Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Matrix-valued distributed stochastic optimization with constraints*

Zicong XIA[1,2], Yang LIU‡[1,2], Wenlian LU[3], Weihua GUI[4]

[1]*Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua 321004, China*

[2]*School of Mathematical Sciences, Zhejiang Normal University, Jinhua 321004, China*

[3]*School of Mathematical Sciences, Fudan University, Shanghai 200433, China*

[4]*School of Automation, Central South University, Changsha 410083, China*

E-mail: 201531700128@zjnu.edu.cn; liuyang@zjnu.edu.cn; wenlian@fudan.edu.cn; gwh@csu.edu.cn

Received Sept. 7, 2022; Revision accepted Oct. 28, 2022; Crosschecked Aug. 5, 2023; Published online Aug. 12, 2023

Abstract: In this paper, we address matrix-valued distributed stochastic optimization with inequality and equality constraints, where the objective function is a sum of multiple matrix-valued functions with stochastic variables and the considered problems are solved in a distributed manner. A penalty method is derived to deal with the constraints, and a selection principle is proposed for choosing feasible penalty functions and penalty gains. A distributed optimization algorithm based on the gossip model is developed for solving the stochastic optimization problem, and its convergence to the optimal solution is analyzed rigorously. Two numerical examples are given to demonstrate the viability of the main results.

Key words: Distributed optimization; Matrix-valued optimization; Stochastic optimization; Penalty method; Gossip model

https://doi.org/10.1631/FITEE.2200381 **CLC number:** O224

1 Introduction

1.1 Distributed optimization

In recent years, distributed optimization has received great attention thanks to its important role in describing a number of collective assignments. As an effective parallel computing method, distributed optimization can tackle large-scale optimization problems which can be decomposed into several subproblems that can be solved in parallel. Its applications and theoretical significance relate to various fields, including sensor networks (Wan and Lemmon, 2009), machine learning (Li H et al., 2020), resource allocation (Deng et al., 2018), and so on (Yang SF et al., 2017; Shi et al., 2019; Yang T et al., 2019; Jiang et al., 2021; Wang D et al., 2021; Wang XY et al., 2021; Yue et al., 2022; Zeng et al., 2022).

Distributed optimization can be regarded as an optimization approach with a multi-agent system. Each agent has its local objective function, and a local decision variable denotes the state of an agent. The objective function of the considered optimization problem is the sum of multiple local objective functions. A distinguishing feature of distributed optimization is that the information exchange among agents is through a network topology graph. In the

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 62173308), the Natural Science Foundation of Zhejiang Province, China (Nos. LR20F030001 and LD19A010001), and the Jinhua Science and Technology Project, China (No. 2022-1-042)

ORCID: Zicong XIA, https://orcid.org/0000-0001-9943-5087; Yang LIU, https://orcid.org/0000-0003-3761-0104; Wenlian LU, https://orcid.org/0000-0003-1880-6240; Weihua GUI, https://orcid.org/0000-0002-5337-6445

graph, a node denotes an agent. Each agent knows only its own information (its local objective function and local decision variable). Compared with centralized optimization, distributed optimization has several desirable features as follows: (1) each agent needs to interact with only neighbor agents, so it can save the communication cost; (2) the information related to the optimization problem is distributed and stored in each agent, so it is more private and safer; (3) there is no single point of failure, which greatly improves the fault tolerance of the system; (4) because it is a parallel computing method, the scalability of the optimization algorithm can be enhanced.

A large number of distributed optimization methods have been developed in recent years. For example, distributed subgradient methods for multi-agent optimization were developed in Nedic and Ozdaglar (2009). In Liu and Wang (2015), a second-order multi-agent system was proposed for distributed optimization with bound constraints. In Zeng et al. (2017), a distributed continuous-time optimization method was presented via non-smooth analysis. In Liu et al. (2017), a recurrent neural network (RNN) system was developed for distributed optimization. Based on an RNN system, decentralized-partial-consensus constrained optimization was addressed in Xia ZC et al. (2023). In Xia ZC et al. (2021, 2022), multi-complex-variable distributed optimization was studied.

Note that the proposed systems in existing works are vector-variable systems. Hence, the computation time relies on the dimension of the state in the optimization problem. However, if the dimension is high, the methods converge slowly. A matrix-valued optimization model can overcome this difficulty in several areas (Bouhamidi and Jbilou, 2012; Bin and Xia, 2014; Xia YS et al., 2014; Li JF et al., 2016; Huang et al., 2021). However, the results of matrix-valued optimization methods are not systematic enough, although several seminal works have been done (Huang et al., 2021; Xing et al., 2021; Zhu ZH et al., 2021; Zhang et al., 2022).

1.2 Constraints and penalty methods

Various types of constraints are studied in distributed optimization. For the resource allocation problem, the choice of each agent is in a certain range, and no agent shares its private information with others (Zeng et al., 2017). In this case, bound constraints and linear equality constraints are needed. Many engineering tasks have complex constraints due to time limitations and technical restrictions. In addition, the limitations of communication capacities cause constraints in social networks. To this end, the handling of constraints has been investigated in many works, including inequality constraints (Zhu MH and Martínez, 2012; Liang et al., 2018a; Li XX et al., 2020), equality constraints (Zhu MH and Martínez, 2012; Liu and Wang, 2013; Liang et al., 2018b; Lv et al., 2020), bound constraints (Zeng et al., 2017; Zhou et al., 2019), and approximate constraints (Jiang et al., 2021).

The exact penalty method is a valid method for dealing with the constraints of the optimization problem. Its core is to choose feasible penalty functions and penalty gains to transform a constrained optimization problem into an equivalent unconstrained one, in which the penalty gains are not so large in contrast to the ones in the conventional penalty methods. There are several related works about the exact penalty methods for distributed optimization. An adaptive exact penalty method was proposed in Zhou et al. (2019) for distributed optimization. A distributed optimization algorithm was developed in Liang et al. (2018a) using an exact penalty function. However, Zhou et al. (2017, 2019) considered the bound constraint and used the distance function as the penalty function. In this paper, we develop an exact penalty method for handling inequality and equality constraints.

1.3 Gossip model

Originating from the social communication network, the gossip model plays an important role in consensus algorithms, and is applied in sensor networks and peer-to-peer networks thanks to its advantages including high fault-tolerance and high scalability (Boyd et al., 2006). In studies of distributed optimization, the gossip model has been applied widely to achieve consensus on the states of agents. In recent years, a large number of works on gossip-like optimization algorithms have been done. For example, a gossip algorithm was designed for the convex consensus optimization in Lu et al. (2011). In Jakovetic et al. (2011), a gossip algorithm was developed to solve cooperative convex optimization in networked systems. In Yuan (2014), a gossip-based gradient-free method was developed, and the gossip

model was regarded as a multi-agent system. In Koloskova et al. (2019), a distributed stochastic optimization algorithm was proposed based on a gossip model with compressed communication.

1.4 Goal and contributions

In this paper, we consider a distributed stochastic optimization problem with N agents as follows:

$$\begin{aligned} \min\ & \sum_{i=1}^{N} f_i(\boldsymbol{X}) = \sum_{i=1}^{N} \mathbb{E}_{\xi_i \in \mathcal{D}_i} F_i(\boldsymbol{X}, \xi_i) \\ \text{s.t. } & g(\boldsymbol{X}) \leq 0,\ h(\boldsymbol{X}) = 0, \end{aligned} \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, $f_i : \mathbb{R}^{n\times m} \to \mathbb{R}$, $g : \mathbb{R}^{n\times m} \to \mathbb{R}$, and $h : \mathbb{R}^{n\times m} \to \mathbb{R}$. $F_i(\boldsymbol{X}, \xi_i)$ is a stochastic function, and $\mathbb{E}_{\xi_i \in \mathcal{D}_i} F_i(\boldsymbol{X}, \xi_i)$ is an experience expectation with ξ_i denoting a sample in a local dataset $\mathcal{D}_i$ of agent i.

Problem (1) is said to be a matrix-valued distributed stochastic optimization problem. For vector-valued stochastic optimization, several works have been done. The problem of distributed stochastic optimization was addressed by Shamir and Srebro (2014). The strongly convex stochastic optimization problem was studied in Rakhlin et al. (2012). Several gossip algorithms with compressed communication were derived for decentralized stochastic optimization in Koloskova et al. (2019).

In this paper, we address the matrix-valued distributed stochastic optimization with inequality and equality constraints using an algorithm based on a gossip model. Specifically, the contributions are summarized as follows:

1. An auxiliary function is proposed to analyze several properties of the matrix-valued functions. Many common properties for vector-valued optimization methods are proposed in a matrix-valued fashion (see in Definitions 1–5 and Lemma 1).

2. A selection principle of the penalty functions and the penalty gains is derived (see in Selection principle 1). Based on the selection principle, an exact penalty method is proposed for transforming a matrix-valued optimization problem with inequality and equality constraints into a problem without inequality or equality constraints (see in Theorems 1 and 2, and Fig. 1).

3. A distributed optimization algorithm based on a gossip model is developed for solving the matrix-valued distributed stochastic optimization problem (see in Algorithm 1), and its convergence is analyzed (see in Theorem 3 and Remark 1). Two numerical examples are provided to illustrate the efficiency of Algorithm 1 for solving matrix-valued distributed stochastic optimization problems (see in Figs. 2 and 3).

2 Preliminaries

2.1 Notations

Let $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ denote the set of all real numbers, the set of all n-dimensional real vectors, and the set of all $(n\times m)$-dimensional real matrices, respectively. $\mathcal{I}_N = \{1, 2, ..., N\}$. $\|\cdot\|$ denotes the Euclidean norm, $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, and $|\cdot|$ denotes the absolute value. $\forall \boldsymbol{X} \in \mathbb{R}^{n\times m}$, $X(i,j)$ denotes the $(i,j)^{\mathrm{th}}$ element of $\boldsymbol{X}$. $\mathrm{vec}(\boldsymbol{X}) = (X(1,1), X(2,1), \ldots, X(n,1), X(1,2), X(2,2), \ldots, X(n,m))^{\mathrm{T}} \in \mathbb{R}^{nm}$. For vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$, $\mathrm{col}(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) = (\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}, ..., \boldsymbol{x}_N^{\mathrm{T}})^{\mathrm{T}}$. $\boldsymbol{A}^{\mathrm{T}}$ denotes the transpose of matrix $\boldsymbol{A}$. $\mathrm{tr}(\boldsymbol{A})$ denotes the trace of the n^{th}-order matrix $\boldsymbol{A}$. $\delta_2(\boldsymbol{A})$ denotes the second largest eigenvalue of the n^{th}-order matrix $\boldsymbol{A}$. $\boldsymbol{I}_n$ denotes the n-dimensional identity matrix, and $\boldsymbol{1}_n$ denotes the n-dimensional vector with all components being 1. "$\otimes$" denotes the Kronecker product operator. $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ denotes a graph with N nodes, where $\mathcal{V} = \mathcal{I}_N$ is the node set, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set, and $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix. $A(i,j) > 0$ if $(i,j) \in \mathcal{E}$; otherwise, $A(i,j) = 0$. Let $\mathcal{N}_i = \{j | A(i,j) \neq 0\}$ be the set of the neighbors of node i. For a set $S \subset \mathbb{R}^{n\times m}$, $P_S(\boldsymbol{X}) = \arg\min_{\boldsymbol{Y}\in S} \|\boldsymbol{X} - \boldsymbol{Y}\|_{\mathrm{F}}$.

2.2 Matrix-valued function

In problem (1), local objective function f_i is a mapping from $\mathbb{R}^{n\times m}$ to $\mathbb{R}$. In addition, the functions g and h in the constraints are the mappings from $\mathbb{R}^{n\times m}$ to $\mathbb{R}$. In this study, we call the function $f : \mathbb{R}^{n\times m} \to \mathbb{R}$ a matrix-valued function. An optimization problem is said to be a matrix-valued optimization problem if its objective function is a matrix-valued function with $n \neq 1$ and $m \neq 1$. Different from the normal vector-valued optimization, the decision variable of a matrix-valued optimization problem is a matrix.

To address matrix-valued optimization problem (1), we need to analyze the properties of

matrix-valued function f. Before analyzing the properties of f, we define an auxiliary function $\alpha(\boldsymbol{x}) = f(\boldsymbol{X})$, where $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{X})$. $\alpha(\boldsymbol{x})$ is a useful tool for proving the properties of $f(\boldsymbol{X})$. Now, we propose several definitions and lemmas of $f(\boldsymbol{X})$ using $\alpha(\boldsymbol{x})$.

Definition 1 (L-Lipschitz continuity) $f:\mathbb{R}^{n\times m}\to\mathbb{R}$ is said to be L-Lipschitz continuous if $\forall \boldsymbol{X},\boldsymbol{Y}\in\mathbb{R}^{n\times m}$, $\exists L>0$, such that $|f(\boldsymbol{X})-f(\boldsymbol{Y})|\le L||\boldsymbol{X}-\boldsymbol{Y}\|_{\mathrm{F}}$, where L is a Lipschitz constant.

Definition 2 (l-smoothness) $f:\mathbb{R}^{n\times m}\to\mathbb{R}$ is said to be l-smooth if $\forall \boldsymbol{X},\boldsymbol{Y}\in\mathbb{R}^{n\times m}$, $\exists l>0$, such that $\|\nabla f(\boldsymbol{X})-\nabla f(\boldsymbol{Y})\|_{\mathrm{F}}\le l||\boldsymbol{X}-\boldsymbol{Y}\|_{\mathrm{F}}$.

Definition 3 (μ-strong convexity) $f:\mathbb{R}^{n\times m}\to\mathbb{R}$ is said to be μ-strongly convex if $\forall \boldsymbol{X},\boldsymbol{Y}\in\mathbb{R}^{n\times m}$, $\exists\mu>0$, such that $f(\boldsymbol{Y})\ge f(\boldsymbol{X})+\mathrm{tr}((\nabla f(\boldsymbol{X}))^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{X}))+\mu\|\boldsymbol{Y}-\boldsymbol{X}\|_{\mathrm{F}}^2/2$.

Note that Definition 3 defines the convexity of f if $\mu=0$.

Definition 4 For any convex function $f:\mathbb{R}^{n\times m}\to\mathbb{R}$, the subdifferential of f with respect to $\boldsymbol{X}$ is defined by

$$\partial f(\boldsymbol{X})=\left\{\boldsymbol{G}|f(\boldsymbol{Y})\ge f(\boldsymbol{X})+\mathrm{tr}\left(\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{X})\right)\right\}.$$

In addition, $\boldsymbol{G}\in\partial f(\boldsymbol{X})$ is called a subgradient of f at $\boldsymbol{X}$.

Furthermore, based on the properties of the Frobenius norm and Definitions 1–3, two lemmas are derived:

Lemma 1 Assume $f:\mathbb{R}^{n\times m}\to\mathbb{R}$ and $\alpha:\mathbb{R}^{nm}\to\mathbb{R}$ with $\mathrm{vec}(\boldsymbol{X})=\boldsymbol{x}$. $\forall \boldsymbol{X},\boldsymbol{Y}\in\mathbb{R}^{n\times m}$, we have the following statements:

(1) $\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})=(\mathrm{vec}(\boldsymbol{X}))^{\mathrm{T}}\mathrm{vec}(\boldsymbol{Y})$;

(2) $\|\boldsymbol{X}\|_{\mathrm{F}}=\|\boldsymbol{x}\|$;

(3) $\forall\eta>0$, $\|\boldsymbol{X}+\boldsymbol{Y}\|_{\mathrm{F}}\le(1+\eta)\|\boldsymbol{X}\|_{\mathrm{F}}+(1+\eta^{-1})\|\boldsymbol{Y}\|_{\mathrm{F}}$;

(4) $f(\boldsymbol{X})$ is l-Lipschitz continuous if and only if $\alpha(\boldsymbol{x})$ is l-Lipschitz continuous;

(5) $f(\boldsymbol{X})$ is l-smooth if and only if $\alpha(\boldsymbol{x})$ is l-smooth;

(6) $f(\boldsymbol{X})$ is μ-strongly convex if and only if $\alpha(\boldsymbol{x})$ is μ-strongly convex.

Proof For statement (1), we can obtain

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}=\begin{bmatrix}\sum\limits_{i=1}^{n}X(i,1)Y(i,1) & \sum\limits_{i=1}^{n}X(i,1)Y(i,2) & \dots & \sum\limits_{i=1}^{n}X(i,1)Y(i,m)\\ \sum\limits_{i=1}^{n}X(i,2)Y(i,1) & \sum\limits_{i=1}^{n}X(i,2)Y(i,2) & \dots & \sum\limits_{i=1}^{n}X(i,2)Y(i,m)\\ \vdots & \vdots & \vdots & \vdots\\ \sum\limits_{i=1}^{n}X(i,m)Y(i,1) & \sum\limits_{i=1}^{n}X(i,m)Y(i,2) & \dots & \sum\limits_{i=1}^{n}X(i,m)Y(i,m)\end{bmatrix}.$$

Next, we have

$$\begin{aligned}\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})&=\sum_{j=1}^{m}\sum_{i=1}^{n}X(i,j)Y(i,j)\\&=(\mathrm{vec}(\boldsymbol{X}))^{\mathrm{T}}\mathrm{vec}(\boldsymbol{Y}).\end{aligned}$$

For statement (2), based on statement (1) we have

$$\begin{aligned}\|\boldsymbol{X}\|_{\mathrm{F}}&=\sqrt{\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})}\\&=\sqrt{(\mathrm{vec}(\boldsymbol{X}))^{\mathrm{T}}\mathrm{vec}(\boldsymbol{X})}\\&=\sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}\\&=\|\boldsymbol{x}\|.\end{aligned}$$

For statement (3), it is a well-known norm inequality. For statements (4)–(6), they can be easily proved using statements (1) and (2).

Lemma 2 If $f(\boldsymbol{X})$ is μ-strongly convex and bounded with M', and $\nabla f(\boldsymbol{X})$ is bounded with M'', then $f(\boldsymbol{X})$ is $2M'\mu/M''$-Lipschitz continuous.

Proof Based on statement (5) in Lemma 1, $\alpha(\boldsymbol{x})$ is μ-strongly convex if $f(\boldsymbol{X})$ is μ-strongly convex. Then, according to Lemma 4.2 in Zhou et al. (2017), we have that $\alpha(\boldsymbol{x})$ is $2M'\mu/M''$-Lipschitz continuous. Based on statement (3) in Lemma 1, $f(\boldsymbol{X})$ is $2M'\mu/M''$-Lipschitz continuous if $\alpha(\boldsymbol{x})$ is $2M'\mu/M''$-Lipschitz continuous.

Definitions 1–3 provide several properties of function $f(\boldsymbol{X})$ which will be commonly considered in the vector-optimization theory. Thus, in matrix-valued optimization, we generalize them into the matrix-valued domain. Based on Lemma 1 and $\alpha(\boldsymbol{x})$, many existing results in the vector-valued domain $\mathbb{R}^{nm}$ can be generalized to the matrix-valued domain $\mathbb{R}^{n\times m}$. For example, Lemma 2 can be proved via statements (4) and (6) in Lemma 1. According to Lemma 2, the strong convexity can lead to Lipschitz continuity under the conditions in Lemma 2.

For a non-smooth function, the subgradient defined in Definition 4 is adopted when the gradient does not exist. In distributed optimization, there are many works about non-smooth analysis with subgradients (Ruszczyński, 2006; Zeng et al., 2017). In the optimization problem $\min\sum_{i=1}^{N}f_i(\boldsymbol{X}_i)$, if for any $i\in\mathcal{I}_N$, $f_i(\boldsymbol{X}_i)$ satisfies the Lipschitz condition in Definition 1, the sum sign and the subdifferential sign can be exchanged, i.e., $\partial\sum_{i=1}^{N}f_i(\boldsymbol{X}_i)=\sum_{i=1}^{N}\partial f_i(\boldsymbol{X}_i)$. This statement can be proved by changing the matrix problem into a vector problem via function $\alpha(\boldsymbol{x})$, and

its proof can be obtained from Ruszczyński (2006) by statement (3) in Lemma 1. In addition, for Definitions 2 and 3, if the function is non-smooth, we can substitute its subgradient for its gradient. In these cases, μ-strong convexity is still called μ-strong convexity, but l-smoothness is now called l-pseudo smoothness and is defined as follows:

Definition 5 (l-pseudo smoothness) $f(\boldsymbol{X})$: $\mathbb{R}^{n\times m} \to \mathbb{R}$ is l-pseudo smooth if for any $\boldsymbol{X},\boldsymbol{Y} \in \Omega_{n\times m}$, $\forall \xi \in \partial f(\boldsymbol{X})$, $\exists l > 0$, such that $f(\boldsymbol{Y}) \le f(\boldsymbol{X}) + \mathrm{tr}(\boldsymbol{\xi}^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{X})) + l\|\boldsymbol{Y}-\boldsymbol{X}\|_{\mathrm{F}}^2/2$.

Lemma 3 If $l\|\boldsymbol{X}\|_{\mathrm{F}}^2/2 - f(\boldsymbol{X})$ is convex, then $f(\boldsymbol{X})$ is l-pseudo smooth.

Proof If $l\|\boldsymbol{X}\|_{\mathrm{F}}^2/2 - f(\boldsymbol{X})$ is convex, then we have

$$\begin{aligned} &l\|\boldsymbol{Y}\|_{\mathrm{F}}^2/2 - f(\boldsymbol{Y}) \\ \ge& l\|\boldsymbol{X}\|_{\mathrm{F}}^2/2 - f(\boldsymbol{X}) + \mathrm{tr}(\boldsymbol{\xi}^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{X})) \\ &+ l\cdot\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}) - l\|\boldsymbol{X}\|_{\mathrm{F}}^2, \end{aligned}$$

which leads to

$$f(\boldsymbol{Y}) \le f(\boldsymbol{X}) + \mathrm{tr}(\boldsymbol{\xi}^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{X})) + l\|\boldsymbol{Y}-\boldsymbol{X}\|_{\mathrm{F}}^2/2.$$

Lemma 3 provides a sufficient condition for judging the l-pseudo smoothness of a non-smooth function.

3 Problem formulation

Consider an optimization problem with inequality and equality constraints as follows:

$$\begin{aligned} &\min \sum_{i=1}^{N} f_i(\boldsymbol{X}_i) \\ &\text{s.t.} \begin{cases} \boldsymbol{X}_i = \boldsymbol{X}_j, & i,j \in \mathcal{I}_N, \\ g(\boldsymbol{X}_i) \le 0, & i \in \mathcal{I}_N, \\ h(\boldsymbol{X}_i) = 0, & i \in \mathcal{I}_N, \end{cases} \end{aligned} \tag{2}$$

where $f_i : \mathbb{R}^{n\times m} \to \mathbb{R}$ is the local objective function for $i \in \mathcal{I}_N$, $g : \mathbb{R}^{n\times m} \to \mathbb{R}$ and $h : \mathbb{R}^{n\times m} \to \mathbb{R}$ are the constraint functions.

Problem (2) is a constrained matrix-valued distributed optimization problem, and the handling of its constraints is a difficult point. For handling bound constraints, a penalty method was proposed in Zhou et al. (2019). In Zhou et al. (2019), the selection of penalty gains relied on the Lipschitz constants of the objective functions, and the continuous-time optimization method was also dependent on the penalty gains. Compared with bound constraints, equality and inequality constraints in problem (2) are more complex, and the selection of penalty gains relies on not only the objective functions f_i, but also g and h, which may be more complex than the optimization in Zhou et al. (2019). Thus, we will develop an exact penalty method for handling the equality and inequality constraints in Section 4.2. The exact penalty method can transform an optimization problem with equality and inequality constraints into an optimization problem without equality or inequality constraints.

Using the penalty method, an optimization problem without inequality constraint $g(\boldsymbol{X}_i) \le 0$ or equality constraint $h(\boldsymbol{X}_i) = 0$ is proposed as follows:

$$\begin{aligned} &\min \sum_{i=1}^{N} \tilde{f}_i(\boldsymbol{X}_i) \\ &\text{s.t. } \boldsymbol{X}_i = \boldsymbol{X}_j,\ i,j \in \mathcal{I}_N, \end{aligned} \tag{3}$$

where $\tilde{f}_i(\boldsymbol{X}_i) = f_i(\boldsymbol{X}_i) + P_{g_i}\mathcal{A}_i(\boldsymbol{X}_i) + P_{h_i}\mathcal{B}_i(\boldsymbol{X}_i)$ with $\mathcal{A}_i(\boldsymbol{X}_i) = (g(\boldsymbol{X}_i) + |g(\boldsymbol{X}_i)|)^2$ and $\mathcal{B}_i(\boldsymbol{X}_i) = |h(\boldsymbol{X}_i)|^2$ for $i \in \mathcal{I}_N$.

The core of the penalty method is to derive feasible penalty gains and penalty functions for guaranteeing the equivalence between the original constrained problems and unconstrained problems. In this study, $\mathcal{A}_i$ and $\mathcal{B}_i$ are penalty functions. P_{g_i} and P_{h_i} $(i = 1, 2, ..., N)$ are the penalty gains that need to be chosen to guarantee the equivalence between problems (2) and (3). Thus, the penalty gains P_{g_i} and P_{h_i} are important for problem transformation, and the selection principle of them is given in Section 4.1.

In addition, we propose a new type of optimization, matrix-valued distributed stochastic optimization, in which stochastic variables are considered into problem (2), and its form is shown as follows:

$$\begin{aligned} &\min \sum_{i=1}^{N} f_i(\boldsymbol{X}_i) = \sum_{i=1}^{N} \mathbb{E}_{\xi_i\in\mathcal{D}_i} F_i(\boldsymbol{X}_i, \xi_i) \\ &\text{s.t.} \begin{cases} \boldsymbol{X}_i = \boldsymbol{X}_j, & i,j \in \mathcal{I}_N, \\ g(\boldsymbol{X}_i) \le 0, & i \in \mathcal{I}_N, \\ h(\boldsymbol{X}_i) = 0, & i \in \mathcal{I}_N. \end{cases} \end{aligned} \tag{4}$$

Note that problems (4) and (1) are identical if we set $\boldsymbol{X}_i = \boldsymbol{X}_j = \boldsymbol{X}$, $\forall i,j \in \mathcal{I}_N$. $F_i(\boldsymbol{X}_i, \xi_i)$ can be regarded as a stochastic function since $\mathcal{D}_i$ satisfies some distribution. We call problem (4) a matrix-valued distributed stochastic optimization with inequality and equality constraints. In this study, we

focus on solving problem (4) by developing a distributed optimization algorithm.

4 Main results

In this section, we address the exact penalty method and the development of a distributed optimization algorithm for solving the matrix-valued distributed stochastic optimization problem (4). In Section 4.1, a selection principle of penalty gains and penalty functions is proposed. Then, we analyze how to obtain feasible penalty gains. In Section 4.2, an exact penalty method is proposed to select the penalty gains and handle the inequality and equality constraints. In Section 4.3, an algorithm based on a gossip model for solving problem (4) is developed.

4.1 Selection principle of penalty gains

Beginning with a centralized matrix-valued optimization problem $\min f(\boldsymbol{X})$ s.t. $\boldsymbol{X} \in S \subset \mathbb{R}^{n\times m}$ (S denotes a feasible set), we propose a selection principle for seeking the penalty gains and penalty functions:

Selection principle 1 Penalty gain c ($c>0$) and penalty function $\tau_S(\boldsymbol{X}) : \mathbb{R}^{n\times m} \to \mathbb{R}$ satisfy the following conditions:

(1) $\forall \boldsymbol{X} \in \mathbb{R}^{n\times m}$, $f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X}) \geq f(P_S(\boldsymbol{X}))$;

(2) $\forall \boldsymbol{X} \in \mathbb{R}^{n\times m}$, $\tau_S(\boldsymbol{X}) \geq 0$;

(3) $\forall \boldsymbol{X} \in S$, $\tau_S(\boldsymbol{X}) = 0$.

Based on Selection principle 1, the equivalence theorem is derived in the following:

Theorem 1 If there exist c and $\tau_S(\boldsymbol{X})$ satisfying Selection principle 1 and the considered problem has at least one solution, then $\arg\min_{\boldsymbol{X}\in S} f(\boldsymbol{X}) = \arg\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} (f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X}))$.

Proof Let $\boldsymbol{X}^* = \arg\min_{\boldsymbol{X}\in S} f(\boldsymbol{X})$. We can determine that $\forall \boldsymbol{X} \in S$, $f(\boldsymbol{X}^*) \leq f(\boldsymbol{X})$. According to condition (3) in Selection principle 1, $f(\boldsymbol{X}^*) + c\tau_S(\boldsymbol{X}^*) = f(\boldsymbol{X}^*) \leq f(\boldsymbol{X}) = f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X})$ holds. Note that $\forall \boldsymbol{X} \in S$, $f(\boldsymbol{X}^*) + c\tau_S(\boldsymbol{X}^*) \leq f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X})$. Then, we have that $\forall \boldsymbol{X}' \notin S$, $f(\boldsymbol{X}^*) + c\tau_S(\boldsymbol{X}^*) \leq f(\boldsymbol{X}') + c\tau_S(\boldsymbol{X}')$. According to condition (1) in Selection principle 1 and $P_S(\boldsymbol{X}) \in S$, we can obtain that $f(\boldsymbol{X}^*) + c\tau_S(\boldsymbol{X}^*) = f(\boldsymbol{X}^*) \leq f(P_S(\boldsymbol{X}')) \leq f(\boldsymbol{X}') + c\tau_S(\boldsymbol{X}')$. Therefore, $\forall \boldsymbol{X} \in S$, $f(\boldsymbol{X}^*) + c\tau_S(\boldsymbol{X}^*) \leq f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X})$, which implies $\boldsymbol{X}^* = \arg\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} (f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X}))$. Conversely, letting $\boldsymbol{X}^* := \arg\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} (f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X}))$, we need to prove that $\forall \boldsymbol{X} \in S$, $f(\boldsymbol{X}^*) \leq f(\boldsymbol{X})$. Based on conditions (2) and (3) in Selection principle 1, $\forall \boldsymbol{X} \in S$, $f(\boldsymbol{X}) = f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X})$. Thus, $\boldsymbol{X}^* = \arg\min_{\boldsymbol{X}\in S} f(\boldsymbol{X})$. To sum up, $\arg\min_{\boldsymbol{X}\in S} f(\boldsymbol{X}) = \arg\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} (f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X}))$.

According to Theorem 1, if the penalty function $\tau_S(\boldsymbol{X})$ and the penalty gain c satisfy Selection principle 1, then the equivalence between the original constrained problem $\min_{\boldsymbol{X}\in S} f(\boldsymbol{X})$ and problem $\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}} (f(\boldsymbol{X}) + c\tau_S(\boldsymbol{X}))$ can be guaranteed. Similarly, we can select the penalty functions and penalty gains by Selection principle 1 to guarantee the equivalence between problems (2) and (3). Actually, when we set $S := \Omega_{g_i} = \{\boldsymbol{X}_i | g(\boldsymbol{X}_i) \leq 0\}$, the function

$$\tau_{\Omega_{g_i}}(\boldsymbol{X}) = \mathcal{A}_i(\boldsymbol{X}_i) = (g(\boldsymbol{X}_i) + |g(\boldsymbol{X}_i)|)^2$$

satisfies conditions (2) and (3) in Selection principle 1. When we set $S := \Omega_{h_i} = \{\boldsymbol{X}_i | h(\boldsymbol{X}_i) = 0\}$, the function

$$\tau_{\Omega_{h_i}}(\boldsymbol{X}_i) = \mathcal{B}_i(\boldsymbol{X}_i) = |h(\boldsymbol{X}_i)|^2$$

satisfies conditions (2) and (3) in Selection principle 1. Hence, if the penalty gains P_{g_i} and P_{h_i} satisfy condition (1) in Selection principle 1, then problems (2) and (3) are equivalent. Therefore, in the next subsection, we propose an exact penalty method to select feasible penalty gains satisfying condition (1) in Selection principle 1.

4.2 A penalty method

In this subsection, we derive the selection of penalty gains for problem (2). Let

$$\hat{\boldsymbol{X}}_i := P_{\Omega_{g_i}\cap\Omega_{h_i}}(\boldsymbol{X}_i) = \arg\min_{\boldsymbol{Y}\in\Omega_{g_i}\cap\Omega_{h_i}} ||\boldsymbol{Y} - \boldsymbol{X}_i||_{\mathrm{F}},$$

$$L_{g_i}(\boldsymbol{X}_i) = \mathcal{A}_i(\boldsymbol{X}_i)/\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i\|_{\mathrm{F}},$$

and

$$L_{h_i}(\boldsymbol{X}_i) = \mathcal{B}_i(\boldsymbol{X}_i)/\|\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i\|_{\mathrm{F}}.$$

Then, Assumption 1 is provided as follows:

Assumption 1 (1) f_i is convex and L_f-Lipschitz continuous for $i \in \mathcal{I}_N$; (2) Slater's condition holds.

Next, we give the equivalence theorem between problems (2) and (3) on the basis of penalty gains P_{g_i} and P_{h_i} satisfying the conditions with $L_{g_i}(\boldsymbol{X}_i)$ and $L_{h_i}(\boldsymbol{X}_i)$:

Theorem 2 Under Assumption 1, if $L_{g_i}(\boldsymbol{X}_i)P_{g_i}+L_{h_i}(\boldsymbol{X}_i)P_{h_i}\geq L_f$ for $\boldsymbol{X}_i\notin\Omega_{g_i}\cap\Omega_{h_i}$, then problem (2) is equivalent to problem (3).

Proof According to Theorem 1, we should prove that P_{g_i} and P_{h_i} satisfy condition (1) in Selection principle 1, $\forall f_i,\ i\in\mathcal{I}_N$. Because $L_{g_i}(\boldsymbol{X})P_{g_i}+L_{h_i}(\boldsymbol{X})P_{h_i}\geq L_f$, we have

$$\begin{aligned}&\tilde{f}_i(\boldsymbol{X}_i)-f_i(\hat{\boldsymbol{X}}_i)\\=&f_i(\boldsymbol{X}_i)+P_{g_i}\mathcal{A}_i(\boldsymbol{X}_i)+P_{h_i}\mathcal{B}_i(\boldsymbol{X}_i)-f_i(\hat{\boldsymbol{X}}_i)\\\geq&-L_f\|\boldsymbol{X}_i-\hat{\boldsymbol{X}}_i\|_{\mathrm{F}}+(P_{g_i}L_{g_i}(\boldsymbol{X}_i)\\&+P_{h_i}L_{h_i}(\boldsymbol{X}_i))\|\boldsymbol{X}_i-\hat{\boldsymbol{X}}_i\|_{\mathrm{F}}\\\geq&(P_{g_i}L_{g_i}(\boldsymbol{X}_i)+P_{h_i}L_{h_i}(\boldsymbol{X}_i)-L_f)\|\boldsymbol{X}_i-\hat{\boldsymbol{X}}_i\|_{\mathrm{F}}\\\geq&0,\end{aligned}$$

which implies that condition (1) in Selection principle 1 holds. Therefore, $\min_{\boldsymbol{X}\in\Omega_g\cap\Omega_h}f_i(\boldsymbol{X})$ is identical to $\min_{\boldsymbol{X}\in\mathbb{R}^{n\times m}}\tilde{f}_i(\boldsymbol{X})$, which indicates that problem (2) is equivalent to problem (3).

Theorem 2 shows that problems (2) and (3) are equivalent if $L_{g_i}(\boldsymbol{X}_i)P_{g_i}+L_{h_i}(\boldsymbol{X}_i)P_{h_i}\geq L_f$ holds. Thus, if we select the penalty gains P_{g_i} and P_{h_i} satisfying $L_{g_i}(\boldsymbol{X}_i)P_{g_i}+L_{h_i}(\boldsymbol{X}_i)P_{h_i}\geq L_f$, then we can solve problem (2) by solving problem (3) which is not subject to inequality constraint $g(\boldsymbol{X}_i)\leq 0$ or equality constraint $h(\boldsymbol{X}_i)=0$.

4.3 Matrix-valued distributed stochastic optimization algorithm based on a gossip model

In this subsection, we develop an algorithm based on a gossip model (its vector-valued form can be found in Xiao and Boyd (2004)). First, we introduce the gossip model as follows:

$$\boldsymbol{X}_i(t+1)=\boldsymbol{X}_i(t)+\kappa\sum_{j\in\mathcal{N}_i}A(i,j)(\boldsymbol{X}_j(k)-\boldsymbol{X}_i(k)),$$

where $i\in\mathcal{I}_N$. $\kappa>0$ limits the rate of the gossip updating. Each state $\boldsymbol{X}_i$ is an $n\times m$ matrix. $\boldsymbol{A}$ is the adjacency matrix of graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\boldsymbol{A})$. In gossip updating, each agent shares its own information with its neighbors and updates by local average, which makes all of the agents reach an agreement on a target solution.

Before proceeding, we propose Assumption 2:

Assumption 2 (1) f_i is bounded with m, μ-strongly convex, and l-smooth for $i\in\mathcal{I}_N$. ∇f_i is bounded with M. (2) $\mathcal{A}_i$ is convex and l_g-pseudo smooth; $\mathcal{B}_i$ is convex and l_h-pseudo smooth. $\partial\mathcal{A}_i$ is bounded with M_g, and $\partial\mathcal{B}_i$ is bounded with M_h. (3) For $i\in\mathcal{I}_N$, $\forall\boldsymbol{X}_i\in\mathbb{R}^{n\times m}$, $\mathbb{E}_{\xi_i}\|\nabla F_i(\boldsymbol{X}_i,\xi_i)-\nabla f_i(\boldsymbol{X}_i)\|^2\leq\sigma_i^2$, $\mathbb{E}_{\xi_i}\|\nabla F_i(\boldsymbol{X}_i,\xi_i)\|^2\leq E^2$, with σ_i and E being known upper bounds. (4) $\boldsymbol{A}$ is a symmetric doubly stochastic matrix. (5) Slater's condition holds.

In Assumption 2, the convexity of the objective functions and constraint functions can lead to a globally optimal solution for optimization problems (Boyd and Vandenberghe, 2004) when the optimal solution exists. The strong convexity and the l-smoothness contribute to the proof of the convergence of the proposed algorithm (Rakhlin et al., 2012). According to Lemma 2, item (3) gives the characteristics of random sampling. Item (4) indicates that the graph is connected and weight-balanced, which contributes to the consensus of states. Slater's condition is a common constraint qualification to guarantee the solvability of an optimization problem (Liu et al., 2017; Xia ZC et al., 2022).

To proceed, based on the gossip model, Algorithm 1 is developed for solving problem (4).

$\boldsymbol{X}_i$ is regarded as the state of the i^{th} agent. Eqs. (5) and (6) are regarded as information update processes of the i^{th} agent, and they originate from the gossip model. Each agent shares its own information with its neighbors by $\boldsymbol{A}$ and updates by local average, which makes all of the agents reach an agreement on a target solution. Therefore, Eq. (5) with Eq. (6) is a multi-agent system, and Algorithm 1 is a distributed optimization method.

In Algorithm 1, the time-varying step $\zeta(k)$ should be chosen to achieve convergence to the optimal value. Thus, we give a convergence theorem to prove the convergence of Algorithm 1, and $\zeta(k)$ is designed in the theorem. Before proposing the convergence theorem, we introduce four necessary lemmas.

Lemma 4 For Algorithm 1, let $\boldsymbol{\mathcal{X}}(k):=\mathrm{col}[\boldsymbol{X}_1(k),\boldsymbol{X}_2(k),...,\boldsymbol{X}_N(k)]\in\mathbb{R}^{nN\times m}$, $\boldsymbol{\mathcal{Y}}(k):=\mathrm{col}[\boldsymbol{Y}_1(k),\boldsymbol{Y}_2(k),...,\boldsymbol{Y}_N(k)]\in\mathbb{R}^{nN\times m}$, and $\bar{\boldsymbol{\mathcal{X}}}(k):=\mathrm{col}[\bar{\boldsymbol{X}}(k),\bar{\boldsymbol{X}}(k),...,\bar{\boldsymbol{X}}(k)]\in\mathbb{R}^{nN\times m}$ with $\bar{\boldsymbol{X}}(k)=\sum_{i=1}^N\boldsymbol{X}_i(k)/N$. If $\boldsymbol{A}$ is a doubly stochastic matrix, then $\bar{\boldsymbol{\mathcal{X}}}(k+1)=\bar{\boldsymbol{\mathcal{X}}}(k)-\mathbf{1}_N^{\mathrm{T}}\otimes(\zeta(k)/N)\sum_{i=1}^N(\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))+\mathcal{H}_i(k))$.

Proof According to Algorithm 1, we have

$$\bar{\boldsymbol{X}}(k+1)$$

Algorithm 1 Distributed stochastic gradient descent algorithm

1: **Initialization**
2: **Input**: $\boldsymbol{X}_i(0)$, time-varying step $\zeta(k)$, total number of iterations K, and $\boldsymbol{A}$
3: **For** $k=1,2,...,K$
4: Sample $\xi_i(k)$, and calculate $\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))$
5: Choose P_{g_i} and P_{h_i} satisfying

$$L_{g_i}(\boldsymbol{X}_i(k))P_{g_i}+L_{h_i}(\boldsymbol{X}_i(k))P_{h_i}\geq\frac{2m\mu}{M}$$

6: Choose $\mathcal{H}_i(k)\in P_{g_i}\partial\mathcal{A}(\boldsymbol{X}_i(k))+P_{h_i}\partial\mathcal{B}(\boldsymbol{X}_i(k))$
7: Calculate

$$\boldsymbol{Y}_i(k)=\boldsymbol{X}_i(k)-\zeta(k)(\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))+\mathcal{H}_i(k)) \tag{5}$$

8: Calculate

$$\boldsymbol{X}_i(k+1)=\boldsymbol{Y}_i(k)+\kappa\sum_{j\in\mathcal{N}_i}A(i,j)(\boldsymbol{Y}_j(k)-\boldsymbol{Y}_i(k)) \tag{6}$$

9: **End for**
10: **Output**: $\boldsymbol{X}_{\mathrm{avg}}(K)$

$$\begin{aligned}&=\bar{\boldsymbol{X}}(k)-\frac{\zeta(k)}{N}\sum_{i=1}^{N}(\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))+\mathcal{H}(k))\\&\quad+\frac{\kappa}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}A(i,j)(\boldsymbol{Y}_j(k)-\boldsymbol{Y}_i(k)).\end{aligned} \tag{7}$$

Note that $\boldsymbol{A}\otimes\boldsymbol{I}_n$ is also a symmetric doubly stochastic matrix. Thus, we can obtain $\mathbf{1}_{nN}^{\mathrm{T}}(\boldsymbol{A}\otimes\boldsymbol{I}_n)/N=\mathbf{1}_{nN}^{\mathrm{T}}/N$, which yields

$$\begin{aligned}&\frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}A(i,j)(\boldsymbol{Y}_j(k)-\boldsymbol{Y}_i(k))\\&=\frac{\mathbf{1}_{nN}^{\mathrm{T}}}{N}((\boldsymbol{A}\otimes\boldsymbol{I}_n)-\boldsymbol{I}_{nN})\boldsymbol{\mathcal{Y}}(t)\\&=0.\end{aligned} \tag{8}$$

Based on Eq. (8), the proof of Lemma 4 is completed.

According to Lemma 2, we have that f_i is $2m\mu/M$-Lipschitz continuous.

Lemma 5 Assume that $\boldsymbol{X}^*$ is an optimal solution to problem (4). Under Assumption 2, $\bar{\boldsymbol{X}}(k)$ in Algorithm 1 satisfies

$$\begin{aligned}&\mathbb{E}_{\xi_1(k),\xi_2(k),\ldots,\xi_N(k)}\left\|\bar{\boldsymbol{X}}(k+1)-\boldsymbol{X}^*\right\|_{\mathrm{F}}^2\\&\leq\left(1-\frac{\zeta(k)\chi_1}{2}\right)\left\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*\right\|_{\mathrm{F}}^2+\zeta^2(k)\left(\frac{\bar{\sigma}^2}{N}+\chi_2\right)\\&\quad-2\zeta(k)\left(\sum_{i=1}^{N}\tilde{f}_i(\bar{\boldsymbol{X}}(k))-\sum_{i=1}^{N}\tilde{f}_i(\boldsymbol{X}^*)\right)\\&\quad+\zeta(k)\frac{\chi_3}{N}\sum_{i=1}^{N}\left\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}_i(k)\right\|_{\mathrm{F}}^2,\end{aligned} \tag{9}$$

where

$$\begin{cases}\chi_1=\mu,\\ \chi_2=2M^2+4P_gM_g+4P_hM_h,\\ \chi_3=l+P_gl_g+P_hl_h+\mu,\\ \bar{\sigma}^2:=\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2,\end{cases}$$

with $P_g=\max_i\{P_{g_i}\}$ and $P_h=\max_i\{P_{h_i}\}$.

Proof Let $\mathcal{H}(k)=P_g\mathcal{H}'(k)+P_h\mathcal{H}''(k)\in P_g\partial\mathcal{A}(\boldsymbol{X}_i(k))+P_h\partial\mathcal{B}(\boldsymbol{X}_i(k))$ with $\mathcal{H}'(k)\in\partial\mathcal{A}(\boldsymbol{X}_i(k))$ and $\mathcal{H}''(k)\in\partial\mathcal{B}(\boldsymbol{X}_i(k))$. Owing to Lemma 4, we have

$$\begin{aligned}&\left\|\bar{\boldsymbol{X}}(k+1)-\boldsymbol{X}^*\right\|_{\mathrm{F}}^2\\&=\left\|\bar{\boldsymbol{X}}(k)-\frac{\zeta(k)}{N}\sum_{i=1}^{N}(\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))+\mathcal{H}(k))-\boldsymbol{X}^*\right\|_{\mathrm{F}}^2\\&=\left\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*-\frac{\zeta(k)}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i(k))+\mathcal{H}(k))\right.\\&\quad+\frac{\zeta(k)}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i(k))+\mathcal{H}(k))\\&\quad\left.-\frac{\zeta(k)}{N}\sum_{i=1}^{N}(\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))+\mathcal{H}(k))\right\|_{\mathrm{F}}^2\\&=\|\boldsymbol{\Delta}_1\|_{\mathrm{F}}^2+\zeta^2(k)\|\boldsymbol{\Delta}_2\|_{\mathrm{F}}^2+\frac{2\zeta(k)}{N}\mathrm{tr}(\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2),\end{aligned} \tag{10}$$

where

$$\boldsymbol{\Delta}_1=\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*-\frac{\zeta(k)}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i(k))+\mathcal{H}(k)) \tag{11}$$

and

$$\begin{aligned}\boldsymbol{\Delta}_2=&\frac{1}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i(k))+\mathcal{H}(k))\\&-\frac{1}{N}\sum_{i=1}^{N}(\nabla F_i(\boldsymbol{X}_i(k),\xi_i(k))+\mathcal{H}(k)).\end{aligned}$$

Since $\sum_{i=1}^{N}f_i(\boldsymbol{X}_i)=\sum_{i=1}^{N}\mathbb{E}_{\xi_i\in\mathcal{D}_i}F_i(\boldsymbol{X}_i,\xi_i)$, we have

$$\mathbb{E}_{\xi_1(k),\xi_2(k),\ldots,\xi_N(k)}\frac{2\zeta(k)}{N}\mathrm{tr}(\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2)=0.$$

Based on item (4) in Assumption 2, we can derive

$$\mathbb{E}_{\xi_1(k),\xi_2(k),\ldots,\xi_N(k)}\zeta^2(k)\|\boldsymbol{\Delta}_2\|_{\mathrm{F}}^2 \le \frac{\zeta^2(k)\bar{\sigma}^2}{N}.$$

For $\|\boldsymbol{\Delta}_1\|_{\mathrm{F}}^2$, we can obtain

$$\begin{aligned}\|\boldsymbol{\Delta}_1\|_{\mathrm{F}}^2 &= \left\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*\right\|_{\mathrm{F}}^2\\ &+\zeta^2(k)\underbrace{\left\|\frac{1}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i(k))+\mathcal{H}(k))\right\|_{\mathrm{F}}^2}_{\Delta_3}\\ &\underbrace{-2\zeta(k)\mathrm{tr}\left(\left(\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*\right)^{\mathrm{T}}\frac{1}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i(k))+\mathcal{H}(k))\right)}_{\Delta_4}.\end{aligned}$$

Then, according to statement (3) in Lemma 1, and items (1) and (2) in Assumption 2, we have

$$\begin{aligned}\Delta_3 &= \left\|\frac{1}{N}\sum_{i=1}^{N}(\nabla f_i(\boldsymbol{X}_i)+\mathcal{H}(k))\right\|_{\mathrm{F}}^2\\ &\le 2\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\boldsymbol{X}_i)\right\|_{\mathrm{F}}^2+4\left\|\frac{1}{N}\sum_{i=1}^{N}P_{g_i}\mathcal{H}'(k)\right\|_{\mathrm{F}}^2\\ &\quad+4\left\|\frac{1}{N}\sum_{i=1}^{N}P_{h_i}\mathcal{H}''(k)\right\|_{\mathrm{F}}^2\\ &\le 2M^2+4P_gM_g+4P_hM_h\\ &=\chi_2.\end{aligned}$$

According to items (1) and (2) in Assumption 2, we can also obtain

$$\begin{aligned}&-\frac{1}{\zeta(k)}\Delta_4\\ &=-\frac{2}{N}\sum_{i=1}^{N}\Big[\mathrm{tr}\left(\left(\bar{\boldsymbol{X}}(k)-\boldsymbol{X}_i(k)\right)^{\mathrm{T}}(\nabla f_i(\boldsymbol{X}_i)+\mathcal{H}(k))\right)\\ &\quad+\mathrm{tr}\left((\boldsymbol{X}_i(k)-\boldsymbol{X}^*)^{\mathrm{T}}(\nabla f_i(\boldsymbol{X}_i)+\mathcal{H}(k))\right)\Big]\\ &\le-\frac{2}{N}\sum_{i=1}^{N}\Big[\tilde{f}_i(\bar{\boldsymbol{X}}(k))-\tilde{f}_i(\boldsymbol{X}_i(k))\\ &\quad-\frac{l+P_gl_g+P_hl_h}{2}\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}_i(k)\|_{\mathrm{F}}^2\\ &\quad+\tilde{f}_i(\boldsymbol{X}_i(k))-\tilde{f}_i(\boldsymbol{X}^*)-\frac{\mu_i}{2}\|\boldsymbol{X}_i(k)-\boldsymbol{X}^*\|_{\mathrm{F}}^2\Big]\\ &\le-\frac{2}{N}\left(\sum_{i=1}^{N}\tilde{f}_i(\bar{\boldsymbol{X}}(k))-\sum_{i=1}^{N}\tilde{f}_i(\boldsymbol{X}^*)\right)\\ &\quad+\sum_{i=1}^{N}\frac{l+P_gl_g+P_hl_h+\mu}{N}\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}_i(k)\|_{\mathrm{F}}^2\\ &\quad-\sum_{i=1}^{N}\frac{\mu}{2N}\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*\|_{\mathrm{F}}^2.\end{aligned}\tag{12}$$

Combining with expressions (10)–(12), the proof is completed.

Lemma 6 (Koloskova et al., 2019) Under Assumption 2, $\{\boldsymbol{\mathcal{X}}(k)\}_{k\ge0}$ in Algorithm 1 with $\zeta(k)=a_1/(k+a_2)$, $a_1>0$, and $a_2\ge5/p$ satisfies

$$\left\|\boldsymbol{\mathcal{X}}(k+1)-\bar{\boldsymbol{\mathcal{X}}}(k+1)\right\|_{\mathrm{F}}^2\le40\zeta^2(k)\frac{NQ}{p^2},\tag{13}$$

where $Q=E^2+2P_g^2M_g^2+2P_h^2M_h^2$ (E, M_g, and M_h are from Assumption 2), and $0<p\le1$ denotes the convergence rate of Algorithm 1 if the conditions of Lemma 4 hold.

From Theorem 1 in Koloskova et al. (2019), we obtain $p=\kappa(1-\delta_2(\boldsymbol{A}))$ (κ is from the gossip model). Based on item (4) in Assumption 2, $\boldsymbol{A}$ is a symmetric doubly stochastic matrix; thus, all its eigenvalues are positive and its largest eigenvalue is 1.

Lemma 7 (Koloskova et al., 2019) Let $\{a(k)\}_{k\ge0}$ with $a(k)\ge0$ and $\{e(k)\}_{k\ge0}$ with $e(k)\ge0$ be the sequences satisfying

$$a(k)\le(1-\chi\zeta(k))\,a(k)-\zeta(k)e(k)A+\zeta^2(k)B+\zeta^3(k)C$$

for step $\zeta(k)=4/(\chi(a+k))$ and constants $A>0$, $B\ge0$, $C\ge0$, $\chi>0$, $a>1$. Then

$$\begin{aligned}&\frac{A}{S(K)}\sum_{k=0}^{K-1}\omega(k)e(k)\\ &\le\frac{\chi a^3}{4S(K)}a(0)+\frac{2K(K+2a)}{\chi S(K)}B+\frac{16K}{\chi^2S(K)}C\end{aligned}$$

for $\omega(k)=(a+k)^2$ and $S(K):=\sum_{k=0}^{K-1}\omega(k)=K\left(2K^2+6aK-3K+6a^2-6a+1\right)/6\ge K^3/3$ with K being the total number of iterations in Algorithm 1.

Summing with Lemmas 4–7, we derive a theorem as follows, which implies that Algorithm 1 can converge to an optimal solution to problem (4) with a proper $\zeta(k)$:

Theorem 3 Under Assumption 2, for $p>0$, Algorithm 1 for $\zeta(k)=4/(\chi_1(a+k))$ with $a\ge5/p$

converges at the rate

$$\begin{aligned}&\sum_{i=1}^{N} f_i\left(\boldsymbol{X}_{\text{avg}}(K)\right)-\sum_{i=1}^{N} f_i(\boldsymbol{X}^*)\\ \leq&\frac{\chi_1 a^3}{8S(K)}\left\|\bar{\boldsymbol{X}}(0)-\boldsymbol{X}^*\right\|_{\text{F}}^2+\frac{K(K+2a)}{\chi_1 S(K)}\left(\frac{\bar{\sigma}^2}{N}+\chi_2\right)\\ &+\frac{320K\chi_3 Q}{\chi_1^2 S(K)p^2},\end{aligned}$$

where $\boldsymbol{X}_{\text{avg}}(K)=\sum_{k=0}^{K-1}\omega(k)\bar{\boldsymbol{X}}(k)/S(K)$ for $\omega(k)=(a+k)^2$, and $S(K)=\sum_{k=0}^{K-1}\omega(k)\geq K^3/3$ ($\bar{\boldsymbol{X}}(k)$ is from Lemma 4).

Proof Substituting inequality (13) into the bound in inequality (9), we can obtain

$$\begin{aligned}&\mathbb{E}_{\xi_1(k),\xi_2(k),\ldots,\xi_N(k)}\left\|\bar{\boldsymbol{X}}(k+1)-\boldsymbol{X}^*\right\|^2\\ \leq&\left(1-\frac{\zeta(k)\chi_1}{2}\right)\left\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*\right\|^2+\zeta^2(k)\left(\frac{\bar{\sigma}^2}{N}+\chi_2\right)\\ &-2\zeta(k)\left(\sum_{i=1}^{N}\tilde{f}_i(\bar{\boldsymbol{X}}(k))-\sum_{i=1}^{N}\tilde{f}_i(\boldsymbol{X}^*)\right)\\ &+40\zeta^3(k)\frac{\chi_3 Q}{p^2}.\end{aligned}$$

Let $A=2$, $B=\bar{\sigma}^2/N+\chi_2$, $C=40\chi_3 Q/p^2$, and $a(k)=\left\|\bar{\boldsymbol{X}}(k)-\boldsymbol{X}^*\right\|_{\text{F}}^2$. According to Lemma 7 with A, B, and C, we can derive

$$\begin{aligned}&\frac{2}{S(K)}\sum_{k=0}^{K-1}\omega(k)\left(\sum_{i=1}^{N}\tilde{f}_i(\bar{\boldsymbol{X}}(k))-\sum_{i=1}^{N}\tilde{f}_i(\boldsymbol{X}^*)\right)\\ \leq&\frac{\chi_1 a^3}{4S(K)}\left\|\bar{\boldsymbol{X}}(0)-\boldsymbol{X}^*\right\|_{\text{F}}^2\\ &+\frac{2K(K+2a)}{\chi_1 S(K)}\left(\frac{\bar{\sigma}^2}{N}+\chi_2\right)+\frac{16K}{\chi_1^2 S(K)}40\frac{\chi_3 Q}{p^2}.\end{aligned}$$

Because of the convexity of f_i, $\mathcal{A}$, and $\mathcal{B}$ (according to items (1) and (2) in Assumption 2), we have that $\tilde{f}_i$ is convex for $i\in\mathcal{I}_N$. In addition, according to Selection principle 1, we have that $\sum_{i=1}^{N} f_i\left(\boldsymbol{X}_{\text{avg}}(K)\right)\leq\sum_{i=1}^{N}\tilde{f}_i\left(\boldsymbol{X}_{\text{avg}}(K)\right)$ and $\sum_{i=1}^{N} f_i\left(\boldsymbol{X}^*\right)=\sum_{i=1}^{N}\tilde{f}_i\left(\boldsymbol{X}^*\right)$. Then, we can derive

$$\begin{aligned}&\sum_{i=1}^{N} f_i\left(\boldsymbol{X}_{\text{avg}}(K)\right)-\sum_{i=1}^{N} f_i(\boldsymbol{X}^*)\\ \leq&\sum_{i=1}^{N}\tilde{f}_i\left(\boldsymbol{X}_{\text{avg}}(K)\right)-\sum_{i=1}^{N}\tilde{f}_i(\boldsymbol{X}^*)\\ \leq&\frac{1}{S(K)}\sum_{k=0}^{K-1}\omega(k)\left(\sum_{i=1}^{N}\tilde{f}_i(\bar{\boldsymbol{X}}(k))-\sum_{i=1}^{N}\tilde{f}_i(\boldsymbol{X}^*)\right)\\ \leq&\frac{\chi_1 a^3}{8S(K)}\left\|\bar{\boldsymbol{X}}(0)-\boldsymbol{X}^*\right\|_{\text{F}}^2+\frac{320K\chi_3 Q}{\chi_1^2 S(K)p^2}\\ &+\frac{K(K+2a)}{\chi_1 S(K)}\left(\frac{\bar{\sigma}^2}{N}+\chi_2\right).\end{aligned}$$

Theorem 3 provides the proper time-varying step $\zeta(k)$ for Algorithm 1. With $\zeta(k)$, Algorithm 1 can solve the matrix-valued distributed stochastic optimization problem (4). In the next section, we will provide two examples to show the validity of the proposed penalty method and Algorithm 1.

Remark 1 (Convergence rate) The convergence rate of Algorithm 1 is $T_1(K)=O(\frac{\bar{\sigma}^2+N\chi_2}{KN\chi_1})$ (see in Theorem 3), where $\chi_1=\mu$ and $\chi_2=2M^2+4P_gM_g+4P_hM_h$. The methods developed in Zhou et al. (2019), Huang et al. (2021), Xia ZC et al. (2021), and Zhang et al. (2022) are continuous-time optimization methods; thus, they are conservative. Therefore, their convergence rate is low. The convergence rate of the algorithm in Koloskova et al. (2019) is $T_2(K)=O(\frac{\bar{\sigma}^2}{KN\mu})$. Because the problems considered in Koloskova et al. (2019) are not subject to any constraint, $T_1(K)$ is larger than $T_2(K)$. The extra part of $T_1(K)$, except $T_2(K)$, is the handling of constraints; note that χ_2, with respect to constraints mainly, is natural to the balance between unconstrained problems and constrained ones.

Remark 2 (Complexity) The numbers of floating points are the same in $\boldsymbol{X}$ and vec($\boldsymbol{X}$). Thus, the spatial complexities of a matrix-valued algorithm and a vector-valued algorithm are the same when there are no other constraints. The complexity of Algorithm 1 is $O(4KN(nm+1))$, while the complexity of the algorithm in Koloskova et al. (2019) is $O(3KNnm)$. The difference is also generated from the handling of constraints naturally.

Compared with conventional distributed optimization methods (see in the references in Section 1.1), we consider matrix-valued optimization and stochastic optimization. In addition, an exact penalty for dealing with (in)equality constraints is employed to distributed optimization. Table 1 presents the comparison results.

5 Simulations

In this section, two numerical examples are presented to illustrate the characteristics of the penalty methods and Algorithm 1. The algorithm and the data are implemented and simulated in MATLAB®

Table 1 Comparison of existing works with this study

Reference	Matrix-valued	Distributed	Stochastic	Constraint
Huang et al. (2021)	✓	×	×	Bound constraints
Zhang et al. (2022)	✓	×	×	Equality constraints
Zhou et al. (2019), Xia ZC et al. (2021)	×	✓	×	Bound constraints
Koloskova et al. (2019)	×	✓	✓	None
This study	✓	✓	✓	Equality constraints; inequality constraints

R2017b and run on Intel® Core™ i5-8257U CPU @1.40 GHz, Intel Iris Plus Graphics 645 1536 MB, 8 GB 2133 MHz LPDDR3, macOS 10.15.7.

Example 1 Consider the following matrix-valued distributed optimization problem with $N=3$:

$$\min \sum_{i=1}^{3} \|\boldsymbol{H}_i\boldsymbol{X}-\boldsymbol{B}_i\|_{\mathrm{F}}^2$$

$$\text{s.t.} \begin{cases} (2,\ 1,\ 3)\boldsymbol{X}(1,\ 1,\ 1)^{\mathrm{T}}=6.22, \\ (5,\ 2,\ 3)\boldsymbol{X}(1,\ 1,\ 1)^{\mathrm{T}}\le 10.38, \end{cases} \tag{14}$$

where $\boldsymbol{X}\in\mathbb{R}^{3\times 3}$,

$$\boldsymbol{H}_1=\begin{bmatrix} 1 & 1 & 1 \\ 1 & 7 & 3 \\ 1 & 5 & 6 \end{bmatrix},\ \boldsymbol{H}_2=\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix},$$

$$\boldsymbol{H}_3=\begin{bmatrix} 0.568 & 1.000 & 0.234 \\ 1.000 & 0.310 & 0.163 \\ 0.234 & 0.163 & 0.550 \end{bmatrix},\ \boldsymbol{B}_1=\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix},$$

$$\boldsymbol{B}_2=\begin{bmatrix} 3 & 3 & 5 \\ 2 & 3 & 5 \\ 1 & 3 & 4 \end{bmatrix},\ \boldsymbol{B}_3=\begin{bmatrix} 2 & 0 & 3 \\ 9 & 0 & 0 \\ 3 & 4 & 5 \end{bmatrix}.$$

Note that the objective functions are all convex and $2\max_{i\in\{1,2,3\}}\|\boldsymbol{H}_i\|_{\mathrm{F}}^2$-strongly convex. The equality constraint function and inequality constraint function are linear. Thus, the penalty functions $\mathcal{A}(\boldsymbol{X})$ and $\mathcal{B}(\boldsymbol{X})$ are convex and pseudo smooth. We can obtain the optimal solution $\bar{\boldsymbol{X}}^*$ without any constraint as follows:

$$\bar{\boldsymbol{X}}^*=\begin{bmatrix} 1.01 & 1.14 & 0.23 \\ 0.18 & 0.00 & 0.38 \\ 0.00 & 0.37 & 0.38 \end{bmatrix},$$

and the optimal value 107.88. Then, we perform Algorithm 1 without random samples where parameters are taken as $\kappa=0.1$, $P_g=4$, $P_h=10$,

$$\boldsymbol{A}=\begin{bmatrix} 0.6357 & 0.0969 & 0.2674 \\ 0.0969 & 0.8751 & 0.0280 \\ 0.2674 & 0.0280 & 0.7047 \end{bmatrix}$$

with $\delta_2(\boldsymbol{A})=0.8212$, and $a=280$ based on $p=0.0178$.

Based on MATLAB, we run Algorithm 1, and we can obtain Figs. 1a–1c depicting the transient states of $X_k(i,j)$, $k,i,j\in\{1,2,3\}$, showing that Algorithm 1 is always globally convergent. We can obtain the optimal solution to problem (14):

$$\boldsymbol{X}^*=\begin{bmatrix} 0.52 & 0.23 & 0.43 \\ 0.29 & 0.14 & 0.29 \\ 0.30 & 0.14 & 0.64 \end{bmatrix},$$

and the optimal value is 147.21. Fig. 1d shows that the objective function value obtained by Algorithm 1 is the same as the optimal solution to problem (14). Therefore, the exact penalty method is valid, and Algorithm 1 can solve problem (14) without random samples.

Then we add random samples to problem (14), and other settings remain the same. We run Algorithm 1 by MATLAB and Fig. 2 is obtained. Fig. 2 depicts the transient objective function values of problem (14) with or without random samples by running Algorithm 1. Note that two trajectories are roughly the same; thus, Algorithm 1 can also solve problem (14) with random samples, which illustrates that Algorithm 1 can be used to solve matrix-valued distributed stochastic optimization problems.

Example 2 Consider a matrix-valued distributed stochastic optimization problem with more agents and higher dimensions ($N=10$ and $\boldsymbol{X}\in\mathbb{R}^{9\times 9}$) as follows:

$$\begin{aligned} &\min\ F(\boldsymbol{X})=\sum_{i=1}^{10}\|\boldsymbol{X}\boldsymbol{H}_i-\boldsymbol{B}_i\|_{\mathrm{F}}^2 \\ &\text{s.t. } \boldsymbol{Q}_1\boldsymbol{X}\boldsymbol{1}_9=0,\ \boldsymbol{Q}_2\boldsymbol{X}\boldsymbol{1}_9\le 0, \end{aligned} \tag{15}$$

where $\boldsymbol{H}_i$, $\boldsymbol{B}_i$, $\boldsymbol{Q}_1^{\mathrm{T}}$, and $\boldsymbol{Q}_2^{\mathrm{T}}$ are generated in $\prod_{i=1}^{9}[0,1]$ randomly.

We run Algorithm 1 to solve problem (15) without or with random samples, and then we obtain Fig. 3. Fig. 3 shows the errors between the transient

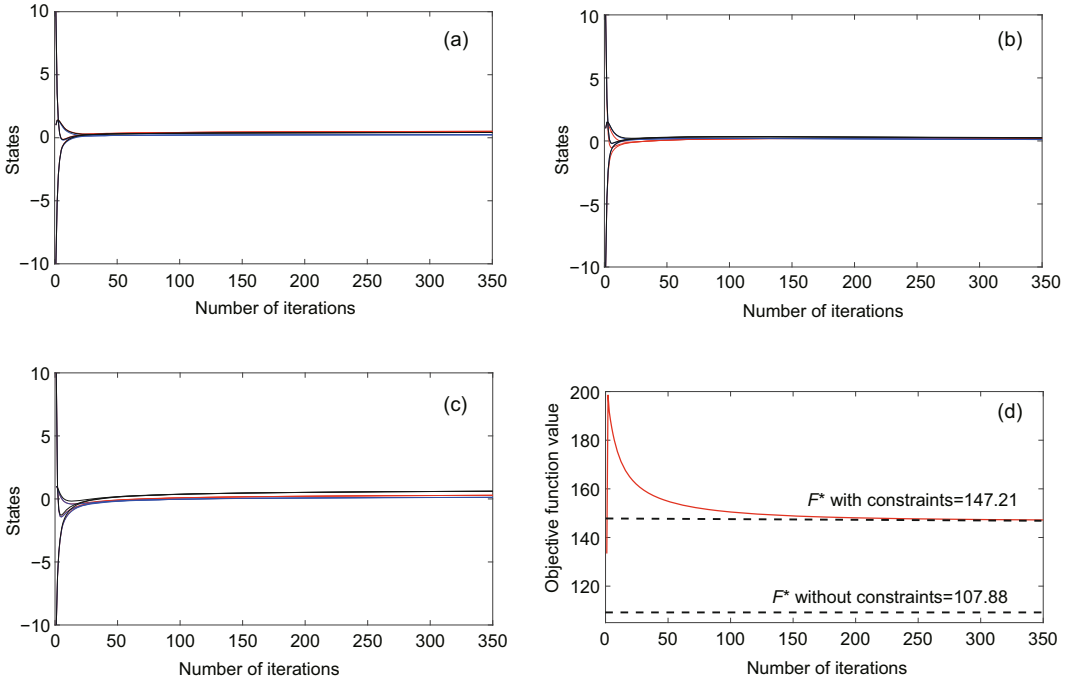

Fig. 1 Transient states of $X_k(1,i)$ (a), $X_k(2,i)$ (b), $X_k(3,i)$ (c), and the transient values of the objective function (d) in Example 1 (k, $i \in \{1,2,3\}$)

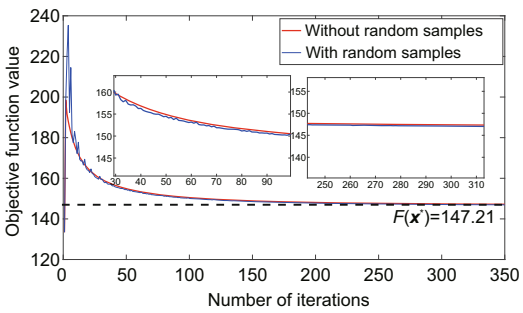

Fig. 2 Transient values of the objective function in Example 1

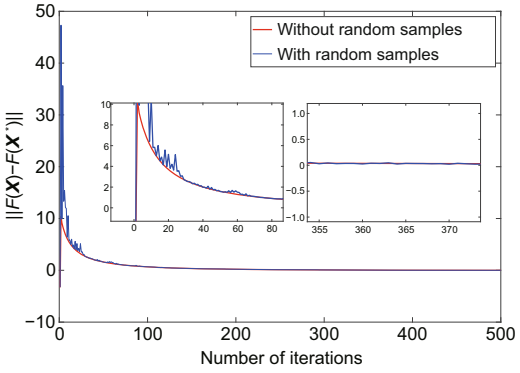

Fig. 3 Errors between the transient values of the objective function obtained by Algorithm 1 and the optimal values of the objective function in Example 2

values of the objective function obtained by Algorithm 1 and the optimal values of the objective function in Example 2. The error converges to 0, and two trajectories are roughly the same, which illustrates the validity of Algorithm 1.

6 Conclusions

In this paper, we have focused on a special constrained optimization called matrix-valued distributed stochastic optimization subject to inequality and equality constraints. We have adopted an exact penalty for the handling of the constraints. Based on a gossip model, we have developed a distributed stochastic gradient descent algorithm and analyzed its stability. Two illustrative examples have been provided to explain the validity of the exact penalty method and the optimization method.

Contributors

Zicong XIA, Yang LIU, and Wenlian LU designed the research. Zicong XIA processed the data. Zicong XIA and Yang LIU drafted the paper. Wenlian LU and Weihua GUI helped organize the paper. Yang LIU and Weihua GUI revised and finalized the paper.

Compliance with ethics guidelines

Yang LIU is a guest editor of this special feature, and he was not involved with the peer review process of this manuscript. Zicong XIA, Yang LIU, Wenlian LU, and Weihua GUI declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Bin SQ, Xia YS, 2014. Fast multi-channel image reconstruction using a novel two-dimensional algorithm. *Multimed Tools Appl*, 71(3):2015-2028. https://doi.org/10.1007/s11042-013-1371-6

Bouhamidi A, Jbilou K, 2012. A kronecker approximation with a convex constrained optimization method for blind image restoration. *Optim Lett*, 6(7):1251-1264. https://doi.org/10.1007/s11590-011-0370-7

Boyd S, Vandenberghe L, 2004. Convex Optimization. Cambridge University Press, Cambridge, UK.

Boyd S, Ghosh A, Prabhakar B, et al., 2006. Randomized gossip algorithms. *IEEE Trans Inform Theory*, 52(6):2508-2530. https://doi.org/10.1109/TIT.2006.874516

Deng ZH, Liang S, Hong YG, 2018. Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs. *IEEE Trans Cybern*, 48(11): 3116-3125. https://doi.org/10.1109/TCYB.2017.2759141

Huang LM, Xia YS, Huang LQ, et al., 2021. Two matrix-type projection neural networks for matrix-valued optimization with application to image restoration. *Neur Process Lett*, 53(3):1685-1707. https://doi.org/10.1007/s11063-019-10086-w

Jakovetic D, Xavier J, Moura JMF, 2011. Cooperative convex optimization in networked systems: augmented Lagrangian algorithms with directed gossip communication. *IEEE Trans Signal Process*, 59(8):3889-3902. https://doi.org/10.1109/TSP.2011.2146776

Jiang XR, Qin ST, Xue XP, 2021. Continuous-time algorithm for approximate distributed optimization with affine equality and convex inequality constraints. *IEEE Trans Syst Man Cybern Syst*, 51(9):5809-5818. https://doi.org/10.1109/TSMC.2019.2957156

Koloskova A, Stich SU, Jaggi M, 2019. Decentralized stochastic optimization and gossip algorithms with compressed communication. Proc 36th Int Conf on Machine Learning, p.3478-3487.

Li H, Fang C, Lin ZC, 2020. Accelerated first-order optimization algorithms for machine learning. *Proc IEEE*, 108(11):2067-2082. https://doi.org/10.1109/JPROC.2020.3007634

Li JF, Li W, Huang R, 2016. An efficient method for solving a matrix least squares problem over a matrix inequality constraint. *Comput Optim Appl*, 63(2):393-423. https://doi.org/10.1007/s10589-015-9783-z

Li XX, Xie LH, Hong YG, 2020. Distributed continuous-time nonsmooth convex optimization with coupled inequality constraints. *IEEE Trans Contr Netw Syst*, 7(1):74-84. https://doi.org/10.1109/TCNS.2019.2915626

Liang S, Zeng XL, Hong YG, 2018a. Distributed nonsmooth optimization with coupled inequality constraints via modified Lagrangian function. *IEEE Trans Autom Contr*, 63(6):1753-1759. https://doi.org/10.1109/TAC.2017.2752001

Liang S, Zeng XL, Hong YG, 2018b. Distributed sub-optimal resource allocation over weight-balanced graph via singular perturbation. *Automatica*, 95:222-228. https://doi.org/10.1016/j.automatica.2018.05.013

Liu QS, Wang J, 2013. A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints. *IEEE Trans Neur Netw Learn Syst*, 24(5):812-824. https://doi.org/10.1109/TNNLS.2013.2244908

Liu QS, Wang J, 2015. A second-order multi-agent network for bound-constrained distributed optimization. *IEEE Trans Autom Contr*, 60(12):3310-3315. https://doi.org/10.1109/TAC.2015.2416927

Liu QS, Yang SF, Wang J, 2017. A collective neurodynamic approach to distributed constrained optimization. *IEEE Trans Neur Netw Learn Syst*, 28(8):1747-1758. https://doi.org/10.1109/TNNLS.2016.2549566

Lu J, Tang CY, Regier PR, et al., 2011. Gossip algorithms for convex consensus optimization over networks. *IEEE Trans Autom Contr*, 56(12):2917-2923. https://doi.org/10.1109/TAC.2011.2160020

Lv YW, Yang GH, Shi CX, 2020. Differentially private distributed optimization for multi-agent systems via the augmented Lagrangian algorithm. *Inform Sci*, 538:39-53. https://doi.org/10.1016/j.ins.2020.05.119

Nedic A, Ozdaglar A, 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Contr*, 54(1):48-61. https://doi.org/10.1109/TAC.2008.2009515

Rakhlin A, Shamir O, Sridharan K, 2012. Making gradient descent optimal for strongly convex stochastic optimization. Proc 29th Int Conf on Machine Learning, p.1571-1578.

Ruszczyński A, 2006. Nonlinear Optimization. Princeton University Press, Princeton, USA.

Shamir O, Srebro N, 2014. Distributed stochastic optimization and learning. Proc 52nd Annual Allerton Conf on Communication, Control, and Computing, p.850-857. https://doi.org/10.1109/ALLERTON.2014.7028543

Shi XL, Cao JD, Wen GH, et al., 2019. Finite-time consensus of opinion dynamics and its applications to distributed optimization over digraph. *IEEE Trans Cybern*, 49(10):3767-3779. https://doi.org/10.1109/TCYB.2018.2850765

Wan P, Lemmon MD, 2009. Event-triggered distributed optimization in sensor networks. Proc Int Conf on Information Processing in Sensor Networks, p.49-60.

Wang D, Wang Z, Wen CY, 2021. Distributed optimal consensus control for a class of uncertain nonlinear multi-agent networks with disturbance rejection using adaptive technique. *IEEE Trans Syst Man Cybern Syst*, 51(7):4389-4399. https://doi.org/10.1109/TSMC.2019.2933005

Wang XY, Wang GD, Li SH, 2021. Distributed finite-time optimization for disturbed second-order multiagent systems. *IEEE Trans Cybern*, 51(9):4634-4647. https://doi.org/10.1109/TCYB.2020.2988490

Xia YS, Chen TP, Shan JJ, 2014. A novel iterative method for computing generalized inverse. *Neur Comput*, 26(2): 449-465. https://doi.org/10.1162/NECO_a_00549

Xia ZC, Liu Y, Lu JQ, et al., 2021. Penalty method for constrained distributed quaternion-variable optimization. *IEEE Trans Cybern*, 51(11):5631-5636. https://doi.org/10.1109/TCYB.2020.3031687

Xia ZC, Liu Y, Kou KI, et al., 2022. Clifford-valued distributed optimization based on recurrent neural networks. *IEEE Trans Neur Netw Learn Syst*, early access. https://doi.org/10.1109/TNNLS.2021.3139865

Xia ZC, Liu Y, Qiu JL, et al., 2023. An RNN-based algorithm for decentralized-partial-consensus constrained optimization. *IEEE Trans Neur Netw Learn Syst*, 34(1):534-542. https://doi.org/10.1109/TNNLS.2021.3098668

Xiao L, Boyd S, 2004. Fast linear iterations for distributed averaging. *Syst Contr Lett*, 53(1):65-78. https://doi.org/10.1016/j.sysconle.2004.02.022

Xing CW, Wang S, Chen S, et al., 2021. Matrix-monotonic optimization – part I: single-variable optimization. *IEEE Trans Signal Process*, 69:738-754. https://doi.org/10.1109/TSP.2020.3037513

Yang SF, Liu QS, Wang J, 2017. Distributed optimization based on a multiagent system in the presence of communication delays. *IEEE Trans Syst Man Cybern Syst*, 47(5):717-728. https://doi.org/10.1109/TSMC.2016.2531649

Yang T, Yi XL, Wu JF, et al., 2019. A survey of distributed optimization. *Annu Rev Contr*, 47:278-305. https://doi.org/10.1016/j.arcontrol.2019.05.006

Yuan DM, 2014. Gossip-based gradient-free method for multi-agent optimization: constant step size analysis. Proc 33rd Chinese Control Conf, p.1349-1353. https://doi.org/10.1109/ChiCC.2014.6896825

Yue DD, Baldi S, Cao JD, et al., 2022. Distributed adaptive optimization with weight-balancing. *IEEE Trans Autom Contr*, 67(4):2068-2075. https://doi.org/10.1109/TAC.2021.3071651

Zeng XL, Yi P, Hong YG, 2017. Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach. *IEEE Trans Autom Contr*, 62(10):5227-5233. https://doi.org/10.1109/TAC.2016.2628807

Zeng XL, Chen J, Hong YG, 2022. Distributed optimization design of iterative refinement technique for algebraic Riccati equations. *IEEE Trans Syst Man Cybern Syst*, 52(5):2833-2847. https://doi.org/10.1109/TSMC.2021.3056871

Zhang SC, Xia YH, Xia YS, et al., 2022. Matrix-form neural networks for complex-variable basis pursuit problem with application to sparse signal reconstruction. *IEEE Trans Cybern*, 52(7):7049-7059. https://doi.org/10.1109/TCYB.2020.3042519

Zhou HB, Zeng XL, Hong YG, 2017. An exact penalty method for constrained distributed optimization. Proc 36th Chinese Control Conf, p.8827-8832. https://doi.org/10.23919/ChiCC.2017.8028760

Zhou HB, Zeng XL, Hong YG, 2019. Adaptive exact penalty design for constrained distributed optimization. *IEEE Trans Autom Contr*, 64(11):4661-4667. https://doi.org/10.1109/TAC.2019.2902612

Zhu MH, Martínez S, 2012. On distributed convex optimization under inequality and equality constraints. *IEEE Trans Autom Contr*, 57(1):151-164. https://doi.org/10.1109/TAC.2011.2167817

Zhu ZH, Li QW, Tang GG, et al., 2021. The global optimization geometry of low-rank matrix optimization. *IEEE Trans Inform Theory*, 67(2):1308-1331. https://doi.org/10.1109/TIT.2021.3049171