KYBERNETIKA — VOLUME *61* (2025), NUMBER 6, PAGES 762–788

ET-DMGING: EVENT-TRIGGERED DISTRIBUTED MOMENTUM-GRADIENT TRACKING OPTIMIZATION ALGORITHM FOR MULTI-AGENT SYSTEMS

Aijuan Wang, Xingmeng Tan and Hai Nan

This paper proposes an event-triggered distributed momentum-gradient tracking optimization algorithm (ET-DMGing) for the collaborative optimization problem of minimizing the sum of all agents' local objective functions in multi-agent systems. Firstly, gradient tracking is employed to precisely track the average momentum gradient for updating agent states, which effectively reduces their dwell time in flat and oscillatory regions. The proposed ET-DMGing exhibits enhanced directional consistency and dynamic stability during optimization by leveraging momentum accumulation effects, achieving a linear convergence rate. Secondly, a new event-triggered condition is proposed, which considers the dual metrics of state error and momentum gradient error. This allows for a more comprehensive assessment of the agents' triggering needs, avoiding instability caused by single-dimensional triggering, and improving the triggering threshold. This event-triggered condition reduces the communication frequency among agents. Thirdly, we rigorously proved that the proposed ET-DMGing converges to the global optimum at a linear convergence rate by employing the small-gain theorem. Furthermore, explicit convergence conditions have been derived for parameter selection, including step size parameters and event-triggered weighting coefficients. Finally, numerical simulations are performed to verify the effectiveness and accuracy of the theoretical results.

Keywords: gradient tracking, event-triggered mechanism, multi-agent systems, distributed optimization

Classification: 68W15, 93D05, 93D21

1. INTRODUCTION

Traditional centralized algorithms face high computational burdens and limited adaptability in multi-agent system optimization, primarily due to their inherent complexity, large-scale nature, and solution complexity [17, 31, 32]. In contrast, distributed optimization algorithms decompose complex and large-scale problems into simpler subproblems. These algorithms provide efficient and stable solutions for large-scale optimization challenges [25] such as cooperative control systems [11], sensor-networks coordination [26], and energy management [12]. Numerous studies have been developed distributed algorithms for multi-agent optimization [9, 18, 28, 29, 30, 34].

DOI: 10.14736/kyb-2025-6-0762

Existing distributed optimization algorithms fall into two categories: diminishing-step-size algorithms [3, 6, 10, 21] and fixed-step-size algorithms [4, 20, 27]. The primary challenge for diminishing-step-size algorithms stems from the difficulty in designing effective decay strategies, resulting in slower convergence rates. Consequently, fixed-step-size algorithms have attracted significant attention due to their structural simplicity and rapid convergence rates. For example, the exact first-order algorithm (EXTRA) [27] incorporates correction terms into decentralized gradient descent. By leveraging a differential structure to eliminate steady-state errors, EXTRA achieved exact convergence with fixed-step-sizes. The stochastic average gradient algorithm (SAGA) [4] integrates stochastic gradient descent with averaged gradient techniques, achieving rapid linear convergence. The predictability and stability of SAGA are enhanced by its fixed-step-size design. Similarly, the algorithm in [20] addresses mixed equilibrium problems under multiple constraints by integrating mirror descent, primal-dual methods, and consensus protocols. This integration ensures asymptotic convergence under fixed-step-size. Many classical distributed gradient tracking algorithms utilize fixed-step-sizes [22, 24]. By integrating consensus-based distributed gradient descent (DGD) with an innovative gradient estimation scheme, these algorithms exploit historical data to achieve fast and precise average gradient estimates. The study in [24] explored the convergence rate of fixed-step-size gradient tracking algorithms under convexity and smoothness conditions, eliminating the need for strong convexity assumptions. The distributed inexact gradient tracking algorithm (DIGing) [22] examines strong convexity and smoothness conditions. The results in [22] demonstrate linear convergence when the fixed-step-size lies within upper bounds, effectively addressing engineering challenges in directed graphs. Building upon [22], the study in [15] incorporates stochastic average gradient technique to propose a fixed-step-size distributed stochastic gradient tracking algorithm (S-DIGing), and provides a novel primal-dual interpretation to prove linear convergence to the global optimum under specific conditions. The distributed gradient tracking algorithm with variance reduction (GT-VR) [8] employs fixed-step-size gradient tracking and variance reduction techniques, introducing Bernoulli distribution into stochastic variance reduced gradient (SVRG) methods to address non-convex optimization. It achieves a convergence rate of $O(1/k)$ and operates without steady-state errors. Furthermore, the study in [16] develops a unified algorithmic framework from a primal-space perspective, which is essentially a generalized gradient tracking method and unifies most existing fixed-step-size distributed optimization algorithms. However, existing studies lack explicit acceleration mechanisms capable of circumventing local oscillations and preventing convergence to shallow local optima. Momentum methods [23] are widely adopted accelerate the convergence of first-order algorithms. The accumulation of historical gradient information causes the agent's state updates to exhibit inertia, helping the algorithm avoid shallow local optima, reducing oscillations during convergence, and accelerating overall convergence. Consequently, researchers have integrated momentum methods with gradient tracking algorithms [1, 7]. For instance, the distributed stochastic momentum tracking (DSMT) algorithm [7] is a single-loop framework that integrates momentum methods, gradient tracking techniques, and loopless Chebyshev acceleration. The gradient tracking with adaptive momentum estimation (GTAdam) distributed algorithm [1] integrates gradient tracking techniques with first- and second-order momentum estimations,

and has demonstrated superior convergence rates. In summary, while the fixed-step-size design simplifies the algorithmic structure, excessively large step sizes may induce oscillations. In gradient-based optimization algorithms, the step size determines the magnitude of updates in each iteration. If set too large, the optimization variable may overshoot the optimal solution region, resulting in repeated oscillations around the optimal point. This behavior undermines convergence stability, slows down the convergence rate, and may even compromise the final optimization result.To address this issue, this paper introduces a momentum gradient tracking methods leverages global historical gradient information to suppress oscillations and enhance convergence rates. This motivates the proposal a momentum-gradient tracking distributed optimization algorithm with fixed-step-sizes, which aims to achieve accelerated convergence while minimizing communication-computation overhead.

In summary, while the fixed-step-size design simplifies the algorithmic structure, excessively large step sizes may cause the variables to oscillate around the optimal point, leading to degraded convergence speed and accuracy. To mitigate this issue, this paper introduces a momentum gradient tracking methods leverages global historical gradient information to suppress oscillations and enhance convergence rates. This motivates the proposal a momentum-gradient tracking distributed optimization algorithm with fixed-step-sizes, which aims to achieve accelerated convergence while minimizing communication-computation overhead.

On the other hand, distributed optimization algorithms rely on inter-agent information exchange to achieve convergence and obtain optimal solutions. However, real-time continuous communication is impractical in real-world scenarios and entails excessive communication overhead. Event-triggered strategies, as a non-real-time control framework [13], demonstrate enhanced compatibility with practical application environments while maintaining resource efficiency, which has attracted substantial research attention [2, 5, 14, 19, 33]. For instance, the event-triggered mechanism in [19] minimizes communication costs by designing dynamic quantizers and broadcast protocols adapted to each agent's bandwidth constraints. However, dynamic quantizers introduce architectural complexity, thereby increasing computational and communication overhead in practical implementations. The event-triggered conditions in [14] are highly dependent on time parameters and impose numerous constraints on other parameter settings. In this regard, the Extended DIGing algorithm proposed in [5] employed an innovative event-triggered condition that depends solely on the local state and sporadic neighbor states. Although its key parameter selection is straightforward, single-dimensional measurement errors can induce frequent false triggering. To address the high reliance on time parameters, elevated complexity, and sensitivity to single error in existing event-triggered conditions, it inspires us to propose an effective event-trigger function aimed at reducing communication costs, minimizing the impact of single errors, and lowering the complexity of event-triggered conditions.

Based on the above discussions, this paper proposes an event-triggered distributed momentum-gradient tracking optimization algorithm for multi-agent systems. The proposed algorithm achieves exact convergence to the global optimum at a linear rate while reducing communication frequency. Compared to the algorithm in [5], the proposed ET-DMGing algorithm exhibits a lower communication frequency and faster convergence.

The main contributions of this paper are as follows:

(1) We propose a fixed-step-size momentum gradient tracking method that accurately tracks the average momentum gradient, thereby accelerating the convergence rate. The accumulation effect of momentum ensures consistent and stable state update directions, reduces oscillations during the optimization process, and effectively prevents convergence to shallow local optima. Experimental results demonstrate a 35.93% reduction in total iteration time compared to the method in [5].

(2) We propose an effective event-triggered condition that relies only on neighbors' sporadic states. This condition is defined as the sum of dual variations in state error and momentum gradient error, taking into account both the magnitude of the agent's own state change and the joint advantages and disadvantages of the momentum gradient bias. Theoretical analysis demonstrates that the proposed constraints are stricter than those in [5]. By increasing the triggering threshold, this approach avoids frequent activations caused by single-error satisfaction and reduces communication volume by 82.32%.

(3) Finally, we employ the small gain theorem to prove that the proposed ET-DMGing algorithm can achieves exact convergence to the global optimal solution at a linear rate. Additionally, convergence conditions about the associated parameters are derived. Simulation results including algorithm convergence analysis, event-triggered mechanism efficiency, and comparative performance validation with baseline methods further validate the effectiveness and correctness of the theoretical results.

The remainder of this paper is organized as follows. Section 2 introduces fundamental preliminary concepts covering notations, graph theory, gradient tracking algorithms, and momentum methods. Section 3 details the algorithm's workflow and distributed event-triggered mechanisms. Section 4 presents the main theorems and rigorous convergence analysis. Section 5 demonstrates numerical simulations for empirical validation of the main theorems. Section 6 provides the conclusions of the paper, summarizing key contributions and future directions.

2. PRELIMINARIES AND PROBLEM FORMULATION

2.1. Notations description

Let $\mathbb{R}$ be the set of real numbers, $\mathbb{R}^n$ be the set of n-dimensional real column vectors, $\mathbf{1}_n$ represent the $n\times 1$ vector with all elements equal to 1, 0_n denote the $n\times 1$ vector with all elements equal to 0, I denotes the unit matrix of $n\times n$, let $J=\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$. For a vector ν, $\|\nu\|$ denotes the Euclidean norm of the vector, and v^T denotes the transpose of the vector ν. For a matrix A, $\sigma_{\max}(A)$ represents its maximum singular value. For a differentiable function $f(x)$, denote by $\nabla f(x)$ its gradient at x. The subscript i denotes the ith agent and $x_i(k)$ denotes the state variable of agent i at the kth iteration. $\langle x,y\rangle$ denotes the inner product of x and y. for an infinite sequence $s_i=\{s_i(0),s_i(1),s_i(2),\ldots\}$, where $s_i(k)\in\mathbb{R}^N$. $\forall k$ such that $\|s_i\|^{\lambda,K}=\max_{k=0,1,\ldots K}\frac{1}{\lambda^k}\|s_i(k)\|$, and $\|s_i\|^{\lambda}=\sup_{k\geq 0}\frac{1}{\lambda^k}\|s_i(k)\|$, where $\lambda\in(0,1)$.

2.2. Network model

Let $G = \{\nu, \varepsilon, A\}$ be defined as an undirected graph composed of n agents, where $\nu = \{\upsilon_1, \upsilon_2, \ldots, \upsilon_n\}$ represents the set of agents, $\varepsilon \in \nu \times \nu$ denotes the set of edges, and $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is the adjacency matrix of G. An edge $e_{ij} = (\upsilon_j, \upsilon_i)$ indicates bidirectional communication between agents i to j. The adjacency matrix A is symmetric, implying mutual accessibility between connected agents. If a bidirectional communication channel exists between agent i and agent j, they are referred to as neighbors, corresponding to $a_{ij} = a_{ji} = 1$, otherwise, they correspond to $a_{ij} = a_{ji} = 0$. The neighbor set of agent i is denoted by N_i. Let n represent the total number of agents in the network, and $|N_i|$ represent the number of neighbors of agent i. The Laplacian matrix of A is defined as $L = (\ell_{ij}) \in \mathbb{R}^{n\times n}$, where $l_{ij} = -a_{ij}$, $i \neq j$, and $\ell_{ii} = \sum_{j=1, i\neq j}^{n} a_{ij}$. L is a symmetric positive semi-definite matrix. The matrix $W = (w_{ij}) \in \mathbb{R}^{n\times n}$ is a doubly stochastic matrix, where all elements are non-negative real numbers, and the sum of the elements in each row and each column is equal to 1, i. e., $w_{ij} \geq 0$ for all i, j, and $\sum_{i=1}^{n} w_{ij} = \sum_{j=1}^{n} w_{ij} = 1$.

2.3. Distributed optimization

This paper addresses an unconstrained optimization problem, aiming to solve it collaboratively through a network of n agents.

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where $f(x) : \mathbb{R}^n \to \mathbb{R}$ denotes the global differentiable convex objective function, and $f_i(x) : \mathbb{R}^n \to \mathbb{R}$ denotes the agent-specific localized convex objective function for agent i. Each agent's local objective function is unique to agent i. The goal is to find the globally optimal solution x^* through local computations and information exchanges among the agents. We make the following assumptions on objective functions in (1):

Assumption 2.1. (Connectivity) Assume that the undirected graph G is connected.

Assumption 2.2. (Doubly Stochasticity) Suppose Assumption 1 hold, let $0 < h < \frac{1}{d^*}$, where $\frac{1}{d^*} = \max_{i=1}^{n}\{d_i\}$ and $d_i = \sum_{j=1}^{n} a_{ij}$, such that the matrix $W = I - hL$ is doubly stochastic, and $\delta = \sigma_{\max}(W - J) < 1$.

Assumption 2.3. (Smoothness) For each agent i, $i = 1, \ldots n$, the local objective function f_i is differentiable, and its gradient is Lipschitz continuous. For all $x, y \in \mathbb{R}^n$, we have:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq l_i \|x - y\|,$$

where $l_i > 0$ is partially defined, we will use $\hat{L} = \max_{i=1}^{n}\{l_i\}$ and $\overline{L} = \frac{1}{n}\sum_{i=1}^{n} l_i$ in our analysis.

Assumption 2.4. (Strong Convexity) For each agent i, $i = 1, \ldots n$, the local objective function f_i is strongly convex, meaning that for all $x, y \in \mathbb{R}^n$, we have:

$$f_i(x) \geq f_i(y) + \langle \nabla f_i(y), (x - y)\rangle + \frac{\mu_i}{2}\|x - y\|^2,$$

where $\mu_i > 0$ is partially defined, we will use $\hat{\mu} = \max_{i=1}^{n} \{\mu_i\}$ and $\bar{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mu_i$ in our analysis.

2.4. DIGing algorithm

The DIGing algorithm Nedic et al. [22] is highly effective in solving distributed optimization problems and achieves exact convergence to the optimal solution at a linear rate. Given initial states $x_i(0) \in \mathbb{R}^n$ and $y_i(0) = \nabla f_i(x_i(0))$ for each agent i, the iterative update rules are:

$$x_i(k+1) = x_i(k) + hu_i(k) - \alpha y_i(k), \tag{2}$$

$$y_i(k+1) = y_i(k) + hv_i(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)), \tag{3}$$

$$u_i(k) = \sum_{j \in N_i} a_{ij} (x_j(k) - x_i(k)), \tag{4}$$

$$v_i(k) = \sum_{j \in N_i} a_{ij} (y_j(k) - y_i(k)), \tag{5}$$

where N_i denotes the set of all neighboring agents of agent i. $x_i(k) \in \mathbb{R}^n$ denotes the state estimate of agent i, $y_i(k) \in \mathbb{R}^n$ denotes the average gradient estimate of agent i, h represents a positive control coefficient, and $u_i(k)$, $v_i(k)$ are two auxiliary variables. In the $k+1$th iteration, each agent updates its state estimate and average gradient estimate using its own kth iteration values $x_i(k)$ and $y_i(k)$, the received values $x_j(k)$ and $y_j(k)$ from neighboring agents, and the gradient difference term $\nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k))$. Finally, each agent broadcasts its updated state estimate $x_i(k+1)$ and average gradient estimate $y_i(k+1)$ to all neighboring agents and receives $x_j(k+1)$ and average gradient estimates $y_j(k+1)$ from its neighbors.

2.5. Momentum method

The momentum method [23] substitutes the original gradient value with the exponentially weighted moving average of historical gradients. This method integrates historical gradient information using exponential weights. When the gradient exhibits rapid directional variations, the state update step size progressively increases in that direction; otherwise, it decreases. By considering the influence of historical gradient trends, the momentum method reduces the sensitivity of the optimization trajectory to the gradient changes of a single iteration, thereby suppressing oscillations in the agent state estimates. This approach facilitates escaping shallow local optima and accelerates the convergence. The momentum update rule are

$$x_i(k+1) = x_i(k) - \alpha y_i(k), y_i(k+1) = \beta y_i(k) + (1-\beta) \nabla f(x_i(k+1)),$$

where $\beta \in (0,1)$ represents the momentum coefficient. With the aid of the β, the influence of historical gradient information on the algorithm's update can be controlled more effectively.

3. ET-DMGING ALGORITHM DESIGN

Considering that the accumulated effect of historical gradients can accelerate convergence, we introduce the momentum method into the distributed gradient tracking algorithm. By employing gradient tracking techniques, the tracked entity shifts from the average gradient to the average of the historically weighted gradients, termed the average momentum gradient. This method reduces oscillations, resulting in enhanced stability and speed throughout the iterative process.

Based on this, an event-triggered mechanism is introduced to reduce communication costs, where each agent i follows a sequence of event-triggered times $\left\{0 = t_0^i, t_1^i, t_2^i, \ldots\right\}$. Thus, the two auxiliary variables in (4) and (5) must be adjusted based on event-triggered times.

$$u_i(k) = \sum_{j \in N_i} a_{ij} \left(x_j\left(t_{k_j}^j\right) - x_i\left(t_{k_i}^i\right)\right), \tag{6}$$

$$v_i(k) = \sum_{j \in N_i} a_{ij} \left(y_j\left(t_{k_j}^j\right) - y_i\left(t_{k_i}^i\right)\right), \tag{7}$$

where $k \in \left[t_{k_i}^i, t_{k_i+1}^i\right)$, $x_i\left(t_{k_i}^i\right)$ denote the state estimates of agent i at the time of the most recent event trigger, while $y_i\left(t_{k_i}^i\right)$ denotes the average momentum gradient estimate of agent i at the same moment. Here, $j \in N_i$ and N_i indicates a neighbor of agent i, where j is the neighbor set. Consequently, the update formula for the ET-MDIGing algorithm can be expressed as follows:

$$x_i(k+1) = x_i(k) + hu_i(k) - \alpha y_i(k), \tag{8}$$

$$g_i(k+1) = (1-\beta)\, g_i(k) + \beta \nabla f_i\left(x_i(k+1)\right), \tag{9}$$

$$y_i(k+1) = y_i(k) + hv_i(k) + g_i(k+1) - g_i(k), \tag{10}$$

where $x_i(k+1)$ denotes the state estimate of agent i at $k+1$th iterations, and $g_i(k+1)$ represents the momentum gradient of agent i at $k+1$th iterations. α is step size, h is a positive control coefficient. $\beta \in (0,1)$ is the momentum coefficient, h is a positive control coefficient. The tracking of the average momentum gradient is effectively facilitated by (10).

We define two measurement errors $e_i^x(k) = x_i\left(t_{k_i}^i\right) - x_i(k)$ and $e_i^y(k) = y_i\left(t_{k_i}^i\right) - y_i(k)$. $t_{k_i}^i$ denotes the most recent triggered time of agent i, the $e_i^x(k)$ quantifies the deviation of the state estimate at the most recent triggered time from that at the kth iteration. The $e_i^y(k)$ quantifies the deviation of the average momentum gradient estimate at the last triggered time from that at the kth iteration. Notably, both measurement errors equal zero when an event is triggered.

Considering that the state error of agents reflects their consensus level, while the average momentum gradient error indicates the accuracy of achieving the optimal solution, relying solely on the state error may lead the algorithm to primarily focus on synchronization among agents, thereby neglecting the precision of the global optimum. Conversely, exclusive reliance on the average momentum gradient error may lead frequent triggers if consensus among agents is not fully established. By incorporating both errors, a more

comprehensive representation of the algorithm's operational status is achieved, mitigating the biases associated with relying solely on either error. This approach enhances the event-triggered threshold, ensuring performance while simultaneously reducing communication overhead. Therefore, the following event-triggered condition is proposed:

$$t_{k+1}^{i} = \inf\left\{t | t > t_{k_i}^{i}, flag_1\left(e_i^x\right) + flag_2\left(e_i^y\right) \geq 0\right\}, \tag{11}$$

$$\begin{aligned} flag_1\left(e_i^x\right) &= \left\|e_i^x\left(k\right)\right\|^2 - \gamma_1 h \sum_{j \in N_i} \left\|x_j\left(t_{k_j}^{j}\right) - x_i\left(t_{k_i}^{i}\right)\right\|^2, \\ flag_2\left(e_i^y\right) &= \left\|e_i^y\left(k\right)\right\|^2 - \gamma_2 h \sum_{j \in N_i} \left\|y_j\left(t_{k_j}^{j}\right) - y_i\left(t_{k_i}^{i}\right)\right\|^2, \end{aligned} \tag{12}$$

where $flag_1\left(e_i^x\right)$ denotes the error of two measurement errors, h represents a positive control coefficient, r_1 and r_2 are event-triggered weighting coefficients, which adjust the weighting ratio between local error and neighboring state discrepancies in the triggering conditions [5] . The first term representing the square of the measurement error of the agent's state estimate, and the second term indicating the sum of the squares of the differences between the state estimate at the agent's most recent triggered time and those of all neighboring agents. This term reflects the consensus level among agents. The difference between the two terms can determine whether the agent is experiencing the fastest state change. A similar logic applies to $flag_2\left(e_i^y\right)$. Utilizing $flag_1\left(e_i^x\right) + flag_2\left(e_i^y\right) \geq 0$ allows for determining whether the agent exhibits the fastest combined change in state information and average gradient information, while balancing the influences of state information and average momentum gradient information on event triggered. This condition raises the event-triggered threshold, further reducing communication frequency among agents. Notably, $t_{k_j}^{j}$ and $t_{k_i}^{i}$ represent the most recent triggered times of agents j and i, respectively, and these times are typically different. Therefore, when assessing the event-triggered conditions, only the state estimate and average momentum gradient estimate at the agent's most recent triggered time for all neighboring agents are required, along with the agent's current iteration state estimate and average momentum gradient estimate. This approach eliminates the need for real-time acquisition of neighboring agents' state information, significantly reducing communication costs and saving network bandwidth.

Remark 3.1. The event-triggered condition

$$t_{k+1}^{i} = \inf\{t | t > t_{k_i}^{i}, \max(flag_1(e_i^x), flag_2(e_i^y)) \geq 0\}$$

in reference [5] is determined by the maximum values of $flag_1\left(e_i^x\right)$ and $flag_2\left(e_i^y\right)$. Since $flag_1\left(e_i^x\right)$ and $flag_2\left(e_i^y\right)$ can be negative, this may lead to the following situation. During the kth iteration. $flag_1\left(e_i^x\right) \geq 0 > flag_2\left(e_i^y\right)$ occurs, and $\left|flag_2\left(e_i^y\right)\right| > \left|flag_1\left(e_i^x\right)\right|$ holds, which means that $flag_1\left(e_i^x\right) + flag_2\left(e_i^y\right) < 0$. This indicates that the advantage brought by the change in agent i's state estimate might not be sufficient to offset the negative impact caused by the change in agent i's average momentum gradient estimate. This design implies that even when the overall system is approaching steady

convergence, a temporary and minor fluctuation in just one direction may still trigger an event, thereby increasing the communication frequency. Since the triggering condition is evaluated independently on a single error metric, it becomes overly sensitive to local disturbances, resulting in frequent and unnecessary communication. In contrast, by jointly considering the variations in both the state estimate and the average momentum gradient estimate, such false triggers can be effectively avoided. The proposed two-dimensional event-triggered mechanism does not rely on the isolated behavior of a single error component; instead, it makes decisions based on the overall error dynamics. This integrated strategy increases the triggering threshold, enables more rational selection of triggering moments, and significantly reduces the number of communication events. Consequently, it enhances communication efficiency while maintaining convergence stability and algorithmic performance.

By substituting (6) into (8) and (7) into (10), a compact form is obtained:

$$x(k+1) = Wx(k) - hLe^{x}(k) - \alpha y(k), \tag{13}$$

$$g(k+1) = (1-\beta)\, g(k) + \beta \nabla f(x(k+1)), \tag{14}$$

$$y(k+1) = Wy(k) - hLe^{y}(k) + g(k+1) - g(k), \tag{15}$$

where $W = I - hL$, $\beta \in (0,1)$,

$$x(k) = \begin{bmatrix} x_1(k) \\ \cdots \\ x_n(k) \end{bmatrix},$$

$$y(k) = \begin{bmatrix} y_1(k) \\ \cdots \\ y_n(k) \end{bmatrix},$$

$$\nabla f(x(k)) = \begin{bmatrix} \nabla f_1(x_1(k)) \\ \cdots \\ \nabla f_n(x_n(k)) \end{bmatrix},$$

$$g(k) = \begin{bmatrix} g_1(k) \\ \cdots \\ g_n(k) \end{bmatrix},$$

$$e^{x}(k) = \begin{bmatrix} e_1^{x}(k) \\ \cdots \\ e_n^{x}(k) \end{bmatrix},$$

$$e^{y}(k) = \begin{bmatrix} e_1^{y}(k) \\ \cdots \\ e_n^{y}(k) \end{bmatrix}.$$

The specific algorithm process is outlined as follows Alogrithm 1:

Algorithm 1 ET-DMGing Algorithm

Initialize: x_i, $x_i(t_0^i)$, $y_i(t_0^i)$
Broadcast $x_i(t_0^i)$, $y_i(t_0^i)$ to all neighboring agents
while exit condition not met **do**
  for $j \in N_i$ **do** ▷ Detect if state estimates are received from neighboring agents
    if $x_i^j(t_{k_j}^j) \neq x_j(t_{k_j}^j)$ **then**
      Update state estimate and average momentum gradient:
      $x_i^j(t_{k_j}^j) \leftarrow x_j(t_{k_j}^j)$, $y_i^j(t_{k_j}^j) \leftarrow y_j(t_{k_j}^j)$
      Update auxiliary variables $u_i(k)$ and $v_i(k)$ according to (6), (7)
    end if
  end for
  Update local agent's state estimate $x_i(k+1)$ according to (8)
  Update local agent's auxiliary gradient $g_i(k+1)$ according to (9)
  Update local agent's average momentum gradient estimate $y_i(k+1)$ according to (10)
  if $flag_1(e_i^x) + flag_2(e_i^y) \geq 0$ **then** ▷ Check if an event is triggered
    Update event time: $t_{k_i}^i = k+1$
    $x_i(t_{k_i}^i) \leftarrow x_i(k+1)$, $y_i(t_{k_i}^i) \leftarrow y_i(k+1)$
    Update auxiliary variables $u_i(t)$ and $v_i(t)$
    Broadcast $x_i(t_{k_i}^i)$, $y_i(t_{k_i}^i)$ to neighboring agents
  end if
  $k \leftarrow k+1$
end while

Remark 3.2. The ET-DMGing algorithm does not exhibit Zeno behavior. In the context of event triggered, Zeno behavior refers to the occurrence of infinitely many triggers within a finite time frame. Since the ET-DMGing algorithm runs on discrete iterations, the triggered times correspond to specific iterations that meet the event-triggered conditions. As the algorithm progresses, the total number of iterations is finite, thus, infinite triggers cannot occur.

4. THEORETICAL ANALYSIS

In this section, we construction a framework for the algorithm's proof based on relevant lemmas and definition of related symbols. Subsequently, we populate this framework with key inferences, ultimately demonstrating that the algorithm converges to the optimal solution at a linear rate.

Lemma 4.1. (Small Gain Theorem) (Nedic et al. [22]) Assume $s_1, s_2, \ldots s_m$ is a sequence, for all positive integers K and any $i = 1, 2, \ldots m$, we have:

$$\left\| s_{(i \bmod m)+1} \right\|^{\lambda,K} \leq \varphi_i \|s_i\|^{\lambda,K} + \omega_i,$$

where $\varphi_1, \varphi_2, \ldots \varphi_m$ and $\omega_1, \omega_2, \ldots \omega_m$ are non-negative constants satisfying $0 \leq \prod_{i=1}^m \varphi_i < 1$, then $\|s_1\|^\lambda \leq \left(\frac{1}{1-\prod_{i=1}^m \varphi_i}\right) \sum_{i=1}^m w_i \prod_{j=i+1}^m \varphi_i$. Since the Small Gain Theorem

employs a cyclic structure for $s_i \to s_{(i \bmod m)+1}$, we can derive the bounds for each $\|s_i\|^{\lambda}$ as $i = 1, 2, \ldots m$ based on the aforementioned.

Based on Lemma 4.1 and (13), (14), (15), we define the sequence $s_1, s_2, \ldots s_5$, as follows $q, \hat{x}, \hat{y}, m, z$, with the cyclic structure composed of:

$$\begin{aligned}
\|z\|^{\lambda,K} &\leq \varphi_1 \|q\|^{\lambda,K} + w_1, \\
\|q\|^{\lambda,K} &\leq \varphi_2 \|\hat{x}\|^{\lambda,K} + w_2, \\
\|\hat{x}\|^{\lambda,K} &\leq \varphi_3 \|\hat{y}\|^{\lambda,K} + w_3, \\
\|\hat{y}\|^{\lambda,K} &\leq \varphi_4 \|m\|^{\lambda,K} + w_4, \\
\|m\|^{\lambda,K} &\leq \varphi_5 \|z\|^{\lambda,K} + w_5,
\end{aligned}$$

where $\bar{x}(k) = \frac{1}{n}\sum_{i=1}^{n} x_i(k)$, $\bar{y}(k) = \frac{1}{n}\sum_{i=1}^{n} y_i(k)$, $\bar{g}(k) = \frac{1}{n}\sum_{i=1}^{n} g_i(k)$, $q(k) = x(k) - \mathbf{1}_n x^*$, $\hat{x}(k) = x(k) - \mathbf{1}_n \bar{x}$, $\hat{y}(k) = y(k) - \mathbf{1}_n \bar{y}$, $z(\mathrm{k}) = \nabla f(x(k)) - \nabla f(x(k-1))$, $m(k) = g(k) - g(k-1)$. Let x^* denote the global optimal solution to problem (1). By proving $0 \leq \prod_{i=1}^{5} \varphi_i < 1$, we can establish the bounds for each $\|s_i\|^{\lambda}$, $i = 1, \ldots, 5$, which demonstrates that the algorithm can converge to the global optimal solution at a linear rate, $w_1, w_2, w_3, w_4, w_5 > 0$.

The following are the necessary inferences for the convergence of the algorithm.

Lemma 4.2. (Nedic et al. [22], Lemma 5) Suppose Assumption 2.2 hold, we have for any $K \geq 0$ and $\lambda \in (0, 1)$ such that:

$$\|z\|^{\lambda,K} \leq \hat{L}\left(1 + \frac{1}{\lambda}\right)\|q\|^{\lambda,K}. \tag{16}$$

Proof. The detailed proof can be found in in [22], so we omit its proof. □

Corollary 4.1. Suppose Assumption 2.4 hold, let W be a doubly stochastic matrix. Define $y(0) = g(0) = \beta \nabla f(x(0))$, we introduce a new auxiliary variable

$$\overline{d}(k) = \begin{cases} \overline{x}(k) & k = 0, \\ \frac{1}{\beta}\overline{x}(k) - \frac{1-\beta}{\beta}\overline{x}(k-1) & k \geq 1. \end{cases} \tag{17}$$

For any $k \geq 0$, we have:

$$\overline{d}(k+1) = \overline{d}(k) - \alpha \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_i(k)). \tag{18}$$

Similar approaches have also been explored in the literature [7].

Proof. Since Assumption 2.4, W is doubly stochastic. Therefore, multiplying both sides of (15) by $\frac{1}{n}\mathbf{1}_n^T$ yields. $\overline{y}(k+1) = \overline{y}(k) + \overline{g}(k+1) - \overline{g}(k)$, which implies $\overline{y}(k+1) -$

$\overline{g}(k+1) = \overline{y}(k) - \overline{g}(k)$. Since $y(0) = g(0) = \beta\nabla f(x(0))$, we have for any $k \geq 0$, such that $\overline{y}(k+1) = \overline{g}(k+1)$. Multiplying both sides of (13) and (14) by $\frac{1}{n}\mathbf{1}_n^T$ yields:

$$\begin{aligned}
\overline{x}(k+1) &= \overline{x}(k) - \alpha\overline{y}(k), \\
\overline{g}(k+1) &= (1-\beta)\overline{g}(k) + \beta\frac{1}{n}\mathbf{1}_n^T\nabla f(x(k+1)) \\
&= (1-\beta)\overline{g}(k) + \beta\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i(k+1)).
\end{aligned}$$

Let $\overline{v}(k) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i(k))$, thus proving $\overline{d}(k+1) = \overline{d}(k) - \alpha\overline{v}(k)$, and we have $\overline{g}(k+1) = (1-\beta)\overline{g}(k) + \beta\overline{v}(k)$. When $k = 0$, we have:

$$\overline{d}_1 - \overline{d}_0 = \frac{1}{\beta}\overline{x}_1 - \frac{1-\beta}{\beta}\overline{x}_0 - \overline{x}_0 = \frac{1}{\beta}(\overline{x}_0 - \alpha\overline{g}_0) - \frac{1}{\beta}\overline{x}_0 = -\alpha\overline{v}_0.$$

When $k \geq 1$, we have:

$$\begin{aligned}
\overline{d}(k+1) - \overline{d}(k) &= \frac{1}{\beta}(\overline{x}(k+1) - (1-\beta)\overline{x}(k) - \overline{x}(k) + (1-\beta)\overline{x}(k-1)) \\
&= \frac{1}{\beta}(-\alpha\overline{g}(k) + (1-\beta)\alpha\overline{g}(k-1)) \\
&= -\alpha\overline{v}(k).
\end{aligned}$$

In summary, for any $k \geq 0$, $\overline{d}(k+1) = \overline{d}(k) - \alpha\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_i(k))$ hold true. □

Corollary 4.2. Based on the event-triggered conditions (11) and (12), we have for any $\gamma_1, \gamma_2 \in \left(0, \frac{1}{4n^2h}\right)$ such that:

$$\|e^x(k)\| \leq b_1\|\hat{x}(k)\|, b_1 = 2n\sqrt{\frac{h\gamma_1}{1-4n^2h\gamma_1}}, \tag{19}$$

$$\|e^y(k)\| \leq b_2\|\hat{y}(k)\|, b_2 = 2n\sqrt{\frac{h\gamma_2}{1-4n^2h\gamma_2}}. \tag{20}$$

Proof. $flag_1(e_i^x)$ and $flag_2(e_i^y)$ are abbreviated as $flag_1$ and $flag_2$. From event-triggered conditions (11) and (12), we derive $flag = flag_1 + flag_2$ and $flag \geq 0$. Analysis shows that when the event is triggered, at least one of $flag_1$ or $flag_2$ exceeds zero. Furthermore, when the event occurs, $e_i^x(k) = x_i\left(t_{k_i}^i\right) - x_i(k) = 0$ and $e_i^y(k) = y_i\left(t_{k_i}^i\right) - y_i(k) = 0$ hold. Thus, the following inequalities are obtained:

$$\|e_i^x(k)\|^2 \leq \gamma_1 h\sum_{j\in N_i}\left\|x_j\left(t_{k_j}^j\right) - x_i\left(t_{k_i}^i\right)\right\|^2, \tag{21}$$

$$\|e_i^y(k)\|^2 \leq \gamma_2 h\sum_{j\in N_i}\left\|y_j\left(t_{k_j}^j\right) - y_i\left(t_{k_i}^i\right)\right\|^2. \tag{22}$$

Based on (21), it follows that:

$$\begin{aligned}\left\|e_i^x(k)\right\|^2 &\leq \gamma_1 h \sum_{j\in N_i} \left\|x_j(k)+e_j^x(k)-x_i(k)-e_i^x(k)\right\|^2 \\ &= \gamma_1 h \sum_{j\in N_i} \left\|\hat{x}_j(k)-\hat{x}_i(k)+e_j^x(k)-e_i^x(k)\right\|^2 \\ &\leq 2\gamma_1 h \sum_{j\in N_i} \left[2\left(\left\|\hat{x}_j(k)\right\|^2+\left\|\hat{x}_i(k)\right\|^2\right)+2\left(\left\|e_j^x(k)\right\|^2+\left\|e_i^x(k)\right\|^2\right)\right] \\ &\leq 4N\gamma_1 h\left(\left\|\hat{x}(k)\right\|^2+\left\|e^x(k)\right\|^2\right). \end{aligned} \tag{23}$$

Similarly, based on (22), we can conclude that:

$$\left\|e_i^y(k)\right\|^2 \leq 4N\gamma_2 h\left(\left\|\hat{y}(k)\right\|^2+\left\|e^y(k)\right\|^2\right). \tag{24}$$

By substituting $\left\|e^x(k)\right\|^2$and $\left\|e^y(k)\right\|^2$ for $\left\|e_j^x(k)\right\|^2$ and $\left\|e_i^y(k)\right\|^2$ in (23) and (24), we obtain:

$$\left\|e^x(k)\right\|^2 \leq 4N\gamma_1 h\left(\left\|\hat{x}(k)\right\|^2+\left\|e^x(k)\right\|^2\right), \tag{25}$$

$$\left\|e^y(k)\right\|^2 \leq 4N\gamma_2 h\left(\left\|\hat{y}(k)\right\|^2+\left\|e^y(k)\right\|^2\right). \tag{26}$$

This is identical to (19) and (20) . □

Corollary 4.3. When $\beta \in (0,1)$ and $\lambda \in (0,1)$hold true, and $\lambda - 1 + \beta > 0$ is satisfied. we have for any $K \geq 0$ such that:

$$\|m\|^{\lambda,K} \leq \frac{\lambda\beta}{\lambda-1+\beta}\|z\|^{\lambda,K} + \frac{\lambda}{\lambda-1+\beta}\left\|m(0)\right\|. \tag{27}$$

Proof. From (14), $g(k+1) = (1-\beta)g(k) + \beta\nabla f(x(k+1))$ follows, leading to $g(k) = (1-\beta)$
$g(k-1) + \beta\nabla f(x(k))$. Subtracting the two gives:

$$\left\|g(k+1)-g(k)\right\| = \left\|(1-\beta)(g(k)-g(k-1)) + \beta(\nabla f(x(k+1)) - \nabla f(x(k)))\right\|. \tag{28}$$

Since $m(k) = g(k) - g(k-1)$ and $z(\mathrm{k}) = \nabla f(x(k)) - \nabla f(x(k-1))$, (28) can be rewritten as:

$$\left\|m(k+1)\right\| \leq \left\|(1-\beta)m(k)\right\| + \left\|\beta z(k+1)\right\|. \tag{29}$$

Multiplying both sides of (29) by $\lambda^{-(k+1)}$ yields:

$$\lambda^{-(k+1)}\left\|m(k+1)\right\| \leq \frac{(1-\beta)}{\lambda}\lambda^{-k}\left\|m(k)\right\| + \beta\lambda^{-(k+1)}\left\|z(k+1)\right\|. \tag{30}$$

Let $\lambda - 1 + \beta > 0$, and taking $\max_{k=0,1,\ldots K-1}\{.\}$ on both sides of (30) yields:

$$\|m\|^{\lambda,K} \leq \frac{\lambda\beta}{\lambda - 1 + \beta}\|z\|^{\lambda,K} + \frac{\lambda}{\lambda - 1 + \beta}\|m(0)\|.$$

□

Corollary 4.4. Suppose Assumption 2.4 hold for $\gamma_2 \in \left(0, \frac{\vartheta}{4n^2h(1+\vartheta)}\right)$, where $\vartheta = \frac{(1-\delta)^2}{h^2\|L\|^2}$ is true. When$\delta + hb_2\|L\| < \lambda < 1$ is satisfied, any $K \geq 0$ such that:

$$\|\hat{y}\|^{\lambda,K} \leq \frac{\lambda}{\lambda - \delta - hb_2\|L\|}\|m\|^{\lambda,K} + \frac{\lambda}{\lambda - \delta - hb_2\|L\|}\|\hat{y}(0)\|. \tag{31}$$

P r o o f. Suppose Assumption 2.4 hold, let W be a doubly stochastic matrix, and $W = I - hL$. Given $\hat{J} = I - J$, $\hat{J}W = W\hat{J}$, $\hat{J}J = 0$, and $\left\|\hat{J}\right\| = 1$, according to (15) and Corollary 4.2, we have:

$$\begin{aligned}
\|\hat{y}(k+1)\| &= \|(I-J)\,y(k+1)\| \\
&= \left\|\hat{J}Wy(k) - hLe^y(k) - \hat{J}m(k+1)\right\| \\
&\leq \left\|W\hat{J}y(k)\right\| + h\|L\|.\|e^y(k)\| + \|m(k+1)\| \\
&= \|(W-J)\,\hat{y}(k)\| + h\|L\|.\|e^y(k)\| + \|m(k+1)\| \\
&\leq (\delta + hb_2\|L\|)\|\hat{y}(k)\| + \|m(k+1)\|.
\end{aligned} \tag{32}$$

Multiplying both sides of (32) by $\lambda^{-(k+1)}$ yields:

$$\lambda^{-(k+1)}\|\hat{y}(k+1)\| \leq \frac{\delta + hb_2\|L\|}{\lambda}\lambda^{-k}\|\hat{y}(k)\| + \lambda^{-(k+1)}\|m(k+1)\|. \tag{33}$$

Let $\lambda - 1 + \beta > 0$, and taking $\max_{k=0,1,\ldots K-1}\{.\}$ on both sides of (33) yields:

$$\|\hat{y}\|^{\lambda,K} \leq \frac{\lambda}{\lambda - \delta - hb_2\|L\|}\|m\|^{\lambda,K} + \frac{\lambda}{\lambda - \delta - hb_2\|L\|}\|\hat{y}(0)\|.$$

□

Corollary 4.5. Suppose Assumption 2.4 hold for $\gamma_1 \in \left(0, \frac{\vartheta}{4n^2h(1+\vartheta)}\right)$, where $\vartheta = \frac{(1-\delta)^2}{h^2\|L\|^2}$ is true. When $\delta + hb_1\|L\| < \lambda < 1$ is satisfied, for any $K \geq 0$, such that:

$$\|\hat{x}\|^{\lambda,K} \leq \frac{\alpha}{\lambda - \delta - hb_1\|L\|}\|\hat{y}\|^{\lambda,K} + \frac{\lambda}{\lambda - \delta - hb_1\|L\|}\|\hat{x}(0)\|. \tag{34}$$

P r o o f. The proof of Corollary 4.5 is omitted as it closely resembles that of Corollary 4.4. □

Corollary 4.6. Suppose Assumption 2.1, 2.2, 2.3 and 2.4 hold, and the step size α and λ satisfy conditions

$$\sqrt{1-\frac{\alpha\bar{\mu}\theta}{1+\theta}} \leq \lambda < 1, 0 < \alpha \leq \frac{1}{(1+\eta)\bar{L}}.$$

Then, for any $K \geq 0$ hold, that:

$$\|q\|^{\lambda,K} \leq \frac{\lambda^2 n\beta}{\phi - \lambda n\psi(1-\beta)} \|\bar{x}(0) - x^*\| + \frac{\phi + \lambda n\psi}{\phi - \lambda n\psi(1-\beta)} \|\bar{x} - x\|^{\lambda,K}, \tag{35}$$

where $\psi = \sqrt{\frac{\hat{L}(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\theta}$, $\phi = \lambda\sqrt{n}(\lambda - 1 + \beta) - (1-\beta)\psi > 0$, $\phi - \lambda n\psi(1-\beta) > 0$.

P r o o f. From Corollary 4.1 and (32) of Lemma 8 in [22], it follows that:

$$\begin{aligned}
&\lambda^{-(k+1)} \left\|\bar{d}(k+1) - d^*\right\| \\
&\leq \left\|\bar{d}(0) - d^*\right\| + \left(\lambda\sqrt{n}\right)^{-1} \left(\sqrt{\frac{\hat{L}(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\theta}\right) \sum_{i=1}^{n} \max_{t=0,..k} \left\{\lambda^{-t} \left\|\bar{d}(k) - x_i^t\right\|\right\},
\end{aligned} \tag{36}$$

where θ and η are positive free variables. Let $\psi = \sqrt{\frac{\hat{L}(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\theta}$, substituting (17) into (36) yields:

$$\begin{aligned}
&\lambda^{-(k+1)} \left\|\frac{1}{\beta}\bar{x}(k+1) - \frac{1-\beta}{\beta}\bar{x}(k) - \left(\frac{1}{\beta}x^* - \frac{1-\beta}{\beta}x^*\right)\right\| \\
&\leq \|\bar{x}(0) - x^*\| + \left(\lambda\sqrt{n}\right)^{-1}\psi \\
&\quad \sum_{i=1}^{n} \max_{t=0,..k} \left\{\lambda^{-t} \left\|\frac{1}{\beta}\bar{x}(t) - \frac{1-\beta}{\beta}\bar{x}(t-1) - \left(\frac{1}{\beta} - \frac{1-\beta}{\beta}\right)x_i^t\right\|\right\},
\end{aligned}$$

$$\begin{aligned}
&\lambda^{-(k+1)}\frac{1}{\beta}\|\bar{x}(k+1) - x^*\| - \lambda^{-(k+1)}\frac{1-\beta}{\beta}\|\bar{x}(k) - x^*\| \\
&\leq \|\bar{x}(0) - x^*\| + \left(\lambda\sqrt{n}\right)^{-1}\psi \\
&\quad \sum_{i=1}^{n} \max_{t=0,...k} \left\{\lambda^{-t}\left(\frac{1}{\beta}\left\|\bar{x}(t) - x_i^t\right\| + \frac{1-\beta}{\beta}\left\|\bar{x}(t-1) - x_i^t\right\|\right)\right\} \\
&\leq \|\bar{x}(0) - x^*\| \\
&\quad + \left(\lambda\sqrt{n}\right)^{-1}\psi \\
&\quad \sum_{i=1}^{n} \max_{t=0,..k} \left\{\lambda^{-t}\left(\frac{1}{\beta}\left\|\bar{x}(t) - x_i^t\right\| + \frac{1-\beta}{\beta}\|\bar{x}(t-1) - x^*\| + \frac{1-\beta}{\beta}\left\|x^* - x_i^t\right\|\right)\right\}.
\end{aligned}$$

Taking $\max_{k=0,1,...K-1}\{.\}$ on both sides of the above expression yields:

$$\frac{1}{\beta}\|\bar{x} - x^*\|^{\lambda,K} - \frac{1-\beta}{\lambda\beta}\|\bar{x} - x^*\|^{\lambda,K-1}$$

$$
\begin{aligned}
&\leq \|\bar{x}(0)-x^*\|+\left(\lambda\sqrt{n}\right)^{-1}\psi \\
&\quad \sum_{i=1}^{n}\left(\frac{1}{\beta}\|\bar{x}-x_i\|^{\lambda,K-1}+\frac{1-\beta}{\lambda\beta}\|\bar{x}-x^*\|^{\lambda,K-2}+\frac{1-\beta}{\beta}\|x_i-x^*\|^{\lambda,K-1}\right) \\
&\leq \|\bar{x}(0)-x^*\|+\left(\lambda\sqrt{n}\right)^{-1}\psi\sqrt{n}\frac{1}{\beta}\sqrt{\sum_{i=1}^{n}\left(\|\bar{x}-x_i\|^{\lambda,K}\right)^2}+\left(\lambda\sqrt{n}\right)^{-1}\psi\frac{1-\beta}{\lambda\beta}\|\bar{x}-x^*\|^{\lambda,K} \\
&\quad +\left(\lambda\sqrt{n}\right)^{-1}\psi\sqrt{n}\frac{1-\beta}{\beta}\sqrt{\sum_{i=1}^{n}\left(\|x^*-x_i\|^{\lambda,K}\right)^2} \\
&\leq \|\bar{x}(0)-x^*\|+(\lambda)^{-1}\psi\frac{1}{\beta}\|\bar{x}-x\|^{\lambda,K}+\left(\lambda\sqrt{n}\right)^{-1}\psi\frac{1-\beta}{\lambda\beta}\|\bar{x}-x^*\|^{\lambda,K} \\
&\quad +(\lambda)^{-1}\psi\frac{1-\beta}{\beta}\|x^*-x\|^{\lambda,K}.
\end{aligned}
$$

Rearranging and combining terms yields:

$$
\begin{aligned}
&\frac{\lambda\sqrt{n}(\lambda-1+\beta)-(1-\beta)\psi}{\lambda^2\sqrt{n}\beta}\|\bar{x}-x^*\|^{\lambda,K} \\
&\leq \|\bar{x}(0)-x^*\|+(\lambda)^{-1}\psi\frac{1}{\beta}\|\bar{x}-x\|^{\lambda,K}+(\lambda)^{-1}\psi\frac{1-\beta}{\beta}\|x^*-x\|^{\lambda,K}.
\end{aligned}
$$

Let $\frac{\lambda\sqrt{n}(\lambda-1+\beta)-(1-\beta)\psi}{\lambda^2\sqrt{n}\beta}>0$, then $\lambda\sqrt{n}(\lambda-1+\beta)$ $-(1-\beta)\psi>0$. Let $\phi=\lambda\sqrt{n}(\lambda-1+\beta)-(1-\beta)\psi>0$, thus:

$$
\begin{aligned}
&\|\bar{x}-x^*\|_F^{\lambda,K} \\
&\leq \frac{\lambda^2\sqrt{n}\beta}{\phi}\|\bar{x}(0)-x^*\|+\frac{\lambda^2\sqrt{n}\beta}{\phi}(\lambda)^{-1}\psi\frac{1}{\beta}\|\bar{x}-x\|^{\lambda,K}+\frac{\lambda^2\sqrt{n}\beta}{\phi}(\lambda)^{-1}\psi\frac{1-\beta}{\beta}\|x^*-x\|^{\lambda,K}.
\end{aligned}
\tag{37}
$$

Note that the following equation holds.

$$
\begin{aligned}
q(k)&=x(k)-\mathbf{1}_n x^* \\
&=x(k)-\mathbf{1}_n\bar{x}(k)+\mathbf{1}_n\bar{x}(k)-\mathbf{1}_n x^* \\
&=\hat{x}(k)+\mathbf{1}_n(\bar{x}(k)-x^*).
\end{aligned}
$$

It can be concluded that:

$$
\|q\|^{\lambda,K}\leq\|\hat{x}\|^{\lambda,K}+\sqrt{n}\|\bar{x}-x^*\|^{\lambda,K}. \tag{38}
$$

Combining (37) and (38), and letting $\frac{\phi-\lambda n\psi(1-\beta)}{\phi}>0$, since$\phi>0$ hold, $\phi-\lambda n\psi(1-\beta)$ >0 follows, we have:

$$
\|q\|^{\lambda,K}\leq\frac{\lambda^2 n\beta}{\phi-\lambda n\psi(1-\beta)}\|\bar{x}(0)-x^*\|+\frac{\phi+\lambda n\psi}{\phi-\lambda n\psi(1-\beta)}\|\bar{x}-x\|^{\lambda,K}.
$$

□

Theorem 4.3. Based on Lemma 4.1 (The Small Gain Theorem), Lemma 4.2 and Corollaries 4.2–4.6, it can be concluded that when the parameters α, γ_1, γ_2 satisfy the following constraints, the sequence generated $\{x(k)\}$ by (8), (9) and (10) under the event-triggered condition (11) will converge to the globally consistent optimal solution $1_n x^*$ at a linear rate of $O\left(\lambda^k\right)$.

$$\gamma_1, \gamma_2 \in \left(0, \frac{\vartheta}{4n^2 h\left(1+\vartheta\right)}\right), \alpha \in \left(0, \frac{C_1(1-\delta)^2}{\bar{\mu}\kappa}\right), \tag{39}$$

where $0 < C_1 < 1$, $\kappa = \frac{2\hat{L}\beta}{\bar{\mu}}$, $\vartheta = \frac{(1-\delta)^2}{h^2\|L\|^2}$, $\hat{L} = \max_{i=1}^n \{l_i\}$, $\bar{\mu} = \frac{1}{n}\sum_{i=1}^n \mu_i$, $\delta = \sigma_{\max}(W-J) < 1$,$0 < h < \frac{1}{\hat{d}}$, $\hat{d} = \max_{i=1}^n \{d_i\}$, $d_i = \sum_{j=1}^n a_{ij}$. Besides, the rate λ is given by

$$\lambda = \max\left\{\sqrt{1-\frac{\alpha\bar{\mu}}{1.5}}, \sqrt{\frac{\kappa\alpha\bar{\mu}}{C_1}} + \delta\right\}. \tag{40}$$

Proof. According to Lemma 4.2 and Corollaries 4.2–4.6, we have:

$$\begin{aligned}
&\|z\|^{\lambda,K} \le \varphi_1\|q\|^{\lambda,K} + w_1, \varphi_1 = \hat{L}\left(1+\frac{1}{\lambda}\right), w_1 = 0,\\
&\|q\|^{\lambda,K} \le \varphi_2\|\hat{x}\|^{\lambda,K} + w_2, \varphi_2 = \frac{\phi+\lambda n\psi}{\phi-\lambda n\psi\left(1-\beta\right)}, w_2 = \frac{\lambda^2 n\beta}{\phi-\lambda n\psi\left(1-\beta\right)},\\
&\|\hat{x}\|^{\lambda,K} \le \varphi_3\|\hat{y}\|^{\lambda,K} + w_3, \varphi_3 = \frac{\alpha}{\lambda-\delta-hb_1\|L\|}, w_3 = \frac{\lambda}{\lambda-\delta-hb_1\|L\|},\\
&\|\hat{y}\|^{\lambda,K} \le \varphi_4\|m\|^{\lambda,K} + w_4, \varphi_4 = \frac{\lambda}{\lambda-\delta-hb_2\|L\|}, w_4 = \frac{\lambda}{\lambda-\delta-hb_2\|L\|},\\
&\|m\|^{\lambda,K} \le \varphi_5\|z\|^{\lambda,K} + w_5, \varphi_5 = \frac{\lambda\beta}{\lambda-1+\beta}, w_5 = \frac{\lambda}{\lambda-1+\beta}.
\end{aligned}$$

According to Lemma 4.1, when $0 < \varphi_1\varphi_2\varphi_3\varphi_4\varphi_5 < 1$ hold, $\|q\|^{\lambda}$ is bounded. Thus:

$$\hat{L}\left(1+\frac{1}{\lambda}\right)\frac{\alpha}{\lambda-\delta-hb_1\|L\|}\frac{\lambda}{\lambda-\delta-hb_2\|L\|}\frac{\lambda\beta}{\lambda-1+\beta}\frac{\phi+\lambda n\psi}{\phi-\lambda n\psi\left(1-\beta\right)} < 1, \tag{41}$$

where the parameters must satisfy the following constraints:

$$\begin{aligned}
&\delta + hb_1\|L\| < \lambda < 1, \delta + hb_2\|L\| < \lambda < 1, \sqrt{1-\frac{\alpha\bar{\mu}\theta}{1+\theta}} \le \lambda < 1,\\
&0 < \alpha \le \frac{1}{\left(1+\eta\right)\bar{L}}, (\lambda-1+\beta) > 0, \phi > 0, \phi - \lambda n\psi\left(1-\beta\right) > 0.
\end{aligned} \tag{42}$$

Considering the constraint $\sqrt{1-\frac{\alpha\bar{\mu}\theta}{1+\theta}} \le \lambda < 1$ in (42), let $1+\frac{1}{\theta} \le 1.5$, then

$$\alpha \ge \frac{1.5\left(1-\lambda^2\right)}{\bar{\mu}}.$$

Additionally, from (41), we have:

$$\alpha \leq \frac{(\lambda-\delta)^2}{\hat{L}(1+\lambda)} \frac{\lambda-1+\beta}{\lambda\beta} \frac{\phi-\lambda n\psi(1-\beta)}{\phi+\lambda n\psi}.$$

It is clear that $\frac{\phi-\lambda n\psi(1-\beta)}{\phi+\lambda n\psi} < 1$. Let $C = \frac{\phi-\lambda n\psi(1-\beta)}{\phi+\lambda n\psi} < 1$, then from (42), $C > 0$ follows. Thus, $0 < C < 1$ leads to

$$\alpha \leq C \frac{(\lambda-\delta)^2}{\hat{L}(1+\lambda)} \frac{\lambda-1+\beta}{\lambda\beta}.$$

In summary, it follows that:

$$\alpha \in \left[\frac{1.5\left(1-\lambda^2\right)}{\bar{\mu}}, C \frac{(\lambda-\delta)^2}{\hat{L}(1+\lambda)} \frac{\lambda-1+\beta}{\lambda\beta} \right].$$

To conveniently obtain the range of λ through α, the range of α is rewritten. Let $\kappa = \frac{2\hat{L}\beta}{\bar{\mu}}$, then

$$\alpha \in \left[\frac{1.5\left(1-\lambda^2\right)}{\bar{\mu}}, C \frac{(\lambda-\delta)^2(\lambda-1+\beta)}{\bar{\mu}\kappa} \right].$$

Since $0 < C < 1$ and $0 < (\lambda - 1 + \beta) < 1$, it follows that $0 < C(\lambda - 1 + \beta) < 1$. Let the constant $C_1 = C(\lambda - 1 + \beta)$, then $0 < C_1 < 1$ is obtained. Therefore

$$\alpha \in \left[\frac{1.5\left(1-\lambda^2\right)}{\bar{\mu}}, \frac{(\lambda-\delta)^2 C_1}{\bar{\mu}\kappa} \right]. \tag{43}$$

Let $\hat{b} = \max\{b_1, b_2\}$. As λ increases from $\delta + h\widehat{b}\,\|L\|$ to 1, the left bound of α decreases monotonically, while the right bound increases monotonically. Let λ^* ensure that the left and right bounds of α are equal to Δ.

$$\lambda^* = \frac{\delta + \sqrt{\left(\frac{1.5\kappa}{C_1}\right)^2 + (1-\delta)^2 \frac{1.5\kappa}{C_1}}}{1+\frac{1.5\kappa}{C_1}}, \Delta = \frac{1.5\left(\sqrt{\left(\frac{1.5\kappa}{C_1}\right)^2 + (1-\delta^2)\frac{1.5\kappa}{C_1}} - \delta\frac{1.5\kappa}{C_1}\right)^2}{\overline{\mu}\frac{1.5\kappa}{C_1}\left(1+\frac{1.5\kappa}{C_1}\right)^2}.$$

When λ increases from$\delta + h\widehat{b}\,\|L\|$ to λ^*, the bounds of α are as follows:

$$\frac{1.5\left(1-\lambda^2\right)}{\bar{\mu}} \in \left[\triangle, \frac{1.5\left(1-\left(\delta+h\hat{b}\,\|L\|\right)^2\right)}{\bar{\mu}} \right], \frac{(\lambda-\delta)^2 C_1}{\bar{\mu}\kappa} \in \left[\frac{C_1\left(h\hat{b}\,\|L\|\right)^2}{\bar{\mu}\kappa}, \triangle \right].$$

It is clear that the minimum value of the left bound of α equals the maximum value of its right bound, thus α is an empty set. As λ increases from λ^* to 1, the bounds of α are as follows:

$$\frac{1.5\,(1-\lambda)}{\bar{\mu}} \in (0, \triangle]\,, \frac{(\lambda-\delta)^2 C_1}{\bar{\mu}\kappa} \in \left[\triangle, \frac{C_1(1-\delta)^2}{\bar{\mu}\kappa}\right).$$

It is evident that α is not an empty set.

This implies that when $\lambda \in [\lambda^*, 1)$ hold, the interval of α is valid, corresponding to the interval $\alpha \in \left(0, \frac{C_1(1-\delta)^2}{\bar{\mu}\kappa}\right)$. Considering the relationship between α and λ in (43), once the step size α is given, the λ can be determined as follows:

$$\lambda = \max\left\{\sqrt{1-\frac{\alpha\bar{\mu}}{1.5}}, \sqrt{\frac{\kappa\alpha\bar{\mu}}{C_1}} + \delta\right\}.$$

□

5. SIMULATION EXPERIMENTS

To validate the convergence and effectiveness of the proposed algorithm, a typical decentralized parameter estimation problem based on an undirected graph was employed. We considered an undirected network consisting of n=20 agents. The goal of all agents is to find an optimal value x^* to achieve the objective $\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, with $f_i(x) = a_i + b_i(x-c_i)^2 + \ln(1+\exp(-d_i x))$.

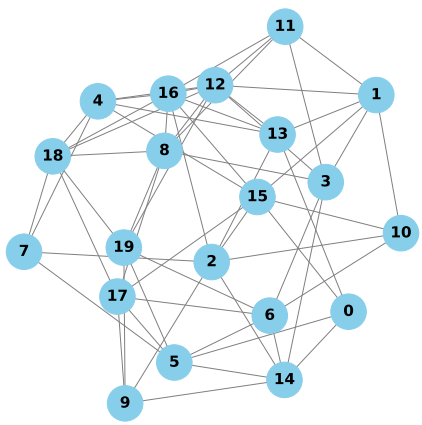


Fig. 1. The network topology consists of 20 agents.

As shown in Figure 1, the network topology consists of 20 agents, where $i = 1, \ldots 20$ represents the 20 agents. The parameters in each objective function $a_i \in (0.5, 2)$, $b_i = 1$, $c_i \in (0, 1)$, and $d_i \in (-1, 1)$ were randomly generated within specified ranges using

random functions, while the initial values of each agent were randomly assigned within the range of $(-1, 1)$. We set $h = 0.05$, step size $\alpha = 0.35$, $\gamma_1 = \gamma_2 = 1$, and momentum coefficient $\beta = 0.9$.

Based on the above parameters and initial settings, the following results were obtained. As shown in Figure 2, the state estimates of all agents gradually converge to the global optimal solution of 0.498, denoted as $x^* = 0.498$. Figure 3 (a) illustrates the changes in the objective function, revealing that the value of f stabilizes at 36.258 during iterations. Figure 3 (b) shows the changes in the squared average gradient $\left(\nabla \bar{f}\left(x\left(k\right)\right)\right)^2$, indicating that it gradually approaches 0. This indicates that our algorithm can achieve precise convergence. Figure 4 shows the event-triggered times for all agents throughout the iteration process, with trigger counts of [30, 33, 40, 35, 59, 38, 44, 23, 46, 33, 34, 34, 103, 46, 40, 56, 95, 33, 54, 38] for each agent. The average trigger count is 45.7, indicating an average communication frequency of 45.7 times among agents, corresponding to an average communication rate of 9.14%, significantly reducing communication costs compared to real-time communication. Figure 5 shows that the event triggered is discrete rather than continuous, with multiple sampling moments included within each event trigger interval, representing multiple iterations. Figure 6 indicates that the auxiliary variables u and v remain unchanged during the non-trigger sampling moments, only changing at the event-triggered sampling times, further confirming the discreteness of the event-triggered. Additionally, the impact of different momentum coefficients β on algorithm performance was considered, with the same stopping condition $\|\epsilon\left(k\right)\| = \|x\left(k\right) - 1_n x^*\| \leq 0.001$ set, where $h = 0.05$, step size $\alpha = 0.35$, and $\gamma_1 = \gamma_2 = 1$ are fixed. The effects of varying on performance are shown in Table 1, which shows that as the momentum coefficient increases, the average communication count reaches its minimum at $\beta = 0.9$.

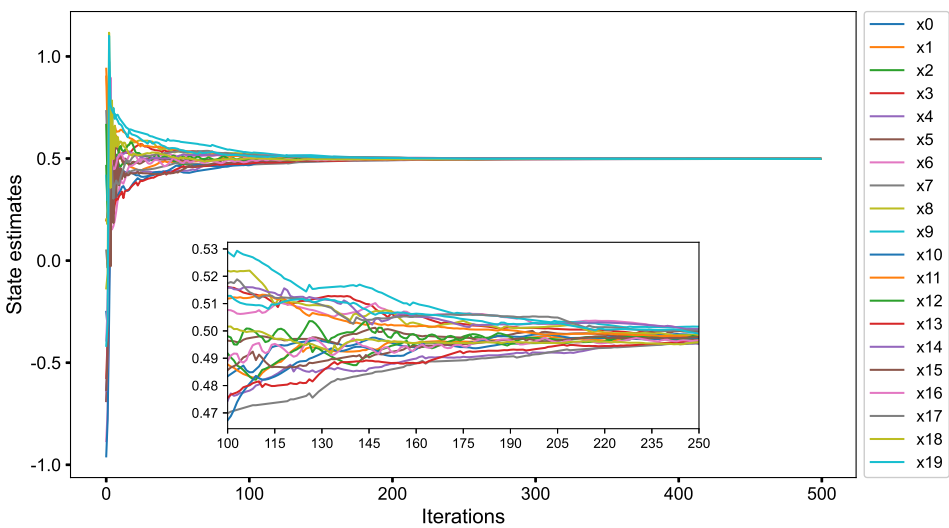


Fig. 2. The state estimates of all agents vary with iterations.

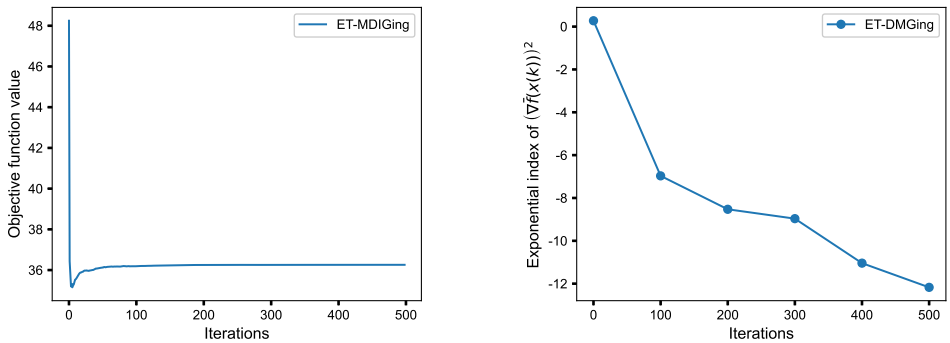


Fig. 3. Iterative optimization characteristics: (a) Trajectory of objective function values, (b) Evolution of average gradient magnitudes.

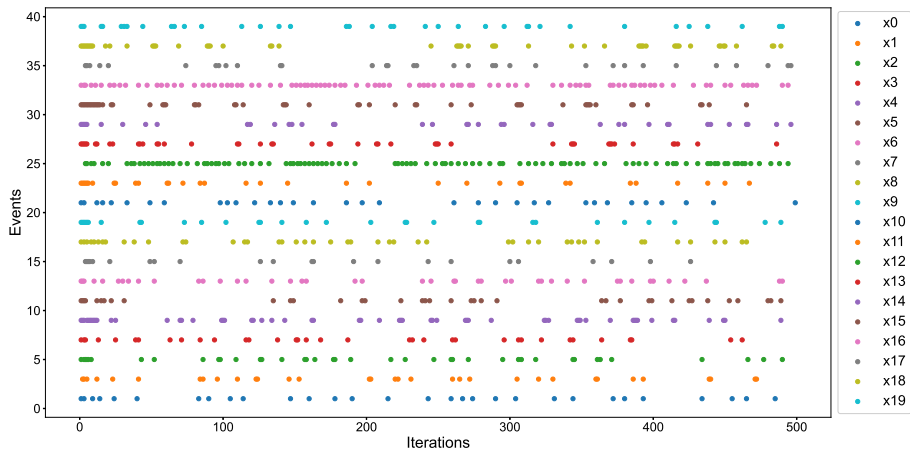


Fig. 4. Event-triggered times of each agent.

β	0.01	0.1	0.3	0.5	0.7	0.9	0.9999
Iterations	1088	429	436	407	424	416	472
Avg. Communication	977.65	92	69.75	49.8	42.7	**39.95**	181.45

Tab. 1. Impact of momentum coefficients β on performance.

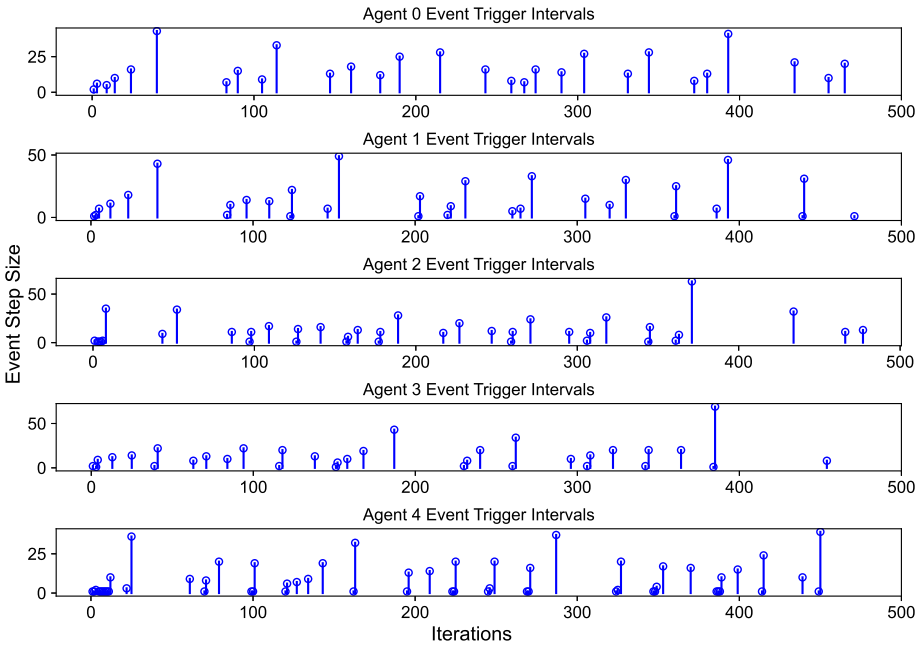
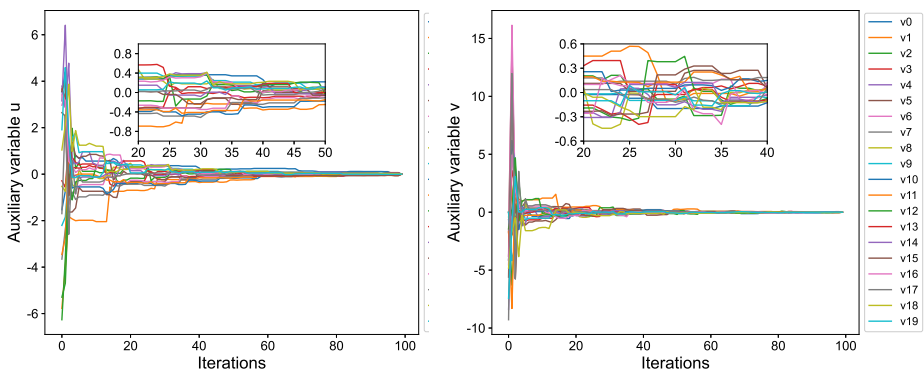


Fig. 5. Event trigger intervals for agent 0-4 with line height representing sampling time intervals per event step.



(a) (b)

Fig. 6. The trajectories of auxiliary variables u_i in figure (a) and v_i in figure (b) for $i = 0, \ldots, 19$.

Algorithm	Iterations	Total Iteration Time	Average Time per Iteration
ET-DMGing (Ours)	**415**	**2.71**	**0.006541**
Extended DIGing [5]	550	4.23	0.007697
ET-DMGing without Momentum	430	3.21	0.007463

Tab. 2. Performance comparison of different algorithms.

Algorithm	Avg. Communication
ET-DMGing-2D (Ours)	**39.95**
ET-DMGing-1D	52.40
Extended DIGing-2D	160.80
Extended DIGing-1D [5]	226.05

Tab. 3. Average number of communications under different algorithms.

To demonstrate the performance differences between our algorithm and the Extended DIGing algorithm [5], we set identical stopping conditions$\|\epsilon(k)\| = \|x(k) - 1_n x^*\| \leq 0.001$, momentum coefficient $\beta = 0.9$, and equal values for the following parameters: step size $\alpha = 0.35$, $h = 0.05$, and $\gamma_1 = \gamma_2 = 1$. As shown in Table 2, the ET-DMGing algorithm reduces the number of iterations by 24.55% and the total iteration time is reduced by 35.93%. compared to the Extended DIGing algorithm [5]. When compared to ET-DMGing without momentum, the iteration count is reduced by 3.49% and the total iteration time is reduced by 15.58%. These results indicate that the momentum mechanism can effectively mitigate oscillations caused by large step sizes and accelerate the convergence speed.

To validate the effectiveness of the proposed two-dimensional event-triggered mechanism, we conduct a comparative experiment under the same parameter settings and termination conditions as described above. Specifically, we adopt the same update rule of the ET-DMGing algorithm and compare the communication frequency under the two mechanisms: the two-dimensional event-triggered mechanism (ET-DMGing-2D) and the single-dimensional event-triggered mechanism (ET-DMGing-1D). Similarly, we employ the update rule of the Extended DIGing algorithm [5] to compare the communication frequency under the two-dimensional (Extended DIGing-2D) and single-dimensional (Extended DIGing-1D) mechanisms. The average number of communications under the four settings is summarized in Table 3. Compared to ET-DMGing-1D, ET-DMGing (our method) reduces the average number of communications from 52.40 to 39.95, achieving a reduction of 23.76%. Likewise, Extended DIGing-2D reduces the communication count from 226.05 to 160.80 compared to Extended DIGing-1D [5], representing a 28.88% reduction. These results demonstrate that the proposed two-dimensional event-triggered mechanism can effectively reduce communication frequency and improve communication

efficiency across different algorithmic frameworks, showing good generality and adaptability.

Figure 7 (a) provides a more intuitive comparison of the average communication frequency between the ET-MDIGing algorithm and the Extended DIGing algorithm [5]. Figure 7 (b) shows the changes in $\frac{\|x(k)-x^*\|}{\|x(0)-x^*\|}$, illustrating that the ET-DMGing algorithm effectively reduces oscillations during the convergence process compared to the Extended DIGing algorithm [5], while also achieving faster convergence.

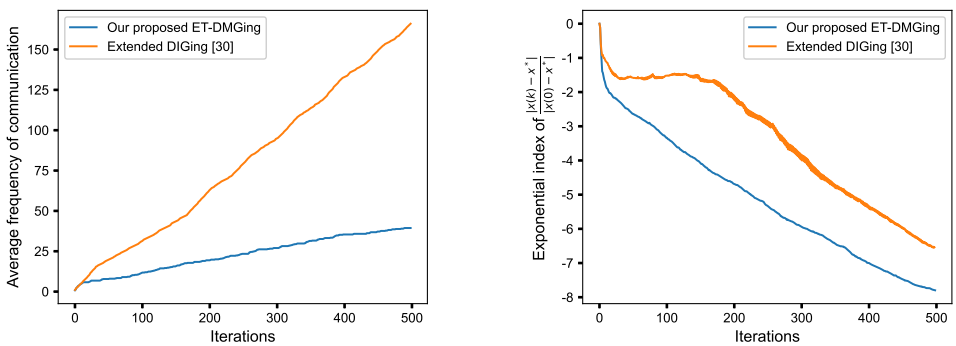


Fig. 7. ET-DMGing vs. Extended DIGing [5]: (a) Average communication count. (b) Exponential evolution curve of normalized solution error $\frac{\|x(k)-x^*\|}{\|x(0)-x^*\|}$.

6. CONCLUSION

This paper proposes an event-triggered distributed momentum-gradient tracking optimization algorithm (ET-DMGing) for multi-agent systems.The algorithm simplifies its structure through the use of a fixed step size and employs gradient tracking techniques to accurately track the average momentum gradient, effectively reducing oscillations during the optimization process and enhancing convergence speed. The design of the event-triggered conditions adequately considers both agent state information and average momentum gradient information, balancing the impact of both on event triggered while ensuring agent consistency and convergence accuracy. This approach significantly reduces the frequency of real-time communications between agents. Numerical simulation results demonstrate that, compared to the Extended DIGing algorithm [5], the ET-DMGing algorithm exhibits lower communication frequency and faster convergence speed. Moreover, a rigorous convergence analysis proves that the ET-DMGing algorithm can converge to the optimal solution with linear precision. Future research will focus on extending event-triggered distributed optimization algorithms to address optimization problems in time-varying multi-agent systems within dynamic and complex environments.

ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (Grant no.62103070), in part by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant no.KJZD-K202301103, Grant no.KJQN202501111), in part by the New Chongqing Youth Innovation Talent Project (Grant no.CSTB2025YITP-QCRCX0015), and in part by Chongqing Natural Science Foundation (Grant no.CSTB2023NSCQ-MSX0539).

(Received April 11, 2025)

REFERENCES

[1] G. Carnevale, F. Farina, I. Notarnicolam and G. Notarstefano: GTAdam: Gradient tracking with adaptive momentum for distributed online optimization. IEEE Trans. Control Network Systems *10* (2022), 3, 1436–1448. DOI:10.1109/TCNS.2022.3232519

[2] W. Chen and W. Ren: Event-triggered zero-gradient-sum distributed consensus optimization over directed networks. Automatica *65* (2016), 90–97. DOI:10.1016/j.automatica.2015.11.015

[3] C. Chen, L. Shen, W. Liu and Z.-Q. Luo: Efficient-Adam: Communication-Efficient Distributed Adam. IEEE Trans. Signal Process. (2023). DOI:10.1109/tsp.2023.3309461

[4] A. Defazio, F. Bach, and S. Lacoste-Julien: SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. Adv. Neural Inform. Process. Systems *27* (2014).

[5] L. Gao, S. Deng, H. Li, and Ch. Li: An event-triggered approach for gradient tracking in consensus-based distributed optimization. IEEE Trans. Network Sci. Engrg. *9* (2021), 2, 510–523. DOI:10.1109/TNSE.2021.3122927

[6] H.-Ch. Huang and J. Lee: A new variable step-size NLMS algorithm and its performance analysis. IEEE Trans. Signal Process. *60* (2011), 4, 2055–2060. DOI:10.1109/TSP.2011.2181505

[7] K. Huang, S. Pu, and A. Nedić: An accelerated distributed stochastic gradient method with momentum. arXiv preprint arXiv:2402.09714 (2024).

[8] X. Jiang, X. Zeng, J. Sun, and Jie Chen: Distributed stochastic gradient tracking algorithm with variance reduction for non-convex optimization. IEEE Trans. Neural Networks Learning Systems *34* (2022), 9, 5310–5321. DOI:10.1109/TNNLS.2022.3170944

[9] H. Lee, S. H. Lee, and T. Q. S. Quek: Deep learning for distributed optimization: Applications to wireless resource management. IEEE J. Select. Areas Commun. *37* (2019), 10, 2251–2266. DOI:10.1109/JSAC.2019.2933890

[10] H.-S. Lee, S.-E. Kim, J.-W. Lee, and W.-J. Song: A variable step-size diffusion LMS algorithm for distributed estimation. IEEE Trans. Signal Process. *63* (2015), 7, 1808–1820. DOI:10.1109/TSP.2015.2401533

[11] A. Lederer, Z. Yang, J. Jiao, and S. Hirche: Cooperative control of uncertain multiagent systems via distributed Gaussian processes. IEEE Trans. Automat. Control *68* (2022), 5, 3091–3098. DOI:10.1109/TAC.2022.3205424

[12] Q. Li, Y. Liao, K. Wu, L. Zhang, J. Lin, M. Chen, J. M. Guerrero, and D. Abbott: Parallel and distributed optimization method with constraint decomposition for energy management of microgrids. IEEE Trans. Smart Grid *12* (2021), 6, 4627–4640. DOI:10.1109/TSG.2021.3097047

[13] H. Li, X. Liao, G. Chen, D. J. Hill, Z. Dong, and T. Huang: Event-triggered asynchronous intermittent communication strategy for synchronization in complex dynamical networks. Neural Networks *66* (2015), 1–10. DOI:10.1016/j.neunet.2015.01.006

[14] H. Li, S. Liu, Y. Ch. Soh, L. Xie, and D. Xia: Achieving linear convergence for distributed optimization with zeno-like-free event-triggered communication scheme. In: Proc. 29th Chinese Control And Decision Conference 2017, pp. 6224–6229.

[15] H. Li, L. Zheng, Z. Wang, Y. Yan, L. Feng, and J. Guo: S-DIGing: A stochastic gradient tracking algorithm for distributed optimization. IEEE Trans. Emerging Topics Comput. Intell. *6* (2020), no. 1, 53–65. DOI:10.1109/tetci.2020.3017242

[16] J. Li and H. Su: Gradient tracking: A unified approach to smooth distributed optimization. arXiv preprint arXiv:2202.09804 (2022).

[17] X. Liu, Ch. Miao, G. Fiumara, and P. De Meo: Information propagation prediction based on spatial-temporal attention and heterogeneous graph convolutional networks. IEEE Trans. Comput. Social Systems *11* (2024), 1, 945–958. DOI:10.1109/TCSS.2023.3244573

[18] Ch. Liu, X. Dou, Y. Fan, and S. Cheng: A penalty ADMM with quantized communication for distributed optimization over multi-agent systems. Kybernetika *59* (2023), 3, 392–417. DOI:10.14736/kyb-2023-3-0392

[19] S. Liu, L. Xie, and D. E. Quevedo: Event-triggered quantized communication-based distributed convex optimization. IEEE Trans. Control Network Systems *5* (2016), 1, 167–178. DOI:10.1109/tcns.2016.2585305

[20] K. Lu and Q. Zhu: Distributed algorithms involving fixed step size for mixed equilibrium problems with multiple set constraints. IEEE Trans. Neural Networks Learn. Systems *32* (2020), 11, 5254–5260. DOI:10.1109/TNNLS.2020.3027288

[21] G. Morral, P. Bianchi, and G. Fort: Success and failure of adaptation-diffusion algorithms with decaying step size in multiagent networks. IEEE Trans. Signal Process. *65* (2017), 11, 2798–2813. DOI:10.1109/TSP.2017.2666771

[22] A. Nedic, A. Olshevsky, and W. Shi: Achieving geometric convergence for distributed optimization over time-varying graphs. SIAM J. Optim. *27* (2017), 4, 2597–2633. DOI:10.1137/16M1084316

[23] N. Qian: On the momentum term in gradient descent learning algorithms. Neural Networks *12* (1999), 1, 145–151. DOI:10.1016/S0893-6080(98)00116-6

[24] G. Qu and N. Li: Harnessing smoothness to accelerate distributed optimization. IEEE Trans. Control Network Systems *5* (2017), 3, 1245–1260. DOI:10.1109/TCNS.2017.2698261

[25] M. Rabbat and R. Nowak: Distributed optimization in sensor networks. In: Proc. 3rd International Symposium on Information Processing in Sensor Networks 2004, pp. 20–27.

[26] Z. Shen and H. Yin: A distributed routing-aware deployment algorithm for underwater sensor networks. IEEE Sensors J. (2024). DOI:10.1109/jsen.2024.3396145

[27] W. Shi, Q. Ling, G. Wu, and W. Yin: Extra: An exact first-order algorithm for decentralized consensus optimization. SIAM J. Optim. *25* (2015), 2, 944–966. DOI:10.1137/14096668X

[28] G. Tychogiorgos, A. Gkelias, and K. K. Leung: A non-convex distributed optimization framework and its application to wireless ad-hoc networks. IEEE Trans. Wireless Commun. *12* (2013), 9, 4286–4296. DOI:10.1109/TW.2013.072313.120739

[29] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar: A distributed optimization framework for localization and formation control: Applications to vision-based measurements. IEEE Control Systems Magazine *36* (2016), 4, 22–44. DOI:10.1109/MCS.2016.2558401

[30] Z. Tu and S. Liang: Distributed dual averaging algorithm for multi-agent optimization with coupled constraints. Kybernetika *60* (2024), 4, 427–445. DOI:10.14736/kyb-2024-4-0427

[31] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, Ho. Wang, Z. Lin, and K. H. Johansson: A survey of distributed optimization. Ann. Rev. Control *47* (2019), 278–305. DOI:10.1016/j.arcontrol.2019.05.006

[32] Q. Yang, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, and J. Zhang: A distributed swarm optimizer with adaptive communication for large-scale optimization. IEEE Trans. Cybernetics *50* (2019), 7, 3393–3408. DOI:10.1109/TCYB.2019.2904543

[33] Y. Yuan, W. He, W. Du, Y.-Ch. Tian, Q.-L. Han, and F. Qian: Distributed gradient tracking for differentially private multi-agent optimization with a dynamic event-triggered mechanism. IEEE Trans. Systems Man Cybernet.: Systems (2024). DOI:10.1109/tsmc.2024.3357253

[34] Y. Wang and S. Cheng: A stochastic mirror-descent algorithm for solving $AXB = C$ over a multi-agent system. Kybernetika *57* (2021), 2, 256–271. DOI:10.14736/kyb-2021-2-0256

Aijuan Wang, College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054. P. R. China.
e-mail: aijuan321@foxmail.com

Xingmeng Tan, College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054. P. R. China.
e-mail: t15123516666@gmail.com

Hai Nan, Corresponding author. College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, 400054. P. R. China.
e-mail: cqu.nn@163.com