IEICE TRANS. FUNDAMENTALS, VOL.E107–A, NO.5 MAY 2024

PAPER *Special Section on Mathematical Systems Science and its Applications*

Distributed Event-Triggered Stochastic Gradient-Tracking for Nonconvex Optimization

Daichi ISHIKAWA[†], *Nonmember*, **Naoki HAYASHI**[††a)], *and* **Shigemasa TAKAI**[†], *Members*

SUMMARY In this paper, we consider a distributed stochastic nonconvex optimization problem for multiagent systems. We propose a distributed stochastic gradient-tracking method with event-triggered communication. A group of agents cooperatively finds a critical point of the sum of local cost functions, which are smooth but not necessarily convex. We show that the proposed algorithm achieves a sublinear convergence rate by appropriately tuning the step size and the trigger threshold. Moreover, we show that agents can effectively solve a nonconvex optimization problem by the proposed event-triggered algorithm with less communication than by the existing time-triggered gradient-tracking algorithm. We confirm the validity of the proposed method by numerical experiments.

key words: *distributed stochastic algorithm, event-triggered communication, nonconvex optimization*

1. Introduction

As typified by the Internet of Things and big data processing, the importance of large-scale systems has been increasing. A large-scale system can be modeled as a multiagent system, in which a group of agents autonomously performs distributed processing [1]. Recently, solving optimization problems in a distributed manner with a multiagent system has attracted great attention [2]–[8]. In distributed optimization, each agent cooperatively estimates an optimal solution by exchanging the estimated values. A number of distributed algorithms have also been considered for nonconvex optimization. Zhu and Martínez proposed a dual subgradient algorithm for constrained optimization [9]. Lorenzo and Scutari considered a distributed iterative algorithm with successive convex approximation [10]. Tatarenko and Touri proposed a push-sum-based algorithm for time-varying and directed graphs [11]. Jiang et al. proposed a proximal gradient algorithm over time-varying multiagent networks [12].

Although these online algorithms have played a crucial role in machine learning and data analysis, deterministic approaches encounter difficulties when handling large-scale problems because they require the computation of a full gradient. In contrast, stochastic gradient descent algorithms, which leverage random sampling to approximate the gradient, offer advantages in reducing the computational burden. Moreover, random sampling can escape non-optimal local minima and lead to faster convergence in practice. Distributed stochastic gradient descent algorithms for nonconvex optimization have also been investigated in [13]–[16].

Many of the existing distributed methods are based on time-triggered algorithms, and agents must exchange information with neighbors at every iteration of the algorithm. However, high-frequency communication by time-triggered algorithms can be a bottleneck due to limited power resources and poor network environments. Distributed algorithms with event-triggered communication have been proposed for convex optimization problems [17]–[22]. Event-triggered communication is a method where agents communicate only when a certain event occurs and can effectively utilize network resources [23], [24]. Event-triggered algorithms for nonconvex optimization problems have also been considered in [25]–[27]. However, the convergence of naive gradient-based algorithms is relatively slow. Therefore, the gradient-tracking method with event-triggered communication is also preferable for distributed optimization to enhance convergence performance.

Motivated by this, we propose a distributed stochastic gradient-tracking algorithm with event-triggered communication. A group of agents cooperatively minimizes the sum of local cost functions, which are smooth but not necessarily convex. Each agent has estimations for a critical point and a gradient of its own local cost function. At each iteration, each agent randomly chooses gradient information with sampled data and exchanges these estimations with the neighboring agents only when the error between the last triggered estimation and the current estimation exceeds a trigger threshold. This is in contrast to the existing distributed stochastic algorithms [13]–[16] that require communication at every iteration. Thus, the proposed algorithm inherits the advantages of the fast convergence of the stochastic gradient-tracking algorithm [15] and the efficient communication of the event-triggered method. We characterize the transient and steady-state performance of the proposed algorithm in terms of the expected time-averaged gradient of the cost function. We also show that the proposed algorithm can achieve a sublinear convergence rate by appropriately tuning the step size and the trigger threshold.

This paper is organized as follows. Section 2 addresses the problem setting of the nonconvex optimization and the distributed event-triggered stochastic gradient-tracking algorithm. Section 3 presents the convergence analysis of the proposed algorithm. Section 4 shows the validity of the

Manuscript received March 24, 2023.
Manuscript revised June 22, 2023.
Manuscript publicized January 18, 2024.

[†]The authors are with the Graduate School of Engineering, Osaka University, Suita-shi, 565-0871 Japan.
[††]The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560-8531 Japan.
a) E-mail: n.hayashi@sys.es.osaka-u.ac.jp

DOI: 10.1587/transfun.2023MAP0002

proposed method through a numerical example. Section 5 concludes this paper.

2. Event-Triggered Stochastic Gradient-Tracking Method

Let $\mathbb{R}$ and $\mathbb{N}$ be the sets of real numbers and nonnegative integers, respectively. $\mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{0}_n \in \mathbb{R}^n$ are vectors whose elements are all 1s and all 0s, respectively. $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ and $\mathbf{O}_n \in \mathbb{R}^{n\times n}$ are the identity matrix and zero matrix, respectively. The transpose of a vector or a matrix is denoted by $[\cdot]^\mathsf{T}$. The Euclidean norm of a vector or the spectral norm of a matrix is denoted by $\|\cdot\|$. For a matrix $\mathbf{X}$, the spectral radius, the adjugate matrix, and the determinant are represented by $\rho(\mathbf{X})$, $\mathrm{adj}(\mathbf{X})$, and $|\mathbf{X}|$, respectively. The diagonal matrix consisting of the diagonal components of $\mathbf{X}$ is represented by $\mathrm{diag}(\mathbf{X})$. The Kronecker product is denoted by $\otimes$. For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, $\mathbf{a} < \mathbf{b}$ and $\mathbf{a} \leq \mathbf{b}$ show the inequality relations of each element.

We consider the following nonconvex optimization problem with n agents:

$$\underset{\mathbf{x}\in\mathbb{R}^p}{\text{minimize}}\ F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where $f_i : \mathbb{R}^p \to \mathbb{R}$ is a local objective function that is not necessarily convex ($i \in \mathcal{V} = \{1, 2, \ldots, n\}$).

The communication between agents is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. In this paper, we make the following assumptions about the local cost function and the graph $\mathcal{G}$.

Assumption 1: The local cost function f_i is ℓ-smooth ($\ell \geq 1$); that is, there exists a positive constant $\ell \geq 1$ such that $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq \ell\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$.

Assumption 2: The directed communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is strongly connected and admits a doubly stochastic weight matrix $\underline{\mathbf{W}} = [\underline{w}_{ij}] \in \mathbb{R}^{n\times n}$, where $\underline{w}_{ij}$ is the weight for a directed edge $(j, i) \in \mathcal{E}$.

Each agent i generates a sequence of estimations $\{\mathbf{x}^i_{(k)}\}$ of the solution of the optimization problem (1). Without loss of generality, we assume that $\mathbf{x}^i_{(0)} = \mathbf{x}^j_{(0)}$ for all $i, j \in \mathcal{V}$. At iteration k, each agent i receives a stochastic gradient $\mathbf{g}_i(\mathbf{x}^i_{(k)}, \xi^i_{(k)})$, where $\xi^i_{(k)} \in \mathbb{R}^q$ is a random vector sampled at iteration k and $\mathbf{g}_i : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p$ is a Borel-measurable function that represents the stochastic gradient evaluated at the estimation $\mathbf{x}^i_{(k)}$ with the sampled data $\xi^i_{(k)}$.

We consider the sub-σ-algebra of $\mathcal{F}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that $\mathcal{F}_0 = \{\Omega, \emptyset\}$ and $\mathcal{F}_k = \sigma(\{\xi^i_{(t)} \mid 0 \leq t \leq k-1,\ i \in \mathcal{V}\})$ for $k \geq 1$.

Assumption 3: The stochastic gradient process $\{\mathbf{g}_i(\mathbf{x}^i_{(k)}, \xi^i_{(k)})\}_{k\geq 0}$ satisfies the following:

(a) The random vector $\xi^i_{(k)}$ is independent.

(b) For all $k \geq 0$ and $i \in \mathcal{V}$, $\mathbb{E}[\mathbf{g}_i(\mathbf{x}^i_{(k)}, \xi^i_{(k)}) \mid \mathcal{F}_k] = \nabla f_i(\mathbf{x}^i_{(k)})$.

(c) For all $k \geq 0$ and $i \in \mathcal{V}$, there exists a constant $\nu_i > 0$ such that $\mathbb{E}[\|\mathbf{g}_i(\mathbf{x}^i_{(k)}, \xi^i_{(k)}) - \nabla f_i(\mathbf{x}^i_{(k)})\|^2 \mid \mathcal{F}_k] \leq \nu_i^2$.

Algorithm 1 Distributed Event-Triggered Stochastic Gradient-Tracking Algorithm

Require: Initial variables $\mathbf{x}^i_{(0)} \in \mathbb{R}^p$, $\mathbf{y}^i_{(0)} = \mathbf{0}_p$ and $\mathbf{g}_j(\mathbf{x}^j_{(-1)}, \xi^j_{(-1)}) = \mathbf{0}_p$ for each agent $i \in \mathcal{V}$. Set $k = 0$.

repeat

For all $i \in \mathcal{V}$,

$$\mathbf{v}^i_{(k+1)} = \mathbf{y}^i_{(k)} + \mathbf{g}_i(\mathbf{x}^i_{(k)}, \xi^i_{(k)}) - \mathbf{g}_i(\mathbf{x}^i_{(k-1)}, \xi^i_{(k-1)})$$

if $\|\mathbf{v}^i_{(k+1)} - \tilde{\mathbf{v}}^i_{(k)}\| \geq E^i_v(k+1)$

agent i sends $\mathbf{v}^i_{(k+1)}$ to agent $j \in \mathcal{N}^{\text{out}}_i$

set $\tilde{\mathbf{v}}^i_{(k+1)} = \mathbf{v}^i_{(k+1)}$

else

$\tilde{\mathbf{v}}^i_{(k+1)} = \tilde{\mathbf{v}}^i_{(k)}$

$$\tilde{\mathbf{v}}^j_{(k+1)} = \begin{cases} \mathbf{v}^j_{(k+1)} & \text{if agent } j \in \mathcal{N}^{\text{in}}_i \text{ sends } \mathbf{v}^j_{(k+1)} \text{ to agent } i \\ \tilde{\mathbf{v}}^j_{(k)} & \text{otherwise} \end{cases}$$

$$\mathbf{y}^i_{(k+1)} = \mathbf{v}^i_{(k+1)} + \sum_{j=1}^{n} \underline{w}_{ij}(\tilde{\mathbf{v}}^j_{(k+1)} - \tilde{\mathbf{v}}^i_{(k+1)})$$

$$\mathbf{u}^i_{(k+1)} = \mathbf{x}^i_{(k)} - \alpha\mathbf{y}^i_{(k+1)}$$

if $\|\mathbf{u}^i_{(k+1)} - \tilde{\mathbf{u}}^i_{(k)}\| \geq E^i_u(k+1)$

agent i sends $\mathbf{u}^i_{(k+1)}$ to agent $j \in \mathcal{N}^{\text{out}}_i$

set $\tilde{\mathbf{u}}^i_{(k+1)} = \mathbf{u}^i_{(k+1)}$

else

$\tilde{\mathbf{u}}^i_{(k+1)} = \tilde{\mathbf{u}}^i_{(k)}$

$$\tilde{\mathbf{u}}^j_{(k+1)} = \begin{cases} \mathbf{u}^j_{(k+1)} & \text{if agent } j \in \mathcal{N}^{\text{in}}_i \text{ sends } \mathbf{u}^j_{(k+1)} \text{ to agent } i \\ \tilde{\mathbf{u}}^j_{(k)} & \text{otherwise} \end{cases}$$

$$\mathbf{x}^i_{(k+1)} = \mathbf{u}^i_{(k+1)} + \sum_{j=1}^{n} \underline{w}_{ij}(\tilde{\mathbf{u}}^j_{(k+1)} - \tilde{\mathbf{u}}^i_{(k+1)})$$

Set $k = k + 1$

until a predefined stopping criterion is satisfied.

The proposed distributed event-triggered stochastic gradient-tracking algorithm is summarized in Algorithm 1, where $\mathcal{N}^{\text{in}}_i$ and $\mathcal{N}^{\text{out}}_i$ are the sets of in-neighbor agents and out-neighbor agents of agent i. Each agent $i \in \mathcal{V}$ has the state $\mathbf{x}^i_{(k)} \in \mathbb{R}^p$ and the variable $\mathbf{y}^i_{(k)} \in \mathbb{R}^p$ at iteration $k \in \mathbb{N}$. These are the estimates of a critical point $x^* \in \mathcal{X}^*$ of the optimization problem (1) and the overall gradient ∇F, respectively, where $\mathcal{X}^* = \{\mathbf{x} \in \mathbb{R}^p \mid F(\mathbf{x}) \leq F(\mathbf{y}), \forall \mathbf{y} \in \mathbb{R}^p\}$. Each agent i updates its state according to Algorithm 1 via local communication on the directed graph $\mathcal{G}$, where α is the step size at iteration k.

Each agent i determines when it communicates with the neighbors depending on the values of the variables $\mathbf{v}^i_{(k)}$ and $\mathbf{u}^i_{(k)}$, which are the internal states for the estimation by the gradient-tracking and gradient descent algorithms. Let $k^i_v(m)$ and $k^i_u(m)$ be the m-th trigger times when agent i sends the internal variables $\mathbf{v}^i_{(k)}$ and $\mathbf{u}^i_{(k)}$ to the neighbors. Suppose that $\tilde{\mathbf{v}}^i_{(k)}$ and $\tilde{\mathbf{u}}^i_{(k)}$ are the latest values that were sent to the

neighbor agent $j \in \mathcal{N}_i^{\text{in}}$ before iteration k; that is,

$$\tilde{\mathbf{v}}_{(k)}^i = \begin{cases} \mathbf{v}_{(k)}^i & \text{if } k \in \kappa_v^i, \\ \tilde{\mathbf{v}}_{(k-1)}^i & \text{otherwise,} \end{cases}$$

$$\tilde{\mathbf{u}}_{(k)}^i = \begin{cases} \mathbf{u}_{(k)}^i & \text{if } k \in \kappa_u^i, \\ \tilde{\mathbf{u}}_{(k-1)}^i & \text{otherwise,} \end{cases}$$

where $\kappa_v^i = \{k_v^i(1), k_v^i(2), \dots\}$ and $\kappa_u^i = \{k_u^i(1), k_u^i(2), \dots\}$ are the sets of trigger times for $\mathbf{v}_{(k)}^i$ and $\mathbf{u}_{(k)}^i$.

In the event-triggered communication, agent i sends the internal variable $\mathbf{v}_{(k)}^i$ to the neighbors in $\mathcal{N}_i^{\text{in}}$ when $\|\mathbf{v}_{(k)}^i - \tilde{\mathbf{v}}_{(k-1)}^i\| \geq E_v^i(k)$, where $E_v^i(k)$ is the trigger threshold for $\mathbf{v}_{(k)}^i$. Similarly, $\mathbf{u}_{(k)}^i$ is sent to the neighbors when $\|\mathbf{u}_{(k)}^i - \tilde{\mathbf{u}}_{(k-1)}^i\| \geq E_u^i(k)$, where $E_u^i(k)$ is the threshold for $\mathbf{u}_{(k)}^i$. The trigger thresholds represent the tolerance of the errors $\mathbf{e}_v^i(k) = \|\mathbf{v}_{(k)}^i - \tilde{\mathbf{v}}_{(k)}^i\|$ and $\mathbf{e}_u^i(k) = \|\mathbf{u}_{(k)}^i - \tilde{\mathbf{u}}_{(k)}^i\|$.

3. Convergence Analysis

We introduce stack vectors such that $\mathbf{x}_k = [(\mathbf{x}_{(k)}^1)^\mathsf{T}, (\mathbf{x}_{(k)}^2)^\mathsf{T}, \dots, (\mathbf{x}_{(k)}^n)^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{np}$, $\mathbf{y}_k = [(\mathbf{y}_{(k)}^1)^\mathsf{T}, (\mathbf{y}_{(k)}^2)^\mathsf{T}, \dots, (\mathbf{y}_{(k)}^n)^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{np}$, $\mathbf{g}_k = [\mathbf{g}_1(\mathbf{x}_{(k)}^1, \xi_{(k)}^1)^\mathsf{T}, \mathbf{g}_2(\mathbf{x}_{(k)}^2, \xi_{(k)}^2)^\mathsf{T}, \dots, \mathbf{g}_n(\mathbf{x}_{(k)}^n, \xi_{(k)}^n)^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{np}$, $\mathbf{e}_k^v = [(\mathbf{e}_v^1(k))^\mathsf{T}, (\mathbf{e}_v^2(k))^\mathsf{T}, \dots, (\mathbf{e}_v^n(k))^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{np}$, and $\mathbf{e}_k^u = [(\mathbf{e}_u^1(k))^\mathsf{T}, (\mathbf{e}_u^2(k))^\mathsf{T}, \dots, (\mathbf{e}_u^n(k))^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{np}$. Then, from Algorithm 1, we have

$$\mathbf{y}_{k+1} = \mathbf{W}(\mathbf{y}_k + \mathbf{g}_k - \mathbf{g}_{k-1}) + \mathbf{L}\mathbf{e}_{k+1}^v, \tag{2}$$

$$\mathbf{x}_{k+1} = \mathbf{W}(\mathbf{x}_k - \alpha \mathbf{y}_{k+1}) + \mathbf{L}\mathbf{e}_{k+1}^u, \tag{3}$$

where $\mathbf{W} = \underline{\mathbf{W}} \otimes \mathbf{I}_p$ and $\mathbf{L} = \mathbf{W} - \mathbf{I}_{np}$. Moreover, the trigger errors satisfy

$$\|\mathbf{e}_k^v\| \leq E_k^v, \quad \|\mathbf{e}_k^u\| \leq E_k^u, \tag{4}$$

where $E_k^v = \sqrt{\sum_{i=1}^n (E_v^i(k))^2}$ and $E_k^u = \sqrt{\sum_{i=1}^n (E_u^i(k))^2}$.

We define the averaging matrix, the spectral of the consensus-error matrix, and the spectral of the Laplacian matrix by $\mathbf{J} = \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\right) \otimes \mathbf{I}_p$, $\lambda = \|\mathbf{W} - \mathbf{J}\|$, and $\lambda_L = \|\mathbf{L}\|$, respectively. From Assumption 2, we have $\lambda \in [0, 1)$ and $\lambda_L \in [0, 1)$. We also consider the averaging vectors as follows: $\overline{\mathbf{x}}_k = \frac{1}{n}(\mathbf{1}_n^\mathsf{T} \otimes \mathbf{I}_p)\mathbf{x}_k$, $\overline{\mathbf{y}}_k = \frac{1}{n}(\mathbf{1}_n^\mathsf{T} \otimes \mathbf{I}_p)\mathbf{y}_k$, $\overline{\nabla \mathbf{f}}_k = \frac{1}{n}(\mathbf{1}_n^\mathsf{T} \otimes \mathbf{I}_p)\nabla \mathbf{f}_k$, and $\overline{\mathbf{g}}_k = \frac{1}{n}(\mathbf{1}_n^\mathsf{T} \otimes \mathbf{I}_p)\mathbf{g}_k$.

In the following argument, we conduct the convergence analysis of the proposed algorithm. We first show the fundamental properties of the gradient of the cost function.

Lemma 1: Under Assumptions 1–3, we have the following:

(a) $\overline{\mathbf{y}}_{k+1} = \overline{\mathbf{g}}_k, \quad \forall k \geq 0.$

(b) $\|\overline{\nabla \mathbf{f}}_k - \nabla F(\overline{\mathbf{x}}_k)\|^2 \leq \frac{\ell^2}{n}\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2, \quad \forall k \geq 0.$

(c) $\mathbb{E}[\|\overline{\mathbf{g}}_k - \overline{\nabla \mathbf{f}}_k\|^2 \mid \mathcal{F}_k] \leq \nu_a^2, \quad \forall k \geq 0,$

where $\nu_a^2 = \frac{1}{n}\sum_{i=1}^n \nu_i^2$.

Lemma 1 can be proven in the same way as Lemma 1 in [15].

The next lemmas show the recursive relation with respect to the estimations.

Lemma 2: Under Assumptions 1–3, we have

$$\mathbf{u}_{k+1} \leq \mathbf{G}\mathbf{u}_k + \mathbf{b}_k, \tag{5}$$

where

$$\mathbf{u}_k = \begin{bmatrix} \mathbb{E}\left[\frac{\|\mathbf{x}_{k+1} - \mathbf{J}\mathbf{x}_{k+1}\|^2}{n}\right] \\ \mathbb{E}\left[\frac{\|\mathbf{y}_{k+1} - \mathbf{J}\mathbf{y}_{k+1}\|^2}{n\ell^2}\right] \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \frac{3+\lambda^2}{4} & \frac{6\lambda^2\alpha^2\ell^2}{1-\lambda^2} \\ \frac{192\lambda^2}{5(1-\lambda^2)} & \frac{3+\lambda^2}{4} \end{bmatrix},$$

$$\mathbf{b}_k = \begin{bmatrix} C_5^*(E_{k+1}^u)^2 \\ C_1^* + C_2^*\alpha^2\mathbb{E}[\|\overline{\nabla \mathbf{f}}_k\|^2] + C_3^*(E_{k+1}^u)^2 + C_4^*(E_{k+2}^v)^2 \end{bmatrix},$$

and

$$C_1^* = \frac{12\nu_a^2}{\ell^2}, \quad C_2^* = \frac{48\lambda^2}{5\left(1-\lambda^2\right)}, \quad C_3^* = \frac{27\lambda_L^2}{n},$$

$$C_4^* = \frac{\lambda_L^2(3+\lambda^2)}{n\ell^2(1-\lambda^2)}, \quad C_5^* = \frac{2(1+\lambda^2)\lambda_L^2}{(1-\lambda^2)n}.$$

Lemma 3: Suppose that $0 < \alpha \leq \min\left\{\frac{1-\lambda^2}{72\lambda^2\ell}, \frac{\sqrt{5}(1-\lambda^2)}{\sqrt{32}\lambda\ell}, \frac{(1-\lambda^2)^2}{36\sqrt{3}\lambda^2\ell}\right\}$ and $0 \leq E_k^u \leq \frac{5}{64}\frac{1-\lambda^2}{\lambda^2\lambda_L\ell}$ for all $k \geq 0$. Then, $\rho(\mathbf{G}) < 1$ and $\sum_{k=0}^\infty \mathbf{G}^k = (\mathbf{I}_2 - \mathbf{G})^{-1}$ hold.

Lemmas 2 and 3 can be proven in the same way as Proposition 1 and Lemma 10 in [15]. To ensure that $\mathbf{G}$ is convergent, it is necessary to enforce the condition on the upper bound of the step size, which depends on the smoothness parameter ℓ of the local cost function. Lemmas 2 and 3 imply that the step size needs to be sufficiently small if the local cost function has a larger smoothness parameter.

The next result evaluates the upper bound on the accumulated consensus errors.

Lemma 4: Suppose that $0 < \alpha \leq \min\left\{\frac{1-\lambda^2}{72\lambda^2\ell}, \frac{\sqrt{5}(1-\lambda^2)}{\sqrt{32}\lambda\ell}, \frac{\sqrt{5}(1-\lambda^2)^2}{192\lambda^2\ell}\right\}$ and $0 \leq E_k^u \leq \frac{5}{64}\frac{1-\lambda^2}{\lambda^2\lambda_L\ell}$ for all $k \geq 0$. Then, under Assumptions 1–3, for all $k \geq 0$, we have

$$\sum_{k=0}^K \mathbb{E}\left[\frac{\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2}{n}\right] \leq C_{11}^* + C_7^* K + C_8^* \sum_{k=0}^{K-1} \mathbb{E}[\|\overline{\nabla \mathbf{f}}_k\|^2] + C_9^* \sum_{k=0}^{K-1} (E_{k+1}^u)^2 + C_{10}^* \sum_{k=0}^{K-1} (E_{k+2}^v)^2, \tag{6}$$

where

$$C_6^* = \frac{192\lambda^2\alpha^2}{(1-\lambda^2)^3}, \quad C_7^* = \frac{192\lambda^2\alpha^2\ell^2}{(1-\lambda^2)^3}C_1^*,$$

$$C_8^* = \frac{192\lambda^2\alpha^2\ell^2}{(1-\lambda^2)^3}C_2^*,$$

$$C_9^* = \frac{8}{1-\lambda^2}C_5^* + \frac{192\lambda^2\alpha^2\ell^2}{(1-\lambda^2)^3}C_3^*,$$
$$C_{10}^* = \frac{192\lambda^2\alpha^2\ell^2}{(1-\lambda^2)^3}C_4^*,$$
$$C_{11}^* = \frac{C_6^*(4\lambda^2 n\nu_a^2 + 4\lambda^2\|\nabla\mathbf{f}_0\|^2 + 2\lambda_L^2(E_0^v)^2)}{n}.$$

Proof : From (5), for all $k \geq 1$, we have $\mathbf{u}_{k+1} \leq \mathbf{G}^k\mathbf{u}_0 + \sum_{t=0}^{k-1}\mathbf{G}^t\mathbf{b}_{k-1-t}$. Then, we have

$$\begin{aligned}\sum_{k=0}^{K-1}\mathbf{u}_{k+1} &\leq \sum_{k=0}^{K-1}\mathbf{G}^k\mathbf{u}_0 + \sum_{k=1}^{K-1}\sum_{t=0}^{k-1}\mathbf{G}^t\mathbf{b}_{k-1-t}\\ &\leq \left(\sum_{k=0}^{\infty}\mathbf{G}^k\right)\mathbf{u}_0 + \left(\sum_{k=0}^{\infty}\mathbf{G}^k\right)\sum_{k=0}^{K-1}\mathbf{b}_k\\ &= (\mathbf{I}_2-\mathbf{G})^{-1}\mathbf{u}_0 + (\mathbf{I}_2-\mathbf{G})^{-1}\sum_{k=0}^{K-1}\mathbf{b}_k, \qquad (7)\end{aligned}$$

where the last equality follows from Lemma 3.

We consider the upper bound of $(\mathbf{I}_2-\mathbf{G})^{-1}$. If $0 < \alpha \leq \frac{\sqrt{5}(1-\lambda^2)^2}{192\lambda^2\ell}$ holds, we have $|\mathbf{I}_2-\mathbf{G}| = \frac{(1-\lambda^2)^2}{16} - \frac{1152\lambda^4\alpha^2\ell^2}{5(1-\lambda^2)^2} \geq \frac{(1-\lambda^2)^2}{32}$. This yields

$$(\mathbf{I}_2-\mathbf{G})^{-1} = \frac{\operatorname{adj}(\mathbf{I}_2-\mathbf{G})}{|\mathbf{I}_2-\mathbf{G}|} \leq \begin{bmatrix}\frac{8}{1-\lambda^2} & \frac{192\lambda^2\alpha^2\ell^2}{(1-\lambda^2)^3}\\ \frac{6144\lambda^2}{5(1-\lambda^2)^3} & \frac{8}{1-\lambda^2}\end{bmatrix}. \qquad (8)$$

By substituting (8) for (7) and using the fact that $\|\mathbf{x}_0-\mathbf{J}\mathbf{x}_0\| = 0$, for $K \geq 1$, we have

$$\begin{aligned}&\sum_{k=0}^{K}\mathbb{E}\left[\frac{\|\mathbf{x}_k-\mathbf{J}\mathbf{x}_k\|^2}{n}\right]\\ &\leq C_6^*\mathbb{E}\left[\frac{\|\mathbf{y}_1-\mathbf{J}\mathbf{y}_1\|^2}{n}\right] + C_7^*K + C_8^*\sum_{k=0}^{K-1}\mathbb{E}[\|\overline{\nabla\mathbf{f}}_k\|^2]\\ &\quad + C_9^*\sum_{k=0}^{K-1}(E_{k+1}^u)^2 + C_{10}^*\sum_{k=0}^{K-1}(E_{k+2}^v)^2. \qquad (9)\end{aligned}$$

Then, from (2), we have

$$\begin{aligned}&\mathbb{E}\left[\|\mathbf{y}_1-\mathbf{J}\mathbf{y}_1\|^2\right]\\ &= \mathbb{E}[\|(\mathbf{W}-\mathbf{J})\mathbf{g}_0 + \mathbf{L}\mathbf{e}_0^v\|^2]\\ &\leq 2\mathbb{E}\left[\mathbb{E}[\|(\mathbf{W}-\mathbf{J})\mathbf{g}_0\|^2 \mid \mathcal{F}_k]\right] + 2\mathbb{E}[\|\mathbf{L}\mathbf{e}_0^v\|^2]\\ &\leq 2\mathbb{E}\left[\mathbb{E}\left[\|(\mathbf{W}-\mathbf{J})(\mathbf{g}_0-\nabla\mathbf{f}_0+\nabla\mathbf{f}_0)\|^2 \mid \mathcal{F}_k\right]\right]\\ &\quad + 2\lambda_L^2(E_0^v)^2\\ &\leq 4\mathbb{E}\left[\|(\mathbf{W}-\mathbf{J})(\mathbf{g}_0-\nabla\mathbf{f}_0)\|^2\right] + 4\mathbb{E}\left[\|(\mathbf{W}-\mathbf{J})\nabla\mathbf{f}_0\|^2\right]\\ &\quad + 2\lambda_L^2(E_0^v)^2\\ &\leq 4\lambda^2 n\nu_a^2 + 4\lambda^2\|\nabla\mathbf{f}_0\|^2 + 2\lambda_L^2(E_0^v)^2, \qquad (10)\end{aligned}$$

where the last inequality follows from Lemma 1 (c).

From (9) and (10), we have (6). □

The next result shows the recursive relation with respect to the cost function.

Lemma 5: Suppose that the step size satisfies $0 < \alpha \leq \frac{1}{2\ell}$ and the trigger threshold satisfies $\sum_{k=0}^{\infty}(E_{k+1}^u)^2 < \infty$ and $\sum_{k=0}^{\infty}(E_{k+2}^v)^2 < \infty$. Then, under Assumptions 1–3, for all $k \geq 0$, we have

$$\begin{aligned}&\mathbb{E}\left[F(\overline{\mathbf{x}}_{k+1})|\mathcal{F}_k\right]\\ &\leq F(\overline{\mathbf{x}}_k) - \frac{\alpha}{2}\|\nabla F(\overline{\mathbf{x}}_k)\|^2 - \frac{\alpha}{4}\|\overline{\nabla\mathbf{f}}_k\|^2\\ &\quad + \frac{\alpha\ell^2}{2}\frac{\|\mathbf{x}_k-\mathbf{J}\mathbf{x}_k\|^2}{n} + \frac{\alpha^2\ell\nu_a^2}{2}. \qquad (11)\end{aligned}$$

Proof : From the ℓ-smoothness of F, for all $\mathbf{x},\mathbf{y} \in \mathbb{R}^p$, we have

$$F(\mathbf{y}) \leq F(\mathbf{x}) + \langle\nabla F(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\ell}{2}\|\mathbf{y}-\mathbf{x}\|^2. \qquad (12)$$

From Lemma 1 (a), for all $k \geq 0$, we also have

$$\overline{\mathbf{x}}_{k+1} = \overline{\mathbf{x}}_k - \alpha\overline{\mathbf{y}}_{k+1} = \overline{\mathbf{x}}_k - \alpha\overline{\mathbf{g}}_k. \qquad (13)$$

Thus, we obtain

$$\begin{aligned}&F(\overline{\mathbf{x}}_{k+1})\\ &\leq F(\overline{\mathbf{x}}_k) + \langle\nabla F(\overline{\mathbf{x}}_k), \overline{\mathbf{x}}_{k+1}-\overline{\mathbf{x}}_k\rangle + \frac{\ell}{2}\|\overline{\mathbf{x}}_{k+1}-\overline{\mathbf{x}}_k\|^2\\ &= F(\overline{\mathbf{x}}_k) - \alpha\langle\nabla F(\overline{\mathbf{x}}_k), \overline{\mathbf{g}}_k\rangle + \frac{\alpha^2\ell}{2}\|\overline{\mathbf{g}}_k\|^2. \qquad (14)\end{aligned}$$

Because $\mathbb{E}\left[\overline{\mathbf{g}}_k\middle|\mathcal{F}_k\right] = \overline{\nabla\mathbf{f}}_k$, we have

$$\begin{aligned}&F(\overline{\mathbf{x}}_{k+1})\\ &\leq F(\overline{\mathbf{x}}_k) - \alpha\langle\nabla F(\overline{\mathbf{x}}_k), \overline{\nabla\mathbf{f}}_k\rangle + \frac{\alpha^2\ell}{2}\mathbb{E}[\|\overline{\mathbf{g}}_k\|^2 \mid \mathcal{F}_k]\\ &= F(\overline{\mathbf{x}}_k) - \frac{\alpha}{2}\|\nabla F(\overline{\mathbf{x}}_k)\|^2 - \frac{\alpha}{2}\|\overline{\nabla\mathbf{f}}_k\|^2\\ &\quad + \frac{\alpha}{2}\|\nabla F(\overline{\mathbf{x}}_k) - \overline{\nabla\mathbf{f}}_k\|^2 + \frac{\alpha^2\ell}{2}\mathbb{E}[\|\overline{\mathbf{g}}_k\|^2 \mid \mathcal{F}_k]\\ &\leq F(\overline{\mathbf{x}}_k) - \frac{\alpha}{2}\|\nabla F(\overline{\mathbf{x}}_k)\|^2 - \frac{\alpha}{2}\|\overline{\nabla\mathbf{f}}_k\|^2\\ &\quad + \frac{\alpha\ell^2}{2}\|\mathbf{x}_k-\mathbf{J}\mathbf{x}_k\|^2 + \frac{\alpha^2\ell}{2}\mathbb{E}[\|\overline{\mathbf{g}}_k\|^2 \mid \mathcal{F}_k], \qquad (15)\end{aligned}$$

where the last inequality follows from Lemma 1 (b). From Lemma 1 (c), we further have

$$\begin{aligned}&\mathbb{E}[\|\overline{\mathbf{g}}_k\|^2 \mid \mathcal{F}_k]\\ &= \mathbb{E}[\|\overline{\mathbf{g}}_k - \overline{\nabla\mathbf{f}}_k + \overline{\nabla\mathbf{f}}_k\|^2 \mid \mathcal{F}_k]\\ &= \mathbb{E}[\|\overline{\mathbf{g}}_k - \overline{\nabla\mathbf{f}}_k\|^2 \mid \mathcal{F}_k] + 2\mathbb{E}[\langle\overline{\mathbf{g}}_k - \overline{\nabla\mathbf{f}}_k, \overline{\nabla\mathbf{f}}_k\rangle \mid \mathcal{F}_k]\\ &\quad + \mathbb{E}[\|\overline{\nabla\mathbf{f}}_k\|^2 \mid \mathcal{F}_k]\\ &\leq \nu_a^2 + \|\overline{\nabla\mathbf{f}}_k\|^2. \qquad (16)\end{aligned}$$

This yields

$$\mathbb{E}\left[F(\overline{\mathbf{x}}_{k+1})|\mathcal{F}_k\right]$$

$$\leq F(\overline{\mathbf{x}}_k) - \frac{\alpha}{2}\|\nabla F(\overline{\mathbf{x}}_k)\|^2 - \frac{\alpha(1-\alpha\ell)}{2}\|\overline{\nabla \mathbf{f}}_k\|^2 + \frac{\alpha\ell^2}{2}\frac{\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2}{n} + \frac{\alpha^2 l \nu_a^2}{2}. \tag{17}$$

Thus, for $1-\alpha\ell \geq \frac{1}{2}$, Lemma 5 holds. □

The next theorem presents the convergence of the estimation of each agent to a critical point.

Theorem 1: Suppose that the step size satisfies $0 < \alpha \leq \min\left\{\frac{1}{2\ell}, \frac{1-\lambda^2}{72\lambda^2\ell}, \frac{\sqrt{5}(1-\lambda^2)}{\sqrt{32}\lambda\ell}, \frac{\sqrt{5}(1-\lambda^2)^2}{192\lambda^2\ell^2}\right\}$ and the trigger threshold satisfies $\sum_{k=0}^{\infty}(E_{k+1}^u)^2 < \infty$ and $\sum_{k=0}^{\infty}(E_{k+2}^v)^2 < \infty$ with $0 \leq E_k^u \leq \frac{5}{64}\frac{1-\lambda^2}{\lambda^2\lambda_L\ell}$. Then, under Assumptions 1–3, we have

$$\lim_{K\to\infty}\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_k^i)\|^2\right] \leq 2\lambda\ell\nu_a^2\alpha + \frac{9126\lambda^2\ell^2\nu_a^2}{(1-\lambda^2)^3}\alpha^2. \tag{18}$$

Proof : From Lemma 5, for $K \geq 1$, we have

$$\mathbb{E}\left[F(\overline{\mathbf{x}}_K)\right] \leq \mathbb{E}\left[F(\overline{\mathbf{x}}_0)\right] - \frac{\alpha}{2}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\overline{\mathbf{x}}_k)\|^2] - \frac{\alpha}{4}\sum_{k=0}^{K-1}\mathbb{E}[\|\overline{\nabla \mathbf{f}}_k\|^2] + \frac{\alpha\ell^2}{2}\sum_{k=0}^{K-1}\mathbb{E}\left[\frac{\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2}{n}\right] + \frac{\alpha^2\ell\nu_a^2 K}{2}. \tag{19}$$

Because $F^* \leq \mathbb{E}\left[F(\overline{\mathbf{x}}_K)\right]$ holds for $K \geq 1$, we have

$$\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla F(\overline{\mathbf{x}}_k)\|^2\right] \leq \frac{2(F(\overline{\mathbf{x}}_0) - F^*)}{\alpha} - \frac{1}{2}\sum_{k=0}^{K-1}\mathbb{E}[\|\overline{\nabla \mathbf{f}}_k\|^2] + \ell^2\sum_{k=0}^{K-1}\mathbb{E}\left[\frac{\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2}{n}\right] + \alpha\ell\nu_a^2 K. \tag{20}$$

From Assumption 1, for $K \geq 1$, we also have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_k^i)\|^2] &\leq \frac{2}{n}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_k^i) - \nabla F(\overline{\mathbf{x}}_k)\|^2 + \|\nabla F(\overline{\mathbf{x}}_k)\|^2\right] \\ &\leq 2\ell^2\sum_{k=0}^{K-1}\mathbb{E}\left[\frac{\sum_{i=1}^{n}\|\mathbf{x}_k^i - \overline{\mathbf{x}}_k\|^2}{n}\right] + 2\sum_{k=0}^{K-1}\|\nabla F(\overline{\mathbf{x}}_k)\|^2 \\ &\leq 2\ell^2\sum_{k=0}^{K-1}\mathbb{E}\left[\frac{\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2}{n}\right] + 2\sum_{k=0}^{K-1}\|\nabla F(\overline{\mathbf{x}}_k)\|^2. \end{aligned} \tag{21}$$

By substituting (20) for (21), we obtain

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_k^i)\|^2] \leq \frac{4(F(\overline{\mathbf{x}}_0) - F^*)}{\alpha} - \sum_{k=0}^{K-1}\mathbb{E}[\|\overline{\nabla \mathbf{f}}_k\|^2] + 4\ell^2\sum_{k=0}^{K-1}\mathbb{E}\left[\frac{\|\mathbf{x}_k - \mathbf{J}\mathbf{x}_k\|^2}{n}\right] + 2\alpha\ell\nu_a^2 K. \tag{22}$$

Then, for $0 < \alpha \leq \min\left\{\frac{1}{2\ell}, \frac{1-\lambda^2}{72\lambda^2\ell}, \frac{\sqrt{5}(1-\lambda^2)}{\sqrt{32}\lambda\ell}, \frac{\sqrt{5}(1-\lambda^2)^2}{192\lambda^2\ell^2}\right\}$ and $0 \leq E_k^u \leq \frac{5}{64}\frac{1-\lambda^2}{\lambda^2\lambda_L\ell}$, we have

$$\begin{aligned}\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_k^i)\|^2] &\leq C_{12}^* + \frac{C_{13}^*}{K} + \frac{C_{14}^*}{K}\sum_{k=0}^{K-1}(E_{k+1}^u)^2 + \frac{C_{15}^*}{K}\sum_{k=0}^{K-1}(E_{k+2}^v)^2 \\ &\quad - (1 - 4\ell^2 C_8^*)\sum_{k=0}^{K-1}\mathbb{E}[\|\overline{\nabla \mathbf{f}}_k\|^2], \end{aligned} \tag{23}$$

where

$$C_{12}^* = 4\ell^2 C_7^* + 2\alpha\lambda\ell\nu_a^2,$$
$$C_{13}^* = \frac{4(F(\overline{\mathbf{x}}_0) - F^*)}{\alpha} + 4\ell^2 C_{11}^*,$$
$$C_{14}^* = 4\ell^2 C_9^*, \quad C_{15}^* = 4\ell^2 C_{10}^*.$$

Moreover, if $0 < \alpha \leq \frac{\sqrt{5}(1-\lambda^2)^2}{192\lambda^2\ell^2}$, then $1 - 4\ell^2 C_8^* \geq 0$ holds. This concludes the proof. □

From the proof of Theorem 1, if $\alpha = \frac{C}{\sqrt{K}}$, $E_k^u = O\left(\frac{1}{\sqrt{k}}\right)$, and $E_k^v = O\left(\frac{1}{\sqrt{k}}\right)$, we further have

$$\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_k^i)\|^2\right] = O\left(\frac{1}{\sqrt{K}}\right), \tag{24}$$

where $O(\cdot)$ represents the big-O notation and C is a positive constant. This implies that the proposed algorithm can achieve sublinear convergence by appropriately tuning the step size and the trigger threshold.

The constants C_p^* $(p = 1, 2, \ldots, 15)$ in the proofs of Lemmas 2–4 and Theorem 1 are closely related to the convergence properties of the algorithm. For example, the regret bound depends on these constants with the network-related parameters λ and λ_L and the smoothness parameter of the cost function ℓ. Further investigation of the relationship between these constants and the convergence property is a future direction of this paper.

4. Numerical Example

This section presents a numerical example of the proposed

Table 1 Trigger thresholds.

	E1	E2	E3
$E_v^i(k)$	$\frac{6000}{\sqrt{k+100}}$	$\frac{9000}{\sqrt{k+100}}$	$\frac{12000}{\sqrt{k+100}}$
$E_u^i(k)$	$\frac{10}{\sqrt{k+10}}$	$\frac{60}{\sqrt{k+10}}$	$\frac{100}{\sqrt{k+10}}$

algorithm. We consider a multiagent system with 3 agents whose cost functions are given by

$$f_1(x) = \begin{cases} x^4 + 6x^3 - 40x^2 + 6x & \text{if } |x| \le R, \\ (4R^3 + 18R^2 - 80R + 6)x & \\ \quad -3R^4 - 12R^3 + 40R^2, & \text{if } R < x, \\ \left(-4R^3 + 18R^2 + 80R + 6\right) x & \\ \quad -3R^4 + 12R^3 + 40R^2, & \text{if } x < -R, \end{cases}$$

$$f_2(x) = 80x,$$

$$f_3(x) = \begin{cases} x^4 + 2x^3 - 24x^2 + 10x + 8 & \text{if } |x| \le R, \\ \left(4R^3 + 6R^2 - 48R + 10\right) x & \\ \quad -3R^4 - 4R^3 - 24R^2 + 8, & \text{if } R < x, \\ \left(-4R^3 + 6R^2 + 48R + 10\right) x & \\ \quad -3R^4 + 4R^3 - 24R^2 + 8, & \text{if } x < -R, \end{cases}$$

where $R = 1000$. In this example, the step size is given by $\alpha = 0.005$.

First, we compare the performance of the proposed algorithm for different trigger thresholds $E1$, $E2$, and $E3$, which are shown in Table 1. Figure 1 shows the time-averaged gradient $\frac{1}{nk}\sum_{i=1}^{n}\sum_{\tau=0}^{k-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_\tau^i)\|^2\right]$, where k is the iteration of the algorithm and Time-Triggered represents the result of the stochastic gradient descent algorithm with gradient-tracking [15], which uses time-triggered communication. Figure 2 shows the total number of communications per agent. Figure 1 shows that the time-averaged gradient approaches 0 by the proposed algorithm. In particular, the case with the threshold E1 is comparable to the time-triggered algorithm, while the cases with E2 and E3 exhibit slower convergence. Moreover, Fig. 2 shows that the event-triggered algorithm can reduce the total number of communications compared to the time-triggered algorithm. Figure 2 shows that the case with E1 requires fewer communications than the other cases. In general, the higher the threshold value is, the lower the number of communications is. However, in the present condition, the opposite occurs. This may be because the thresholds for E2 and E3 are set too high, and it takes time for the agents to reach an agreement. This indicates that if the threshold is set appropriately, the number of communications can be significantly reduced while maintaining the same convergence as a time-triggered algorithm.

Next, we compare the performance of the proposed gradient-tracking algorithm (GTDSGD) with the distributed stochastic gradient descent algorithm (DSGD) [3]. In this example, the trigger threshold of the event-triggered algorithm is set as E1 and the step size for the DSGD algorithm is set as $\alpha = 0.003$. Figures 3 and 4 show the time-averaged gradient and the total number of communications per agent, respectively. Figure 4 shows that the proposed

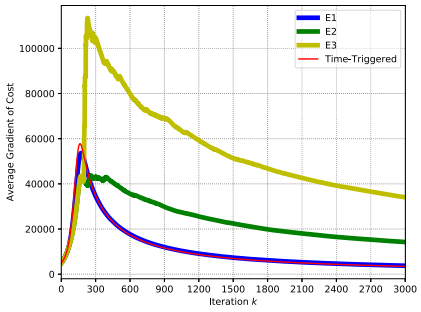

Fig. 1 Comparison of the time-averaged gradient.

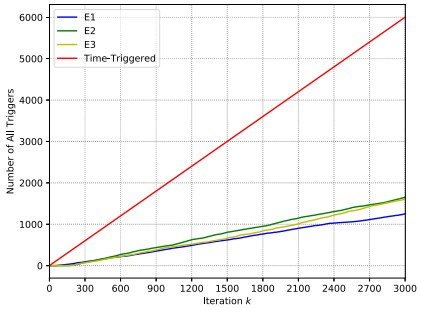

Fig. 2 Total number of communications per agent.

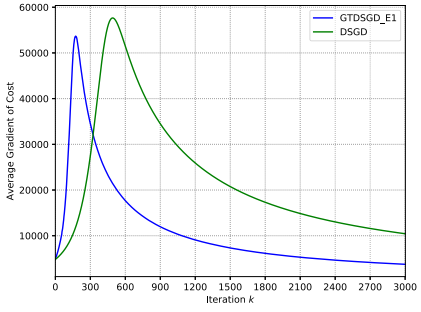

Fig. 3 Comparison of the time-averaged gradient.

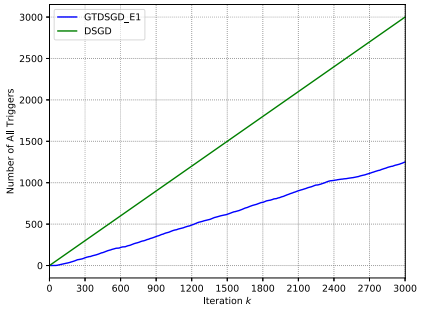

Fig. 4 Total number of communications per agent.

event-triggered algorithm requires fewer total communications than the time-triggered DSGD algorithm. Figure 3 shows that the GTDSGD algorithm converges faster than the DSGD algorithm even though the number of communications is small. This is due to the improved convergence rate achieved by the gradient-tracking step of the proposed

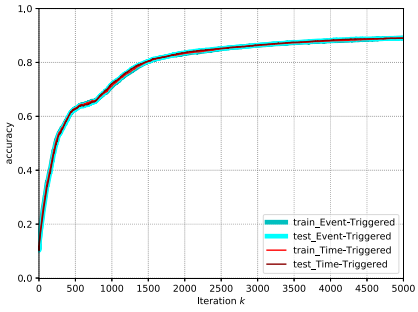

Fig. 5 Accuracy rate.

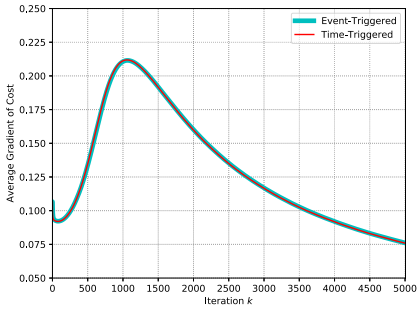

Fig. 6 Comparison of the time-averaged gradient.

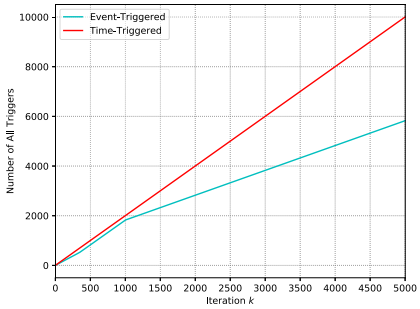

Fig. 7 Total number of communications per agent.

algorithm.

Finally, we consider an application to a distributed multiclass logistic regression problem with 10 agents for the MNIST dataset [28]. The image feature vector of agent i is given by $\mathbf{h}^i_{(k)} \in \mathbb{R}^{784}$. Let $\mathbf{X}^i_{(k)} \in \mathbb{R}^{784\times 10}$ be the weight matrix of agent i at iteration k, whose elements are estimated by the distributed online algorithms after being vectorized as $\mathbf{x}^i_{(k)}$. The output of agent i at iteration k is given by the softmax function $\mathbf{p}^i_{(k)} = e^{\check{\mathbf{x}}^i_{(k)}}/\sum_{\ell=1}^{10} e^{[\check{\mathbf{x}}^i_{(k)}]_\ell}$, where $\check{\mathbf{x}}^i_{(k)} = (\mathbf{X}^i_{(k)})^\mathsf{T}\mathbf{h}^i_{(k)} \in \mathbb{R}^{10}$ and $[\mathbf{a}]_\ell$ is the ℓ-th element of the vector $\mathbf{a}$. Then, the local cost function of agent i at iteration k is given by the loss function $f_i(\mathbf{x}^i_{(k)}) = -\sum_{m=1}^{M}\sum_{c=1}^{10}[\mathbf{q}^{i,m}_{(k)}]_c \log[\mathbf{p}^{i,m}_{(k)}]_c$, where $\mathbf{p}^{i,m}_{(k)}$ is the output for the m-th data value assigned to agent i at iteration k, $\mathbf{q}^{i,m}_{(k)}$ is the label for the data $\mathbf{p}^{i,m}_{(k)}$, and M is the batch size of each agent. In this example, 60,000 images are used for training, and 10,000 images are used as the test data. The batch size is given as $M = 100$. The step size and the trigger thresholds are given as $\alpha = 0.005$ and

$$E^i_v(k) = \begin{cases} 0 & k \le 1000, \\ \frac{20}{(k+10^4)^{0.26}} & k > 1000, \end{cases}$$
$$E^i_u(k) = \frac{0.01}{(k+100)^{0.26}}.$$

Figure 5 shows the accuracy rate for the test data with the proposed event-triggered algorithm and the time-triggered stochastic gradient-tracking algorithm [15]. Figure 6 shows the time-averaged gradient. From these results, the accuracy of the event-triggered algorithm is comparable to that of the time-triggered algorithm. Figure 7 shows the total number of communications per agent. In this numerical example, the number of communications with the proposed algorithm can be reduced by more than 40% compared to that with the time-triggered algorithm.

5. Conclusion

In this paper, we presented a distributed event-triggered method for nonconvex optimization on multiagent networks. We proposed a stochastic gradient-tracking algorithm by which the estimation of every agent converges to a critical point. We showed that the proposed algorithm achieves a sublinear convergence rate by appropriately setting the step size and the trigger threshold. We also showed that the trigger times can be effectively reduced by event-triggered communication compared to the time-triggered method. An extension to a more general network topology is one of our future research directions.

Acknowledgments

This work is supported in part by JSPS KAKENHI Grant Number JP21K04121.

References

[1] K. Sakurama and T. Sugie, "Generalized coordination of multi-robot systems," Foundations and Trends in Systems and Control, vol.9, no.1, pp.1–170, 2021.
[2] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," IEEE Trans. Autom. Control, vol.54, no.1, pp.48–61, 2009.
[3] S.S. Ram, A. Nedić, and V.V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," J. Optim. Theory Appl., vol.147, no.3, pp.516–545, 2010.
[4] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," Proc. 55th IEEE Conference on Decision and Control, pp.159–166, 2016.
[5] A. Nedić, A. Olshevsky, and M.G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," Proc. IEEE, vol.106, no.5, pp.953–976, 2018.
[6] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K.H. Johansson, "A survey of distributed optimization," Annual Reviews in Control, vol.47, pp.278–305, 2019.
[7] R. Adachi and Y. Wakasa, "Distributed filter using ADMM for optimal estimation over wireless sensor network," IEICE Trans. Fundamentals, vol.E105-A, no.11, pp.1458–1465, Nov. 2022.

[8] N. Hayashi and K. Sakurama, "Communication-aware distributed rebalancing for cooperative car-sharing service," IET Control Theory & Applications, vol.17, no.7, pp.850–867, 2023.
[9] M. Zhu and S. Martínez, "An approximate dual subgradient algorithm for multi-agent non-convex optimization," IEEE Trans. Autom. Control, vol.58, no.6, pp.1534–1539, 2013.
[10] P.D. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," IEEE Trans. Signal Inf. Process. over Netw., vol.2, no.2, pp.120–136, 2016.
[11] T. Tatarenko and B. Touri, "Non-convex distributed optimization," IEEE Trans. Autom. Control, vol.62, no.8, pp.3744–3757, 2017.
[12] X. Jiang, X. Zeng, J. Sun, and J. Chen, "Distributed proximal gradient algorithm for non-convex optimization over time-varying networks," IEEE Trans. Control Netw. Syst., vol.10, no.2, pp.1005–1017, 2022.
[13] S. Vlaski and A.H. Sayed, "Distributed learning in non-convex environments — Part I: Agreement at a linear rate," IEEE Trans. Signal Process., vol.69, pp.1242–1256, 2021.
[14] S. Vlaski and A.H. Sayed, "Distributed learning in non-convex environments — Part II: Polynomial escape from saddle-points," IEEE Trans. Signal Process., vol.69, pp.1257–1270, 2021.
[15] R. Xin, U.A. Khan, and S. Kar, "An improved convergence analysis for decentralized online stochastic non-convex optimization," IEEE Trans. Signal Process., vol.69, pp.1842–1858, 2021.
[16] J. Gao, X.-W. Liu, Y.-H. Dai, Y. Huang, and J. Gu, "Distributed stochastic gradient tracking methods with momentum acceleration for non-convex optimization," Comput. Optim. Appl., vol.84, pp.531–572, 2022.
[17] Y. Kajiyama, N. Hayashi, and S. Takai, "Distributed subgradient method with edge-based event-triggered communication," IEEE Trans. Autom. Control, vol.63, no.7, pp.2248–2255, 2018.
[18] S. Liu, L. Xie, and D.E. Quevedo, "Event-triggered quantized communication-based distributed convex optimization," IEEE Trans. Control Netw. Syst., vol.5, no.1, pp.167–178, 2018.
[19] K. Ishikawa, N. Hayashi, and S. Takai, "Consensus-based distributed particle swarm optimization with event-triggered communication," IEICE Trans. Fundamentals, vol.E101-A, no.2, pp.338–344, Feb. 2018.
[20] N. Hayashi, T. Sugiura, Y. Kajiyama, and S. Takai, "Distributed event-triggered algorithm for unconstrained convex optimization over weight-balanced directed networks," IET Control Theory & Applications, vol.14, no.2, pp.253–261, 2020.
[21] S. Ghosh, B. Aquino, and V. Gupta, "EventGraD: Event-triggered communication in parallel machine learning," Neurocomputing, vol.483, pp.474–487, 2022.
[22] G. Carnevale, I. Notarnicola, L. Marconi, and G. Notarstefano, "Triggered gradient tracking for asynchronous distributed optimization," Automatica, vol.147, p.110726, 2023.
[23] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus," Automatica, vol.105, pp.1–27, 2019.
[24] K. Kitamura, K. Kobayashi, and Y. Yamashita, "LMI-based design of output feedback controllers with decentralized event-triggering," IEICE Trans. Fundamentals, vol.E105-A, no.5, pp.816–822, May 2022.
[25] J. George and P. Gurram, "Distributed deep learning with event-triggered communication," arXiv preprint arXiv:1909.05020, 2019.
[26] T. Adachi, N. Hayashi, and S. Takai, "Distributed gradient descent method with edge-based event-driven communication for non-convex optimization," IET Control Theory & Applications, vol.15, no.12, pp.1588–1598, 2021.
[27] S. Mao, Z. Dong, W. Du, Y.-C. Tian, C. Liang, and Y. Tang, "Distributed nonconvex event-triggered optimization over time-varying directed networks," IEEE Trans. Ind. Inf., vol.18, no.7, pp.4737–4748, 2022.
[28] L. Deng, "The MNIST database of handwritten digit images for machine learning research," IEEE Signal Process. Mag., vol.29, no.6, pp.141–142, 2012.

Daichi Ishikawa received a B.E. and M.E. from Osaka University in 2020 and 2022. His research interests include distributed optimization in multiagent systems.

Naoki Hayashi received B.E., M.E., and Ph.D. degrees from Osaka University in 2006, 2008, and 2011, respectively. He was a Research Assistant at Kyoto University in 2011. From 2012 to 2020, he was an Assistant Professor at Osaka University. He is currently an Associate Professor at Osaka University. His research interests include cooperative control and distributed optimization. He is a member of ISCIE, SICE, and IEEE.

Shigemasa Takai received B.E. and M.E. degrees from Kobe University in 1989 and 1991, respectively, and a Ph.D. degree from Osaka University in 1995. From 1992 to 1998, he was a Research Associate at Osaka University. He joined Wakayama University as a Lecturer in 1998 and became an Associate Professor in 1999. From 2004 to 2009, he was an Associate Professor at Kyoto Institute of Technology. Since 2009, he has been a Professor at Osaka University. His research interests include supervisory control and fault diagnosis of discrete event systems. He is a member of ISCIE, SICE, and IEEE.