}^+$ is the regularization parameter, $\Phi\left(D_xX, D_yX\right) = \sum_i ((D_xX)_i^2 + (D_yX)_i^2)^{1/2}$ is the regularization term about the image X, and D_x, D_y are gradient operators along the horizontal and vertical directions, respectively.

According to **??**, we take the gradient operator matrix as

$$
D_x = D_y^T = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & 0 & -1 & 1 \\ -0.5 & 0 & \cdots & \cdots & 0 & 0.5 \end{bmatrix}_{N\times N}
\tag{26}
$$

To realize the image restoration in a distributed way, we formulate three distributed optimization subproblems based on the matrix-variable optimization problem (25) in the following three subsections.

5.2.1. Subproblem 1 for matrix-variable optimization problem (25)

Define $X_i \in \mathbb{R}^{m\times n}$ and introduce a replaceable variable $Y_i \in \mathbb{R}^{m\times n}$, $Y_i := H_1X_i$, such that

$$
X := \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \in \mathbb{R}^{Nm\times n},\ Y := \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix} \in \mathbb{R}^{Nm\times n}.
$$

Next, define matrices $H_1^{v_i} \in \mathbb{R}^{m_i\times m}$, $Y_i^{v_i} \in \mathbb{R}^{m_i\times n}$, $H_2^{l_i} \in \mathbb{R}^{n\times n_i}$ and $G^{l_i} \in \mathbb{R}^{m\times n_i}$ as

$$
\begin{aligned}
Y_i &:= \begin{bmatrix} Y_i^{v_1} \\ \vdots \\ Y_i^{v_N} \end{bmatrix} \in \mathbb{R}^{m\times n}, \quad H_1 := \begin{bmatrix} H_1^{v_1} \\ \vdots \\ H_1^{v_N} \end{bmatrix} \in \mathbb{R}^{m\times m}, \\
H_2 &:= \left[H_2^{l_1}, \cdots, H_2^{l_N}\right] \in \mathbb{R}^{n\times n}, \quad G := \left[G^{l_1} \cdots, G^{l_N}\right] \in \mathbb{R}^{m\times n}
\end{aligned}
\tag{27}
$$

where $\sum_{i=1}^N m_i = m$ and $\sum_{i=1}^N n_i = n$. Then, the degradation model is like $H_1^{v_i}X_i = Y_i^{v_i}$.

Let $f^i(X_i, Y_i) : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be a convex cost function. We formulate Subproblem 1 as

Subproblem 1: Propose a distributed algorithm to solve

$$
\begin{aligned}
\min_{X_i,Y_i} & \sum_{i=1}^N f_i(X_i, Y_i) = \sum_{i=1}^N \left[\frac{\beta}{2}\left\|Y_iH_2^{l_i} - G^{l_i}\right\|_F^2 + \left[(D_xX_i)^2 + (D_yX_i)^2\right]^{\frac{1}{2}}\right] \\
\text{s.t } & X_i = X_j, \quad Y_i = Y_j \\
& H_1^{v_i}X_i = Y_i^{v_i}, \quad i, j \in [1, \ldots, N] \\
& X_i \in \Omega := \{X_i \in \mathcal{R}^{m\times n} | L \le X_i \le U\}
\end{aligned}
\tag{28}
$$

where L is a zero matrix, U is a matrix with elements being 255. Then, as the analysis in Section 4, the event-triggered algorithm for (28) is presented as following:

$$
\begin{cases}
\dot{X}_i \in 2\left[-X_i + P_{\Omega_i}(X_i - \partial f^i(X_i) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{X}_i - \tilde{X}_j) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{\lambda}_1^i - \tilde{\lambda}_1^j) + {H_1^{v_i}}^T(\mu_i - (H_1^{v_i}X_i - Y_i^{v_i}))\right] \\
\dot{Y}_i \in 2\left[-Y_i + P_{\Omega_i}(Y_i - \partial f^i(Y_i) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{Y}_i - \tilde{Y}_j) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{\lambda}_2^i - \tilde{\lambda}_2^j) - \mathscr{I}_i^R(H_1^{v_i}X_i - Y_i^{v_i} + \mu_i)\right] \\
\dot{\lambda}_1^i = X_i \\
\dot{\lambda}_2^i = Y_i \\
\dot{\mu}_i = -H_1^{v_i}X_i + Y_i^{v_i}
\end{cases}
\tag{29}
$$

Some pseudocodes for distributed event-triggered algorithm (29) are list in Algorithm 1. In Algorithm 1, variable X_i and Y_i viewed as an estimated state, is computed by using the proximal-gradient primal–dual dynamics by alternating updates.

Algorithm 1 Distributed event-triggered algorithm for subproblem 1

Input: The initial image $X(0)$; System parameters; arbitrary initial guess $X_i(0)$, $Y_i(0)$, $\lambda_i^1(0)$, $\lambda_i^1(0)$, $\mu_i(0)$; maximum time interval T.

Each agent $\forall i \in \mathcal{I}$ broadcasts initial states to its neighbors.
for each agent $\forall i \in \mathcal{I}$ **do**
    Update the states of primal variables $X_i(t)$, $Y_i(t)$ via (29a) and (29b);
  Update the states of dual variables $\lambda_i^1(t)$, $\lambda_i^1(t)$, $\mu_i(t)$ via (29c) - (29e);
    Text the event-triggered condition in (9);
    if triggered **then**
        Broadcast $X_i(\tau_k^i)$, $\lambda_1^i(\tau_k^i)$ and $\lambda_1^i(\tau_k^i)$ to it neighbors;
    end if
end for

Output: the final state $X_i(T)$.

5.2.2. Subproblem 2 for matrix-variable optimization problem (25)

Define $H_1^i \in \mathbb{R}^{m\times m}$ where, $H_1 = col\{H_1^1, H_1^2, ..., H_1^N\} \in \mathbb{R}^{Nm\times m}$. For each H_1^i, it satisfies $E_1 H_1^i = E_1$. Then we devide H_2 to $\{H_2^{l_1}, H_2^{l_2}, \cdots, H_2^{l_N}\}$ where $H_2^{l_i}$ is $\in \mathbb{R}^{n\times n_i}$ and $\sum_{i=1}^N n_i = n$. And $G^{l_i} \in R^{m\times n_i}$ is defined as in (27). Let $f^i(H_1^i) : \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times n}$ be a convex cost function chosen by our preference. As a result, we can formulate Sub-problem 2 as

Subproblem 2: Propose a distributed algorithm to solve

$$
\begin{aligned}
\min_{H_1^i} \sum_{i=1}^N f_i(H_1^i) &= \sum_{i=1}^N \left[\left\| H_1^i X_i H_2^{l_i} - G^{l_i}\right\|_F^2 + \frac{\beta_1}{2}\left\|H_1^i\right\|_F^2\right] \\
\text{s.t.} \quad & E_1 H_1^i = E_1, \quad H_1^i \geqslant 0, \\
& H_1^i = H_1^j \\
& H_1^i \in \mathbb{R}^{m\times m}.
\end{aligned}
\tag{30}
$$

Then, the event-triggered algorithm for (30) is presented as following:

$$\begin{cases} \dot{H}_1^i \in 2\left[-H_1^i + P_{\Omega_i}(H_1^i - \partial f^i(H_1^i) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{H}_1^i - \tilde{H}_1^j) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{\lambda}_i - \tilde{\lambda}_j) + {E_1}^T(\mu_i - (E_1 H_1^i - E_1)) \right] \\ \dot{\lambda}_i = H_1^i \\ \dot{\mu}_i = -E_1 H_1^i + E_1 \end{cases} \tag{31}$$

Some pseudocodes for distributed event-triggered algorithm (31) are list in Algorithm 2. Compared with Algorithm 1, which the update rule follows the alternating primal–dual order. The update rule of Algorithm 2 follows the alternating dual–primal order. Specifically, Algorithm 2 first computes auxiliary dual variable, then updates auxiliary primal variables.

Algorithm 2 Distributed event-triggered algorithm for subproblem 2

Input: The output matrices $X_i(k)$, $H_2^{(k-1)}$ from the Algorithm 1 and 3 and $H_1^{(k-1)}$; System parameters; arbitrary initial guess; maximum time interval T'.

```
Each agent ∀i ∈ I broadcasts initial states to its neighbors.
for each agent ∀i ∈ I do
    Update the states of dual variables λ_i(t), μ_i(t) via (31b) and (31c);
  Update the states of primal variables H_1^i(t) via (31a);
    Text the event-triggered condition in (9);
    if triggered then
        Broadcast X_i(τ_k^i) and λ_i(τ_k^i) to it neighbors;
    end if
end for
```

Output: the final state $H_1^i(T')$.

5.2.3. Subproblem 3 for matrix-variable optimization problem (25)

Define $H_2^i \in \mathbb{R}^{n\times n}$, $H_2 = \mathrm{col}\{H_2^1, H_2^2, ..., H_2^N\} \in \mathbb{R}^{Nn\times n}$. For each H_2^i, it satisfies $E_2 H_2^i = E_2$. Then we divide H_1 to $\{H_1^{l_1}, H_1^{l_2}, \cdots, H_1^{l_N}\}$ where $H_1^{l_i} \in \mathbb{R}^{m_i\times m}$ and $\sum_{i=1}^N m_i = m$. And $G^{v_i} \in \mathbb{R}^{m_i\times n}$ is defined as follows:

$$G := \begin{bmatrix} G^{v_1} \\ \vdots \\ G^{v_N} \end{bmatrix} \in \mathbb{R}^{m\times n}, G^{v_i} \in R^{m_i\times n}$$

Let $f_i(H_2^i) : \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times n}$ be a convex cost function chosen by our preference. Thus, we can formulate Subproblem 3 as

Subproblem 3: Propose a distributed algorithm to solve

$$\begin{aligned} \min_{H_2^i} \quad & \sum_{i=1}^N f_i(H_2^i) = \min_{H_2^i} \sum_{i=1}^N \left[\left\| H_1^{l_i} X_i H_2^i - G^{v_i} \right\|_F^2 + \frac{\beta_2}{2} \left\| H_2^i \right\|_F^2 \right] \\ \text{s.t. } & E_2 H_2^i = E_2; \quad H_2^i \geqslant 0, \\ & H_2^i = H_2^j \\ & H_2^i \in \mathbb{R}^{n\times n}. \end{aligned} \tag{32}$$

Then, the event-triggered algorithm for (32) is presented as following:

$$
\begin{cases}
\dot{H}_2^i \in 2\left[-H_2^i + P_{\Omega_i}(H_2^i - \partial f^i(H_2^i) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{H}_2^i - \tilde{H}_2^j) - \sum\limits_{j\in\mathbf{N_i}} a_{ij}(\tilde{\lambda}_i - \tilde{\lambda}_j) + {E_1}^T(\mu_i - (E_2H_2^i - E_2))\right] \\
\dot{\lambda}_i = H_2^i \\
\dot{\mu}_i = -E_2H_1^i + E_2
\end{cases}
\tag{33}
$$

Some pseudocodes for distributed event-triggered algorithm (33) are list in Algorithm 3. The update rule of Algorithm 3 is the same as the order in Algorithm 2 by firstly computes auxiliary dual variable, then updates auxiliary primal variables.

Algorithm 3 Distributed event-triggered algorithm for subproblem 3

Input: The output matrices $X_i(k)$, $H_1^{(k-1)}$ from the Algorithm 1 and 2 and $H_2^{(k-1)}$; System parameters; arbitrary initial guess; maximum time interval T''.

Each agent $\forall i \in \mathcal{I}$ broadcasts initial states to its neighbors.
for each agent $\forall i \in \mathcal{I}$ **do**
    Update the states of dual variables $\lambda_i(t)$, $\mu_i(t)$ via (33b) and (33c);
  Update the states of primal variables $H_2^i(t)$ via (33a);
    Text the event-triggered condition in (9);
    if triggered **then**
        Broadcast $X_i(\tau_k^i)$ and $\lambda_i(\tau_k^i)$ to it neighbors;
    end if
end for

Output: the final state $H_2^i(T'')$.

5.2.4. Distributed event-triggered algorithm for solving (25)

In this section, we finally propose a distributed computation method for solving (25). In this method, Subproblem 1, Subproblems 2 and 3 are solved iteratively to obtain a final estimated image X. In all subproblems, the information of matrices is distributed among agents. Next, we use our three algorithms for three subproblems for steps in the following algorithm for the whole image restoration problem. The key details are listed in Algorithm 4

Algorithm 4 Distributed event-triggered algorithm for solving (25)

Input: Set parameters $\beta = 0.001, \beta_1 = 0.001, \beta_2 = 0.001$; degraded image G; the initial image $X_i(0) = G$; Initialize $H_1^{(0)}$ and $H_2^{(0)}$ by using SVD [42].

for each agent $\forall i \in \mathcal{I}$ **do**
    Update $X_i(t)$ by solving Subproblem 1 using 1;
    Update $H_1^{(t)}$ by solving Subproblem 2 using 2;
    Update $H_2^{(t)}$ by solving Subproblem 3 using 3;
end for

Output: The estimated image X.

We tested our algorithm 4 on six 256×256 pixel images displayed in Fig.5. We degraded the first three images by using 7×7 Gaussian blur with variance being 1.5 and adding random noise with the variance being 10^{-4} and the last three were degraded by 9×9 Gaussian blur with variance being 1.5 and contaminated by Gaussian noise with variance being 10^{-4} shown as "Degraded Image" in Fig.5(b). And we compared the performance of our algorithm with that of traditional vector-form RNN in [44]. The output images are displayed respectively in Fig.5(c) and Fig5(d) as "matrix-form algorithm" and "vector-form algorithm".

Figure 5: Restoration results on two algorithms

Table 1: Computed result of image restoration

Image	Evaluation index	vector-form algorithm	centralized matrix-form algorithm	distributed matrix-form algorithm
Horse	MSE	0.0053	0.0034	0.0029
	PSNR	32.4398	39.5792	41.2732
	SSIM	0.910	0.922	0.922
	TIME	312.0213	23.1472	18.2346
Leopard	MSE	0.0091	0.0076	0.0072
	PSNR	29.3275	35.2837	37.1742
	SSIM	0.892	0.903	0.905
	TIME	323.2731	31.2682	28.2712
Cat	MSE	0.0082	0.0056	0.0051
	PSNR	34.3826	37.3729	39.2174
	SSIM	0.903	0.913	0.915
	TIME	330.8675	36.2616	34.7293
Dog	MSE	0.0061	0.0042	0.0039
	PSNR	31.2738	34.8729	35.1728
	SSIM	0.908	0.912	0.914
	TIME	302.7439	30.2742	29.3782

Furthermore, several computed results of image restoration by both vector-form and matrix-form algorithm 4 are listed in Table 1. We evaluated their performance using key metrics: Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural SIMilarity (SSIM), and TIME. It's notable that the distributed matrix-form algorithm in 4 obtains better evaluation index in values of MSE, PSNR and SSIM than the vector-form algorithm in [41**?**], which means the proposed algorithm effectively reduces the noise in the degraded images and maintains the structural details. On top of that, the proposed matrix-form distributed algorithm can obviously save computation time as demonstrated in table1.

6. Conclusions

This paper introduces the matrix-variable optimization and its importance in solving linear matrix equations and blind image restoration. We present a distributed matrix-form algorithm for solving optimization problems with general equality, inequality and set constraints. By introducing the event triggering mechanism, the communication burden is successfully reduced and the global convergence of the algorithm is guaranteed. The algorithm has low computational complexity, is suitable for parallel implementation, and has shown excellent performance in experiments. Compared with the traditional centralized matrix-variable and vector-variable methods, the distributed event-triggered algorithm proposed in this study has significant advantages in computation speed and accuracy. This research is of great significance for the further application of matrix-variable optimization and the solution of practical problems.

Appendix

Proof. Consider the following Lyapunov function candidate:

$$\begin{aligned} V =& f(X) - f(X^*) - \sum_{i=1}^{N} \langle X_i - X_i^*, \eta_i^* \rangle_F + \frac{1}{2} \sum_{i=1}^{N} \langle X_i - X_i^* + \lambda_i - \lambda_i^* \\ &, a_{ij}(X_i - X_i^* + \lambda_i - \lambda_i^* - (X_j - X_j^* + \lambda_j - \lambda_j^*)) \rangle + \frac{1}{2} \sum_{i=1}^{N} \|\mu_i - \mu_i^* - (AX_i - b_i)\|_F^2 \\ &+ \frac{1}{2} \sum_{i=1}^{N} \|(\gamma_i + g^i(X_i))^+\|_F^2 - \frac{1}{2} \sum_{i=1}^{N} \|\gamma_i^*\|_F^2 - \sum_{i=1}^{*} \langle \gamma_i - \gamma_i^*, \gamma_i^* \rangle_F - \sum_{i=1}^{N} \langle X_i - X_i^*, \xi_i^* \gamma_i^* \rangle_F \\ &+ \frac{1}{2} \sum_{i=1}^{N} \|X_i - X_i^*\|_F^* + \frac{1}{2} \|\mu_i - \mu_i^*\|_F^2 + \frac{1}{2} \|\gamma_i - \gamma_i^*\|_F^2 \\ &+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_\rangle} \langle \gamma_i^* - \gamma_i^*, a_{ij}(\lambda_i - \lambda_i^* - \lambda_j + \lambda_j^*) \rangle_F \end{aligned} \tag{34}$$

where $\eta_i \in \partial f^i(X_i)$, $\eta_i^* \in \partial f^i(X_i^*)$, $\xi_i \in \partial g^i(X_i)$ and $\xi_i^* \in \partial g^i(X_i^*)$. By the convexity of f and g, it has

$$f(X) - f(X^*) - \sum_{i=1}^{N} \langle X_i - X_i^*, \eta_i^* \rangle_F \geq 0$$

and

$$\frac{1}{2} \sum_{i=1}^{N} \|X_i - X_i^*\|_F^* + \frac{1}{2} \|\mu_i - \mu_i^*\|_F^2 + \frac{1}{2} \|\gamma_i - \gamma_i^*\|_F^2 \geq 0.$$

Combining the above inequalities with the positivness of a_{ij}, it yields that $V \geq 0$. Then, differentiating $V(t)$ along algorithm (10) satisfies

$$\dot{V} = D_{X_i} + D_{\lambda_i} + D_{\mu_i} + D_{\gamma_i}$$

where

$$
\begin{aligned}
D_{X_i} &= \sum_{i=1}^{N} \langle \eta_i - \eta_i^* + \sum_{j\in N_i} a_{ij}\left[X_i - X_i^* + \lambda_i - \lambda_i^* - (X_j - X_j^* + \lambda_j - \lambda_j^*)\right] - A_i^{\mathrm{T}}(\mu_i - \mu_i^*) \\
&+ A_i^{\mathrm{T}}(A_iX_i - b_i) + (\gamma_i + g^i(X_i)^+)\xi_i - \xi_i^*\gamma_i^* + X_i - X_i^*, \dot{X}_i\rangle_F \\
&= \sum_{i=1}^{N} \langle \left[\eta_i + \sum_{j\in N_i} a_{ij}(X_i - X_j + \lambda_i - \lambda_j) - A_i^{\mathrm{T}}\mu_i + A_i^{\mathrm{T}}(A_iX_i - b_i) + (\gamma_i + g^i(X_i))^+\xi_i\right] \\
&- \left[\mu_i^* + \sum_{j\in N_i} a_{ij}(X_i^* - X_j^* + \lambda_i^* - \lambda_j^*) - A_i^{\mathrm{T}}\mu_i^* - \xi_i^*\gamma_i^*\right] + X_i - X_i^*, \dot{X}_i\rangle_F \\
&= \sum_{i=1}^{N} \langle \Delta_i - \Delta_i^* + X_i - X_i^*, \dot{X}_i\rangle_F
\end{aligned}
$$

$$
\begin{aligned}
D_{\lambda_i} &= \sum_{i=1}^{n} \langle \sum_{j\in N_i} a_{ij}\left[X_i - X_i^* + \lambda_i - \lambda_i^* - \left(X_j - X_j^* + \lambda_j - \lambda_j^*\right)\right] + \sum_{j\in N_i} a_{ij}\left(\lambda_i - \lambda_i^* - \lambda_j + \lambda_j^*\right), \dot{\lambda}_i\rangle_F \\
D_{\mu_i} &= \sum_{i=1}^{N} \langle 2(\mu_i - \mu_i^*) - (AX_i - b_i), \dot{\mu}_i\rangle_F \\
D_{\gamma_i} &= \sum_{i=1}^{N} \langle \left(\gamma_i + g^i(x_i)\right)^+ + \gamma_i - 2\gamma_i^*, \dot{\gamma}_i\rangle_F
\end{aligned}
$$

in which $\Delta_i = \eta_i + \sum_{j\in N_i} a_{ij}\left(X_i - X_j + \lambda_i - \lambda_j\right) - A_i^{\mathrm{T}}m_i + A_i^{\mathrm{T}}\left(A_iX_i - b_i\right) + \left(\gamma_i + \dot{g}^i\left(X_i\right)\right)^+\xi_i$, $\tilde{\Delta}_i = \partial f^i(X_i) + \sum_{j\in N_i} a_{ij}(\tilde{X}_i - \tilde{X}_j) + \sum_{j\in N_i} a_{ij}(\tilde{\lambda}_i - \tilde{\lambda}_j) - A_i^{\mathrm{T}}\mu_i + A_i^{T}(A_iX_i - b_i) + \xi_i(X_i)(\gamma_i + g_i)^+$ and $\Delta_i^* = \eta_i^* + \sum_{j\in N_i} a_{ij}(X_i^* - X_j^* + \lambda_i^* - \lambda_j^*) - A_i^{\mathrm{T}}\mu_i^* - \xi_i^*\gamma_i^* + X_i - X_i^*$. Thus, we have

$$
\Delta_i = \tilde{\Delta}_i + \sum_{j\in N_i} a_{ij}(e_i^x - e_j^x + e_i^\lambda - e_j^\lambda).
$$

Then, each variable is calculated one by one.

$$
\begin{aligned}
D_{X_i} =&\sum_{i=1}^{N}\langle \Delta_i-\Delta_i^*+X_i-X_i^*,2\left[-X_i+P_{\Omega_i}(X_i-\tilde{\Delta}_i)\right]\rangle_F\\
=&\sum_{i=1}^{N}\langle \tilde{\Delta}_i+\sum_{j\in N_i}a_{ij}(e_i^x-e_j^x+e_i^\lambda-e_j^\lambda)+X_i-X_i^*-\Delta_i^*,2\left[-X_i+P_{\Omega_i}(X_i-\tilde{\Delta}_i)\right]\rangle_F\\
=&\sum_{i=1}^{N}\langle \tilde{\Delta}_i+\sum_{j\in N_i}a_{ij}(e_i^x-e_j^x+e_i^\lambda-e_j^\lambda)+X_i-X_i^*-\Delta_i^*,2\left[-X_i^*+P_{\Omega_i}(X_i-\tilde{\Delta}_i)\right]\rangle_F\\
&+\sum_{i=1}^{N}\langle \tilde{\Delta}_i+\sum_{j\in N_i}a_{ij}(e_i^x-e_j^x+e_i^\lambda-e_j^\lambda)+X_i-X_i^*-\Delta_i^*,2(X_i^*-X_i)\rangle_F\\
=&\sum_{i=1}^{N}\langle P_{\Omega_i}(X_i-\tilde{\Delta}_i)-(X_i-\tilde{\Delta}_i),2\left[P_{\Omega_i}(X_i-\tilde{\Delta}_i-X_i^*\right]\rangle_F\\
&+\sum_{i=1}^{N}\langle X_i-P_{\Omega_i}(X_i-\tilde{\Delta}_i+X_i-X_i^*,2\left[P_{\Omega_i}(X_i-\tilde{\Delta}_i-X_i^*\right]\rangle_F-\sum_{i=1}^{N}\langle \Delta_i^*,2\left[P_{\Omega_i}(X_i-\tilde{\Delta}_i-X_i^*\right]\rangle_F\\
&+\langle\sum_{j\in N_i}a_{ij}(e_i^x-e_j^*+e_i^\lambda-e_j^\lambda),2\left[P_{\Omega_i}(X_i-\tilde{\Delta}_i-X_i^*\right]\rangle_F\\
&+\sum_{i=1}^{N}2\langle\tilde{\Delta}_i-\Delta_i^*+\sum_{j\in N_i}a_{ij}(e_i^x-e_j^*+e_i^\lambda-e_j^\lambda),X_i^*-X_i\rangle_F-2\sum_{i=1}^{N}\|X_i-X_i^*\|^2
\end{aligned}
$$

Then, we denote

$$
\begin{aligned}
\Psi_1 &= \sum_{i=1}^{N}\left\langle X_i-P_{\Omega_i}\left(X_i-\widetilde{\Delta}_i\right)+X_i-X_i^*,2\left[P_{\Omega_i}\left(X_i-\widetilde{\Delta}_i\right)-X_i^*\right]\right\rangle_F\\
\Psi_2 &= \sum_{i=1}^{N}\langle\widetilde{\Delta}_i-\Delta_i^*,X_i^*-X_i\rangle_F\\
\Psi_3 &= \sum_{i=1}^{N}2\langle\sum_{j\in N_i}a_{ij}(e_i^X-e_j^X+e_i^\lambda-e_j^\lambda),X_i^*-X_i\rangle_F\\
\Psi_4 &= \sum_{i=1}^{N}2\langle\sum_{j\in N_i}a_{ij}(e_i^X-e_j^X+e_i^\lambda-e_j^\lambda),P_{\Omega_i}(X_i-\widetilde{\Delta}_i-X_i^*\rangle_F
\end{aligned}
$$

We have

$$
\begin{aligned}
\Psi_1 =&\sum_{i=1}^{N}\langle X_i-P_{\Omega_i}\left(X_i-\widetilde{\Delta}_i\right)+X_i-X_i^*,2\left[P_{\Omega_i}\left(X_i-\widetilde{\Delta}_i\right)-X_i+X_i-X_i^*\right]\rangle_F\\
=&\sum_{i=1}^{N}\left[-2\|X_i-P_{\Omega_i}(X_i-\tilde{\Delta}_i)\|^2+\langle X_i-X_i^*,2P_{\Omega_i}(X_i-\tilde{\Delta}_i)-X_i\rangle+\langle X_i-X_i^*,2\left(X_i-P_{\Omega_i}\left(X_i-\tilde{\Delta}_i\right)\right)\rangle_F\right]\\
&+2\sum_{i=1}^{N}\|X_i-X_i^*\|^2
\end{aligned}
$$

and

$$\Psi_2 =\sum_{i=1}^{N}\langle\eta_i-\eta_i^*, X_i-X_i^*\rangle+\sum_{i=1}^{N}\langle\sum_{j\in N_i}a_{ij}(\tilde{X}_i-\tilde{X}_j+\tilde{\lambda}_i-\tilde{\lambda}_j)-\sum_{j\in N_i}a_{ij}(X_i^*-X_j^*+\lambda_i^*-\lambda_j^*), X_i-X_i^*\rangle$$
$$+\sum_{i=1}^{N}\langle A_i^{\mathrm{T}}(\mu_i-\mu_i^*), X_i-X_i^*\rangle-\sum_{i=1}^{N}\langle A_i^{\mathrm{T}}(A_iX_i-b_i), X_i-X_i^*\rangle-\sum_{i=1}^{N}\langle\xi_i(\gamma_i+g_i)^+-\xi_i^*\gamma_i^*, X_i-X_i^*\rangle_F$$

One of the projection property in (4b) implies

$$\left\langle P_{\Omega_i}\left(X_i-\widetilde{\Delta}_i\right)-(X_i-\tilde{\Delta}_i), 2\left[P_{\Omega_i}\left(X_i-\widetilde{\Delta}_i\right)-X_i^*\right]\right\rangle_F\leq 0$$
$$\left\langle \Delta_i^*, 2\left[P_{\Omega_i}(X_i-\widetilde{\Delta}_i)-X_i^*\right]\right\rangle_F\geq 0$$

Moreover, X^* satisfies that $A_iX^*=b_i$, so we have

$$D_{X_i} \leq\Psi_1+\Psi_2+\Psi_3+\Psi_4-2\sum_{i=1}^{N}\|X_i-X_i^*\|^2$$
$$\leq -2\sum_{i=1}^{N}\|x_i-P_{\Omega_i}(X_i-\widetilde{\Delta}_i)\|^2-2\sum_{i=1}^{N}\langle\sum_{j\in N_i}a_{ij}(\tilde{X}_i-\tilde{X}_j+\tilde{\lambda}_i-\tilde{\lambda}_j-(X_i^*-X_j^*+\lambda_i^*-\lambda_j^*)), X_i-X_i^*\rangle_F$$
$$+\sum_{i=1}^{N}2\langle\mu_i-\mu_i^*, A_i^{\mathrm{T}}X_i-b_i\rangle_F-2\sum_{i=1}^{N}\|A_iX_i-b_i\|_F^2-2\sum_{i=1}^{N}\langle\xi_i(\gamma_i+g^i(X_i))^+-\xi_i^*\gamma_i^*, X_i-X_i^*\rangle_F+\Psi_3+\Psi_4$$

Furthermore,

$$D_{\lambda_i} =\sum_{i=1}^{N}\langle\sum_{j\in N_i}a_{ij}\left[(X_i-X_j+\lambda_i-\lambda_j)-(X_i^*-X_j^*+\lambda_i^*-\lambda_j^*)\right]+\sum_{j\in N_i}a_{ij}\left[\lambda_i-\lambda_j-(\lambda_i^*-\lambda_j^*\right], X_i\rangle_F$$
$$=\sum_{i=1}^{N}\left\langle 2\sum_{j\in N_i}a_{ij}\left(\lambda_i-\lambda_j-\left(\lambda_i^*-\lambda_j^*\right)\right)+\sum_{j\in N_i}a_{ij}\left(X_i-X_j-\left(X_i^*-X_j^*\right)\right), X_i\right\rangle_F$$
$$=\sum_{i=1}^{N}2\langle\sum_{j\in N_i}a_{ij}\left[(\lambda_i-\lambda_j)-\left(\lambda_i^*-\lambda_j^*\right)\right], X_i-X_i^*\rangle_F+\sum_{i=1}^{N}\langle X_i-X_i^*, \sum_{j\in N_i}a_{ij}\left[X_i-X_j-\left(X_i^*-X_j^*\right)\right]\rangle_F$$

$$D_{\mu_i} =\sum_{i=1}^{N}\langle 2\left(\mu_i-\mu_i^*\right)-\left(Ax_i-b_i\right), -A_iX_i+b_i\rangle F$$
$$=-2\sum_{i=1}^{N}\langle\mu_i-\mu_i^*, A_iX_i-b_i\rangle_F+\sum_{i=1}^{N}\left\|A_iX_i-b_i\right\|_F^2$$

$$D_{\gamma_i} =\sum_{i=1}^{N}\langle(\gamma_i+g^i(X_i))^+-2\gamma_i^*+\gamma_i, -\gamma_i+(\gamma_i+g^i(X_i))^+\rangle_F$$
$$=\sum_{i=1}^{N}\langle 2(\gamma_i+g^i(X_i))^+-2\gamma_i^*+\gamma_i-(\gamma_i+g^i(X_i))^+, -\gamma_i+(\gamma_i+g^i(X_i))^+\rangle_F$$
$$=-\sum_{i=1}^{N}\|\gamma_i-(\gamma_i+g^i(X_i))^+\|^2+2\sum_{i=1}^{N}\langle(\gamma_i+g^i(X_i))^+, (\gamma_i+g^i(X_i))^+-\gamma_i\rangle-2\sum_{i=1}^{N}\langle\gamma_i^*, (\gamma_i+g^i(X_i))^+-\gamma_i\rangle$$

Thus

$$\begin{aligned}
\dot{V} \leq & \sum_{i=1}^{N}(-2)\|x_i - P_{\Omega_i}(X_i - \widetilde{\Delta}_i)\|^2 - 2\sum_{i=1}^{N}\langle \sum_{j\in N_i} a_{ij}(\tilde{X}_i - \tilde{X}_j + \tilde{\lambda}_i - \tilde{\lambda}_j - (X_i^* - X_j^* + \lambda_i^* - \lambda_j^*)) \\
& , X_i - X_i^*\rangle_F + \sum_{i=1}^{N} 2\langle \sum_{j\in N_i} a_{ij}\left[(\lambda_i - \lambda_j) - \left(\lambda_i^* - \lambda_j^*\right)\right], X_i - X_i^*\rangle_F \\
& - \sum_{i=1}^{N}\|A_i X_i - b_i\|_F^2 - 2\sum_{i=1}^{N}\langle \xi_i(\gamma_i + g^i(X_i))^+ - \xi_i^*\gamma_i^*, X_i - X_i^*\rangle_F \\
& + 2\sum_{i=1}^{N}\langle (\gamma_i + g^i(X_i))^+, (\gamma_i + g^i(X_i))^+ - \gamma_i\rangle - 2\sum_{i=1}^{N}\langle \gamma_i^*, (\gamma_i + g^i(X_i))^+ - \gamma_i\rangle \\
& - \sum_{i=1}^{N}\|\gamma_i - (\gamma_i + g^i(X_i))^+\|^2 + \sum_{i=1}^{N}\langle X_i - X_i^*, \sum_{j\in N_i} a_{ij}\left[X_i - X_j - \left(X_i^* - X_j^*\right)\right]\rangle_F \\
& + \Psi_3 + \Psi_4 + 2\sum_{i=1}^{N}\langle \sum_{j\in N_i} a_{ij}(e_i^x - e_j^x + e_i^\lambda - e_j^\lambda), -X_i + P_{\Omega_i}(X_i - \tilde{\Delta}_i)\rangle_F \\
\leq & -2\sum_{i=1}^{N}\|x_i - P_{\Omega_i}(X_i - \widetilde{\Delta}_i)\|^2 - \sum_{i=1}^{N}\|A_i X_i - b_i\|_F^2 - \sum_{i=1}^{N}\|\gamma_i - (\gamma_i + g^i(X_i))^+\|^2 \\
& - \sum_{i=1}^{N}\langle X_i - X_i^*, \sum_{j\in N_i} a_{ij}\left[X_i - X_j - \left(X_i^* - X_j^*\right)\right]\rangle_F \\
& - 2\sum_{i=1}^{N}\langle \sum_{j\in N_i} a_{ij}(e_i^X - e_j^X + e_i^\lambda - e_j^\lambda), X_i - X_i^*\rangle_F \\
& + 2\sum_{i=1}^{N}\langle \sum_{j\in N_i} a_{ij}(e_i^x - e_j^x + e_i^\lambda - e_j^\lambda), -X_i + P_{\Omega_i}(X_i - \tilde{\Delta}_i)\rangle_F
\end{aligned}$$

when $c_1 > \lambda_{max}(L)$

$$\begin{aligned}
& - 2\sum_{i=1}^{N}\langle \sum_{j\in N_i} a_{ij}(e_i^X - e_j^X + e_i^\lambda - e_j^\lambda), X_i - X_i^*\rangle_F \\
& = -2\sum_{i=1}^{N}\langle e_i^X + e_i^\lambda, \sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\rangle_F \\
& \leq 2\sum_{i=1}^{N}(c_1\|e_i^X + e_i^\lambda\|^2 + \frac{1}{4c_1}\|\sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\|_F^2) \\
& \leq 2\sum_{i=1}^{N}(c_1\|e_i^X + e_i^\lambda\|^2 + \frac{\lambda_{max}(L)}{4c_1}\|\sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\|_F^2) \\
& = 2\sum_{i=1}^{N}(c_1\|e_i^X + e_i^\lambda\|^2 + \frac{\lambda_{max}(L)}{2c_1}\langle X_i - X_i^*, \sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\rangle_F)
\end{aligned}$$

when $c_1 > \lambda_{max}(L)$

$$
\begin{aligned}
&-2\sum_{i=1}^{N}\langle\sum_{j\in N_i} a_{ij}(e_i^X - e_j^X + e_i^\lambda - e_j^\lambda), X_i - X_i^*\rangle_F \\
=&-2\sum_{i=1}^{N}\langle e_i^X + e_i^\lambda, \sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\rangle_F \\
\leq&2\sum_{i=1}^{N}(c_1\|e_i^X + e_i^\lambda\|^2 + \frac{1}{4c_1}\|\sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\|_F^2) \\
\leq&2\sum_{i=1}^{N}(c_1\|e_i^X + e_i^\lambda\|^2 + \frac{\lambda_{max}(L)}{4c_1}\|\sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\|_F^2) \\
=&2\sum_{i=1}^{N}(c_1\|e_i^X + e_i^\lambda\|^2 + \frac{\lambda_{max}(L)}{2c_1}\langle X_i - X_i^*, \sum_{j\in N_i} a_{ij}(X_i - X_j - X_i^* + X_j^*)\rangle_F)
\end{aligned}
$$

and

$$
\begin{aligned}
&2\sum_{i=1}^{N}\langle\sum_{j\in N_i} a_{ij}(e_i^x - e_j^x + e_i^\lambda - e_j^\lambda), -X_i + P_{\Omega_i}(X_i - \tilde{\Delta}_i)\rangle_F \\
\leq&2\sum_{i=1}^{N}\langle\sum_{i=1}^{N} a_{ij}^2\|e_i^X - e_j^X + e_i^\lambda - e_j^\lambda\|_F^2 + \frac{1}{4}\| - X_i + P_{\Omega_i}(X_i - \tilde{\Delta}_i)\|_F^2) \\
\leq&2\sum_{i=1}^{N}(4\sum_{j\in N_i} a_{ij}^2\|e_i^X + e_i^\lambda\|_F^2) + \frac{1}{2}\|X_i - P_{\Omega_i}(X_i - \tilde{\Delta}_i\|_F^2
\end{aligned}
$$

By considering the triggering condition (9), for $\forall i$, there is

$$
\left(\alpha + \sum_{jak} a_{ij}^2\right)\left\|e_i^X + e_i^\lambda\right\|_F^2 \leq \mu e^{-\tau t}
$$

Thus,

$$
\begin{aligned}
\dot{V} \leq& -\frac{3}{2}\sum_{i=1}^{N}\|X_i - P_{\Omega_i}(X_i - \tilde{\Delta}_i)\|^2 - \sum_{i=1}^{N}\|A_iX_i - b_i\|_F^2 - \sum_{i=1}^{N}\|\gamma_i - (\gamma_i + g^i)^+\|_F^2 \\
&-(1 - \frac{\lambda_{max}(L)}{2c_1})\sum_{i=1}^{N}\left\langle x_i - x_i^*, \sum_{j=1v_i} a_{ij}\left(x_i - x_j - x_i^* + x_j^*\right)\right\rangle_F \\
&+2c_1\sum_{i=1}^{N}\left\|e_i^X + e_i^\lambda\right\|_F^2 + 8\sum_{i=1}^{N}\sum_{j\in N_i} a_{ij}^2\left\|e_i^X + e_i^\lambda\right\|_F^2 \\
=& -\frac{3}{2}\sum_{i=1}^{N}\|X_i - P_{\Omega_i}(X_i - \tilde{\Delta}_i)\|^2 - \sum_{i=1}^{N}\|A_iX_i - b_i\|_F^2 - \sum_{i=1}^{N}\|\gamma_i - (\gamma_i + g^i)^+\|_F^2 \\
&-(1 - \frac{\lambda_{max}(L)}{2c_1})\sum_{i=1}^{N}\left\langle x_i - x_i^*, \sum_{j=1v_i} a_{ij}\left(x_i - x_j - x_i^* + x_j^*\right)\right\rangle_F \\
&+8\sum_{i=1}^{N}\left[\left(\frac{c_1}{4}\right)\left\|e_i^X + e_i^\lambda\right\|_F^2 + \sum_{j\in V_i} a_{ij}^2\left\|e_i^X + e_i^\lambda\right\|_F\right]^2
\end{aligned}
$$

Let $\alpha = \frac{c_1}{4}$, there has

$$8\sum_{i=1}^{n}\left(\alpha+\sum_{j\in N_i}a_{ij}^2\right)\left\|e_i^X+e_i^\lambda\right\|_F^2 \leqslant 8N\omega e^{-\varsigma t}$$

Define $\tilde{V} = \frac{8N\omega}{\tau}e^{-\varsigma t}$ and $V' = V + \tilde{V}$

$$\begin{aligned}\dot{V}' \leq & -\frac{3}{2}\sum_{i=1}^{N}\|X_i - P_{\Omega_i}(X_i-\Delta_i)\|^2 - \sum_{i=1}^{N}\|A_iX_i-b_i\|_F^2 - \sum_{i=1}^{N}\|\gamma_i-(\gamma_i+g^i)^+\|_F^2 \\ & -(1-\frac{\lambda_{\max}L}{2c_1})\sum_{i=1}^{N}\left\langle X_i-X_i^*, \sum_{j\in N_i}a_{ij}\left(X_i-X_j-X_i^*+X_j^*\right)\right\rangle_F\end{aligned} \tag{35}$$

Define

$$\begin{aligned}H(t) =& \frac{3}{2}\sum_{i=1}^{N}\|X_i - P_{\Omega_i}(X_i-\Delta_i)\|^2 + \sum_{i=1}^{N}\|A_iX_i-b_i\|_F^2 + \sum_{i=1}^{N}\|\gamma_i-(\gamma_i+g^i)^+\|_F^2 \\ & +(1-\frac{\lambda_{\max}L}{2c_1})\sum_{i=1}^{N}\left\langle X_i-X_i^*, \sum_{j\in N_i}a_{ij}\left(X_i-X_j-X_i^*+X_j^*\right)\right\rangle_F\end{aligned}$$

According to (35), we have

$$\int_0^{+\infty}H(t)dt \leq V'(0) - \lim_{t\to\infty}V'(t).$$

Among them, $V'(0)$ is finite and $\lim_{t\to\infty}V'(t)$ exists. Recall that $\lim_{t\to+\infty}(2Nk\omega/\varsigma)e^{-\varsigma t} = 0$. Thus, $\lim_{t\to+\infty}V(t)$ exists.

Next, we certify that

$$\liminf_{t\to+\infty}H(t) = 0. \tag{36}$$

We explain it by contraction. If (36) does not hold, there exist an increasing sequence $\{t_k\}$ and a constant $\varepsilon > 0$ such that

$$\liminf_{t\to+\infty}H(t_k) = \varepsilon > 0.$$

That is, there is $T > 0$ satisfying that $H(t_k) \geq \varepsilon/2, \forall t \in [T, +\infty)$. According to $\dot{V}'(t) \leq -H(t) \leq 0$, it can be arrived at $(d/dt)\tilde{V}(t) \leq -\varepsilon/2$, for a.e. $t \in [T, +\infty)$. Integrating above inequality from T to t, one has

$$V'(t) \leq V'(T) - (\varepsilon/2)\left(t-T\right).$$

Thus, we have $\lim_{t\to+\infty}V'(t) = -\infty$. Obviously, it leads a contradiction with the boundedness of $V'(t)$. Therefore, the equality (36) holds.

Thus, there exists an increasing sequence $\{\tilde{t}_k\}$, such that $\|X_i(\tilde{t}_k) - P_{\Omega_i}(X_i(\tilde{t}_k)-\Delta_i(\tilde{t}_k))\|_F^2 = 0$, $\sum_{i=1}^{N}\|A_iX_i(\tilde{t}_k)-b_i\|_F^2 = 0$, $\sum_{i=1}^{N}\|\gamma_i(\tilde{t}_k)-(\gamma_i(\tilde{t}_k)+g^i(X_i(\tilde{t}_k)))^+\|_F^2 = 0$ and $\sum_{i=1}^{N}\langle X_i(\tilde{t}_k)-X_i^*, \sum_{j\in N_i}a_{ij}\left(X_i(\tilde{t}_k)-X_j(\tilde{t}_k)-X_i^*+X_j^*\right)\rangle_F = 0$ for $i = 1, 2, \ldots, N$. By above analysis, $\lim_{t\to+\infty}V(t)$ exists. It can be obtained that By Theorem 4.1, we can obtain that $X_i(t)$ produced by distributed event-triggered algorithm (10) converges to the optimal solution X_i^* of matrix-variable optimization problem (7).

□

References

References

[1] O. Shorinwa, M. Schwager, Distributed model predictive control via separable optimization in multi-agent networks, IEEE Transactions on Automatic Control 69 (1) (2024) 230–245.

[2] E. Espina, R. J. Cárdenas-Dobson, J. W. Simpson-Porco, M. Kazerani, D. Sáez, A consensus-based distributed secondary control optimization strategy for hybrid microgrids, IEEE Transactions on Smart Grid 14 (6) (2023) 4242–4255. doi:10.1109/TSG.2023.3263107.

[3] B. Jin, J. Song, Y. Liao, Comparative study on the improvement of a distributed energy system performance by optimized operation strategy from design to operation, Energy Reports 10 (2023) 1726–1738.

[4] J. Xu, X. He, X. Han, H. Wen, A two-layer distributed algorithm using neurodynamic system for solving L1-Minimization, IEEE Transactions on Circuits and Systems II: Express Briefs 69 (8) (2022) 3490–3494.

[5] S. Sun, J. Xu, W. Ren, Distributed continuous-time algorithms for time-varying constrained convex optimization, IEEE Transactions on Automatic Control 68 (7) (2023) 3931–3946.

[6] T. Colombo, S. Sagratella, Distributed algorithms for convex problems with linear coupling constraints, JOURNAL OF GLOBAL OPTIMIZATION 77 (1, SI) (2020) 53–73.

[7] Z. Deng, X. Wang, Y. Hong, Distributed optimisation design with triggers for disturbed continuous-time multi-agent systems, IET Control Theory & Applications 11 (2) (2017) 282–290.

[8] L. Huang, Y. Xia, L. Huang, S. Zhang, Two matrix-type projection neural networks for matrix-valued optimization with application to image restoration, NEURAL PROCESSING LETTERS 53 (3, SI) (2021) 1685–1707.

[9] K. Fletcher, Regularizing matrix factorization with implicit user preference embeddings for Web API Recommendation, in: 2019 IEEE International Conference on Services Computing (SCC), 2019, pp. 1–8. doi:10.1109/SCC.2019.00014.

[10] R. Ji, A. Heidarzadeh, K. R. Narayanan, Sparse random Khatri-Rao product codes for distributed matrix multiplication, in: 2022 IEEE Information Theory Workshop (ITW), 2022, pp. 416–421. doi:10.1109/ITW54588.2022.9965842.

[11] G. Albi, S. Bicego, D. Kalise, Gradient-augmented supervised learning of optimal feedback laws using state-dependent riccati equations, IEEE Control Systems Letters 6 (2022) 836–841. doi:10.1109/LCSYS.2021.3086697.

[12] A. Barrau, G. Rance, Y. Bouzidi, A. Quadrat, A. Quadrat, Using symbolic computation to solve algebraic Riccati equations arising in invariant filtering, in: 2018 European Control Conference (ECC), 2018, pp. 2284–2289. doi:10.23919/ECC.2018.8550411.

[13] S. Mou, J. Liu, A. S. Morse, A distributed algorithm for solving a linear algebraic equation, in: 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2013, pp. 267–274. doi:10.1109/Allerton.2013.6736534.

[14] G. Chen, X. Zeng, Y. Hong, Distributed optimisation design for solving the Stein equation with constraints, IET Control Theory & Applications 13 (15) (2019) 2492–2499.

[15] S. Cheng, S. Liang, Y. Fan, Distributed solving Sylvester equations with fractional order dynamics, CONTROL THEORY AND TECHNOLOGY 19 (2) (2021) 249–259. doi:10.1007/s11768-021-00044-0.

[16] X. Zeng, S. Liang, Y. Hong, J. Chen, Distributed computation of linear matrix equations: An optimization perspective, IEEE Transactions on Automatic Control 64 (5) (2019) 1858–1873. doi:10.1109/TAC.2018.2847603.

[17] Y. Huang, Z. Meng, J. Sun, Scalable distributed least square algorithms for large-scale linear equations via an optimization approach, Automatica 146 (2022) 110572.

[18] H. Fan, Y. Li, H. Zhang, X. Zhu, Preconditioners based on matrix splitting for the structured systems from elliptic pde-constrained optimization problems, Applied Mathematics and Computation 463 (2024) 128341.

[19] Numerical simulation and structural optimization of transpiration cooling with gradient porous matrix in supersonic condition, International Journal of Thermal Sciences 198 (2024) 108871.

[20] M. Sørensen, N. D. Sidiropoulos, A. Swami, Overlapping community detection via semi-binary matrix factorization: Identifiability and algorithms, IEEE Transactions on Signal Processing 70 (2022) 4321–4336.

[21] Y. Xia, T. Ye, L. Huang, Analysis and application of matrix-form neural networks for fast matrix-variable convex optimization, IEEE Transactions on Neural Networks and Learning Systems (2023) 1–15doi:10.1109/TNNLS.2023.3340730.

[22] Z. Feng, G. Hu, C. G. Cassandras, Finite-time distributed convex optimization for continuous-time multiagent systems with disturbance rejection, IEEE Transactions on Control of Network Systems 7 (2) (2020) 686–698.

[23] W. Yu, Z. Deng, H. Zhou, X. Zeng, Distributed event-triggered algorithm for optimal resource allocation of multi-agent systems, Kybernetica 53 (5) (2017) 747–764.

[24] W. Chen, W. Ren, Event-triggered zero-gradient-sum distributed consensus optimization over directed networks, AUTOMATICA 65 (2016) 90–97.

[25] A. Wang, X. Liao, T. Dong, Event-triggered gradient-based distributed optimisation for multi-agent systems with state consensus constraint, IET Control Theory & Applications (2018).
URL `https://api.semanticscholar.org/CorpusID:125098698`

[26] S. Mao, Z. Dong, W. Du, Y.-C. Tian, C. Liang, Y. Tang, Distributed nonconvex event-triggered optimization over time-varying directed networks, IEEE Transactions on Industrial Informatics 18 (7) (2022) 4737–4748. doi:10.1109/TII.2021.3103747.

[27] J. Liu, W. Chen, H. Dai, Sampled-data based distributed convex optimization with event-triggered communication, INTERNATIONAL JOURNAL OF CONTROL AUTOMATION AND SYSTEMS 14 (6) (2016) 1421–1429.

[28] W. Liu, Y. Li, J. Sun, G. Wang, J. Chen, Data-driven self-triggering mechanism for state feedback control, IEEE Control Systems Letters 7 (2023) 1975–1980.

[29] X. Zeng, J. Chen, Y. Hong, Distributed optimization design of iterative refinement technique for algebraic riccati equations, IEEE Transactions on Systems, Man, and Cybernetics: Systems 52 (5) (2022) 2833–2847. doi:10.1109/TSMC.2021.3056871.

[30] X. Zeng, J. Chen, J. Sun, Y. Hong, Distributed optimization approach for solving continuous-time lyapunov equations with exponential rate of convergence, IEEE Transactions on Systems, Man, and Cybernetics: Systems 52 (3) (2022) 1684–1691.

[31] S. Arif, V. J. Moulin, Extracellular vesicles on the move: Traversing the complex matrix of tissues, European Journal of Cell Biology 102 (4) (2023) 151372.

[32] X. Ma, C. Ma, Y. Jiang, C. Ge, Improved privacy-preserving pca using optimized homomorphic matrix multiplication, Computers & Security 138 (2024) 103658.

[33] X. J. Li, G. S. Deng, X. Z. Wang, X. L. Wu, Q. W. Zeng, A hybrid recommendation algorithm based on user comment sentiment and matrix decomposition, Information Systems 117 (2023) 102244.

[34] B. Gharesifard, J. Cortés, Distributed continuous-time convex optimization on weight-balanced digraphs, IEEE Transactions on Automatic Control 59 (3) (2014) 781–786.

[35] Y. Zhu, W. Yu, G. Wen, G. Chen, W. Ren, Continuous-time distributed subgradient algorithm for convex optimization with general constraints, IEEE Transactions on Automatic Control 64 (4) (2019) 1694–1701. doi:10.1109/TAC.2018.2852602.

[36] X. Chen, X. Hu, T. Yue, Semi-blindly enhancing extremely noisy videos with recurrent Spatio-Temporal large-span network, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (7) (2023) 8984–9003. doi:10.1109/TPAMI.2023.3234026.

[37] D. Xie, W. Dong, J. Zheng, C. Tian, Spatially-variant image deconvolution for photoacoustic tomography, Opt. Express 31 (13) (2023) 21641–21657.

[38] K. Li, L. Wu, Q. Qi, W. Liu, X. Gao, L. Zhou, D. Song, Beyond single reference for training: Underwater image enhancement via comparative learning, IEEE Transactions on Circuits and Systems for Video Technology 33 (6) (2023) 2561–2576. doi:10.1109/TCSVT.2022.3225376.

[39] F. Qin, Y. Xie, L. Jiang, X. Wen, C. Li, J. Zeng, L. Cao, Blind restoration of vertical motion blurred image based on point spread function estimation, in: L. Xiao, D. Xu (Eds.), Thirteenth International Conference on Graphics and Image Processing (ICGIP 2021), Vol. 12083, International Society for Optics and Photonics, SPIE, 2022, p. 1208314.

[40] Liu, W. Tian, Y. Luo, J. Zou, S. Tang, A Windowed-Total-Variation regularization constraint model for blind image restoration, Journal of Information Processing Systems 18 (1) (2022) 48–58.

[41] F. Xue, T. Blu, A novel sure-based criterion for parametric PSF estimation, IEEE Transactions on Image Processing 24 (2) (2015) 595–607.

[42] L. Huang, Y. Xia, T. Ye, Effective blind image deblurring using matrix-variable optimization, IEEE Transactions on Image Processing 30 (2021) 4653–4666. doi:10.1109/TIP.2021.3073856.

[43] L. Huang, Y. Xia, Fast blind image super resolution using matrix-variable optimization, IEEE Transactions on Circuits and Systems for Video Technology 31 (3) (2021) 945–955. doi:10.1109/TCSVT.2020.2996592.

[44] L. Cheng, Z.-G. Hou, Y. Lin, M. Tan, W. C. Zhang, F.-X. Wu, Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks, IEEE Transactions on Neural Networks 22 (5) (2011) 714–726.