Kybernetika

Ziqin Chen; Shu Liang

Distributed aggregative optimization with quantized communication

DISTRIBUTED AGGREGATIVE OPTIMIZATION WITH QUANTIZED COMMUNICATION

Ziqin Chen and Shu Liang

In this paper, we focus on an aggregative optimization problem under the communication bottleneck. The aggregative optimization is to minimize the sum of local cost functions. Each cost function depends on not only local state variables but also the sum of functions of global state variables. The goal is to solve the aggregative optimization problem through distributed computation and local efficient communication over a network of agents without a central coordinator. Using the variable tracking method to seek the global state variables and the quantization scheme to reduce the communication cost spent in the optimization process, we develop a novel distributed quantized algorithm, called D-QAGT, to track the optimal variables with finite bits communication. Although quantization may lose transmitting information, our algorithm can still achieve the exact optimal solution with linear convergence rate. Simulation experiments on an optimal placement problem is carried out to verify the correctness of the theoretical results.

1. INTRODUCTION

Distributed optimization has gained much research attention due to its wide applications in multi-agent network systems, such as resource allocation [12, 28], machine learning [18, 25] and cloud computing [1]. In distributed optimization, each agent only uses local data and transmits information with its neighbors to minimize a global cost function cooperatively. To solve it, many efficient algorithms have been proposed, such as consensus-based algorithms [3, 4, 19, 24, 29] and dual-decomposition-based algorithms [8, 11, 22, 26].

However, when the number of agents grows, neighboring communication is much slower than computation, which induces a bottleneck to run the above distributed optimization algorithm [10]. Therefore, developing communication-efficient algorithms becomes a new research hotspot. Quantization techniques aim to compress information by reducing the number of bits per communication, and have been successfully applied to

several large-scale engineering tasks recently [15, 17]. With regards to distributed optimization problems, the static quantization method has been utilized in [20, 31], which find the optimal solution with some fixed error bound. For removing the quantization errors, the 3-bit dynamic quantization method with an adjustable quantization level was developed in [27]. It enables distributed quantized subgradient algorithms to achieve the exact optimal solution. After that, various distributed quantized optimization algorithms have arisen, including distributed alternating direction method of multipliers with dynamic quantization methods [16], distributed quantized gradient tracking algorithms [9] and distributed subgradient descent algorithms with amplified-differential compression methods [30].

It should be noted that the local cost function studied in the aforementioned works only depends on its own state variable, like the form as $f_i(x_i)$ associated with inequality constraints $x_i = x_j$, $i \neq j$. However, in many practical applications, such as the multi-agent formation control, optimal placement problems and transportation networks, local cost functions f_i are not only determined by its own variable x_i but also influenced by any other agents' variables x_j. Hence, a novel framework for distributed optimization, called aggregative optimization, was investigated in [2, 13, 14]. The form of the aggregative optimization is often described as follows,

$$
\begin{aligned}
\min_{\boldsymbol{x} \in \mathbb{R}^{Nn}} f(\boldsymbol{x}) &:= \sum_{i=1}^{N} f_i\left(x_i, \chi(\boldsymbol{x})\right), \\
\chi(\boldsymbol{x}) &:= \frac{\sum_{i=1}^{N} g_i\left(x_i\right)}{N},
\end{aligned}
\tag{1}
$$

where $\boldsymbol{x} = [x_i]_{i \in \{1, \ldots, N\}}$ is the global state variable with the local state variable $x_i \in \mathbb{R}^n$. The term $\chi(\boldsymbol{x}) : \mathbb{R}^{Nn} \to \mathbb{R}^r$ is an aggregative variables associated with all agent's state variables. In the distributed aggregative optimization, the local cost function $f_i\left(x_i, \chi(\boldsymbol{x})\right) : \mathbb{R}^{Nn} \to \mathbb{R}$ and $g_i(x_i) : \mathbb{R}^n \to \mathbb{R}^r$ are only privately known by agent i.

For solving the problem (1), the related work [14] proposes the distributed aggregative gradient tracking (D-AGT) algorithm with infinity precision communication. However, in many complex and harsh environments, the existence of communication data rate bottleneck causes that the D-AGT algorithm can not be directly applied. So far as we know, the main difficulties of distributed aggregative optimization with quantized communication may lie in the following three points: i) Each local cost function f_i depends on aggregative variables $\chi(\boldsymbol{x})$, while the gobal aggregative variables and their gradient can not be access to any individual agent i. ii) Only finite bits quantized variables are transmitted between neighboring agents, and it further provides less available information for each agent i. iii) Quantization introduces noise to algorithm updates and thus deteriorates convergence in general. Based on these difficulties, the distributed quantized algorithm for solving aggregative optimization in (1) has not been proposed. Motivated by the above facts, the contributions of this paper are summarized as follows.

- This paper studies distributed aggregative optimization problem (1). Particularly, the local cost function of each agent depends both on its own variable and the aggregative variable, which is the global information that any individual agent can not know. This problem can cover much transitional optimization problems in [1,

4, 18, 19, 24, 28, 29]. Meanwhile, the communication bottleneck is also considered. The only finite number of bits are allowed to interact between the neighboring agents, making it possible to employ the practical bandwidth constraints situation rather than just the infinity communication [2, 13, 14].

- Based on the classical gradient descent algorithm and the quantized technique, we propose a distributed quantized aggregative gradient tracking (D-QAGT) algorithm for solving the problem (1). The novel algorithm uses the variable tracking method to estimate the global aggregative term and its corresponding gradient such that it can converge to the optimal solution at the linear convergence rate. Furthermore, the proposed D-QAGT is a communication-efficient algorithm that significantly saving in the communication overhead.

- A practical optimal placement problem is used in simulation to demonstrate the convergence of the D-QAGT algorithm. Numerical tests show that under 5-bits transmission in each communication step, the proposed D-QAGT algorithm could keep the same linear convergence with the D-AGT algorithm in [14], which utilizes infinity precision communication.

This paper is organized as follows. Section 2 introduces some related preliminaries on basic notations, graph theory and formulates the distributed aggregative optimization problem via two examples. Second 3 provides the proposed distributed algorithm and analyzes its convergence performance. Then, Section 4 gives a numerical experiment and Section 5 concludes the paper.

2. PRELIMINARIES AND PROBLEM FORMULATION

2.1. Basic notations and notions

Denote by $\mathbb{R}^n$ the set of real vectors with n-dimension, $\mathbb{R}^n_+$ the set of vectors with nonnegative coordinates of n-dimension and $\mathbb{R}^{n \times m}$ the set of real matrices with n-rows and m-columns. Denote by $\mathbb{N}$ an integer set. Let $\|\cdot\|$ and $\otimes$ be the standard Euclidean norm and the Kronecker product, respectively. Denote by $\mathbf{1}_N$ and $\mathbf{0}_N$ the column vectors of N dimension with all entries being 1 and 0, respectively. Let I_n be the compatible identity matrix with dimension n. For any vector $x, y \in \mathbb{R}^n$, let x^T be the transpose of x. Denote by $[x_i]_{i \in \Omega}$ the column vector by stacking up x_i associated with $i \in \Omega$. For any square matrix H, denote by $\rho(H)$ the spectral radius.

For a differentiable function $f(x) : \mathbb{R}^n \to \mathbb{R}$, its gradient $\nabla_x f(x) = \text{col}\{\frac{\partial f}{\partial x_i}\}_{i \in \{1,\ldots,n\}} \in \mathbb{R}^n$. We say that a function f is μ-strongly convex for $\mu > 0$ if for any $x, y \in \mathbb{R}^n$, $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|x - y\|^2$, a function f is m-Lipschitz if $\|\nabla_x f(x)\| \leq m$, a function f is l-smooth for any $l > 0$ if for any $x, y \in \mathbb{R}^n$, $\|\nabla_x f(x) - \nabla_y f(y)\|_2 \leq l \|x - y\|_2$ and a function $f(x)$ is radially unbounded on $\mathbb{R}^n$ if for every $x_n \in \mathbb{R}^n$ such that $\|x_n\| \to \infty$, we also have $f(x_n) \to \infty$.

2.2. Graph theory

A directed and strongly connected graph of a multi-agent system is denoted by $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where the node set $\mathcal{V} = \{1, \ldots, N\}$ and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. $(j, i) \in \mathcal{E}$ with

$i \neq j$ means that agent j can sent information to i. Denote $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ is the neighbor set of agent i. The adjacency matrix $A = (a_{ij}) \in R^{N \times N}$ is defined by $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. Let $d_i = |N_i|$ denote the degree of node i and D be the $N \times N$ diagonal matrix such that $D_{ii} = d_i$. Then, the Laplacian matrix is denoted by $L_{\mathcal{G}} = D - A$. It is necessary to display the following assumption and lemma.

Assumption 1. The graph $\mathcal{G}$ is directed and strongly connected, and the adjacency matrix $A \in \mathbb{R}^{N \times N}$ is doubly stochastic, i.e, $\mathbf{1}_N^T A = \mathbf{1}_N^T$ and $A\mathbf{1}_N = \mathbf{1}_N$.

Lemma 2.1. Horn et al. [7] Under Assumption 1, the following properties hold.

i) $(A \otimes I_n)(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_n) = (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_n)(A \otimes I_n) = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_n$.

ii) $\|A - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\| < 1$.

iii) $\|A - I_N\| \leq 2$.

iv) $\|(A \otimes I_n)x - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_n)x\| \leq \|A - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\|\|x - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_n)x\|$.

2.3. Problem formulations

For aggregative optimization problem (1), the goal of the agents is to collectively seek an optimal solution of the global function, which is the sum of the local cost functions. It is different from the aggregative game [5, 6], where each agent selfishly minimizes its own local cost function. In the most situation, the global performance of (1) may be better than that of the aggregative game, which is illustrated by the following example.

Example 1.   (Bandwidth sharing problem) Consider N users in the bandwidth sharing problem, in which each user wants to have part of a shared resource. Assume that the maximum capacity of a shared channel says 1 and the strategy of each user i, $i \in \{\mathcal{V}\}$ is to send x_i units of flow along the channel for some value $x_i \in [0, 1]$. If the total bandwidth $\sum_{i=1}^N x_i \geq 1$, no user gets any benefit. If $\sum_{i=1}^N x_i < 1$, then the local payoff function of each user i is $f_i(x_i, \chi(\boldsymbol{x})) = x_i(1 - \chi(\boldsymbol{x}))$ with $\chi(\boldsymbol{x}) = \sum_{j \in N} x_j$. Based on [23], when each user selfishly maximizes its own payoff, the optimal strategy for user i, $i \in \{1, \ldots, N\}$ is $x_i^* = 1/(N+1)$. Thus, the Nash equilibrium is $[1/(N+1); \ldots; 1/(N+1)]$. In this aggregative game, the local payoff of each user i is $f_i = (1/(N+1))^2$ and the global payoff is $\sum_{i=1}^N f_i = N/(N+1)^2$.

However, If all users cooperate to maximize the global payoff function, the optimal solution is computed as $\boldsymbol{x}^* = [1/2N; \ldots; 1/2N]$. In this setting, the local payoff of each user i is $f_i = 1/4N$ and the global payoff is $\sum_{i=1}^N f_i = 1/4 > N/(N+1)^2$. This example indicates that all agents will perform better in a cooperative manner compared with the aggregative game in a noncooperative manner, which motivates us to study the aggregative optimization as (1).

Note that in the cooperation process among agents, neighboring infinite precision communication often incurs expensive communication overheads. In order to solve (1) with the cost of all communications as low as possible, we adopt quantized communication. The main idea of quantization is to map the input data to a countable set of code values. More specifically, we use a simple yet effective uniform quantizer to divide

the input domain into the same size quantization cells, in which a unique code value represents all elements within a cell. For a pair of $(j,i) \in \mathcal{E}$, agent j transmits this code value to its neighbors i, and the agents i recovers the state of agent j based on the received code value. To illustrate the problem (1) with quantized communication better, we present the following optimal placement example.

Example 2. (Optimal placement problem) Consider N agents to protect a target at position $p_0 \in \mathbb{R}^2_+$. We hope that some weighted center of all agents could track the target p_0. Denote the position of agent i by $x_i \in \mathbb{R}^2_+$ and the weight center of all agents by $\chi(\boldsymbol{x}) = \sum_{i=1}^{N} \sqrt{Nx_i^2}/N$. Then the problem is cast as problem (1) with $\sum_{i=1}^{N} f_i(x_i, \chi(\boldsymbol{x})) = \sum_{i=1}^{N} \alpha_i \|\chi(\boldsymbol{x}) - p_0\|^2$, where $\alpha_i > 0$ is the weight constant.

In the distributed framework, $\chi(\boldsymbol{x})$ is the global information and cannot be known directly for all agents. Hence, $\chi_i \in \mathbb{R}^2$, $i \in \{1, \ldots, N\}$ is leveraged for each agent i to track the $\chi(\boldsymbol{x})$, and χ_i need to be interacted between neighbors over network graph $\mathcal{G}$. However, transmitting χ_i would consume amount of bandwidth resource, so we quantize χ_i before transmitting it. Particularly, by using quantizer $Q(\cdot)$, each coordinate of real value χ_i is mapped into a countable set of code values. i. e., $Q(\chi_i) : \mathbb{R}^2_+ \to \mathbb{N}^2_+$. Then agent i just transmits the code of quantized message $Q(\chi_i)$ rather than that of the raw vectors χ_i so as to the reduction of communication cost. More details on the design of the quantization scheme will be given subsequently.

To move forward, we define $f(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{N} f_i(x_i, z_i) : \mathbb{R}^{Nn+Nr} \to \mathbb{R}$ for any $\boldsymbol{x} = [x_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nn}$ and $\boldsymbol{z} = [z_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr}$. The gradients of $f(\boldsymbol{x}, \boldsymbol{z})$ is defined by $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{z}) = [\nabla_{x_i} f_i(x_i, z_i)]_{i \in \mathcal{V}}$ and $\nabla_{\boldsymbol{z}} f(\boldsymbol{x}, \boldsymbol{z}) = [\nabla_{z_i} f_i(x_i, z_i)]_{i \in \mathcal{V}}$, respectively. We make the following necessary assumption.

Assumption 2. The functions arisen in (1) satisfy

i) The cost function f is differentiable, μ-strongly convex and locally l_1-smooth.

ii) $\nabla_{\boldsymbol{z}} f(\boldsymbol{x}, \boldsymbol{z})$ is locally l_2-Lipschitz continuous.

iii) For all $i \in \mathcal{V}$, $g_i(x_i)$ are differentiable and $\nabla_{x_i} g_i(x_i) \in R^{n \times r}$ is locally bounded, i. e., $\|\nabla_{x_i} g_i(x_i)\| \le \frac{l_3}{N}$ for some positive l_3.

Note that Property i) means that ∇f and

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{z}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) \mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_{z_i} f_i(x_i, z_i), \tag{2}$$

are locally l_1-Lipschitz continuous with $g(\boldsymbol{x}) = [g_i(x_i)]_{i \in \mathcal{V}} : \mathbb{R}^{Nn} \to \mathbb{R}^{Nr}$. At the same time, Property iii) ensures that $\|\nabla_{\boldsymbol{x}} g(\boldsymbol{x})\| \le l_3$.

3. MAIN RESULT

In this section, we present the design of distributed quantized algorithm and give its convergence analysis.

3.1. Distributed quantized algorithm design

For solving the problem (1), each agent j, $j \in \mathcal{V}$ holds two states $\chi_j \in \mathbb{R}^r$ and $y_j \in \mathbb{R}^r$ to track the average $\chi(\boldsymbol{x})$ and the gradient sum $\frac{1}{N} \sum_{j=1}^{N} \nabla_\chi f_j(x_j, \chi(\boldsymbol{x}))$, respectively. Meanwhile, each agent quantizes χ_j and y_j for the information interaction. Thus, each agent is associated with an encoder and its neighbors possess the corresponding decoder. Firstly, the uniform quantizer $q(\cdot)$ installed in the encoder is introduced as follows.

Quantizer: A uniform quantizer is described by the function $q(\cdot) : \mathbb{R} \to \mathbb{N}$, in which

$$
q(x) = \begin{cases} 0, & \text{if } -\frac{1}{2} \le x \le \frac{1}{2}, \\ i, & \text{if } \frac{2i-1}{2} < x \le \frac{2i+1}{2}, i = 1, \ldots, L, \\ L, & \text{if } x > \frac{2L+1}{2}, \\ -q(-x), & \text{if } x < -\frac{1}{2}. \end{cases} \tag{3}
$$

At each time instant, the encoder outputs one symbol from a finite and possibly set $\mathcal{S}_k$ to the neighbors. Then, the data rate is defined as follows.

Definition 3.1. The communication data rate to quantify the information rate at which the channel supports is defined in the asymptotically average sense

$$
R = \liminf_{k \to \infty} \frac{1}{k} \sum_{j=1}^{k} \log_2 (|\mathcal{S}_j|) \tag{4}
$$

where $|\mathcal{S}_j|$ denotes the cardinality of set $\mathcal{S}_j$.

In practice, it is not necessary to transmit any information when the output of the quantizer is zero, thus, the communication process of each agent is required to transmit $\lceil \log_2 (|\mathcal{S}_j|) \rceil = \lceil \log_2(2L) \rceil$ bits at most for the above $2L + 1$-level quantizer (3).

Next, we design an encoder-decoder scheme for each pair of agents (j, i), $j \in \mathcal{N}_i$. For clarify, define $Q(\mathbf{a}) = [q(a_i)]_{i \in \{1,\ldots,N\}} \in \mathbb{R}^N$ for the vector $\mathbf{a} = [a_i]_{i \in \{1,\ldots,N\}} \in \mathbb{R}^N$. Agent j quantizes its states χ_j and y_j, then transmits the code of $Q(\chi_j)$ and $Q(y_j)$ to its neighbors i. Agent i receives the code of $Q(\chi_j)$ and $Q(y_j)$ and estimates the states of agent j denoted by $\hat{\chi}_j$ and $\hat{y}_j$, $j \in \mathcal{N}_i$.

Remark 3.2. In the dynamic quantized control, the scaling function represents the quantized precision and is designed as a decaying sequence to adaptively adjust the encoder. For convergence analysis of the proposed algorithm, the scaling function must be designed carefully such that the agents gradually increase the accuracy of states recovery of its neighbors and the quantization error gradually decays to zero. Noted that for different agent i, the scaling function $l_i(k)$ can be designed differently, which is only needed to satisfy (5) and (15). It does not bring additional difficulty in the convergence analysis of our algorithm, but it causes the notation to be much more complicated. For clarity, we use an unified notation $l(k)$ for different agent i. For much same reason, we also use the $l(k)$ instead of $l_{\chi_j}(k)$ or $l_{y_j}(k)$ for vectors χ_j or y_j.

Encoder: The encoder is installed in agent j, $j \in \mathcal{V}$.

Agent j generates r-dimensional quantized outputs $s_{\chi j}(k)$ and $s_{yj}(k)$, and then transmits them to neighbors i for any $k \in \mathbb{N}$,

$$s_{\chi j}(0) = Q\left(\frac{\chi_j(0)}{l(0)}\right), \qquad s_{yj}(0) = Q\left(\frac{y_j(0)}{l(0)}\right), \qquad l(k) = l(0)\gamma^k, \qquad (5)$$

$$s_{\chi j}(k+1) := Q\left(\frac{\chi_j(k+1) - \hat{\chi}_j(k)}{l(k+1)}\right), \qquad (6)$$

$$s_{yj}(k+1) := Q\left(\frac{y_j(k+1) - \hat{y}_j(k)}{l(k+1)}\right), \qquad (7)$$

where $l(k)$ is the decaying scaling function with any positive initial constant $l(0) > 0$. The rate of the scaling function γ is given in the following (15).

Decoder: The decoder is installed in agent i, $j \in \mathcal{N}_i$.

Agent i receives r-dimensional quantized outputs $s_{\chi j}(k)$ and $s_{yj}(k)$ from its neighbors j, $j \in \mathcal{N}_i$, and then recovers its neighbors' states $\hat{\chi}_j$ and $\hat{y}_j$ as follows,

$$\hat{\chi}_j(0) = l(0)s_{\chi j}(0), \qquad \hat{y}_j(0) = l(0)s_{yj}(0), \qquad (8)$$

$$\hat{\chi}_j(k+1) = l(k+1)s_{\chi j}(k+1) + \hat{\chi}_j(k), \qquad (9)$$

$$\hat{y}_j(k+1) = l(k+1)s_{yj}(k+1) + \hat{y}_j(k), \ k \in \mathbb{N}, \qquad (10)$$

where the design of $l(k)$ is the same as the encoder.

Tab. 1. Dataset Similarity Measurement Based on Euclidean Distance.

Agent i receives quantized outputs $s_{\chi j}$ and s_{yj} from its neighbors j, and then estimates its neighbors's states through a decoder defined as follows.

Based on the quantized communication associated with the above encoder-decoder pair, for $i \in \mathcal{V}$, the ith agent updates its real-valued state $x_i \in \mathbb{R}^n$ via the following distributed quantized algorithm.

It should be noted that the Q-DAGT algorithm combines the classical gradient descent algorithm, the variables tracking techniques and quantization communication methods. It is the first proposed to solve the aggregative network optimization problem with the communication bottleneck.

Remark 3.3. Compared with the existed D-AGT algorithm in [14], in which the actual neighbors' states of χ_j and y_j, $j \in \mathcal{N}_i$ are needed, our D-QAGT algorithm just relies on the quantized communication via using the estimated neighbors' states $\hat{\chi}_j$ and $\hat{y}_j$, $j \in \mathcal{N}_i$. These estimated states come from discrete-time communication with finite bits at each round of communication. Regarding this fact, the D-QAGT algorithm can significantly save the communication resources and resolve a finite bandwidth bottleneck, which broadens the range of applications of the D-AGT algorithm.

Algorithm 1: Distributed Quantized Aggregative Gradient Tracking (D-QAGT)

For any $k \in \mathbb{N}$, each agent i, $i \in \mathcal{V}$ updates its states x_i, χ_i and y_i as follows,

$$x_i(k+1) = x_i(k) - \alpha \left[\nabla_{x_i} f_i(x_i(k), \chi_i(k)) + \nabla_{x_i} g_i(x_i(k)) y_i(k) \right], \tag{11}$$

$$\chi_i(k+1) = \sum_{j=1}^{N} a_{ij} \hat{\chi}_j(k) + g_i(x_i(k+1)) - g_i(x_i(k)) + \chi_i(k) - \hat{\chi}_i(k), \tag{12}$$

$$y_i(k+1) = \sum_{j=1}^{N} a_{ij} \hat{y}_j(k) + \nabla_{\chi_i} f_i(x_i(k+1), \chi_i(k+1)) - \nabla_{\chi_i} f_i(x_i(k), \chi_i(k)) + y_i(k) - \hat{y}_i(k), \tag{13}$$

where the stepsize α is given in the following (14).

The parameters in Algorithm 1 and Encoder-Decoder scheme are chosen as follows.

- The initial states $x_i(0)$, $\chi_i(0)$ and $y_i(0)$ satisfy

  i) $\|x_i(0) - x^*\|_\infty \leq c_0$.
  ii) $\chi_i(0) = g_i(x_i(0))$ and $\|\chi_i(0)\|_\infty \leq c_1$.
  iii) $y_i(0) = \nabla_{\chi_i} f_i(x_i(0), \chi_i(0))$ and $\|y_i(0)\|_\infty \leq c_2$.

- The stepsize α satisfies

$$\alpha \in \left(0, \frac{\mu(1-\kappa)^2}{l_3(\mu + l_1 + l_2 l_3)((1-\kappa)(l_1 + l_2 + l_2 l_3) + 2 l_2 l_3)} \right), \tag{14}$$

  with the constant $\kappa = \|A - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T\| < 1$, which derived from Lemma 2.1.

- The rate of the scaling function γ satisfies

$$\gamma \in (\rho(H), 1), \tag{15}$$

  where $\rho(H)$ is the spectral radius of $H(\alpha)$ defined as

$$H(\alpha) = \begin{bmatrix} (1 - \mu\alpha) & \alpha l_1 & \alpha l_3 \\ \alpha l_1 l_3 (1 + l_3) & \kappa + \alpha l_1 l_3 & \alpha l_3^2 \\ \alpha l_1 l_2 (1 + l_3)^2 & \alpha l_1 l_2 (1 + l_3) + 2 l_2 & \kappa + \alpha l_2 l_3 (1 + l_3) \end{bmatrix}. \tag{16}$$

Remark 3.4. Noted that in quantized control of the multi-agent systems, the upper bound of the initial states is needed to estimate the upper bound of the tracking errors. These errors determines the design of the scaling function, which is required to satisfy the unsaturation of the quantizer. Similarly, in distributed aggregative optimization problem, the tracking errors rely on the upper bound of the initial states of $x_i(0)$, $\chi_i(0)$ and $y_i(0)$. Thus, we still need the upper bounds of these variables.

3.2. Convergence analysis

In this section, we present the convergence analysis for the D-QAGT algorithm. For explicit illustration, we introduce the following notations:

$$\boldsymbol{x} = [x_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nn}, \qquad\qquad \boldsymbol{L} = (L_{\mathcal{G}} \otimes I_r) \in \mathbb{R}^{Nr \times Nr},$$

$$\boldsymbol{\chi} = [\chi_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr}, \qquad\qquad \boldsymbol{y} = [y_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr},$$

$$\bar{\chi} = \frac{1}{N} \sum_{i=1}^{N} \chi_i \in \mathbb{R}^r, \qquad\qquad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \in \mathbb{R}^r,$$

$$\hat{\boldsymbol{\chi}} = [\hat{\chi}_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr}, \qquad\qquad \hat{\boldsymbol{y}} = [\hat{y}_i]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr},$$

$$\boldsymbol{e_\chi} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}} \in \mathbb{R}^{Nr}, \qquad\qquad \boldsymbol{e_y} = \boldsymbol{y} - \hat{\boldsymbol{y}} \in \mathbb{R}^{Nr},$$

$$g(\boldsymbol{x}) = [g_i(x_i)]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr}, \qquad \nabla_{\boldsymbol{x}} g(\boldsymbol{x}) = [\nabla_{x_i} g_i(x_i)]_{i \in \mathcal{V}} \in \mathbb{R}^{Nn \times Nr},$$

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\chi}) = [\nabla_{x_i} f_i(x_i, \chi_i)]_{i \in \mathcal{V}} \in \mathbb{R}^{Nn}, \qquad \nabla_{\boldsymbol{\chi}} f(\boldsymbol{x}, \boldsymbol{\chi}) = [\nabla_{\chi_i} f_i(x_i, \chi_i)]_{i \in \mathcal{V}} \in \mathbb{R}^{Nr}.$$

By using $\boldsymbol{L} = I_{Nr} - (A \otimes I_r)$, we first formulate a compact form of dynamics $(11)-(13)$ as the follows,

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) - \alpha \left[\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(k), \boldsymbol{\chi}(k)) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(k)) \boldsymbol{y}(k) \right], \tag{17}$$

$$\boldsymbol{\chi}(k+1) = \boldsymbol{L} \boldsymbol{e_\chi}(k) + g(\boldsymbol{x}(k+1)) - g(\boldsymbol{x}(k)) + (A \otimes I_r) \boldsymbol{\chi}(k), \tag{18}$$

$$\boldsymbol{y}(k+1) = \boldsymbol{L} \boldsymbol{e_y}(k) + \nabla_{\boldsymbol{\chi}} f(\boldsymbol{x}(k+1), \boldsymbol{\chi}(k+1)) - \nabla_{\boldsymbol{\chi}} f(\boldsymbol{x}(k), \boldsymbol{\chi}(k)) + (A \otimes I_r) \boldsymbol{y}(k). \tag{19}$$

Next, we establish the equivalence of the fixed point of the D-QAGT algorithm and the optimal solution to the problem (1).

Lemma 3.5. Denote $\boldsymbol{x}^*$, $\boldsymbol{\chi}^*$ and $\boldsymbol{y}^*$ as the fixed points of $(17)-(19)$. Under Assumptions 1 and 2, $\boldsymbol{x}^*$ is the optimal solution to problem (1).

P r o o f. Based on Assumption 1 on $\mathbf{1}_{Nr}^T \boldsymbol{L} = \mathbf{0}_N^T$, we multiply $\frac{1}{N} \mathbf{1}_{Nr}^T$ on both side of (18) and (19) to obtain that

$$\bar{\chi}(k+1) = \bar{\chi}(k) + \frac{1}{N} \sum_{i=1}^{N} g_i(x_i(k+1)) - \frac{1}{N} \sum_{i=1}^{N} g_i(x_i(k)), \tag{20}$$

$$\bar{y}(k+1) = \bar{y}(k) + \frac{1}{N} \sum_{i=1}^{N} \nabla_{\chi_i} f_i(x_i(k+1), \chi_i(k+1)) - \frac{1}{N} \sum_{i=1}^{N} \nabla_{\chi_i} f_i(x_i(k), \chi_i(k)). \tag{21}$$

It follows from a simple recursion that

$$\bar{\chi}(k) - \frac{1}{N} \sum_{i=1}^{N} g_i(x_i(k)) = \bar{\chi}(0) - \frac{1}{N} \sum_{i=1}^{N} g_i(x_i(0)), \tag{22}$$

$$\bar{y}(k) - \frac{1}{N} \sum_{i=1}^{N} \nabla_{\chi_i} f_i(x_i(k), \chi_i(k)) = \bar{y}(0) - \frac{1}{N} \sum_{i=1}^{N} \nabla_{\chi_i} f_i(x_i(0), \chi_i(0)). \tag{23}$$

Combing with $\chi_i(0) = g_i(x_i(0))$ and $y_i(0) = \nabla_{\chi_i} f_i(x_i(0), \chi_i(0))$,

$$\bar{\chi}(k) \;=\; \frac{1}{N} \sum_{i=1}^{N} g_i(x_i(k)) = \chi(\boldsymbol{x}), \tag{24}$$

$$\bar{y}(k) \;=\; \frac{1}{N} \sum_{i=1}^{N} \nabla_{\chi_i} f_i(x_i(k), \chi_i(k)). \tag{25}$$

By substituting the fixed point $\boldsymbol{x}^*$, $\boldsymbol{\chi}^*$ and $\boldsymbol{y}^*$ into $(17)-(19)$,

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*, \boldsymbol{\chi}^*) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^*) \boldsymbol{y}^* = \mathbf{0}_{Nn}, \tag{26}$$

$$\boldsymbol{L} \boldsymbol{\chi}^* = \mathbf{0}_{Nr}, \quad \boldsymbol{L} \boldsymbol{y}^* = \mathbf{0}_{Nr}, \tag{27}$$

where $\boldsymbol{e}_{\boldsymbol{\chi}^*} = \boldsymbol{e}_{\boldsymbol{y}^*} = \mathbf{0}_{Nr}$ are used. Noticing from (27) that there exist $\chi_i^* = \chi_j^* = \chi^*$ and $y_i^* = y_j^* = y^*$. Meanwhile, in view of $(24)-(25)$,

$$\chi^* \;=\; \bar{\chi}^* = \frac{1}{N} \sum_{i=1}^{N} g_i(x_i^*) = \chi(\boldsymbol{x}^*), \tag{28}$$

$$y^* \;=\; \bar{y}^* = \frac{1}{N} \sum_{i=1}^{N} \nabla_\chi f_i(x_i^*, \chi(\boldsymbol{x}^*)). \tag{29}$$

Based on Assumption 2, we compute the gradient of $f(\boldsymbol{x})$ in $\boldsymbol{x}^*$ as follows,

$$
\begin{aligned}
\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*) \;&=\; \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*, \mathbf{1}_N \otimes \chi^*) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^*)[\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_{\chi_i} f_i(x_i^*, \chi_i^*)], \\
&=\; \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*, \mathbf{1}_N \otimes \chi^*) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^*) \left[\mathbf{1}_N \otimes y^* \right], \\
&=\; \mathbf{0}_{Nn},
\end{aligned}
\tag{30}
$$

where (28) and (29) are used in the second equality and (26) is used in the third equality. (30) implies that $\boldsymbol{x}^*$ is the optimal solution to problem (1).   □

Remark 3.6. The optimal condition (27) implies that the fixed points $\boldsymbol{\chi}^* = \mathbf{1}_N \otimes \chi^*$ and $\boldsymbol{y}^* = \mathbf{1}_N \otimes y^*$. Meanwhile, $(28)-(29)$ means that $\boldsymbol{\chi}^* = \mathbf{1}_N \otimes \chi(\boldsymbol{x}^*)$ and $\boldsymbol{y}^* = \mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_\chi f_i(x_i^*, \chi(\boldsymbol{x}^*))$. Hence, in the D-QAGT algorithm, $\chi_i(k)$ is leveraged for agent i to track t*he global information $\chi(\boldsymbol{x})$ and $y_i(k)$ is leveraged for agent i to seek the gradient sum $\frac{1}{N} \sum_{i=1}^{N} \nabla_\chi f_i(x_i, \chi(\boldsymbol{x}))$.

Then, the following lemmas give the convergence analysis framework of the D-QAGT algorithm with two intermediate results.

Lemma 3.7. Define $\Theta(k) = [\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|; \|\boldsymbol{\chi}(k) - \mathbf{1}_N \otimes \bar{\chi}(k)\|; \|\boldsymbol{y}(k) - \mathbf{1}_N \otimes \bar{y}(k)\|]$ and $E(k) = [0; 2\|\boldsymbol{e}_{\boldsymbol{\chi}}(k)\|; 2l_2\|\boldsymbol{e}_{\boldsymbol{\chi}}(k)\| + 2\|\boldsymbol{e}_{\boldsymbol{y}}(k)\|]$. Under Assumptions 1 and 2 and considering the iterations on $(11)-(13)$, the following inequality holds

$$\Theta(k+1) \leq H(\alpha) \Theta(k) + E(k). \tag{31}$$

Furthermore, the spectral radius $\rho(H) < 1$.

P r o o f . See Appendix 6.1. □

The 3-dimensions vector $\Theta(k)$ describes the distance between the iteration $\boldsymbol{x}(k)$, $\boldsymbol{\chi}(k)$, $\boldsymbol{y}(k)$ and the fixed point $\boldsymbol{x}^*$, $\boldsymbol{\chi}^*$, $\boldsymbol{y}^*$, respectively. Lemma 3.7 provides the upper bound of $\Theta(k)$, which is related to the stepsize matrix $H(\alpha)$ and the quantization error vector $E(k)$. Hence, we present the following lemma for analyzing the convergence of $\|H^k(\alpha)\|$.

Lemma 3.8. Under Assumptions 1 and 2 and considering the iterations on $(11)-(13)$, for any $\epsilon \in (0, \min(\gamma - \rho(H), 2\|H(\alpha)\|))$, the following inequality holds

$$\|H^k(\alpha)\| \leq c_3(\rho(H) + \epsilon)^k, \tag{32}$$

where the constant $c_3 = 3\sqrt{3}\max\{\frac{4\|H(\alpha)\|^2}{\epsilon^2}, \frac{\epsilon^2}{4\|H(\alpha)\|^2}\}$.

P r o o f . See Appendix 6.2. □

Note that the chosen of the constant ϵ can ensure $\rho(H) + \epsilon < 1$ due to $\gamma \in (\rho(H), 1)$. Then Lemma 3.7 provides the linear convergence of $\|H^k(\alpha)\|$. Summarizing the above lemmas, we give the convergence of the D-QAGT algorithm in the following theorem.

Theorem 3.9. Under Assumptions 1 and 2, if the number of the quantization levels

$$2L + 1 \geq 2\max\{\frac{\zeta C_0 + 3C_1}{l_0\gamma}, \frac{\sqrt{4c_1^2 + 4c_2^2}}{l_0}\} + 1, \tag{33}$$

with

$$\zeta = \max\{\alpha l_1 l_2(1 + l_3) + \alpha l_1 l_3(1 + l_3), l_2 + \alpha l_1 l_2 + \alpha l_1 l_3 + 2, \alpha l_3 l_2 + \alpha l_3^2 + 2\}, \tag{34}$$

$$C_0 = c_3\sqrt{Nnc_0^2 + 4Nr(c_1^2 + c_2^2)} + \frac{c_3 C_1}{\gamma - \rho(H) - \epsilon}, \tag{35}$$

$$C_1 = 2(l_2 + 1)\sqrt{Nr}l_0, \tag{36}$$

and $\epsilon \in (0, \min(\gamma - \rho(H), 2\|H(\alpha)\|))$, then $\boldsymbol{x} = [x_i]_{i \in \mathcal{V}}$ generated by the D-QAGT algorithm can converge to the optimizer of problem (1) at a linear convergence rate.

P r o o f . We prove Theorem 3.9 by proving the following inequalities hold simultaneously.

$$\|\Theta(k)\| \leq C_0\gamma^k, \tag{37}$$

$$\|E(k)\| \leq C_1\gamma^k, \tag{38}$$

where $\Theta(k)$ and $E(k)$ are defined the same as Lemma 3.7.

When $k = 0$, based on the definition of $\Theta(k)$ and the initial values of $x_i(0)$, $\chi_i(0)$ and $y_i(0)$, we have that

$$\|\Theta(0)\| \leq \sqrt{Nnc_0^2 + 4Nr(c_1^2 + c_2^2)}. \tag{39}$$

It follows from $c_3 > 1$ and $\frac{c_3 C_1}{\gamma - \rho(H) - \epsilon} > 0$ that

$$\|\Theta(0)\| \leq c_3 \sqrt{Nnc_0^2 + 4Nr(c_1^2 + c_2^2)} + \frac{c_3 C_1}{\gamma - \rho(H) - \epsilon} = C_0,$$

which ensures that (37) holds at $k = 0$.

When $k \in \mathbb{N}^+$, we assume that (37) and (38) hold for any $k \leq k_1$, $k_1 \in \mathbb{N}^+$. We first prove that (37) holds for $k = k_1 + 1$. Iterating (31) yields

$$\Theta(k_1 + 1) \leq H^{k_1+1}(\alpha)\Theta(0) + \sum_{i=0}^{k_1} H^i(\alpha)E(k_1 - i). \tag{40}$$

Substituting (32) into (40) and using (38) holds for any $k \leq k_1$, $k_1 \in \mathbb{N}^+$, we have that

$$\|\Theta(k_1 + 1)\| \leq c_3 \bar{\gamma}^{k_1+1}\|\Theta(0)\| + \sum_{i=0}^{k_1} c_3 \bar{\gamma}^i \|E(k_1 - i)\|,$$

$$\leq c_3 \bar{\gamma}^{k_1+1}\|\Theta(0)\| + \sum_{i=0}^{k_1} c_3 \bar{\gamma}^i C_1 \gamma^{k_1-i},$$

$$\leq (c_3\|\Theta(0)\| + \frac{aC_1}{\gamma - \bar{\gamma}})\gamma^{k_1+1}, \tag{41}$$

where $\bar{\gamma} = \rho(H(\alpha)) + \epsilon$. Based on the definition of C_0, it follows that

$$\|\Theta(k_1 + 1)\| \leq C_0 \gamma^{k_1+1}. \tag{42}$$

Then, we prove (38) for $k = k_1 + 1$. Define

$$W(k + 1) = [\boldsymbol{\chi}(k + 1) - \hat{\boldsymbol{\chi}}(k); \boldsymbol{y}(k + 1) - \hat{\boldsymbol{y}}(k)].$$

For $k = k_1 + 1$, taking norm on $W(k_1 + 1)$ yields that

$$\|W(k_1 + 1)\| = \|\boldsymbol{\chi}(k_1 + 1) - \hat{\boldsymbol{\chi}}(k_1)\| + \|\boldsymbol{y}(k_1 + 1) - \hat{\boldsymbol{y}}(k_1)\|,$$
$$\leq \|\boldsymbol{\chi}(k_1 + 1) - \boldsymbol{\chi}(k_1)\| + \|\boldsymbol{y}(k_1 + 1) - \boldsymbol{y}(k_1)\| + \|\boldsymbol{e}_{\boldsymbol{\chi}}(k)\| + \|\boldsymbol{e}_{\boldsymbol{y}}(k)\|. \tag{43}$$

Considering the first term of (43), in light of $\|A - I_N\| = \|-L\| \leq 2$, it follows from (18) that

$$\|\boldsymbol{\chi}(k_1 + 1) - \boldsymbol{\chi}(k_1)\|$$
$$\leq \|(A - I_N)(\boldsymbol{\chi}(k_1) - \mathbf{1}_N \otimes \bar{\chi}(k_1)) + (I_N - A)\boldsymbol{e}_{\boldsymbol{\chi}}(k_1)\| + l_3\|\boldsymbol{x}(k_1 + 1) - \boldsymbol{x}(k_1)\|,$$
$$\leq 2\|\boldsymbol{\chi}(k_1) - \mathbf{1}_N \otimes \bar{\chi}(k_1)\| + 2\|\boldsymbol{e}_{\boldsymbol{\chi}}(k_1)\| + l_3\|\boldsymbol{x}(k_1 + 1) - \boldsymbol{x}(k_1)\|. \tag{44}$$

Substituting (60) into (44) yields

$$\|\boldsymbol{\chi}(k_1 + 1) - \boldsymbol{\chi}(k_1)\|$$
$$\leq \alpha l_1 l_3(1 + l_3)\|\boldsymbol{x}(k_1) - \boldsymbol{x}^*\| + (\alpha l_1 l_3 + 2)\|\boldsymbol{\chi}(k_1) - \mathbf{1}_N \otimes \bar{\chi}(k)\|$$
$$+ \alpha l_3^2\|\boldsymbol{y}(k_1) - \mathbf{1}_N \otimes \bar{y}(k_1)\| + 2\|\boldsymbol{e}_{\boldsymbol{\chi}}(k_1)\|. \tag{45}$$

Similarly, considering the second term of (43) and using (60) again, it follows from (19) that

$$
\begin{aligned}
&\|\boldsymbol{y}(k_1+1) - \boldsymbol{y}(k_1)\| \\
&\leq \|(I_N - A)(\boldsymbol{y}(k_1) - 1_N \otimes \bar{y}(k_1))\| + \|(A - I_N)\boldsymbol{e_y}(k_1) + \nabla_{\boldsymbol{\chi}} f(\boldsymbol{x}(k_1+1), \boldsymbol{\chi}(k_1+1)) \\
&\quad - \nabla_{\boldsymbol{\chi}} f(\boldsymbol{x}(k_1), \boldsymbol{\chi}(k_1))\|, \\
&\leq 2\|\boldsymbol{y}(k_1) - 1_N \otimes \bar{y}(k_1)\| + 2\|\boldsymbol{e_y}(k_1)\| + l_2\|\boldsymbol{x}(k_1+1) - \boldsymbol{x}(k_1)\| + l_2\|\boldsymbol{\chi}(k_1+1) - \boldsymbol{\chi}(k_1)\|, \\
&\leq \alpha l_1 l_2 (1 + l_3)\|\boldsymbol{x}(k_1) - \boldsymbol{x}^*\| + (l_2 + \alpha l_1 l_2)\|\boldsymbol{\chi}(k_1) - 1_N \otimes \bar{\chi}(k_1)\| \\
&\quad + (2 + \alpha l_3 l_2)\|\boldsymbol{y}(k_1) - 1_N \otimes \bar{y}(k_1)\| + 2\|\boldsymbol{e_y}(k_1)\|.
\end{aligned}
\tag{46}
$$

Combing with (45) and (46), we bound $\|W(k_1+1)\|$ as

$$
\begin{aligned}
&\|W(k_1+1)\| \\
&\leq (\alpha l_1 l_2 (1 + l_3) + \alpha l_1 l_3 (1 + l_3))\|\boldsymbol{x}(k_1) - \boldsymbol{x}^*\| + (l_2 + \alpha l_1 l_2 + \alpha l_1 l_3 + 2) \\
&\quad \|\boldsymbol{\chi}(k_1) - 1_N \otimes \bar{\chi}(k_1)\| + (\alpha l_3 l_2 + \alpha l_3^2 + 2)\|\boldsymbol{y}(k_1) - 1_N \otimes \bar{y}(k_1)\| \\
&\quad + 3\|\boldsymbol{e_y}(k_1)\| + 3\|\boldsymbol{e_\chi}(k_1)\|.
\end{aligned}
\tag{47}
$$

Since (37) and (38) hold for any $k = k_1$, we conclude that

$$
\begin{aligned}
\|W(k+1)\| &\leq \zeta \Theta(k_1) + 3(\|\boldsymbol{e_\chi}(k_1)\| + \|\boldsymbol{e_y}(k_1)\|), \\
&\leq (\zeta C_0 + 3C_1)\gamma^{k_1}.
\end{aligned}
\tag{48}
$$

It further yields

$$
\frac{\|W(k_1+1)\|_\infty}{l(k_1+1)} \leq \frac{\|W(k_1+1)\|}{l(k_1+1)} \leq \frac{\zeta C_0 + 3C_1}{l_0 \gamma},
\tag{49}
$$

which means that $\frac{\|W(k_1+1)\|_\infty}{l(k_1+1)} \leq L$ and the quantizer is unsaturated at $k = k_1 + 1$ by recalling from the definition of L in (33). According to the design of the encoder in (6) and (7), when the quantizer is unsaturated at $k_1 + 1$, there are $|\hat{\chi}_{im}(k_1+1) - \chi_{im}(k_1+1)| \leq \frac{l(k_1+1)}{2}$, $m = 1, \ldots, r$ and $|\hat{y}_{im}(k_1+1) - y_{im}(k_1+1)| \leq \frac{l(k_1+1)}{2}$, $m = 1, \ldots, r$ with χ_{im} and y_{im} are the m-elements of the vector χ_i and y_i. Hence, based on the definition of $\boldsymbol{e_\chi}$ and $\boldsymbol{e_y}$, we have $\|\boldsymbol{e_\chi}(k_1+1)\|_\infty \leq l(k_1+1)$ and $\|\boldsymbol{e_y}(k_1+1)\|_\infty \leq l(k_1+1)$. Together with the definition of $E(k)$, one has that

$$
\begin{aligned}
\|E(k_1+1)\| &\leq (2 + 2l_2)\|\boldsymbol{e_\chi}(k_1+1)\| + 2\|\boldsymbol{e_y}(k_1+1)\|, \\
&\leq (2 + 2l_2)(\|\boldsymbol{e_\chi}(k_1+1)\| + \|\boldsymbol{e_y}(k_1+1)\|), \\
&\leq (1 + l_2)\sqrt{N r} l(k_1+1) = C_1 \gamma^{k_1+1}.
\end{aligned}
\tag{50}
$$

To sum up, based on (42) and (50), we conclude that (37) and (38) hold for $k = k_1 + 1$. Following by the principle of induction, we can conclude that (37) and (38) hold for any $k \in \mathbb{N}$. Based on the definition of $\Theta(k)$, it can guarantee the linear convergence rate of $\boldsymbol{x}(k)$, which ends the proof. $\qquad\square$

Remark 3.10. Noted that the exponentially decaying scaling function in (5) is used for achieving exponential convergence of the D-QAGT algorithm. In the convergence process, for avoiding the quantizer saturated, $l(k)$ could not converge too fast. Hence, the design convergence rate γ of the scaling function $l(k)$ matches that of the tracking errors $\Theta(k)$. Theorem 3.9 proves $\|\Theta(k)\| \leq C_0 \gamma^k$ for any $k \in \mathbb{N}$. Based on the definition of $\Theta(k)$, it should be noted that $\|\boldsymbol{x}(k) - \boldsymbol{x}^*\| \leq \|\Theta(k)\| \leq C_0 \gamma^k$, which ensures the convergence rate of the D-QAGT algorithm is at least γ. Furthermore, since $\gamma \in (\rho(H), 1)$, the linear convergence rate of the D-QAGT algorithm is no less than $\rho(H)$.

In addition, according to the selection of the quantization levels $2L + 1$ in (33), the required maximum data rate for ensuring the linear convergence of the D-QAGT is $R = \lceil \log_2(2 \max\{\frac{\zeta C_0 + 3C_1}{l_0 \rho(H)}, \frac{\sqrt{4c_1^2 + 4c_2^2}}{l_0}\} + 1) \rceil$.

4. SIMULATION

To evaluate the performance of the proposed algorithm, we verify its communication saving in the optimal placement problem. In an optimal placement problem, assume that there are $M = 5$ entities, which are located at $r_1 = [3; 5]$, $r_2 = [6; 9]$, $r_3 = [9; 8]$, $r_4 = [6; 2]$ and $r_5 = [9; 2]$. There are distributed across $N = 5$ free entities, each of that privately knows some of the fixed M entities. The aim is to determine the optimal position $\boldsymbol{x} = [x_i]_{i \in \{1,\ldots,5\}}$ of the free entities for minimizing the sum of all distance from the current position of each free entity to a corresponding fixed entitys position and the distances from each agent to the weighted center of all free entities. In this case, the cost function of each free entity is modeled as follows

$$f_i(x_i, \chi(\boldsymbol{x})) = \gamma_i \|x_i - r_i\|^2 + \|x_i - \chi(\boldsymbol{x})\|^2, \ i \in \{1, \ldots, 5\}, \tag{51}$$

where $\gamma_i = 100$ represents the weight constant and $\chi(\boldsymbol{x}) = \sqrt{\sum_{i=1}^5 x_i^2 / N}$. The communication graph among the free entities is randomly chosen to be strongly connected and the communication channel between free entities exists finite bandwidth constraints. The stepsize is chosen as $\alpha = 0.01$ in the D-QAGT algorithm.

The scaling functions is chosen as $l(k) = 10e^{-0.1k}$ and the quantization levels is chosen as $L = 10$. Figure 1 shows that the evolution of $x_i(k)$ associated with iterations $k = 100$. The results shows that all free entities in the plane converge to their optimal positions $\boldsymbol{x}^* = [x_i^*]_{i \in \{1,\ldots,5\}}$. Figure 2 shows that the evolution of $\chi_i(k)$. The results shows that the estimate $\chi_i(k)$ of each free entity converges to the optimal $\chi(\boldsymbol{x}^*) = \sqrt{\sum_{i=1}^5 (x_i^*)^2 / N}$.

In order to show the linear convergence, we take a performance index J as $J(t) = e^{\gamma k} \|f(\boldsymbol{x}) - f(\boldsymbol{x}^*)\|$ with $\gamma = 0.1$. The trajectory of $J(t)$ for the D-QAGT is shown in Figure 3(a). The results illustrates linear convergence of D-QAGT with a rate no less than the constant 0.1. To compare the convergence with the related D-AGT algorithm in [14], in which the communication channel is ideal, the evolutions of $f(\boldsymbol{x}) = \sum_{i=1}^5 f_i(\boldsymbol{x})$ for D-AGT and D-QAGT are shown in Figure 3(b). The results shows that the influence of quantization on the convergence speed is slight under the quantization levels $L = 10$.
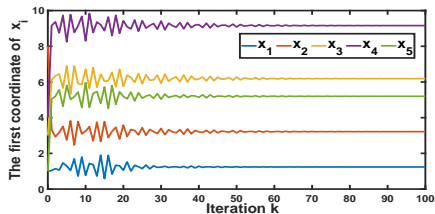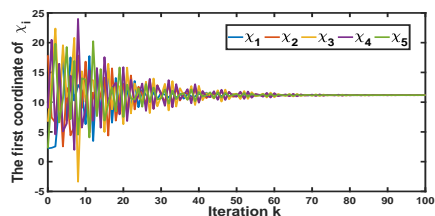
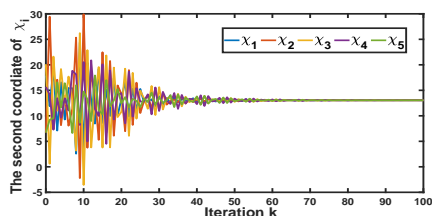(a) The first coordinate of $x_i(k)$.
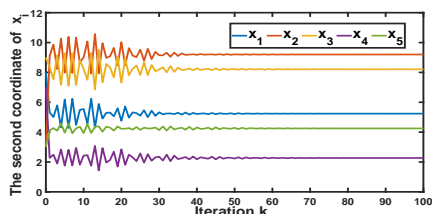
(b) The second coordinate of $x_i(k)$.

Fig. 1. The evolution of $x_i(k) \in \mathbb{R}^2$, $i \in \{1,\ldots,5\}$. The optimal positions of each agent are $x_1^* = [1.2376; 5.2393]$, $x_2^* = [3.2178; 9.1997]$, $x_3^* = [6.1881; 8.2096]$, $x_4^* = [9.1584; 2.2690]$ and $x_5^* = [5.1980; 4.2492]$.
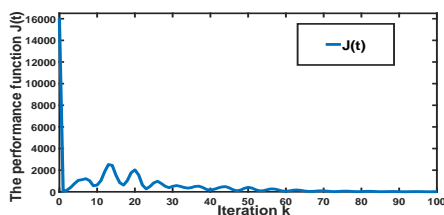

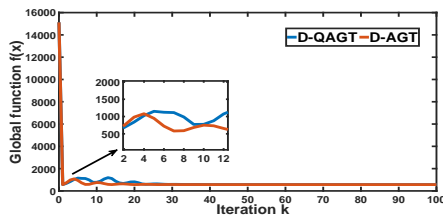(a) The first coordinate of $\chi_i(k)$ v.s. iteration.

(b) The second coordinate of $\chi_i(k)$ v.s. iteration.

Fig. 2. The evolution of the estimate $\chi_i(k) \in \mathbb{R}^2$, $i \in \{1,\ldots,5\}$. The optimal aggregative information is $\chi(\boldsymbol{x}^*) = [11.1803; 13.0437]$.

(a) The perfmance function $J(t)$ v.s. interation

(b) The global functions $J(t)$ v.s. iteration.

Fig. 3. (a) represents the trajectory of the performance function $J(t)$. (b) represents the convergence trajectories of $f(\boldsymbol{x})$ for D-QAGT (blue line) and D-AGT (red line), respectively.

5. CONCLUSION

This paper proposed a novel distributed quantized algorithm to solve the aggregative optimization problem, in which agents collectively minimize the sum of the local cost function that depends on the global aggregative variable. Particularly, we combined gradient descent and variables tracking methods to estimate the global aggregative variable, and also introduced the quantization technology to overcome communication bottleneck. By using dynamic encoder-decoder schemes, the influence of quantization errors is eliminated so that the linear convergence of the proposed Q-DGAT algorithms is achieved. Future works would include exploiting multiple information compression technologies for distributed aggregative optimization problems, such as sparsification and even-trigger technologies.

6. APPENDIX

6.1. Proof of Lemma 3.7

For explicit illustration, we prove the Lemma 3.7 by the following two steps.

Step 1. We first bound $\|\boldsymbol{x}(k+1) - \boldsymbol{x}^*\|$, $\|\boldsymbol{\chi}(k+1) - \mathbf{1}_N \otimes \bar{\chi}(k+1)\|$ and $\|\boldsymbol{y}(k+1) - \mathbf{1}_N \otimes \bar{y}(k+1)\|$, respectively.

- Invoking (17) into $\|\boldsymbol{x}(k+1) - \boldsymbol{x}^*\|$ yields that

$$
\begin{aligned}
&\|\boldsymbol{x}(k+1) - \boldsymbol{x}^*\| \\
&= \left\| \boldsymbol{x}(k) - \boldsymbol{x}^* - \alpha \left[\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}(k), \boldsymbol{\chi}(k)\right) + \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) \boldsymbol{y}(k) \right] \right\|, \\
&\le \left\| \boldsymbol{x}(k) - \boldsymbol{x}^* - \alpha \left[\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}(k), \mathbf{1}_N \otimes \bar{\chi}(k)\right) + \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) [\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_{\bar{\chi}} f_i\left(x_i(k), \bar{\chi}(k)\right)] \right] \right. \\
&\quad \left. + \alpha \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^*\right) \right\| + \alpha \left\| \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}(k), \boldsymbol{\chi}(k)\right) + \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) \left(\mathbf{1}_N \otimes \bar{y}(k)\right) \right. \\
&\quad \left. - \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}(k), \mathbf{1}_N \otimes \bar{\chi}(k)\right) - \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) \left[\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_{\bar{\chi}} f_i\left(x_i(k), \bar{\chi}(k)\right)\right] \right\| \\
&\quad + \alpha \left\| \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) \boldsymbol{y}(k) - \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) \left(\mathbf{1}_N \otimes \bar{y}(k)\right) \right\|. \quad (52)
\end{aligned}
$$

Following from Lemma 3 in [14], for a μ-strongly convex and l-smooth function $f : \mathbb{R}^n \to \mathbb{R}$, there is

$$
\|x - \alpha \nabla_x f(x) - (y - \alpha \nabla_y f(y))\| \le (1 - \mu\alpha)\|x - y\|, \ \forall x, y \in \mathbb{R}^n,
$$

with $\alpha \in (0, 1/l)$. Recalling (24) that $\bar{\chi}(k) = \chi(\boldsymbol{x}(k))$ and using $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(k)) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x}(k), \mathbf{1}_N \otimes \bar{\chi}(k)) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}(k)) \left[\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_{\bar{\chi}} f_i(x_i(k), \bar{\chi}(k))\right]$, we bound the first term of the right of (52) as follows,

$$
\begin{aligned}
&\left\| \boldsymbol{x}(k) - \alpha \left[\nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}(k), \mathbf{1}_N \otimes \bar{\chi}(k)\right) + \nabla_{\boldsymbol{x}} g\left(\boldsymbol{x}(k)\right) \left[\mathbf{1}_N \otimes \frac{1}{N} \sum_{i=1}^{N} \nabla_{\bar{\chi}} f_i\left(x_i(k), \bar{\chi}(k)\right)\right] \right] \right. \\
&\quad \left. - \left(\boldsymbol{x}^* - \alpha \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}^*\right)\right) \right\| \le (1 - \mu\alpha)\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|. \quad (53)
\end{aligned}
$$

Similarly, using (25) and Assumption 2, the second term of the right of (52) is bounded as

$$\alpha\big\|(\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}(k),\boldsymbol{\chi}(k)\right)+\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)(\mathbf{1}_N\otimes\bar{y}(k)))-(\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}(k),\mathbf{1}_N\otimes\bar{\chi}(k)\right)$$

$$+\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)[\mathbf{1}_N\otimes\frac{1}{N}\sum_{i=1}^{N}\nabla_{\bar{\chi}}f_i\left(x_i(k),\bar{\chi}(k)\right)])\big\|$$

$$\leq \alpha l_1\|\boldsymbol{\chi}(k)-\mathbf{1}_N\otimes\bar{\chi}(k)\|. \tag{54}$$

Combining with (53) and (54), in light of $\|\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)\|\leq l_3$, we conclude that

$$\|\boldsymbol{x}(k+1)-\boldsymbol{x}^*\|\leq(1-\mu\alpha)\|\boldsymbol{x}(k)-\boldsymbol{x}^*\|+\alpha l_1\|\boldsymbol{\chi}(k)-\mathbf{1}_N\otimes\bar{\chi}(k)\|$$
$$+\alpha l_3\|\boldsymbol{y}(k)-\mathbf{1}_N\otimes\bar{y}(k)\|. \tag{55}$$

- Invoking (18) into $\boldsymbol{\chi}(k+1)-\mathbf{1}_N\otimes\bar{\chi}(k+1)$ yields that

$$\big\|\boldsymbol{\chi}(k+1)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{\chi}(k+1)\big\|$$

$$=\big\|\boldsymbol{L}\boldsymbol{e}_{\boldsymbol{\chi}}(k)+g(\boldsymbol{x}(k+1))-g(\boldsymbol{x}(k))+(A\otimes I_r)\boldsymbol{\chi}(k)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)$$

$$\big[\boldsymbol{L}\boldsymbol{e}_{\boldsymbol{\chi}}(k)+g(\boldsymbol{x}(k+1))-g(\boldsymbol{x}(k))+(A\otimes I_r)\boldsymbol{\chi}(k)\big]\big\|,$$

$$\leq\big\|(A\otimes I_r)\boldsymbol{\chi}(k)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)(A\otimes I_r)\boldsymbol{\chi}(k)\big\|+\big\|I_{Nr}-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\big\|$$

$$\big\|\boldsymbol{L}\boldsymbol{e}_{\boldsymbol{\chi}}(k)\big\|+\big\|I_{Nr}-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r))\big\|\big\|g(\boldsymbol{x}(k+1))-g(\boldsymbol{x}(k))\big\|, \tag{56}$$

where the fact that $I_{Nr}-\boldsymbol{L}=A\otimes I_r$ and $\mathbf{1}_N\otimes\bar{\chi}(k)=(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{\chi}(k)$ are used. Based on Lemma 2.1 and $\|I_{Nr}-\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r\|=1$, it follows that

$$\big\|\boldsymbol{\chi}(k+1)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{\chi}(k+1)\big\|$$

$$\leq\kappa\big\|\boldsymbol{\chi}(k)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{\chi}(k)\big\|+\big\|I_N-A\big\|\big\|\boldsymbol{e}_{\boldsymbol{\chi}}(k)\big\|+\big\|g(\boldsymbol{x}(k+1))-g(\boldsymbol{x}(k))\big\|,$$

$$\leq\kappa\big\|\boldsymbol{\chi}(k)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{\chi}(k)\big\|+2\|\boldsymbol{e}_{\boldsymbol{\chi}}(k)\|+l_3\|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|, \tag{57}$$

where $\kappa=\|A-1/N\mathbf{1}_N\mathbf{1}_N^T\|<1$. We now compute the upper bound of $\|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|$. Invoking (17) into $\|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|$ yields that

$$\|\boldsymbol{x}(k+1)-\boldsymbol{x}(k)\|=\alpha\left\|\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}(k),\boldsymbol{\chi}(k)\right)+\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)\boldsymbol{y}(k)\right\|$$

$$\leq\alpha\big\|\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}(k),\boldsymbol{\chi}(k)\right)+\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{y}(k)-\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}^*,\mathbf{1}_N\otimes\chi^*\right)$$

$$-\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}^*\right)\big[\mathbf{1}_N\otimes\frac{1}{N}\sum_{i=1}^{N}\nabla_{\chi_i}f_i\left(x_i^*,\chi_i^*\right)\big]\big\|+\alpha\big\|\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)\left(\boldsymbol{y}(k)-(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\otimes I_r)\boldsymbol{y}(k)\right)\big\|,$$

$$=\alpha\big\|\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}(k),\boldsymbol{\chi}(k)\right)+\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)(\mathbf{1}_N\otimes\bar{y}(k))-\nabla_{\boldsymbol{x}}f\left(\boldsymbol{x}^*,\mathbf{1}_N\otimes\chi^*\right)-\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}^*\right)$$

$$\big[\mathbf{1}_N\otimes\frac{1}{N}\sum_{i=1}^{N}\nabla_{\chi_i}f_i\left(x_i^*,\chi_i^*\right)\big]\big\|+\alpha\big\|\nabla_{\boldsymbol{x}}g\left(\boldsymbol{x}(k)\right)\left(\boldsymbol{y}(k)-\mathbf{1}_N\otimes\bar{y}(k)\right)\big\|.$$

It follows from Assumption 2 that

$$\|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\|$$
$$\leq \alpha l_1 \left(\|\boldsymbol{x}(k) - \boldsymbol{x}^*\| + \|\boldsymbol{\chi}(k) - \mathbf{1}_N \otimes \chi^*\|\right) + \alpha l_3 \|\boldsymbol{y}(k) - \mathbf{1}_N \otimes \bar{y}(k)\|,$$
$$\leq \alpha l_1 \left(\|\boldsymbol{x}(k) - \boldsymbol{x}^*\| + \left\|\boldsymbol{\chi}(k) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k)\right\|\right) + \alpha l_3 \|\boldsymbol{y}(k) - \mathbf{1}_N \otimes \bar{y}(k)\|$$
$$+ \alpha l_1 \left\|(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k) - \mathbf{1}_N \otimes \chi^*\right\|. \tag{58}$$

Using again $(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k) = \mathbf{1}_N \otimes \bar{\chi}(k)$, we compute the upper bound of the last term in (58) as follows,

$$\left\|(\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k) - \mathbf{1}_N \otimes \chi^*\right\|^2 = \left\|\mathbf{1}_N \otimes (\bar{\chi}(k) - \chi^*)\right\|^2$$
$$= N\left\|\frac{1}{N}\sum_{i=1}^{N}(g_i(x_i(k)) - g_i(x_i^*))\right\|^2 \leq \frac{1}{N}\left(\sum_{i=1}^{N}\|g_i(x_i(k)) - g_i(x_i^*)\|\right)^2,$$
$$\leq \frac{1}{N}\left(\sum_{i=1}^{N}l_3\|x_i(k) - x_i^*\|\right)^2 \leq l_3^2 \sum_{i=1}^{N}\|x_i(k) - x_i^*\|^2 = l_3^2 \|\boldsymbol{x}(k) - \boldsymbol{x}^*\|^2. \tag{59}$$

Substituting (59) into (58), we conclude that

$$\|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\| \leq \alpha l_1(1+l_3)\|\boldsymbol{x}(k) - \boldsymbol{x}^*\| + \alpha l_1\left\|\boldsymbol{\chi}(k) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k)\right\|$$
$$+ \alpha l_3\left\|\boldsymbol{y}(k) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{y}(k)\right\|. \tag{60}$$

Further, substituting (60) into (57), we conclude that

$$\left\|\boldsymbol{\chi}(k+1) - \mathbf{1}_N \otimes \bar{\chi}(k+1)\right\|$$
$$\leq (\kappa + \alpha l_1 l_3)\left\|\boldsymbol{\chi}(k) - \mathbf{1}_N \otimes \bar{\chi}(k)\right\| + 2\|\boldsymbol{e_\chi}(k)\| + \alpha l_1 l_3(1+l_3)\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|$$
$$+ \alpha l_3^2\|\boldsymbol{y}(k) - \mathbf{1}_N \otimes \bar{y}(k)\|. \tag{61}$$

• Invoking (19) into $\boldsymbol{y}(k+1) - \mathbf{1}_N \otimes \bar{y}(k+1)$ yields that

$$\|\boldsymbol{y}(k+1) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{y}(k+1)\| \leq \kappa\|\boldsymbol{y}(k) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{y}(k)\|$$
$$+ \|\nabla_{\boldsymbol{\chi}}f(\boldsymbol{x}(k+1), \boldsymbol{\chi}(k+1)) - \nabla_{\boldsymbol{\chi}}f(\boldsymbol{x}(k), \boldsymbol{\chi}(k))\| + 2\|\boldsymbol{e_y}(k)\|. \tag{62}$$

Using (12), via Assumption 2, we obtain that

$$\|\nabla_{\boldsymbol{\chi}}f(\boldsymbol{x}(k+1), \boldsymbol{\chi}(k+1)) - \nabla_{\boldsymbol{\chi}}f(\boldsymbol{x}(k), \boldsymbol{\chi}(k))\|$$
$$\leq l_2\left(\|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\| + \|\boldsymbol{\chi}(k+1) - \boldsymbol{\chi}(k)\|\right),$$
$$\leq l_2\left(\|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\| + \|\boldsymbol{Le_\chi}(k) + g(\boldsymbol{x}(k+1)) - g(\boldsymbol{x}(k)) + (A \otimes I_r)\boldsymbol{\chi}(k) - \boldsymbol{\chi}(k)\|\right),$$
$$\leq l_2\left(\|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\| + \|\boldsymbol{Le_\chi}(k) + g(\boldsymbol{x}(k+1)) - g(\boldsymbol{x}(k))\right. \tag{63}$$
$$\left. + (A \otimes I_r - I_{Nr})(\boldsymbol{\chi}(k) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k))\|\right),$$
$$\leq l_2(1+l_3)\|\boldsymbol{x}(k+1) - \boldsymbol{x}(k)\| + 2l_2\|\boldsymbol{\chi}(k) - (\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T \otimes I_r)\boldsymbol{\chi}(k)\| + 2l_2\|\boldsymbol{e_\chi}(k)\|.$$

Substituting (63) into (62), one has that

$$\|\boldsymbol{y}(k+1)-\mathbf{1}_N \otimes \bar{y}(k+1)\|$$
$$\leq (\kappa+\alpha l_2 l_3(1+l_3))\|\boldsymbol{y}(k)-\mathbf{1}_N \otimes \bar{y}(k)\|+\alpha l_1 l_2(1+l_3)^2\|\boldsymbol{x}(k) - \boldsymbol{x}^*\|+(\alpha l_1 l_2$$
$$(1+l_3)+2l_2)\|\boldsymbol{\chi}(k)-\mathbf{1}_N \otimes \bar{\chi}(k)\|+2l_2\|\boldsymbol{e}_{\boldsymbol{\chi}}(k)\|+2\|\boldsymbol{e}_{\boldsymbol{y}}(k)\|. \qquad (64)$$

Step 2. To sum up, we have bounded the $\|\boldsymbol{x}(k+1) - \boldsymbol{x}^*\|$, $\|\boldsymbol{\chi}(k+1) - \mathbf{1}_N \otimes \bar{\chi}(k+1)\|$ and $\|\boldsymbol{y}(k+1) - \mathbf{1}_N \otimes \bar{y}(k+1)\|$. Based on the definition of $\Theta(k)$, by substituting the upper bound of the norm of each coordinate, we conclude that

$$\Theta(k+1) \leq H(\alpha)\Theta(k) + E(k). \qquad (65)$$

We now prove $\rho(H) < 1$. Let $\lambda(\alpha)$ be the eigenvalue of $H(\alpha)$. It is obvious that 1 is a simple eigenvalue of $H(0)$ and $\nu = [1;0;0]$ is 1's corresponding left and right eigenvectors. According to [7] and Lemma 5 in [14], there is $\frac{d\lambda(\alpha)}{d\alpha}|_{\alpha=0} = -\mu < 0$, which means that the spectral radius of $H(\alpha)$ is less than 1 for sufficiently small positive α. Furthermore, the strongly connected graph makes the related $H(\alpha)$ is irreducible [21]. Combining with Lemmas 1-2 in [14] can ensure that 1 will be a simple eigenvalue of $H(\alpha)$ as α increases from zero to some positive value. By computing $\det(I - H(\alpha)) = 0$, we obtain that

$$\alpha = \frac{\mu(1-\kappa)^2}{l_3(\mu + l_1 + l_2 l_3)\big((1-\kappa)(l_1 + l_2 + l_2 l_3) + 2l_2 l_3\big)}.$$

Hence, when $\alpha \in \big(0, \frac{\mu(1-\kappa)^2}{l_3(\mu+l_1+l_2 l_3)((1-\kappa)(l_1+l_2+l_2 l_3)+2l_2 l_3)}\big)$, all eigenvalues of $H(\alpha)$ have absolute values less than 1.

6.2. Proof of Lemma 3.8

By Shur theorem, there exists an unitary matrix $U = [u_1, u_2, u_3] \in \mathbb{R}^{3\times 3}$ such that $U^H H(\alpha)U$ is an upper triangular matrix as the following form

$$U^H H(\alpha)U = \begin{bmatrix} \lambda_{H1} & T_{12} & T_{13} \\ 0 & \lambda_{H2} & T_{23} \\ 0 & 0 & \lambda_{H3} \end{bmatrix},$$

where $T_{12} = u_1^H H(\alpha)u_2$, $T_{13} = u_1^H H(\alpha)u_3$ and $T_{23} = u_2^H H(\alpha)u_3$. Further define $D = \text{diag}(1, \eta, \eta^2)$ with $\eta = \epsilon/2\|H(\alpha)\|$, then it follows that

$$D^{-1}U^H H(\alpha)UD = \begin{bmatrix} \lambda_{H1} & T_{12}\eta & T_{13}\eta^2 \\ 0 & \lambda_{H2} & T_{23}\eta \\ 0 & 0 & \lambda_{H3} \end{bmatrix}.$$

It should be noted that $\max\{\|T_{12}\|, \|T_{13}\|, \|T_{23}\|\} \leq \|H(\alpha)\|$ such that

$$\|D^{-1}U^H H(\alpha)UD\|_\infty \leq \rho(H(\alpha)) + \epsilon,$$
$$\|D^{-1}U^H H^k(\alpha)UD\|_\infty \leq \|(D^{-1}U^H H(\alpha)UD)^k\|_\infty \leq (\rho(H(\alpha)) + \epsilon)^k. \qquad (66)$$

We can calculate $\|H^k(\alpha)\|_\infty$ as follows

$$
\begin{aligned}
\|H^k(\alpha)\|_\infty &= \|UDD^{-1}U^H H^k(\alpha) UDD^{-1}U^H\|_\infty, \\
&\leq \|U\|_\infty \|U^H\|_\infty \|D^{-1}\|_\infty \|D\|_\infty (\rho(H(\alpha)) + \epsilon)^k, \\
&\leq 3\max\Big\{\frac{4\|H(\alpha)\|^2}{\epsilon^2}, \frac{\epsilon^2}{4\|H(\alpha)\|^2}\Big\}(\rho(H(\alpha)) + \epsilon)^k,
\end{aligned}
\tag{67}
$$

which further yields that

$$
\|H^k(\alpha)\| \leq \sqrt{3}\|H^k(\alpha)\|_\infty = 3\sqrt{3}\max\Big\{\frac{4\|H(\alpha)\|^2}{\epsilon^2}, \frac{\epsilon^2}{4\|H(\alpha)\|^2}\Big\}(\rho(H(\alpha)) + \epsilon)^k.
\tag{68}
$$

ACKNOWLEDGEMENT

R E F E R E N C E S

[1] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo: Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks. IEEE Signal Process. Mag. *31* (2014), 45–55. DOI:10.1109/MSP.2014.2334709

[2] X. Cao and K. J. R. Liu: Distributed Newton's method for network cost minimization. IEEE Trans. Automat. Control *66* (2021), 1278–1285. DOI:10.1109/TAC.2020.2989266

[3] S. Cheng and S. Liang: Distributed optimization for multi-agent system over unbalanced graphs with linear convergence rate. Kybernetika *56* (2020), 559–577. DOI:10.14736/kyb-2020-3-0559

[4] J. Chen and A. Sayed: Diffusion adaptation strategies for distributed optimization and learning over networks. IEEE Trans. Signal Process. *60* (2012), 4289–4305. DOI:10.1109/TSP.2012.2198470

[5] Z. Deng and S. Liang: Distributed algorithms for aggregative games of multiple heterogeneous Euler–Lagrange systems. Automatica *99* (2019), 246–252. DOI:10.1016/j.automatica.2018.10.041

[6] C. De Persis and S. Grammatico: Continuous-time integral dynamics for a class of aggregative games with coupling constraints. IEEE Trans. Automat. Control *65* (2020), 2171–2176. DOI:10.1109/TAC.2019.2939639

[7] R. A. Horn and C. R Johnson: Matrix Analysis. Cambridge University Press, New York 2012.

[8] D. Jakovetic, J. M. F. Moura, and J. Xavier: Linear convergence rate of a class of distributed augmented Lagrangian algorithms. IEEE Trans. Automat. Control *60* (2015), 922–936. DOI:10.1109/TAC.2014.2363299

[9] Y. Kajiyama, N. Hayashi, and S. Takai: Linear convergence of consensus-based quantized optimization for smooth and strongly convex cost functions. IEEE Trans. Automat. Control *66* (2020), 1254–1261. DOI:10.1109/TAC.2020.2989281

[10] G. Lan, S. Lee, and Y. Zhou: Communication-efficient algorithms for decentralized and stochastic optimization. Math. Program. *180* (2020), 237–284. DOI:10.1007/s10107-018-1355-4

[11] P. Li and J. Hu: A solution strategy for distributed uncertain economic dispatch problems via scenario theory. Control Theory Technol. *19* (2021), 499–506. DOI:10.1007/s11768-021-00068-6

[12] P. Li, J. Hu, L. Qiu, Y. Zhao, and B. Ghosh: A distributed economic dispatch strategy for power-water networks. IEEE Trans. Control Netw. Syst., Early Access (2021). DOI:10.1109/tcns.2021.3104103

[13] X. Li, L. Xie, and Y. Hong: Distributed continuous-time algorithm for a general non-smooth monotropic optimization problem. Int. J. Robust Nonlinear Control *29* (2019), 3252–3266. DOI:10.1002/rnc.4547

[14] X. Li, L. Xie, and Y. Hong: Distributed aggregative optimization over multi-agent networks. IEEE Trans. Automat. Control. Early Access (2021). DOI:10.1109/TAC.2021.3095456

[15] J. Ma, X. Yu, L. Liu, and G. Feng: Global cooperative output regulation of linear multi-agent systems with limited bandwidth. IEEE Trans. Control Netw. Syst., Early Access (2021). DOI:10.1109/TCNS.2021.3128500

[16] S. Magnussson, H. Shokri-Ghadikolaei, and N. Li: On maintaining linear convergence of distributed learning and optimization under limited communication. IEEE Trans. Signal Process. *68* (2020), 6101–6116. DOI:10.1109/TSP.2020.3031073

[17] E. J. Msechu and G. B. Giannakis: Sensor-centric data reduction for estimation with WSNs via censoring and quantization. IEEE Trans. Signal Process. *60* (2011), 400–414. DOI:10.1109/TSP.2011.2171686

[18] A. Nedic: Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. IEEE Signal Process. Mag. *73* (2020), 92–101.

[19] A. Nedic and A. Ozdaglar: Distributed subgradient methods for multi-agent optimization. IEEE Trans. Autom. Control *54* (2009), 48–61. DOI:10.1109/TAC.2008.2009515

[20] Y. Pu, M. N. Zeilinger, and C. N. Jones: Quantization design for distributed optimization. IEEE Trans. Automat. Control *62* (2016), 2107–2120. DOI:10.1109/TAC.2016.2600597

[21] W. Ren and R. W. Beard: Distributed consensus in multi-vehicle cooperative control. Springer, London 2008.

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

[23] E. Tardos and V. V. Vazirani: Basic solution concepts and computational issues. Algorithmic Game Theory (2007), 3–28. DOI:10.1017/cbo9780511800481.003

[24] X. Wang, Z. Deng, S. Ma, and D. Xian: Event-triggered design for multi-agent optimal consensus of Euler–Lagrangian systems. Kybernetika *53* (2017), 179–194. DOI:10.14736/kyb-2017-1-0179

[25] Y. Wang, P. Lin, and H. Qin: Distributed classification learning based on nonlinear vector support machines for switching networks. Kybernetika *53* (2017), 595–611. DOI:10.14736/kyb-2017-4-0595

[26] J. Xu, S. Zhu, Y. C. Soh, and L. Xie: A Bregman splitting scheme for distributed optimization over networks. IEEE Trans. Automat. Control *63* (2018), 3809–3824. DOI:10.1109/TAC.2018.2805260

[27] P. Yi and Y. Hong: Quantized subgradient algorithm and data-rate analysis for distributed optimization. IEEE Trans. Control Netw. Syst. *1* (2014), 380–392. DOI:10.1109/TCNS.2014.2357513

[28] P. Yi, Y. Hong, and F. Liu: Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and its application to economic dispatch of power systems. Automatica *74* (2016), 259–269. DOI:10.1016/j.automatica.2016.08.007

[29] D. Yuan, Y. Hong, D. Ho, and G. Jiang: Optimal distributed stochastic mirror descent for strongly convex optimization. Automatica *90* (2018), 196–203. DOI:10.1016/j.automatica.2017.12.053

[30] X. Zhang, J. Liu, Z. Zhu, and E. S. Bentley: Compressed distributed gradient descent: Communication-efficient consensus over networks. In: Proc. IEEE Conf. Comput. Commun. 2019, 2431–2439. DOI:10.1109/infocom.2019.8737489

[31] S. Zhu, M. Hong, and B. Chen: Quantized consensus ADMM for multi-agent distributed optimization. In: Proc. IEEE Int. Conf. Acoust., Speech Signal Process 2016, pp. 4134–4138. DOI:10.1109/icassp.2016.7472455

Ziqin Chen, The Department of Control Science and Engineering, Tongji University, Shanghai, 201210. P. R. China.
    e-mail: cxq0915@tongji.edu.cn

Shu Liang, The Department of Control Science and Engineering, Tongji University, Shanghai, 201210. P. R. China.
    e-mail: sliang@tongji.edu.cn