Research Article

A Distributed Conjugate Gradient Online Learning Method over Networks

Cuixia Xu,^{1,2} Junlong Zhu ,¹ Youlin Shang,² and Qingtao Wu ¹

¹School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

²School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang 471023, China

Correspondence should be addressed to Junlong Zhu; jlzhu@haust.edu.cn

Received 4 November 2019; Revised 1 January 2020; Accepted 10 February 2020; Published 11 March 2020

Academic Editor: Diego R. Amancio

Copyright © 2020 Cuixia Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In a distributed online optimization problem with a convex constrained set over an undirected multiagent network, the local objective functions are convex and vary over time. Most of the existing methods used to solve this problem are based on the fastest gradient descent method. However, the convergence speed of these methods is decreased with an increase in the number of iterations. To accelerate the convergence speed of the algorithm, we present a distributed online conjugate gradient algorithm, different from a gradient method, in which the search directions are a set of vectors that are conjugated to each other and the step sizes are obtained through an accurate line search. We analyzed the convergence of the algorithm theoretically and obtained a regret bound of $O(\sqrt{T})$, where T is the number of iterations. Finally, numerical experiments conducted on a sensor network demonstrate the performance of the proposed algorithm.

1. Introduction

Distributed optimization has received considerable interest in science and engineering, which can be applied in numerous fields such as distributed tracking and localization [1], multiagent coordination [2], distributed estimations using sensor networks [3–5], and machine learning [6]. Such problems can be modeled to minimize or maximize the summation of some of the local convex functions, and these local functions only use a local computation and communication in a distributed manner. With an increase in the network size and data volume, more effective distributed algorithms have become a hot research topic. In recent years, many scholars have proposed various distributed optimization algorithms to solve such problems [7–15].

Most of the existing algorithms assume that the cost function at each agent is fixed. However, in practical problems, the environment of an agent is uncertain and the cost function of each agent changes over time, requiring us to solve such problems through an online setting. To be more precise, in a distributed online optimization, the cost function of each agent changes during each step, and with t

iterations, before a decision is given, the cost function for each agent is unknown; only when we obtain a decision from a constrained set, we can obtain the information of the cost function. In addition, we also obtain a loss at the same time. Such loss reflects the error in the cost of the objective function between the current decision point and the best decision in hindsight, which we call regret. Regret is an important criterion in evaluating a distributed online algorithm. A well-performing distributed online optimization algorithm should decrease the average total regret approach to zero over time.

Because an online distributed optimization algorithm is more consistent with that used for practical problems, many scholars have conducted numerous studies and some effective algorithms have been proposed [16–24]. Yan et al. [20] introduced a distributed autonomous online learning algorithm, namely, a subgradient descent method using a projection. When the local objective functions are strongly convex and convex, the regret bounds of the proposed algorithm are obtained, respectively. The authors in [22] introduced an online distributed push-sum algorithm in which the search direction is a negative subgradient in each

iteration, which achieves regret $O((\log(T))^2)$ when the local function is strongly convex. For a time-varying directed network problem, Zhu et al. [25, 26] proposed a distributed online optimization algorithm. During each iteration, the negative subgradient is randomly selected as the search direction. The authors in [27] presented a distributed online algorithm based on a primal-dual dynamic mirror descent for a problem with time-varying coupling inequality constraints and obtained a dynamic regret bound. The authors in [28] proposed a distributed online conditional gradient algorithm for a constrained distributed online optimization problem in the Internet of Things.

The existing distributed online optimization algorithm based on the gradient method is simple to calculate and requires little storage; however, to ensure the convergence of the algorithm, the iterative step length usually needs to decrease with an increase in the number of iterations, which will lead to a zigzag path at the end of the algorithm. That is, the algorithm will carry out multiple iterations in the same direction or approximate direction, which greatly increases the computational time of the algorithm. The conjugate gradient algorithm also has the advantages of simple calculations and guaranteed convergence under certain conditions [29–31] but differs from the gradient method in that the search direction of the conjugate gradient algorithm is a group of conjugate or approximately conjugate vectors, and during the later stage of the algorithm, there are no additional repeated iterations in the same or approximate direction. Thus, the convergence of the conjugate gradient method is generally faster than that of the gradient descent method. In particular, for an objective functional quadratic, the conjugate gradient method has a quadratic termination. Based on these advantages, the conjugate gradient method has been used to solve numerous centralized offline optimization problems [32–35]. According to the existing literature, however, the conjugate gradient method has not been applied to distributed online optimization problems. To fill in this gap, herein, we present a distributed online conjugate gradient algorithm.

There are two main contributions provided by the present study. First, a new algorithm for a distributed online constrained convex optimization problem, namely, a distributed online conjugate gradient algorithm, is proposed. In our algorithm, a set of conjugate directions is used to replace the gradient directions used in a traditional gradient descent method, and the step size is obtained through an accurate linear search, thus effectively avoiding the slow convergence speed of a traditional gradient descent algorithm during the later stage. Second, we provide a careful analysis of the convergence of the proposed algorithm and obtain the square root of the regret bound.

The remainder of this paper is organized as follows: in Section 2, we first briefly introduce the distributed online optimization model, followed by some necessary mathematical preliminaries and assumptions used in this study. We also provide a detailed statement of our algorithm in Section 3 and an analysis of the convergence of the algorithm in Section 4. The simulation results of our algorithm are then presented in Section 5. Finally, we provide some concluding

remarks in Section 6. In addition, further detailed proofs of some of the lemmas applied can be found in the Appendix.

2. Preliminaries

In this section, we provide a brief background on the distributed online optimization and the conjugate gradient method. At the same time, some constructs used in this study and some relevant assumptions regarding our analysis are provided.

2.1. Distributed Online Optimization. Consider a network system with multiple agents; in this network, each agent i is associated with a convex function $f_{ti}(x): \mathbb{R}^n \rightarrow \mathbb{R}$. All agents aim to solve the following general consensus problem cooperatively:

$$\begin{aligned} \min \quad & \sum_{i=1}^n f_{ti}(x) \\ \text{subject to} \quad & x \in \mathcal{X}. \end{aligned} \quad (1)$$

During each round $t \in \{1, \dots, T\}$, the i th agent is required to generate a decision point $x_i(t)$ from a convex compact set $\mathcal{X}$. Then, the adversary replies to each agent's decision with a cost function $f_{ti}(x): \mathcal{X} \rightarrow \mathbb{R}$, and each agent has a loss of $f_{ti}(x_i(t))$ simultaneously. The communication between agents is specified by a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. Each agent i can only communicate with its immediate neighbors $\mathcal{N}(i) = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. The goal of the agents is to seek a sequence of decision points $\{x_i(t)\}, i \in \mathcal{V}$ such that the cumulative regret with respect to each agent i regarding any fixed decision $x^* \in \mathcal{X}$ in hindsight

$$R_T(x_i(t), x^*) = \sum_{t=1}^T (f_{ti}(x_i(t)) - f_{ti}(x^*)), \quad (2)$$

is sublinear in T , that is, $\lim_{T \rightarrow \infty} R_T/T = 0$.

2.2. Conjugate Gradient Method. For the following optimization problem,

$$\min_{x \in \mathbb{R}^n} f(x), \quad (3)$$

where $f(x)$ is a quadratic continuous differentiable. The iterative form of the conjugate gradient (CG) method is usually designed as

$$x(k+1) = x(k) + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (4)$$

where $x(k)$ is the point from the k th iteration, $\alpha_k > 0$ is the step length, and the search direction d_k is defined as

$$d_k = \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k \geq 1. \end{cases} \quad (5)$$

in which, g_k is the gradient of the objective function at the current iterate point $x(k)$, $\beta_k \in \mathbb{R}$ is a scalar, and the different definitions of β_k represent different methods of a conjugate gradient [27]. Well-known conjugate gradient methods include the Polak–Ribiere–Polyak (PRP) method and the

Fletcher–Reeves (FR) method. In this study, we define the parameter β_k using the PRP method, the specific form of which is as follows:

$$\beta_k^{\text{PRP}} = \frac{(g_k - g_{k-1})^T g_k}{g_{k-1}^T g_{k-1}}, \quad (6)$$

$$\beta_k = \max(0, \beta_k^{\text{PRP}}).$$

Gilbert and Nocedal [36] proved that if the parameter β_k is appropriately bounded in magnitude, the CG method can converge globally. Therefore, the CG method satisfies the sufficient descent condition under this hypothesis.

To analyze the convergence of our algorithm, we provide the bound of the conjugate gradient as follows.

Lemma 1 (see [37]). *Let $f(x)$ be a quadratic continuous difference convex function, and $\nabla^2 f(x)$ be a Hessian matrix of the function. For any $\hat{x} \in \mathbb{R}^n$, when $x \in C = \{x \mid f(x) \leq f(\hat{x}), x \in \mathbb{R}^n\}$ and $y \in \mathbb{R}^n$, there exist two positive numbers m and M such that*

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2. \quad (7)$$

Taking an initial point $x_{(1)} \in C$, where x_k, d_k , and β_k are all defined using the PRP method,

$$\|g_k\| \leq \|d_k\| \leq \frac{m+M}{m} \|g_k\|, \quad (8)$$

in which $g_k = \nabla f(x_k)$.

2.3. Some Constructs and Assumptions. The following assumptions are given throughout this paper:

- (i) Each cost function $f_{ii}(x)$ is a convex and twice continuous differentiable L -Lipschitz on the convex set $\mathcal{X}$.
- (ii) The set $\mathcal{X}$ is compact and convex, and $\mathbf{0} \in \mathcal{X}$, $\mathbf{0}$ denotes a vector with all entries equal to zero.
- (iii) The Euclidean diameter of $\mathcal{X}$ is bounded by R .

As the Lipschitz condition in (i) implies, for any $x \in \mathcal{X}$ and any gradient g_i , we have the following:

$$\|g_i\|_* \leq L, \quad (9)$$

Where $\|\cdot\|_* = \sup_{\|u\| \leq 1} \langle \cdot, u \rangle$ denotes the dual norm.

The next definition is used throughout this paper.

Definition 1 (see [38]). Let $f(x)$ be a function difference on an open set $C \subseteq \mathbb{R}^n$, and let $\mathcal{X}$ be a convex subset of C . Then, $f(x)$ is convex on $\mathcal{X}$ if and only if

$$f(x) \geq f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle, \quad (10)$$

for all $(x_0, x) \in \mathcal{X} \times \mathcal{X}$.

Now, we give an important inequality in [39] that is often used in optimization problems.

Let $f(x)$ be a first-order continuous differentiable function on the set $\mathbb{R}^n$, whose first derivative satisfies the Lipschitz condition, and thus $\forall x, y \in \mathbb{R}^n$,

$$|f(y) - f(x) - \langle f'(x), y - x \rangle| \leq \frac{L}{2} \|y - x\|^2, \quad (11)$$

where L is the Lipschitz constant and $\|\cdot\|$ denotes the European norm.

3. Distributed Online Conjugate Gradient Algorithm

For the distributed online optimization problem (1), each locally cost function $f_{ii}(x)$ satisfies the assumptions in Section 2. The network topology relationship among agents is specified by an undirected graph $G = (\mathcal{V}, \mathcal{E})$, that is, if $(i, j) \in \mathcal{E}$, it implies that $(j, i) \in \mathcal{E}$. Each agent i can only communicate with its immediate neighbors. The adjacency matrix of the undirected graph is a doubly stochastic symmetric $P = (p_{ij})_{n \times n}$ such that $p_{ij} \geq 0$ only if $(i, j) \in \mathcal{E}$; otherwise, $p_{ij} = 0$ and $\sum_{j \in \mathcal{N}(i)} p_{ij} = \sum_{j=1}^n p_{ij} = 1$ for all $i \in \mathcal{V}$ and $\sum_{i \in \mathcal{N}(j)} p_{ij} = \sum_{i=1}^n p_{ij} = 1$ for all $j \in \mathcal{V}$.

To solve (1), we present a distributed online conjugate gradient algorithm. After giving a decision $x_i(t) \in \mathcal{X}$ based on the current information, we can obtain the cost function $f_{ii}(x)$ and compute the gradient $g_i(t) = \nabla f_{ii}(x(t))$. We can then calculate the value of $\beta_i(t)$ using the gradients in the current iteration point $x_i(t)$ and the previous iteration point $x_i(t-1)$. If $\beta_i(t) > 0$, a new search direction $d_i(t)$, computed using a Gram–Schmidt conjugate of the gradients in the current iteration point $x_i(t)$ and the previous search direction $d_i(t-1)$, can be constructed. If the parameter is $\beta_i(t) \leq 0$, we then obtain the new search direction $d_i(t) = -g_i(t)$, which is equivalent to restarting the distributed online conjugate gradient algorithm in the direction of the steepest descent. The iteration step length $\alpha_i(t)$ can be obtained through an exact line search, and the next iteration point $x_i(t+1)$ can be obtained using the conjugate direction vector $d_i(t)$ and step $\alpha_i(t)$. The specific algorithm is summarized in Algorithm 1.

Here, we define the projection function used in this algorithm as follows:

$$\prod_{\mathcal{X}}^{\varphi(x)}(u, \alpha(t)) = \arg \min_{x \in \mathcal{X}} \left\{ \langle u, x \rangle + \frac{1}{\alpha(t)} \varphi(x) \right\}. \quad (12)$$

4. Regret Bound Analysis

To analyze the regret bound for D-OCG, we provide some preliminary remarks and a few definitions. Using Algorithm 1, we can determine the following:

$$z_i(t+1) = \sum_{j=1}^n p_{ij} z_j(t) - d_i(t), \quad (13)$$

$$x_i(t+1) = \prod_{\mathcal{X}}^{\varphi(x)}(-z_i(t+1), \alpha_i(t)).$$

Now, we define

```

(1) Input: convex set  $\mathcal{X}$ , maximum round number  $T$ 
(2) Initialize:  $x_i(1) \in \mathcal{X}$ ,  $z_i(1) = 0$ ,  $\forall i \in V$ 
(3) for  $i = 1, \dots, T$  do
(4)   The adversary reveals  $f_{ti}$ ,  $\forall i \in V$ 
(5)   Compute the gradients  $g_i(t) \in \partial f_{ti}(x_i(t))$ ,  $\forall i \in V$ 
(6)   Compute:
        $\beta_i(t)^{\text{PRP}} = (g_i(t)^T (g_i(t) - g_i(t-1))) / (\|g_i(t-1)\|^2)$ ,
        $\beta_i(t) = \max\{0, \beta_i(t)^{\text{PRP}}\}$ ,
        $d_i(t) = \begin{cases} -g_i(1), & t = 1 \\ -g_i(t) + \beta_i(t)d_i(t-1), & t > 1 \end{cases}$ 
(7)    $\lambda_i(t) = \arg \min_{\lambda \geq 0} f_{ti}(x_i(t) + (1/(2\sqrt{t}))\lambda d_i(t))$ 
        $\alpha_i(t) = \lambda_i(t)/\sqrt{t}$ ,  $\forall i \in V$ 
(8)   for each  $i \in V$  do
(9)      $z_i(t+1) = \sum_{j=1}^n p_{ij} z_j(t) - d_i(t)$ 
(10)     $x_i(t+1) = \prod_{\mathcal{X}}^{\varphi(x)} (-z_i(t+1), \alpha_i(t))$ 
(11)   end for
(12) end for

```

ALGORITHM 1: Distributed Online Conjugate Gradient Algorithm (D-OCG).

$$\begin{aligned}
\bar{z}(t) &= \frac{1}{n} \sum_{i=1}^n z_i(t), \\
\bar{d}(t) &= \frac{1}{n} \sum_{i=1}^n d_i(t), \\
y(t) &= \prod_{\mathcal{X}}^{\varphi(x)} (-\bar{z}(t+1), \alpha(t)),
\end{aligned} \tag{14}$$

and from the evolution of $z_i(t+1)$, we can obtain

$$\bar{z}(t+1) = \bar{z}(t) - \bar{d}(t). \tag{15}$$

Now, the main results in our paper can be stated.

Theorem 1. *The sequences of $x_i(t)$ and $z_i(t)$ generated by Algorithm 1 are given for all $i \in \mathcal{V}$, where $\varphi(x) = (1/2)\|x\|^2$, $\varphi(x^*) \leq R^2$, and $\alpha_i(t) = (1/\sqrt{t})\lambda_i(t)$, and we thus have the cumulative regret owing to the action of agent i s*

$$\begin{aligned}
R_T(x^*, x_i(t)) &= \sum_{t=1}^T (f_t(x_i(t)) - f_t(x^*)) \\
&\leq \left[\frac{(n+2)(M+m)}{m(1-\sigma_2(P))\sqrt{n}} + \frac{(b+1)(M+m)^2}{2m^2} \right] \lambda L^2 \sqrt{T} \\
&\quad + \frac{1}{\lambda} (b+1)R^2 \sqrt{T} + bD,
\end{aligned} \tag{16}$$

where $\lambda = \max_{1 \leq i \leq n, 1 \leq t \leq T} \{\lambda_i(t)\}$, b and D are two nonnegative constants, M and m are as defined in Lemma 1, n is the number of agents, and $\sigma_2(P)$ is the second largest eigenvalue of the adjacency matrix P .

From Theorem 1, we obtain a regret bound of the proposed algorithm under the local convexity, which is sublinear to T , i.e., the regret bound of the D-OCG algorithm can approach zero as the value of T increases, where T is the number of iterations. It is evident that the

value of the regret bound is related to the upper bound L of the gradient of the local objective functions and the diameter R of the constraint set $\mathcal{X}$. By Lemma 1, we know that the regret bound is also related to the Hessian matrix of the local objective functions. Moreover, the value of the regret bound is also related to the scale and topology of the network.

To prove Theorem 1, we now present the following lemmas.

Lemma 2. *For any $i \in \mathcal{V}$ and $x^* \in \mathcal{X}$, we can obtain the following inequality:*

$$\begin{aligned}
\sum_{t=1}^T (f_t(x_i(t)) - f_t(x^*)) &\leq \left(L + \frac{L}{n} \right) \sum_{t=1}^T \|x_i(t) - y(t)\| \\
&\quad + \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n (f_t(x_i(t)) - f_t(x^*)).
\end{aligned} \tag{17}$$

Proof. Based on assumption (i), the function $f_t(x_i(t))$ is L -Lipschitz continuous on the convex set $\mathcal{X}$, that is,

$$|f_t(x_i(t)) - f_t(y(t))| \leq L \|x_i(t) - y(t)\|, \tag{18}$$

and thus

$$\begin{aligned}
&\sum_{t=1}^T (f_t(x_i(t)) - f_t(x^*)) \\
&= \sum_{t=1}^T [f_t(x_i(t)) - f_t(y(t)) + f_t(y(t)) - f_t(x^*)] \\
&\leq L \sum_{t=1}^T \|x_i(t) - y(t)\| + \sum_{t=1}^T (f_t(y(t)) - f_t(x^*)).
\end{aligned} \tag{19}$$

By contrast,

$$\begin{aligned}
& \sum_{t=1}^T (f_t(y(t)) - f_t(x^*)) \\
&= \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n [(f_t(y(t))) - f_t(x_i(t)) + (f_t(x_i(t)) - f_t(x^*))] \\
&\leq \frac{L}{n} \sum_{t=1}^T \sum_{i=1}^n \|x_i(t) - y(t)\| + \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n (f_t(x_i(t)) - f_t(x^*)).
\end{aligned} \tag{20}$$

Combining equations (19) and (20), the proof of Lemma 2 is completed.

Now, we prove that the last term of inequality (20) has a particular bound. $\square$

Lemma 3. For any $i \in \mathcal{V}$ and $x^* \in \mathcal{X}$,

$$\begin{aligned}
& \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n (f_t(x_i(t)) - f_t(x^*)) \\
&\leq \frac{L}{n} \sum_{t=1}^T \sum_{i=1}^n \|x_i(t) - y(t)\| + \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), y(t) - x^* \rangle.
\end{aligned} \tag{21}$$

Proof. Because $g_i(t) = \nabla f_t(x_i(t))$ is a gradient of $f_t(x)$ at $x_i(t)$, using the convexity for function $f_t(x)$, we have

$$f_t(x_i(t)) - f_t(x^*) \leq \langle g_i(t), x_i(t) - x^* \rangle. \tag{22}$$

Then, based on assumption (i), we know that $\|g_i(t)\| \leq L$. We can then obtain the following:

$$\begin{aligned}
\langle g_i(t), x_i(t) - x^* \rangle &= \langle g_i(t), x_i(t) - y(t) \rangle + \langle g_i(t), y(t) - x^* \rangle \\
&\leq L \|x_i(t) - y(t)\| + \langle g_i(t), y(t) - x^* \rangle.
\end{aligned} \tag{23}$$

Summing for $t = 1, \dots, T$ for the average of $\langle g_i(t), x_i(t) - x^* \rangle$, the following is obtained:

$$\begin{aligned}
& \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), x_i(t) - x^* \rangle \\
&= \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), x_i(t) - y(t) + y(t) - x^* \rangle \\
&\leq \frac{L}{n} \sum_{t=1}^T \sum_{i=1}^n \|x_i(t) - y(t)\| + \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), y(t) - x^* \rangle.
\end{aligned} \tag{24}$$

The proof of Lemma 3 is completed.

Now, we turn our attention to the following term:

$$\sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), y(t) - x^* \rangle. \tag{25}$$

According to the definition of the conjugate gradient, we give the bound of equation (25) in Lemma 4. $\square$

Lemma 4. For any $i \in \mathcal{V}$ and $\beta_i(t) \leq b$ (where b is a non-negative constant), the following bound holds:

$$\begin{aligned}
& \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), y(t) - x^* \rangle \\
&\leq (b+1) \left[\frac{1}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t-1)\|_*^2 + \frac{1}{\alpha(T)} \varphi(x^*) \right] \\
&\quad + b \varphi_{\alpha(T)}^*(\bar{z}(T)).
\end{aligned} \tag{26}$$

Proof. Based on the definitions of $d_i(t)$ and $\bar{d}(t)$, the left-hand side of the above inequality can be split into two:

$$\begin{aligned}
& \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i(t), y(t) - x^* \rangle \\
&= \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle -d_i(t) + \beta_i(t) d_i(t-1), y(t) - x^* \rangle \\
&= \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle -d_i(t), y(t) - x^* \rangle \\
&\quad + \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \\
&= \sum_{t=1}^T \langle -\bar{d}(t), y(t) - x^* \rangle \\
&\quad + \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle.
\end{aligned} \tag{27}$$

Thus, we prove that the first term in equation (27) has a bound. For any function $f(x)$, we know that

$$f(x) \leq \sup_{x \in \text{dom} f} f(x), \tag{28}$$

where $\text{dom} f$ is the domain of the function $f(x)$. Therefore, for the function

$$G(x) = \sum_{t=1}^T \langle \bar{d}(t), x \rangle - \frac{1}{\alpha(T)} \varphi(x), \quad x \in \mathcal{X}, \tag{29}$$

we can obtain for any $x^* \in \mathcal{X}$,

$$G(x^*) \leq \sup_{x \in \mathcal{X}} G(x), \tag{30}$$

that is,

$$\sum_{t=1}^T \langle \bar{d}(t), x^* \rangle - \frac{1}{\alpha(T)} \varphi(x^*) \leq \sup_{x \in \mathcal{X}} \left\{ \sum_{t=1}^T \langle \bar{d}(t), x \rangle - \frac{1}{\alpha(T)} \varphi(x) \right\}, \tag{31}$$

so

$$\sum_{t=1}^T \langle \bar{d}(t), x^* \rangle \leq \sup_{x \in \mathcal{X}} \left\{ \sum_{t=1}^T \langle \bar{d}(t), x \rangle - \frac{1}{\alpha(T)} \varphi(x) \right\} + \frac{1}{\alpha(T)} \varphi(x^*). \quad (32)$$

Based on the definition of the conjugate function [40] and the updates for $\bar{z}(t)$, we have the following:

$$\begin{aligned} \sum_{t=1}^T \langle \bar{d}(t), x^* \rangle &\leq \varphi_{\alpha(T)}^* \left(\sum_{t=1}^T (\bar{d}(t)) \right) + \frac{1}{\alpha(T)} \varphi(x^*) \\ &\leq \varphi_{\alpha(T)}^* (-\bar{z}(T+1)) + \frac{1}{\alpha(T)} \varphi(x^*). \end{aligned} \quad (33)$$

Because $\alpha(t)$ is a nonincreasing sequence, based on the definition of the conjugate function $\varphi_{\alpha}^*(z)$, we can obtain for all $z \in \mathbb{R}^d$,

$$\varphi_{\alpha(t)}^*(z) \leq \varphi_{\alpha(t-1)}^*(z), \quad (34)$$

and thus we obtain the following:

$$\varphi_{\alpha(t)}^*(-\bar{z}(t+1)) \leq \varphi_{\alpha(t-1)}^*(-\bar{z}(t) + \bar{d}(t)). \quad (35)$$

According to the inequality (11), we know that

$$\begin{aligned} \varphi_{\alpha(t-1)}^*(-\bar{z}(t) + \bar{d}(t)) &\leq \varphi_{\alpha(t-1)}^*(-\bar{z}(t)) + \langle \bar{d}(t), x \rangle \\ &\quad + \frac{1}{2} \alpha(t-1) \|\bar{d}(t)\|_*^2. \end{aligned} \quad (36)$$

A detailed proof of equation (36) is provided in Appendix A. The following inequality is then established:

$$\begin{aligned} \langle -\bar{d}(t), x \rangle &\leq \varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t)}^*(-\bar{z}(t+1)) \\ &\quad + \frac{1}{2} \alpha(t-1) \|\bar{d}(t)\|_*^2. \end{aligned} \quad (37)$$

Summing both sides of the above inequation from $t = 1$ to T , we obtain the following:

$$\begin{aligned} \sum_{t=1}^T \langle -\bar{d}(t), x \rangle &\leq \sum_{t=1}^T [\varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t)}^*(-\bar{z}(t+1))] \\ &\quad + \frac{1}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t)\|_*^2. \end{aligned} \quad (38)$$

Through equations (33) and (38), we can write the following:

$$\begin{aligned} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle -d_i(t), y(t) - x^* \rangle &= \sum_{t=1}^T \langle -\bar{d}(t), y(t) - x^* \rangle \\ &= \sum_{t=1}^T \langle -\bar{d}(t), y(t) \rangle + \sum_{t=1}^T \langle \bar{d}(t), x^* \rangle \\ &\leq \sum_{t=1}^T [\varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t)}^*(-\bar{z}(t+1))] \\ &\quad + \frac{1}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t)\|_*^2 + \varphi_{\alpha(T)}^*(-\bar{z}(T+1)) + \frac{1}{\alpha(T)} \varphi(x^*) \\ &\leq \frac{1}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t)\|_*^2 + \frac{1}{\alpha(T)} \varphi(x^*). \end{aligned} \quad (39)$$

We then analyze the bound on the second term in inequality (27). Because $\beta_i(t) = \max_{1 \leq i \leq n} \{0, \beta_i(t)^{\text{PRP}}\}$, and $\beta_i(t) \leq b$, we then analyze the following two situations. $\square$

Case 1. If

$$\frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \leq 0, \quad (40)$$

then,

$$\sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \leq 0. \quad (41)$$

The conclusion therefore clearly holds.

Case 2. If

$$\frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle > 0, \quad (42)$$

without loss of generality, we assume that for all $i \in \mathcal{V}$,

$$\langle d_i(t-1), y(t) - x^* \rangle > 0, \quad (43)$$

so

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \\ &\leq \frac{1}{n} \sum_{i=1}^n b \langle d_i(t-1), y(t) - x^* \rangle \\ &\leq b \langle \frac{1}{n} \sum_{i=1}^n d_i(t-1), y(t) - x^* \rangle. \end{aligned} \quad (44)$$

That is,

$$\frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \leq b \langle \bar{d}(t-1), y(t) - x^* \rangle. \quad (45)$$

Summing for $t = 1, 2, \dots, T$,

$$\begin{aligned} & \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \\ & \leq b \sum_{t=1}^T \langle \bar{d}(t-1), y(t) - x^* \rangle. \end{aligned} \quad (46)$$

Next, we give the bound of $\sum_{t=1}^T \langle \bar{d}(t-1), y(t) - x^* \rangle$. Because

$$\begin{aligned} & \sum_{t=1}^T \langle \bar{d}(t-1), y(t) - x^* \rangle \\ & = \sum_{t=1}^T \langle \bar{d}(t-1), y(t) \rangle + \sum_{t=1}^T \langle -\bar{d}(t-1), x^* \rangle, \end{aligned} \quad (47)$$

similar to the proof of inequality (33), we can obtain the following:

$$\sum_{t=1}^T \langle -\bar{d}(t-1), x^* \rangle \leq \varphi_{\alpha(T)}^*(\bar{z}(T)) + \frac{1}{\alpha(T)} \varphi(x^*). \quad (48)$$

Through inequality (11), we obtain the following:

$$\begin{aligned} \langle \bar{d}(t-1), x \rangle & \leq \varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t-1)}^*(-\bar{z}(t-1)) \\ & \quad + \frac{\alpha}{2} \|\bar{d}(t-1)\|_*^2. \end{aligned} \quad (49)$$

In addition to the definition of function $\varphi_{\alpha(t-1)}^*(z)$, when $z = \mathbf{0}$,

$$\begin{aligned} \varphi_{\alpha(t-1)}^*(\mathbf{0}) & = \sup_{x \in \mathcal{X}} \left\{ -\frac{1}{\alpha(t-1)} \varphi(x) \right\} \\ & = \sup_x \left\{ -\frac{1}{\alpha(t-1)} \varphi(x) - \delta(x \in \mathcal{X}) \right\}. \end{aligned} \quad (50)$$

The definition of the projection shows that the supremum of the above equation is uniquely attained through $\prod_{\mathcal{X}}^{\varphi}(\mathbf{0}, \alpha(t-1))$. In addition, for all fixed z , when $x = \mathbf{0}$, $\{\langle z, x \rangle - (1/\alpha(t-1))\varphi(x) - \delta(x \in \mathcal{X})\} = 0$, and thus we can obtain the supremum of $\varphi_{\alpha(t-1)}^*(\mathbf{0})$ to the set.

$$\begin{aligned} & \left\{ x \mid \frac{1}{\alpha(t-1)} \varphi(x) + \delta(x \in \mathcal{X}) - \langle z, x \rangle \leq 0 \right\} \\ & = \mathcal{X} \cap \left\{ x \mid \frac{1}{\alpha(t-1)} \varphi(x) - \langle z, x \rangle \leq 0 \right\}. \end{aligned} \quad (51)$$

Because the set $\mathcal{X}$ is closed and $\varphi(x)$ is strongly convex (for the definition of strongly convex, see [40]), the set described above is compact. By contrast, we know that $\langle z, x \rangle$ is differentiable in z , and the supremum is unique, and thus we can obtain the following: $\nabla \varphi_{\alpha(t-1)}^*(\mathbf{0}) = \prod_{\mathcal{X}}^{\varphi}(\mathbf{0}, \alpha(t-1)) = \arg \min_{x \in \mathcal{X}} \{(1/\alpha(t-1))\varphi(x)\}$.

Then, we derive the next two equations through Taylor expansion:

$$\begin{aligned} \varphi_{\alpha(t-1)}^*(-\bar{z}(t)) & \approx \varphi_{\alpha(t-1)}^*(\mathbf{0}) + \nabla \varphi_{\alpha(t-1)}^*(\mathbf{0})(-\bar{z}(t)), \\ \varphi_{\alpha(t-1)}^*(-\bar{z}(t-1)) & \approx \varphi_{\alpha(t-1)}^*(\mathbf{0}) + \nabla \varphi_{\alpha(t-1)}^*(\mathbf{0})(-\bar{z}(t-1)), \end{aligned} \quad (52)$$

and thus

$$\begin{aligned} & \varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t-1)}^*(-\bar{z}(t-1)) \\ & = \nabla \varphi_{\alpha(t-1)}^*(\mathbf{0})(-\bar{z}(t) - (-\bar{z}(t-1))) \\ & \leq \|\nabla \varphi_{\alpha(t-1)}^*(\mathbf{0})\| \|\bar{z}(t) - \bar{z}(t-1)\| \end{aligned} \quad (53)$$

$$\leq \left\| \arg \min_{x \in \mathcal{X}} \left(\frac{1}{\alpha(t-1)} \varphi(x) \right) \right\| \|\bar{z}(t) - \bar{z}(t-1)\|.$$

Because $\mathbf{0} \in \mathcal{X}$, $\varphi(x) = (1/2)\|x\|^2$,

$$\left\| \arg \min_{x \in \mathcal{X}} \left(\frac{1}{\alpha(t-1)} \varphi(x) \right) \right\| = 0, \quad (54)$$

and therefore

$$\varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t-1)}^*(-\bar{z}(t-1)) \leq 0. \quad (55)$$

Thus,

$$\begin{aligned} \langle \bar{d}(t-1), x \rangle & \leq \varphi_{\alpha(t-1)}^*(-\bar{z}(t)) - \varphi_{\alpha(t-1)}^*(-\bar{z}(t-1)) \\ & \quad + \frac{\alpha(t-1)}{2} \|\bar{d}(t-1)\|_*^2 \\ & \leq \frac{\alpha}{2} \|\bar{d}(t-1)\|_*^2. \end{aligned} \quad (56)$$

Summing both sides of the above inequation from $t = 1$ to T , we obtain

$$\sum_{t=1}^T \langle \bar{d}(t-1), x \rangle \leq \frac{1}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t-1)\|_*^2, \quad (57)$$

and combining equations (46)–(57), we obtain the following:

$$\begin{aligned} & \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle \beta_i(t) d_i(t-1), y(t) - x^* \rangle \\ & \leq \frac{b}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t-1)\|_*^2 \\ & \quad + b \varphi_{\alpha(T)}^*(\bar{z}(T)) + \frac{b}{\alpha(T)} \varphi(x^*). \end{aligned} \quad (58)$$

Through equations (27), (39), and (58), we finalize the proof of Lemma 4.

Mark:

- (1) The definition of the conjugate function for $(1/\alpha(T))\varphi(x)$ is as follows:

$$\varphi_{\alpha(T)}^*(z) = \sup_{x \in \mathcal{X}} \left\{ \langle z, x \rangle - \frac{1}{\alpha(T)} \varphi(x) \right\}. \quad (59)$$

- (2) Through inequality (15) and step 2 in Algorithm 1, for all $i \in \mathcal{V}$, $\bar{z}(1) = (1/n) \sum_{i=1}^n z_i(1) = \mathbf{0}$, we can obtain the following:

$$\begin{aligned} \bar{z}(T+1) &= \bar{z}(T) - \bar{d}(T) = \dots \\ &= \bar{z}(1) - \sum_{t=1}^T \bar{d}(t) = - \sum_{t=1}^T \bar{d}(t). \end{aligned} \quad (60)$$

Next, we provide an important inequality, which will be used in the proof of Lemma 6.

Lemma 5. (α -Lipschitz continuity of the projections). For any pair $u, v \in \mathbb{R}^n$, $\alpha(t) = \max_{1 \leq i \leq n} \alpha_i(t)$, we have the following:

$$\left\| \prod_{\mathcal{X}}^{\varphi}(u, \alpha_i(t)) - \prod_{\mathcal{X}}^{\varphi}(v, \alpha(t)) \right\| \leq \alpha(t) \|u - v\|. \quad (61)$$

A detailed proof of this Lemma can be seen in Appendix B.

Now, we focus on an analysis of a key result concerning regret, i.e., $\|x_i(t) - y(t)\|$ in Lemma 6.

Lemma 6. For all $i \in \mathcal{V}$ and $t \in \{0, \dots, T\}$, the following inequality is true:

$$\|x_i(t) - y(t)\| \leq \frac{(m+M)\sqrt{n}L}{m(1-\sigma_2(P))} \alpha(t-1). \quad (62)$$

Proof. Because $x_i(t)$ and $y(t)$ are the projections of $z_i(t)$ and $\bar{z}(t)$ onto the set $\mathcal{X}$, through Lemma 5, we have the following:

$$\begin{aligned} &\|x_i(t) - y(t)\| \\ &= \left\| \prod_{\mathcal{X}}^{\varphi}(-z_i(t), \alpha_i(t-1)) - \prod_{\mathcal{X}}^{\varphi}(-\bar{z}(t), \alpha(t-1)) \right\| \\ &\leq \alpha(t-1) \|z_i(t) - \bar{z}(t)\|. \end{aligned} \quad (63)$$

Now, considering the evolution of sequence $\{z_i(t)\}$ in Algorithm 1, we obtain the following:

$$\begin{aligned} z_i(t) &= \sum_{j=1}^n p_{ij} z_j(t-1) - d_i(t-1) \\ &= \sum_{j=1}^n p_{ij} \left[\sum_{j=1}^n p_{ij} z_j(t-2) - d_i(t-2) \right] - d_i(t-1) \\ &= \dots = - \sum_{r=1}^{t-1} \sum_{j=1}^n p_{ij}^{t-r-1} d_j(r). \end{aligned} \quad (64)$$

Because p_{ij} is an element of a doubly stochastic matrix, $\sum_{i=1}^n p_{ij} = 1$, then we have

$$\begin{aligned} \bar{z}(t+1) &= \frac{1}{n} \sum_{i=1}^n z_i(t+1) \\ &= \frac{1}{n} \sum_{i=1}^n \left[\sum_{j=1}^n p_{ij} z_j(t) - d_i(t) \right] \\ &= \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^n p_{ij} z_j(t) - \frac{1}{n} \sum_{i=1}^n d_i(t) \\ &= \frac{1}{n} \sum_{j=1}^n z_j(t) - \frac{1}{n} \sum_{i=1}^n d_i(t) \\ &= \bar{z}(t) - \bar{d}(t). \end{aligned} \quad (65)$$

Based on Algorithm 1 and the definition of $\bar{z}(t)$, we can determine that $\bar{z}(1) = \mathbf{0}$, and thus

$$\begin{aligned} \bar{z}(t+1) &= \bar{z}(t) - \bar{d}(t) = \bar{z}(t-1) - \bar{d}(t-1) - \bar{d}(t) \\ &= \dots = - \sum_{r=1}^t \bar{d}(r). \end{aligned} \quad (66)$$

In addition, we can obtain

$$\bar{z}(t) = - \sum_{r=1}^{t-1} \bar{d}(r) = - \frac{1}{n} \sum_{r=1}^{t-1} \sum_{j=1}^n d_j(r), \quad (67)$$

and based on the definition for the 1 norm of the vector (see [41]),

$$\begin{aligned} \|z_i(t) - \bar{z}(t)\| &= \left\| \sum_{r=1}^{t-1} \sum_{j=1}^n \left(p_{ij}^{t-r-1} - \frac{1}{n} \right) d_j(r) \right\| \\ &\leq \sum_{r=1}^{t-1} \sum_{j=1}^n \left| p_{ij}^{t-r-1} - \frac{1}{n} \right| \|d_j(r)\| \\ &\leq \frac{m+M}{m} L \sum_{r=1}^{t-1} \left\| P_i^{t-r-1} - \frac{1}{n} \mathbf{1} \right\|_1 \\ &= \frac{m+M}{m} L \sum_{r=1}^{t-1} \left\| P^{t-r-1} e_i - \frac{1}{n} \mathbf{1} \right\|_1. \end{aligned} \quad (68)$$

To obtain a more specific bound of equation (46), we introduce a useful property of a stochastic matrix as follows [12]:

$$\left\| P^{t-r-1} e_i - \frac{1}{n} \mathbf{1} \right\|_1 \leq \sigma_2(P)^{t-r-1} \sqrt{n}, \quad (69)$$

where P^{t-r-1} denotes the $(t-r-1)$ -th power of matrix P , e_i is the i th basis coordinate of an n -dimensional space $\mathbb{R}^n$, $\mathbf{1}$ denotes a vector with all entries equal to 1, and $\sigma_2(P)$ is the second largest eigenvalue of stochastic matrices P and $\sigma_2(P) \leq 1$, through which we obtain the following inequality:

$$\begin{aligned}
& \frac{m+M}{m} L \sum_{r=1}^{t-1} \left\| P^{t-r-1} e_i - \frac{\mathbf{1}}{n} \right\|_1 \\
& \leq \frac{m+M}{m} L \sum_{r=1}^{t-1} \sigma_2(P)^{t-r-1} \sqrt{n} \\
& \leq \frac{(m+M)\sqrt{n}L}{(1-\sigma_2(P))m}.
\end{aligned} \tag{70}$$

Combining equations (63), (68), and (70) yields the following:

$$\|x_i(t) - y(t)\| \leq \frac{(m+M)\sqrt{n}L}{m(1-\sigma_2(P))} \alpha(t-1). \tag{71}$$

Thus, we complete the proof of Lemma 6

Now, we can provide a brief proof of Theorem 1. $\square$

Proof of Theorem 1. Combining lemmata 2–6 yields the following regret bound:

$$\begin{aligned}
R_T(x^*, x_i(t)) &= \sum_{t=1}^T (f_{t,i}(x_i(t)) - f_t(x^*)) \\
&\leq \left(L + \frac{L}{n}\right) \sum_{t=1}^T \|x_i(t) - y(t)\| + \frac{L}{n} \sum_{t=1}^T \sum_{i=1}^n \|x_i(t) - y(t)\| \\
&\quad + (b+1) \left[\frac{1}{2} \sum_{t=1}^T \alpha(t-1) \|\bar{d}(t-1)\|_*^2 + \frac{1}{\alpha(T)} \varphi(x^*) \right] \\
&\quad + b\varphi_{\alpha(T)}^*(\bar{z}(T)).
\end{aligned} \tag{72}$$

By equation (71) and based on $\alpha(t) = (\lambda/\sqrt{t})$, $\varphi(x^*) \leq R^2$, $\varphi_{\alpha(T)}^*(\bar{z}(T)) \leq D$, and $\|\bar{d}(t-1)\|_*^2 \leq ((M+m)^2/m^2)L^2$, we can obtain the conclusion to Theorem 1. $\square$

5. Simulation Experiments

To verify the performance of the D-OCG, we consider a problem of a distributed sensor network [18], which has n sensors and aims at the estimation of a random vector $x \in \mathcal{X} = \{x \in \mathbb{R}^d \mid \|x\|_2^2 \leq x_{\max}^2\}$. In this network, at each time $t \in \{1, 2, \dots, T\}$, each sensor i receives an observation vector $v_{ti}: \mathbb{R}^d \rightarrow \mathbb{R}^m$, in which the vector v_{ti} is time-varying owing to the effect of the observed noise. Assume that each sensor i has a linear model $\phi_i(x) = A_i x$, where A_i is the observation matrix of sensor i , and $A_i \in \mathbb{R}^{m \times d}$ and $\|A_i\|_1 \leq \phi_{\max}$. The local cost function in sensor i is defined as $f_{ti}(\hat{x}) = (1/2)\|v_{ti} - A_i \hat{x}\|_2^2$, where $v_{ti} = A_i x + \eta_{ti}$ in which η_{ti} is white noise. The mathematical model of this problem is

$$\min f(\hat{x}) \quad \sum_{t=1}^T \sum_{i=1}^n \frac{1}{2} \|v_{ti} - A_i \hat{x}\|_2^2 \tag{73}$$

subject to $\hat{x} \in \mathcal{X}$.

In an offline case, the cost function in each sensor i is fixed, and because we can know all information of the cost function in advance, the centralized optimal estimate for this problem can be obtained by

$$x^* = \frac{1}{T} \sum_{t=1}^T \left(\sum_{i=1}^n A_i^T A_i \right)^{-1} \left(\sum_{i=1}^n A_i v_{ti} \right). \tag{74}$$

In a practical problem, the characteristics of the white noise may be unknown, or some sensors might not work properly for a particular reason, and we therefore need to find an estimate for vector x using a distributed online algorithm. Here, we set $d = 1$ and $A_i = 1/2$, and sensor i observes $v_{ti} = a_{ti}x + b_{ti}$, where $a_{ti} \sim \mathbf{U}(0, 1)$ and $b_{ti} \sim \mathbf{U}(-(1/4), 1/4)$ (in which $x \sim \mathbf{U}(a, b)$ indicates a random vector x uniformly distributed on $(0, 1)$). Then, the cost function for sensor i at each time t is given by $f_{ti}: \mathbb{R} \rightarrow \mathbb{R}$, where $f_{ti} = (1/2)(v_{ti} - A_i \hat{x})^2$ and $A_i \in \mathbb{R}$.

We verified the performance of the proposed algorithm based on the following three aspects:

- (1) First, we determined how the number of nodes in the network affects the performance of the D-OCG. We can see from Figure 1 that the average regret decreases slowly when we increase the number of nodes, and the algorithm is convergent on different scaled networks. When $n = 1$, the problem is equivalent to a centralized optimization problem, and our distributed optimization algorithm can reach the same effect as the centralized algorithm.
- (2) We then checked how the network topology influences the performance of the D-OCG. We implemented the algorithm on three types of graphs with nine nodes. In a complete graph, each node is connected to the remaining nodes, that is, all nodes can exchange information with each other. In a cycle graph, each node is only connected to two nodes directly adjacent to it. The connectivity of a Watts–Strogatz graph is between the complete graph and the cycle graph. From Figure 2, it can be seen that a better connectivity can lead to a slightly faster convergence.
- (3) We next compared our algorithm with the class algorithm D-OGD in [20]. The parameters used in these two algorithms are based on their theoretical proofs. The network topology relationship among nodes is complete, whereas for nodes $n = 9$, the step size is $\alpha(t) = 1/\sqrt{t}$. As shown in Figure 3, the convergence speed of the two algorithms is initially close, but with an increase in the number of iterations, the D-OCG converges faster than the D-OGD, which fully reflects the excellent performance of the proposed algorithm.

FIGURE 1: The performance of our algorithm with different numbers of nodes.

FIGURE 2: The performance of our algorithm with different topologies.

6. Conclusion

We proposed a distributed online conjugate gradient algorithm to solve the distributed optimization problem with a convex constraint in a network. With this algorithm, the conjugate gradient is used to replace the gradient or subgradient in a traditional gradient decent method. Because the search direction is mutually conjugated throughout the entire algorithm iteration process, we can remove the disadvantage that a slow convergence has in the later stage of a gradient decent. We also presented a detailed analysis of the convergence for the proposed algorithm and obtained a regret bound for the optimization problem. The regret bound has a sublinear convergence.

FIGURE 3: Comparison of D-OCG and D-OGD with nine nodes.

We applied the proposed algorithm (D-OCG) to a distributed sensor estimation problem. The numerical results show that our algorithm is feasible and effective, and under the same assumptions, the D-OCG has a better convergence rate than the traditional D-OGD gradient method.

Appendix

A. Proof of Lemma 5

Let $\omega = \prod_{\mathcal{X}}^{\varphi}(u, \alpha_i(t))$, $x = \prod_{\mathcal{X}}^{\varphi}(v, \alpha(t))$. Based on the first-order optimality condition for convex optimization, for any $y \in \mathcal{X}$, we obtain the following two inequalities:

$$\langle u + \frac{1}{\alpha_i(t)} \nabla \varphi(\omega), y - \omega \rangle \geq 0, \quad (\text{A.1})$$

$$\langle v + \frac{1}{\alpha(t)} \nabla \varphi(x), y - x \rangle \geq 0. \quad (\text{A.2})$$

Through equation (A.1), we obtain the following:

$$\begin{aligned} & \langle u + \frac{1}{\alpha_i(t)} \nabla \varphi(\omega), y - \omega \rangle \\ &= \langle u, y - \omega \rangle + \langle \frac{1}{\alpha_i(t)} \nabla \varphi(\omega), y - \omega \rangle \geq 0, \end{aligned} \quad (\text{A.3})$$

and thus

$$\langle u, y - \omega \rangle \geq \langle -\frac{1}{\alpha_i(t)} \nabla \varphi(\omega), \omega - y \rangle. \quad (\text{A.5})$$

In addition, because $\varphi(x) = (1/2)\|x\|^2$ is a strong convex function, we obtain the following: $\forall \omega, y \in \mathcal{X}$,

$$\varphi(y) \geq \varphi(\omega) + \langle \nabla \varphi(\omega), y - \omega \rangle + \frac{1}{2}\|y - \omega\|^2, \quad (\text{A.6})$$

namely,

$$\begin{aligned}
\langle \nabla \varphi(\omega), \omega - y \rangle &\geq \varphi(\omega) - \varphi(y) + \frac{1}{2} \|y - \omega\|^2 \\
&= \frac{1}{2} \|\omega\|^2 - \frac{1}{2} \|y\|^2 + \frac{1}{2} \|y - \omega\|^2 \\
&= \frac{1}{2} [(\|\omega\| + \|y\|)(\|\omega\| - \|y\|) + \|y - \omega\|^2] \\
&\geq \frac{1}{2} [(\|y - \omega\|)(\|\omega\| - \|y\|) + \|y - \omega\|^2] \\
&= \frac{1}{2} \|y - \omega\| [\|\omega\| - \|y\| + \|y - \omega\|] \\
&\geq \frac{1}{2} \|y - \omega\| [(\|\omega\| - \|y\|) + (\|y\| - \|\omega\|)] = 0.
\end{aligned} \tag{A.7}$$

We therefore have

$$\langle \nabla \varphi(\omega), \omega - y \rangle \geq 0, \tag{A.8}$$

and because we know $\alpha(t) \geq \alpha_i(t) \geq 0$, therefore

$$\frac{1}{\alpha_i(t)} \langle \nabla \varphi(\omega), \omega - y \rangle \geq \frac{1}{\alpha(t)} \langle \nabla \varphi(\omega), \omega - y \rangle,$$

$$\begin{aligned}
\langle u, y - \omega \rangle &\geq \frac{1}{\alpha_i(t)} \langle \nabla \varphi(\omega), \omega - y \rangle \\
&\geq \frac{1}{\alpha(t)} \langle \nabla \varphi(\omega), \omega - y \rangle.
\end{aligned} \tag{A.9}$$

In addition, we have

$$\langle u, y - \omega \rangle \geq \frac{1}{\alpha(t)} \langle \nabla \varphi(\omega), \omega - y \rangle, \tag{A.10}$$

that is,

$$\langle u + \frac{1}{\alpha(t)} \nabla \varphi(\omega), y - \omega \rangle \geq 0. \tag{A.11}$$

Setting $y = \omega$ in equation (A.2) and $y = x$ in equation (A.11) yields

$$\begin{aligned}
\langle u\alpha(t) + \nabla \varphi(\omega), x - \omega \rangle &\geq 0, \\
\langle v\alpha(t) + \nabla \varphi(x), \omega - x \rangle &\geq 0.
\end{aligned} \tag{A.12}$$

Adding the above two inequalities, we obtain the following bound:

$$\alpha(t) \langle u - v, x - \omega \rangle \geq \langle \nabla \varphi(\omega) - \nabla \varphi(x), \omega - x \rangle. \tag{A.13}$$

By contrast, $\varphi(x)$ is a strong convexity function, which implies that

$$\varphi(\omega) \geq \varphi(x) + \langle \nabla \varphi(x), \omega - x \rangle + \frac{1}{2} \|\omega - x\|^2, \tag{A.14}$$

$$\varphi(x) \geq \varphi(\omega) + \langle \nabla \varphi(\omega), x - \omega \rangle + \frac{1}{2} \|\omega - x\|^2.$$

Adding the above two inequalities, we have the following:

$$\langle \nabla \varphi(\omega) - \nabla \varphi(x), x - \omega \rangle + \|\omega - x\|^2 \leq 0, \tag{A.15}$$

and thus

$$\langle \nabla \varphi(\omega) - \nabla \varphi(x), \omega - x \rangle \geq \|\omega - x\|^2. \tag{A.16}$$

Combining equations (A.13) and (A.16), we can obtain

$$\alpha(t) \langle u - v, x - \omega \rangle \geq \langle \nabla \varphi(\omega) - \nabla \varphi(x), \omega - x \rangle \geq \|\omega - x\|^2, \tag{A.17}$$

namely,

$$\begin{aligned}
\|\omega - x\|^2 &\leq \alpha(t) \langle u - v, x - \omega \rangle \\
&\leq \alpha(t) \|u - v\| \|x - \omega\|.
\end{aligned} \tag{A.18}$$

Thus,

$$\|\omega - x\| \leq \alpha(t) \|u - v\|, \tag{A.19}$$

and we therefore obtain

$$\left\| \prod_{\mathcal{X}}^{\varphi(x)} (u, \alpha_i(t)) - \prod_{\mathcal{X}}^{\varphi(x)} (v, \alpha(t)) \right\| \leq \alpha(t) \|u - v\|. \tag{A.20}$$

This completes the proof of the claim in Lemma 5.

B. Proof of Inequality (36)

Based on the definition of the conjugate function, we can obtain

$$\begin{aligned}
\varphi_{\alpha}^*(-z) &= \sup_{x \in \mathcal{X}} \left\{ \langle -z, x \rangle - \frac{1}{\alpha} \varphi(x) \right\} \\
&= \sup_x \left\{ \langle z, x \rangle - \frac{1}{\alpha} \varphi(x) - \delta(x \in \mathcal{X}) \right\},
\end{aligned} \tag{A.21}$$

where $\delta(x \in \mathcal{X})$ is the indicator function of set $\mathcal{X}$. In addition, $\delta(x \in \mathcal{X}) = 0$ when $x \in \mathcal{X}$; otherwise, $\delta(x \in \mathcal{X}) = +\infty$. By contrast,

$$\prod_{\mathcal{X}}^{\varphi(x)} (z, \alpha) = \arg \min_{x \in \mathcal{X}} \{ \langle z, x \rangle + (1/\alpha) \varphi(x) \}, \tag{A.22}$$

and $\mathcal{X}$ is compact, and thus the supremum of $\varphi_{\alpha}^*(-z)$ can be uniquely attained through $\prod_{\mathcal{X}}^{\varphi(x)} (z, \alpha)$. Here, $\langle z, x \rangle$ is differentiable in z , $\nabla \varphi_{\alpha}^*(-z) = \prod_{\mathcal{X}}^{\varphi(x)} (z, \alpha)$.

Because the projection is Lipschitz continuous, we have the following:

$$\|\nabla \varphi_{\alpha}^*(-\bar{z}) - \nabla \varphi_{\alpha}^*(-\bar{z} - \bar{d})\| = \left\| \prod_{\mathcal{X}}^{\varphi} (\bar{z}, \alpha) - \prod_{\mathcal{X}}^{\varphi} (\bar{z} + \bar{d}, \alpha) \right\| \leq \alpha \|\bar{d}\|_*. \tag{A.23}$$

Through Lemma 1.2.3 and Corollary 4.4.5 in [39], we can obtain

$$\varphi_{\alpha}^*(\bar{z} + \bar{d}) \leq \varphi_{\alpha}^*(\bar{z}) - \langle -\bar{d}, \nabla \varphi_{\alpha}^*(\bar{z}) \rangle + \frac{1}{2} \alpha \|\bar{d}\|_*^2. \tag{A.24}$$

However,

$$\nabla \varphi_{\alpha}^*(-\bar{z}) = x, \quad (\text{A.25})$$

and thus

$$\varphi_{\alpha}^*(-\bar{z} + \bar{d}) \leq \varphi_{\alpha}^*(-\bar{z}) - \langle -\bar{d}, x \rangle + \frac{1}{2} \alpha \|\bar{d}\|_*^2. \quad (\text{A.26})$$

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC), under Grant nos. 11471102, 61976243 and 61871430, the basic research projects in the University of Henan Province, under Grant nos. 19zx010 and 20zx001, and the Science and Technology Development Programs of Henan Province, under Grant no. 192102210284.

References

- [1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
- [2] B. Charrow, N. Michael, and V. Kumar, "Cooperative multi-robot estimation and control for radio source localization," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 569–580, 2014.
- [3] V. Lesser, C. Ortiz, and M. Tambe, *Distributed Sensor Networks: A Multiagent Perspective*, Vol. 9, Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.
- [4] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pp. 20–27, Berkeley, CA, USA, April 2004.
- [5] D. Li, K. Wong, Y. Hu, and A. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.
- [6] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, 2018.
- [7] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.
- [8] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.
- [9] S. Pu and A. Nedić, "A distributed stochastic gradient tracking method," 2019, <http://arxiv.org/abs/1805.11454>.
- [10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [11] D. P. Bertsekas, "Centralized and distributed Newton methods for network optimization and extensions," *Mathematics*, vol. 3, no. 3, pp. 5917–5922, 2015.
- [12] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.
- [13] D. Jakoveti, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.
- [14] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [15] J. Zhu, P. Xie, M. Zhang, R. Zheng, L. Xing, and Q. Wu, "Distributed stochastic subgradient projection algorithms based on weight-balancing over time-varying directed graphs," *Complexity*, vol. 2019, Article ID 8030792, 16 pages, 2019.
- [16] H. Gokcesu and S. S. Kozat, "Efficient online convex optimization with adaptively minimax optimal dynamic regret," 2019, <http://arxiv.org/abs/1907.00497>.
- [17] D. Mateos-Nunez and J. Cortes, "Distributed online convex optimization over jointly connected digraphs," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 1, pp. 23–37, 2014.
- [18] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed convex optimization on dynamic networks," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3545–3550, 2016.
- [19] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *Proceedings of the 2nd IEEE Conference on Decision and Control*, pp. 1484–1489, Florence, Italy, December 2013.
- [20] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2013.
- [21] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized online convex programming with local information," in *Proceedings of the 2011 American Control Conference*, pp. 5363–5369, San Francisco, CA, USA, June 2011.
- [22] M. Akbari, B. Ghahsifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 417–428, 2017.
- [23] D. Yuan, D. W. C. Ho, and G.-P. Jiang, "An adaptive primal-dual subgradient algorithm for online distributed constrained optimization," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3045–3055, 2018.
- [24] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.
- [25] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Transactions on Signal and Information Processing Over Networks*, vol. 4, no. 1, pp. 4–17, 2018.
- [26] C. Xu, J. Zhu, and D. O. Wu, "Decentralized online learning methods based on weight-balancing over time-varying digraphs," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1–13, 2019.
- [27] X. Yi, X. Li, L. Xie, and H. J. Karl, "Distributed online convex optimization with time-varying coupled inequality constraints," 2019, <http://arxiv.org/abs/1903.04277>.
- [28] M. Zhang, W. Quan, N. Cheng et al., "Distributed conditional gradient online learning for IoT optimization," *IEEE Internet of Things Journal*, 2019.

- [29] M. J. D. Powell, "Convergence properties of algorithms for nonlinear optimization," *SIAM Review*, vol. 28, no. 4, pp. 487–500, 1986.
- [30] P. Faramarzi and K. Amini, "A modified spectral conjugate gradient method with global convergence," *Journal of Optimization Theory and Applications*, vol. 182, no. 2, pp. 667–690, 2019.
- [31] G. Yuan, "Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems," *Optimization Letters*, vol. 3, no. 1, pp. 11–21, 2009.
- [32] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," *IEE Proceedings G Circuits, Devices and Systems*, vol. 139, no. 3, pp. 301–310, 1992.
- [33] M. Al-Baali, Y. Narushima, and H. Yabe, "A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization," *Computational Optimization and Applications*, vol. 60, no. 1, pp. 89–110, 2015.
- [34] W. Rodi and R. L. Mackie, "Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion," *Geophysics*, vol. 66, no. 1, pp. 174–187, 2001.
- [35] G. Li, C. Tang, and Z. Wei, "New conjugacy condition and related new conjugate gradient methods for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 202, no. 2, pp. 523–539, 2007.
- [36] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.
- [37] B. Chen, *Optimization Theory and Algorithm*, Tsinghua University Press, Beijing, China, 2015.
- [38] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [39] J. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I*, Springer, Berlin, Germany, 1996.
- [40] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Kluwer Academic Publishers, Dordrecht, Netherlands, 2004.
- [41] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, USA, 1985.