Poincare Journal of Analysis & Applications
Vol. 10, No. 3 (2023), Special issue, 137-163

DOI: 10.46753/pjaa.2023.v010i03.009

A CODE-BASED DISTRIBUTED GRADIENT DESCENT SCHEME FOR DECENTRALIZED CONVEX OPTIMIZATION

ELIE ATALLAH, NAZANIN RAHNAVARD, AND QIYU SUN†

Date of Receiving : 11. 03. 2023
Date of Revision : 01. 02. 2024
Date of Acceptance : 09. 02. 2024

Abstract. In this paper, we consider a large network containing many regions such that each region is equipped with a worker with some data processing and communication capability. For such a network, some workers may become stragglers due to the failure or heavy delay on computing or communicating. To resolve the above straggling problem, a coded scheme that introduces certain redundancy for every worker was recently proposed, and a gradient coding paradigm was developed to solve convex optimization problems when the network has a centralized fusion center. In this paper, we propose an iterative distributed algorithm, referred as Code-Based Distributed Gradient Descent algorithm (CoDGraD), to solve convex optimization problems over distributed networks. In each iteration of the proposed algorithm, an active worker shares the coded local gradient and approximated solution of the convex optimization problem with non-straggling workers at the adjacent regions only. In this paper, we provide the consensus and convergence analysis for the CoDGraD algorithm and we demonstrate its performance via numerical simulations.

1. Introduction

Convex optimization on a network of large size has played a significant role for solving various problems, such as big-data processing in machine learning, distributed parameter estimation in wireless sensor networks, distributed sampling and signal reconstruction, distributed design of filter banks, distributed spectrum sensing in cognitive radio networks, source localization in cellular networks [7, 8, 10, 12, 18, 19, 23, 24]. The objective functions f in such optimization problems,

$$f(\mathbf{x}) = \sum_{l=1}^{m} f_l(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^N, \tag{1.1}$$

2010 *Mathematics Subject Classification.* 65K99, 65Y05.
Key words and phrases. Convex optimization, Distributed gradient descent algorithm, Networks with stragglers.
Communicated by: Michael Unser
† *Corresponding author*

are often the summation of some local objective functions $f_l, 1 \leq l \leq m$, related to a partition of the network. In this paper, we consider the scenario that each region of the partition is equipped with a worker that has some data processing and communication ability, while the whole network has a fusion center with limited computing capacity or it does not have a fusion center at all.

The optimization problem

$$\bar{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \sum_{l=1}^{m} f_l(\mathbf{x}) \tag{1.2}$$

associated with the objective function $f = \sum_{l=1}^{m} f_l$ has been well studied, see [2, 3, 6, 10, 14, 23, 24, 25, 26, 30, 29, 32] and references therein for various algorithms implementable in a strong fusion center or in local workers distributed over the network. Denote the gradient of g on $\mathbb{R}^N$ by ∇g. For a network equipped with one data processing unit only, a conventional approach to the optimization problem (1.2) is the *gradient descent algorithm*,

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \frac{\alpha_k}{m} \sum_{l=1}^{m} \nabla f_l(\mathbf{x}(k)),\ k \geq 0, \tag{1.3}$$

where $\{\alpha_k\}_{k=0}^{\infty}$ is a positive sequence chosen appropriately [4, 11, 16, 20, 22, 28, 31, 33, 36]. Our illustrative examples of step sizes $\alpha_k, k \geq 0$, are

$$\alpha_k = (k+a)^{-\theta},\ k \geq 0, \tag{1.4}$$

for some $\theta \in (1/2, 1]$ and $a \geq 1$.

Several distributed versions of the gradient descent algorithm (1.3) have been proposed, including the Adapt-Then-Combine algorithm (ATC) and the Combine-Then-Adapt algorithm (CTA) [6, 30], where implementation of the worker at agent l is given by

$$\begin{cases} \mathbf{y}_l(k) = \mathbf{x}_l(k) - \alpha_k \nabla f_l(\mathbf{x}_l(k)) \\ \mathbf{x}_l(k+1) = \sum_{j\in\mathcal{N}_l} w_{lj}(k)\mathbf{y}_j(k) \end{cases} \tag{1.5}$$

and

$$\begin{cases} \mathbf{y}_l(k) = \sum_{j\in\mathcal{N}_l} w_{lj}(k)\mathbf{x}_j(k) \\ \mathbf{x}_l(k+1) = \mathbf{y}_l(k) - \alpha_k \nabla f_l(\mathbf{y}_l(k)) \end{cases} \tag{1.6}$$

respectively, where $\mathcal{N}_l$ contains all adjacent agents of the agent l for data sharing, and $(w_{lj})_{1\leq l,j\leq m}$ is a consensus matrix. The above ATC and CTA algorithms may reach consensus over all nodes to the optimal solution $\bar{\mathbf{x}}$ in (1.2). They are essentially the same as the gradient descent algorithm (1.3) if the graph to describe topological structure of the network is complete and the consensus weights $w_{lj} = 1/m, 1 \leq l, j \leq m$ (in this case, $\mathcal{N}_l = \{1, \ldots, m\}$ and $\mathbf{x}_l(k)$ in (1.5) and $\mathbf{y}_l(k)$ in (1.6) for different agents l are the same for $k \geq 1$). However, comparing with the gradient descent algorithm (1.3), in the implementation of the ATC and CTA algorithms, we circumvent the expansive evaluation of gradient ∇f of the global objective function by evaluating gradients $\nabla f_l, 1 \leq l \leq m$, of local objective functions at each worker node and then communicating local gradients to the neighbors with nonzero consensus weights.

In applications such as distributed learning and optimization over the cloud, some workers in the network may become inactive due to the failure or heavy delay on computing or communicating [9, 17, 21]. To resolve the above problem, uncoded and

coded local gradients have been proposed in [15, 27, 34] to recover the full gradient from local gradients on active nodes $\Delta \subset \{1, \ldots, m\}$. Without loss of generality, we assume that $\Delta \subset \{1, \ldots, n\}$, where $n \leq m$ is the number of active nodes. In this paper, we follow the coded scheme in [34] and consider the paradigm that for every active node i, the coded scheme for non-stragglers is given by

$$g_i(\mathbf{x}) = \sum_{l=1}^{m} b(i,l) f_l(\mathbf{x}),\ 1 \leq i \leq n, \tag{1.7}$$

and the global objective function can be recovered linearly from coded objective functions g_j on non-stragglers relative to node i, i.e.,

$$f(\mathbf{x}) = \sum_{j=1}^{n} a(i,j) g_j(\mathbf{x}). \tag{1.8}$$

Denote the decoding and coding matrices of the above paradigm by

$$\mathbf{A} = (a(i,j))_{1 \leq i,j \leq n} \ \text{ and } \ \mathbf{B} = (b(i,l))_{1 \leq i \leq n, 1 \leq l \leq m}.$$

Then the requirement in the paradigm to recover the global objective function at active nodes can be reformulated in the following matrix form,

$$\mathbf{AB} = \mathbf{1}_{n \times m}, \tag{1.9}$$

where $\mathbf{1}_{n \times m}$ is the $n \times m$ matrix with all entries taking value 1, since

$$\sum_{j=1}^{n} a(i,j) g_j(\mathbf{x}) = \sum_{l=1}^{m} \sum_{j=1}^{n} a(i,j) b(j,l) f_l(\mathbf{x}) = \sum_{l=1}^{m} f_l(\mathbf{x}) = f(\mathbf{x}).$$

We note that the code matrix $\mathbf{B}$ must be constructed such that the m-dimensional vector $\mathbf{1}_m$ with all one entries is in the column space of the transpose $\mathbf{B}^T$ of the code matrix $\mathbf{B}$. This can be shown to be a necessary and sufficient condition for the existence of a decoding matrix $\mathbf{A}$ that satisfies (1.9). We also see from (1.9) that the difference of any two rows of the decoding matrix $\mathbf{A}$ belongs to the left null space of the code matrix $\mathbf{B}$, and therefore the decoding matrix $\mathbf{A}$ could be selected to have more zero entries when the code matrix $\mathbf{B}$ has lower rank. We can see how sparse the decoding matrix $\mathbf{A}$ is by looking at two extreme cases. In the first case, every active node gets a copy of the global objective functions (and hence $\mathbf{B}$ has rank one) and the identity matrix can be used as the decoding matrix (so each row has only one nonzero element). In the second case, which is used in the decentralized descent algorithm, all nodes are active and each node gets only the local objective function ($\mathbf{B}$ is the identity matrix in this case), and then all entries of the decoding matrix take value one and are nonzero. The sparsity of the decode matrix $\mathbf{A}$ will reduce the amount of data exchange among the active nodes in the proposed iterative code-based distributed gradient descent algorithm (CoDGraD). However, to meet the requirements in Assumption 2.1 for the convergence of the iterative CoDGraD, the decode matrix $\mathbf{A}$ should not be too sparse and it should be designed such that the normalized decoding matrix $|\tilde{\mathbf{A}}|$ given in (2.14) corresponds to the weighted adjacent matrix of a strongly connected digraph, see Structures 2.4 and 2.6.

The coding and decoding scheme (1.7) and (1.8) has been used in [34] to solve the global optimization problem (1.2), where the worker at each region evaluates the coded

local gradients and sends them to the worker at the master node (the fusion center); then the worker at the master node aggregates a weighted sum of coded local gradients to form the gradient of the global objective function, and applies a centralized gradient descent approach similar to (1.3) to update the approximation to the optimal solution $\bar{\mathbf{x}}$ in (1.2); and finally the worker at the master node sends the updated approximation to all local workers on the network for the next iteration. In this paper, based on the coding and decoding scheme (1.7) and (1.8), we propose a *distributed algorithm* to solve the global convex optimization problem (1.2).

1.1. Objectives and contributions. Since the focus of distributed optimization is overwhelmingly occupied with first-order methods, it is worth trying to enhance such methods rather than employing methods of another type. Our initial aim of this paper is to improve the performance of distributed gradient descent algorithm through utilizing coding. In particular, we focus on the leveraging of coding on the performance of the distributed optimization algorithm and how the convergence rate of these algorithms can be better enhanced through using an appropriate coding scheme based on the network topology. And how resilience to stragglers and information privacy is also established.

While distributed optimization is implementable through distributing the local functions among the nodes to sum up to the global optimization function as in (1.1), we follow here an alternative route. We decompose the global function among the nodes in a coded manner and try to implement an algorithm which allows optimization under such scheme (1.8). To this end, we adapt the stochastic form of the decoding matrix $\mathbf{A}$ in our proposed algorithm, carry the negativity in some of its coefficients to the gradient operation, and then utilize a gradient descent and a gradient ascent step, see (2.4) and (2.5).

In this paper, we do not impose any algebraic structure of the coding matrix $\mathbf{B}$ and the decoding matrix $\mathbf{A}$, and thus our approach applies for a broader class of coding/decoding matrices as long as the allowed number of stragglers is fulfilled, except that the normalized decoding matrix $|\tilde{\mathbf{A}}|$ in (2.14) is assumed to have simple eigenvalue one, see Assumption 2.1 and Proposition 2.3. Our work has the assumptions of strongly convex global function f, Lipschitz continuous (coded) gradients ∇g_i, and uniformly bounded (coded) gradients ∇g_i. We believe that if the coding/decoding matrices have additional structure properties [35], our algorithm could converge under weak requirements on the objective functions and coded gradients.

In this paper, we utilize exact gradient coding for fusion centralized networks and investigate the implications of such coding used in a multi-agent setting. This serves well to our initial aim of showing how enhancement to the convergence rate can be accomplished, see the consensus and convergence conclusions in Sections 3 and 4 of our proposed algorithm. The paradigm of approximate gradient coding is discussed in [13]. It could be an interesting problem to extend our convergence conclusions in the above setting.

In this paper, we work on distributed optimization problem for multi-agent systems on fixed static network topology. Thus, the active nodes are the non-straggler nodes which are fixed throughout the proposed distributed algorithm and where coded local functions are used according to the coding scheme (1.8) instead of the brute decomposition as in

(1.1). We postpone the consideration of time-varying straggler networks to a later endeavor.

The implementation of distributed algorithms on static/time-varying networks with stragglers is of importance. We wish that this work may serve as a starting point for a full-fledged investigation to distributed algorithms on static/time-varying networks with stragglers, with applications to the engineering field, such as federated decentralized learning and distributed machine learning.

1.2. Organization and notations. The remainder of the paper is organized as follows. In Section 2, we formulate our code-based distributed gradient descent algorithm (CoDGraD) to solve the optimization problem (1.2). In Sections 3 and 4, we consider the consensus and convergence properties of the proposed CoDGraD algorithm. Afterwards, we demonstrate the performance of CoDGraD through simulations in Section 5. We conclude the paper in Section 6 and include proofs of some conclusions in the Appendix.

In what follows, we use bold capital letters, bold lower case letters and lower case letters for matrices, vectors and scalar variables, respectively. Denote the positive part of a real number t by $t_+ = \max(t, 0)$, the matrix of dimension $n \times m$ with all entries taking value one by $\mathbf{1}_{n\times m}$, the unit matrix of size $n \times n$ by $\mathbf{I}_n$, the column vector of size n with all entries taking value 1 by $\mathbf{1}_n$, and the zero matrix of size $n \times m$ by $\mathbf{0}_{n\times m}$ respectively. Denote the transpose of an matrix $\mathbf{A}$ by $\mathbf{A}^T$, and the transpose and standard ℓ^p-norm of a vector $\mathbf{x}$ by $\mathbf{x}^T$ and $\|\mathbf{x}\|_p, 1 \le p \le \infty$, respectively.

2. A Code-based Distributed Gradient Descent Algorithm

Let the coded objective functions $g_j, 1 \le j \le n$, and the decoding matrix $\mathbf{A} = (a(i,j))_{1\le i,j\le n}$ be as in the coding and decording scheme (1.7) and (1.8). Set decoding weights

$$w_i = \Big(\sum_{j=1}^{n} |a(i,j)|\Big)^{-1} = \Big(\sum_{j\in\Gamma_i} |a(i,j)|\Big)^{-1},\ 1 \le i \le n, \tag{2.1}$$

where

$$\Gamma_i = \{j,\ a(i,j) \ne 0\}. \tag{2.2}$$

For the decoding matrix $\mathbf{A}$ in (1.8), we define its normalized decoding matrix $\mathbf{A}_{\rm nde} = \big(\tilde{a}(i,j)\big)_{1\le i,j\le n}$ of size $n \times n$ by

$$\tilde{a}(i,j) = w_i a(i,j),\ 1 \le i,j \le n, \tag{2.3}$$

and its row stochastic decoding matrix $\mathbf{A}_{\rm sde}$ of size $2n \times 2n$ by

$$\mathbf{A}_{\rm sde} = \begin{pmatrix} \tilde{\mathbf{A}}_+ & \tilde{\mathbf{A}}_- \\ \tilde{\mathbf{A}}_+ & \tilde{\mathbf{A}}_- \end{pmatrix}, \tag{2.4}$$

where

$$\tilde{\mathbf{A}}_+ = \big((\tilde{a}(i,j))_+\big)_{1\le i,j\le n} \ \text{ and } \ \tilde{\mathbf{A}}_- = \big((-\tilde{a}(i,j))_+\big)_{1\le i,j\le n}$$

are positive/negative parts of the matrix $\mathbf{A}_{\rm nde} = (\tilde{a}(i,j))_{1\le i,j\le n}$ respectively. In this section, we introduce a code-based distributed gradient descent algorithm (2.5) to solve the optimization problem (1.2). For the consensus and convergence properties of $\mathbf{x}_i(k), k \ge 0$, in the proposed CoDGraD algorithm (2.5), we make the following assumptions on the row stochastic decoding matrix $\mathbf{A}_{\rm sde}$, the global objective function f and the (coded) local objective functions $g_i, 1 \le i \le n$.

Assumption 2.1. (i) The row stochastic matrix $\mathbf{A}_{\rm sde}$ in (2.4) has simple eigenvalue one and all other eigenvalues contained in the open unit complex disk centered at the origin.

(ii) The global objective function f is differentiable and strongly convex in the sense that

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle_N \geq A\|\mathbf{x} - \mathbf{y}\|_2^2$$

for all $\mathbf{x}, \mathbf{y} \in B(\bar{\mathbf{x}}, C_2)$, where $\langle \cdot, \cdot \rangle_N$ is the inner product on $\mathbb{R}^N$, the constant C_2 is defined in Theorem 2, and $B(\bar{\mathbf{x}}, C_2)$ is the ball with center $\bar{\mathbf{x}}$ and radius C_2.

(iii) The (coded) local objective functions $g_i, 1 \leq i \leq n$, have bounded gradients, i.e., there exists a positive constant M such that

$$\|\nabla g_i(\mathbf{x})\|_2 \leq M, \ \mathbf{x} \in \mathbb{R}^N.$$

(iv) The (coded) local objective functions $g_j, 1 \leq j \leq n$ are differentiable and have L-smoothly continuous gradients, i.e.,

$$\|\nabla g_i(x) - \nabla g_i(y)\|_2 \leq L\|x - y\|_2, \quad x, y \in \mathbb{R}^N, 1 \leq i \leq n,$$

for some positive constant L.

The rest of this section is organized as follows. In Subsections 2.1 and 2.2, we introduce the code-based distributed gradient descent algorithm (2.5) to solve the optimization problem (1.2) and we propose its implementation on a distributed network with all communicating and computing conducted only at active nodes, see Algorithm 2.2. Here active nodes are those workers over the network that don't witness delay or failure on computing or communicating. The eigenvalue assumption on the row stochastic matrix $\mathbf{A}_{\rm sde}$ in (2.4) plays a crucial role to establish the consensus and convergence properties of the proposed CoDGraD algorithm (2.5), see Theorems 3.1, 4.1 and 4.2. In Subsection 2.3, we discuss two structures on the decoding matrix $\mathbf{A}$ so that the row stochastic matrix $\mathbf{A}_{\rm sde}$ meets the requirement (i) in Assumption 2.1, see Propositions 2.5 and 2.7.

2.1. Code-based distributed gradient descent algorithm. To solve the optimization problem (1.2), we propose an iterative distributed algorithm with the implementation of the worker at the region i given by

$$\begin{cases} \mathbf{v}_i(k) = \nabla g_i(\mathbf{x}_i(k)), \\ \mathbf{y}_i^+(k) = \mathbf{x}_i(k) - \alpha_k \mathbf{v}_i(k), \\ \mathbf{y}_i^-(k) = \mathbf{x}_i(k) + \alpha_k \mathbf{v}_i(k), \\ \mathbf{x}_i(k+1) = \sum_{j \in \Gamma_i} w_i \big\{ (a(i,j))_+ \mathbf{y}_j^+(k) + (-a(i,j))_+ \mathbf{y}_j^-(k) \big\}, \quad k \geq 0, \end{cases} \tag{2.5}$$

where initials $\mathbf{x}_i(0)$ are chosen randomly or set zero initially, and step sizes $\alpha_k, k \geq 0$, are so chosen that

$$\sum_{k=0}^{\infty} \alpha_k = \infty \ \text{ and } \ \sum_{k=0}^{\infty} \alpha_k^2 < \infty \tag{2.6}$$

[3, 4, 28], and the weights w_i and the sets Γ_i of active neighboring nodes are given in (2.1) and (2.2) respectively. We call the above algorithm as a *code-based distributed gradient descent algorithm* and use CoDGraD for abbreviation. Our illustrative examples of step sizes are $\alpha_k = (k+a)^{-\theta}, k \geq 0$, for some $\theta \in (1/2, 1]$ and $a \geq 1$, see (1.4).

Write $\mathbf{A}_{\rm sde} = (q(i,j))_{1\le i,j\le 2n}$ and for $1 \le i \le n$, set

$$\mathbf{z}_i(k) = \mathbf{z}_{i+n}(k) = \mathbf{x}_i(k),\ k \ge 0. \tag{2.7}$$

Then the CoDGraD algorithm (2.5) can be rewritten as

$$\mathbf{z}_i(k+1) = \sum_{j=1}^{2n} q(i,j)\big(\mathbf{z}_j(k) - \alpha_k \mathbf{h}_j(k)\big),\ 1 \le i \le 2n, \tag{2.8}$$

where

$$\mathbf{h}_i(k) = \begin{cases} \nabla g_i(\mathbf{z}_i(k)) & \text{if } 1 \le i \le n \\ -\nabla g_{i-n}(\mathbf{z}_i(k)) & \text{if } n+1 \le i \le 2n. \end{cases} \tag{2.9}$$

Set

$$\mathbf{z}(k) := \big(\mathbf{z}_i(k)\big)_{1\le i\le 2n} \ \text{ and } \ \mathbf{h}(k) := \big(\mathbf{h}_i(k)\big)_{1\le i\le 2n} \tag{2.10}$$

with vectors $\mathbf{z}_i(k)$ and $\mathbf{h}_i(k) \in \mathbb{R}^N, 1 \le i \le 2n$, as their i-th entries respectively. Then the iterative algorithm (2.8) can be reformulated in a matrix form:

$$\mathbf{z}(k+1) = \mathbf{A}_{\rm sde}\mathbf{z}(k) - \alpha_k \mathbf{A}_{\rm sde}\mathbf{h}(k),\ k \ge 0. \tag{2.11}$$

Let $\mathbf{a}_{\rm sde}$ be the stationary probability vector invariant under the row stochastic matrix $\mathbf{A}_{\rm sde}$, i.e., the left eigenvector of $\mathbf{A}_{\rm sde}$ associated with eigenvalue one that satisfies

$$\mathbf{a}_{\rm sde}^T \mathbf{A}_{\rm sde} = \mathbf{a}_{\rm sde}^T \ \text{ and } \ \mathbf{a}_{\rm sde}^T \mathbf{1}_{2n} = 1. \tag{2.12}$$

Define

$$\bar{\mathbf{x}}(k) = \mathbf{a}_{\rm sde}^T \mathbf{z}(k). \tag{2.13}$$

In Theorem 3.1, we show that

$$\lim_{k\to\infty} \mathbf{x}_i(k) - \bar{\mathbf{x}}(k) = 0,\ 1 \le i \le n$$

and hence $\mathbf{x}_i(k), k \ge 1$, in the CoDGraD algorithm (2.5) has the consensus property.

By (2.11), the weighted average $\bar{\mathbf{x}}(k)$ of $\mathbf{x}_i(k), 1 \le i \le n$, satisfies

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k) - \alpha_k \mathbf{a}_{\rm sde}^T \mathbf{h}(k),\ k \ge 0.$$

We observe that $\mathbf{a}_{\rm sde}^T \mathbf{h}(k)$ is an inexact estimate of the scaled global gradient $\nabla f(\bar{\mathbf{x}}(k))$, see Proposition 4.4. As a consequence, we show that $\bar{\mathbf{x}}(k), k \ge 1$ converges to the solution $\bar{\mathbf{x}}$ of the optimization problem (1.2), see Theorems 4.1 and 4.2.

2.2. Distributed implementation of the code-based distributed gradient descent algorithm. For the distributed implementation of the CoDGraD (2.5), we describe the network topology by an undirected graph $\mathcal{G} = (V, E)$, where the worker in each region is represented by a vertex in V and each edge $(l, l') \in E$ means that workers in the regions l and $l' \in V$ have direct communication for data sharing. However, for each node i, any node $j \in \mathcal{N}_i = \{j : (i,j) \in E\} \cup \{i\}$ may (however does not always) contribute in the decoding process then any such node can be considered as *a non-straggling node of the active node i*, with some abuse of meaning. Similarly, any node $j \in V \backslash \mathcal{N}_i$ is considered as *a straggler of the active node i*. Thus, only nodes from the non-straggling set of nodes of an active node i can contribute in the decoding process at i. Therefore the topological matching property of the coded scheme (1.8) and (1.7) is satisfied if the decoding weight $a(i,j)$ takes zero value whenever there is no direct communication between active workers in the region i and j, i.e.,

$$a(i,j) = 0 \ \text{ if } \ (i,j) \notin E.$$

This means that any non-straggling node of any active node i which is contributing in the decoding process must be a neighbor of the active node i in the whole network, i.e.,

$$\Gamma_i \subset \mathcal{N}_i,$$

where Γ_i is given in (2.2). For the above scenario of the coded scheme (1.8) and (1.7), the active worker at each region first evaluates the coded local gradient, then it updates its estimate through gradient descent/ascent step, next it shares the updated estimate approximations with neighboring active workers at the adjacent regions, henceforth it updates the estimate towards the optimal solution $\bar{\mathbf{x}}$. Hence the CoDGradD algorithm is implementable in the network that does not have a fusion center at all. As observed from Algorithm 2.2 below, all communicating and computing in the CoDGradD algorithm could be conducted *only* at active nodes.

Algorithm 2.2. (CoDGraD algorithm)

1. **INPUT**: Set tolerance value ϵ for halting the algorithm and the number of iteration $k = 0$. Initialize an estimate $\mathbf{x}_i(0)$ of the optimal solution $\bar{\mathbf{x}}$ and approximation error $e_i(0) = \epsilon$ at the worker i.
2. **ITERATION**:
   - WHILE $e_i(k) \geq \epsilon$
   - Use (2.5) to update the estimate $\mathbf{x}_i(k)$ at the worker i.
   - Find error $e_i(k+1) = \|\mathbf{x}_i(k+1) - \mathbf{x}_i(k)\|$ for all $1 \leq i \leq n$.
   - Let $k = k + 1$
   - ENDWHILE
3. **OUTPUT**: $\mathbf{x}_i(k)$ and k.

2.3. Conditions on the network topology and (de)coding scheme. Denote the normalized decoding matrix of the decoding matrix $\mathbf{A}$ by

$$|\tilde{\mathbf{A}}| = (|\tilde{a}(i,j)|)_{1 \leq i,j \leq n}, \tag{2.14}$$

where $\tilde{a}(i,j), 1 \leq i,j \leq n$, are given in (2.3). By (2.3), the normalized decoding matrix $|\tilde{\mathbf{A}}|$ has row stochastic property. To meet the first requirement in Assumption 2.1 for the row stochastic matrix $\mathbf{A}_{\text{sde}}$, we need an equivalence between the simple eigenvalue properties for the row stochastic matrices $\mathbf{A}_{\text{sde}}$ and $|\tilde{\mathbf{A}}|$.

Proposition 2.3. The row stochastic decoding matrix $\mathbf{A}_{\text{sde}}$ has simple eigenvalue one and all other eigenvalues contained in the open unit complex disk centered at the origin if and only if $|\tilde{\mathbf{A}}|$ has the same property for its eigenvalues.

The proof of the above proposition will be given in Appendix A.1.

As $|\tilde{\mathbf{A}}|$ is a row stochastic matrix, it follows from Perron-Frobenius theorem that $|\tilde{\mathbf{A}}|$ has simple eigenvalue one if it is irreducible. In order for $|\tilde{\mathbf{A}}|$ to be irreducible, the digraph with weighted adjacent matrix $(|\tilde{a}(i,j)|)_{1 \leq i,j \leq n}$ should be strongly connected. But since $\Gamma_i = \{j, a(i,j) \neq 0\} = \{j, |\tilde{a}(i,j)| \neq 0\}$ and $\Gamma_i \subset \mathcal{N}_i$ then a necessary condition for this digraph to be strongly connected is that the network graph $\mathcal{G}$ is strongly connected.

In the following, we consider two scenarios of network topology in which the first requirement in Assumption 2.1 is met, see Structure 2.4 and 2.6. Denote the minimum

degree of the network graph $\mathcal{G}$ by

$$\delta(\mathcal{G}) = \min_{1\leq i\leq n} \deg(i)$$

where $\deg(i)$ is the degree of a node i, i.e., the number of edges connected with the node $i \in \mathcal{G}$. Let $s_i = n - |\mathcal{N}_i|$ be the number of nodes that are not connected to i. Thus, $s = \max_{1\leq i\leq n} s_i = n - \delta(\mathcal{G})$. In the first restrictive structure, the decoding matrix $\mathbf{A}$ can be described as follows.

Structure 2.4. The decoding matrix $\mathbf{A}$ formed for an n-node network and computed by Algorithm 1 of coding scheme in [34] which is robust to a maximum of s-stragglers should be formed such that at least $s+1$ entries are nonzero per each column and at least $s+1$ non zero entries per each row.

For the above scenario, we have the following property for the corresponding row stochastic matrix $\mathbf{A}_{\text{sde}}$, see Appendix A.2 for the proof.

Proposition 2.5. Let $\delta(\mathcal{G}) > n/2-1$, and the decoding matrix $\mathbf{A}$ satisfy Structure 2.4. Then the matrix $\mathbf{A}_{\text{sde}}$ used in the CoDGraD algorithm (2.5) has simple eigenvalue one and all other eigenvalues contained in the open unit complex disk centered at the origin.

Next we consider the scenario of full cyclic assignment of the active workers after certain permutation, i.e., the decoding matrix $\mathbf{A} = (a(i,j))_{1\leq i,j\leq n}$ with the first row having the $n-s$ entries after the first assigned as non-zero, and as we move down the rows, the positions of the $n-s$ non-zero entries shifting one step to the right, and cycle around until the last row.

Structure 2.6. The decoding matrix $\mathbf{A}$ formed for an n-node network and computed by Algorithm 1 of coding scheme in [34] satisfies the following conditions:

$$a(i,j) \neq 0 \tag{2.15}$$

for all (i,j) satisfying either $i+1 \leq j \leq \min(i+n-s, n)$ or $j+s \leq i$.

In this following proposition, we discuss the following eigenvalue property for the corresponding row stochastic matrix $\mathbf{A}_{\text{sde}}$, see Appendix A.3 for the proof.

Proposition 2.7. Let $1 \leq s \leq n-1$ and the decoding matrix $\mathbf{A}$ satisfy (2.15). Then the matrix $\mathbf{A}_{\text{sde}}$ used in the CoDGraD algorithm (2.5) has simple eigenvalue one and all other eigenvalues contained in the open unit complex disk centered at the origin.

3. Consensus property of the Code-based Distributed Gradient Descent Algorithm

In this section, we establish the consensus property of $\mathbf{x}_i(k),\ k \geq 1$, in the CoDGraD algorithm (2.5).

Theorem 3.1. *Let $\mathbf{x}_i(k),\ k \geq 1$, be in the CoDGraD algorithm* (2.5)*. If the row stochastic matrix $\mathbf{A}_{\text{sde}}$ in* (2.4) *has simple eigenvalue one and all other eigenvalues contained in the open unit complex disk centered at the origin, the sequence $\{\alpha_k\}_{k=0}^{\infty}$ satisfies $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, and the local objective functions $g_i, 1 \leq i \leq n$, have bounded gradients, i.e., there exists a positive constant M such that*

$$\|\nabla g_i(\mathbf{x})\|_2 \leq M,\ \mathbf{x} \in \mathbb{R}^N, \tag{3.1}$$

then

$$\lim_{k\to\infty}(\mathbf{x}_i(k)-\mathbf{x}_j(k))=0 \tag{3.2}$$

and

$$\lim_{k\to\infty}(f(\mathbf{x}_i(k))-f(\mathbf{x}_j(k)))=0 \tag{3.3}$$

for all $1\le i,j\le n$.

Define

$$\tilde{\mathbf{z}}(k)=\mathbf{z}(k)-\mathbf{P}\mathbf{z}(k),\ k\ge 0, \tag{3.4}$$

where $\mathbf{z}_k, k\ge 0$ are defined by (2.10), the stationary probability vector $\mathbf{a}_{\rm sde}$ is given in (2.12), and

$$\mathbf{P}=\mathbf{1}_{2n}(\mathbf{a}_{\rm sde})^T. \tag{3.5}$$

Then the consensus property (3.2) reduces to establishing

$$\lim_{k\to\infty}\|\tilde{\mathbf{z}}(k)\|_{2,\infty}=0, \tag{3.6}$$

where $\|\mathbf{z}\|_{2,\infty}=\sup_{1\le i\le 2n}\|\mathbf{z}_i\|_2$ for a vector $\mathbf{z}=(\mathbf{z}_i)_{1\le i\le 2n}$ with entries $\mathbf{z}_i\in\mathbb{R}^N, 1\le i\le 2n$.

Let the row stochastic matrix $\mathbf{A}_{\rm sde}$ in (2.4) have simple eigenvalue one and $\lambda_m(\mathbf{A}_{\rm sde}), 1\le m\le 2n$, be its eigenvalues listed in the order that

$$1=\lambda_1(\mathbf{A}_{\rm sde})>|\lambda_2(\mathbf{A}_{\rm sde})|\ge\ldots\ge|\lambda_{2n}(\mathbf{A}_{\rm sde})|. \tag{3.7}$$

For $\|\tilde{\mathbf{z}}(k)\|_{2,\infty}, k\ge 1$, in (3.4), we have

Proposition 3.2. Let $\mathbf{A}_{\rm sde}$, $\lambda_2(\mathbf{A}_{\rm sde})$ and $\mathbf{P}$ be as in (2.4), (3.7) and (3.5) respectively. Assume that the row stochastic matrix $\mathbf{A}_{\rm sde}$ has simple eigenvalue one, and that the local objective functions $g_i, 1\le i\le n$, satisfy (3.1). Then there exists a positive constant C_1 such that

$$\begin{aligned}\|\tilde{\mathbf{z}}(k)\|_{2,\infty}\ \le\ & C_1M\sum_{l=0}^{k-1}\Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l}\alpha_l\\ &+C_1\Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k}\|\tilde{\mathbf{z}}(0)\|_{2,\infty}\end{aligned} \tag{3.8}$$

hold for all $k\ge 1$.

The detailed proof of Proposition 3.2 is given in Appendix A.4.

Let $\bar{\mathbf{x}}(k)$ be as in (2.13). By (3.5), we can write

$$\mathbf{P}\mathbf{z}(k)=\bar{\mathbf{x}}(k)\mathbf{1}_{2n}. \tag{3.9}$$

Observe that

$$\|\tilde{\mathbf{z}}(k)\|_{2,\infty}=\max_{1\le i\le n}\|\mathbf{x}_i(k)-\bar{\mathbf{x}}(k)\|_2,\ k\ge 1.$$

Then by Proposition 3.2 we obtain the following estimate about the consensus property of $\mathbf{x}_i(k)$ for different $1\le i\le n$:

$$\begin{aligned}\max_{1\le i\le n}\|\mathbf{x}_i(k)-\bar{\mathbf{x}}(k)\|_2\ \ \le\ & C_1\Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k}\max_{1\le i\le n}\|\mathbf{x}_i(0)-\bar{\mathbf{x}}(0)\|_2\\ &+C_1M\sum_{l=0}^{k-1}\Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l}\alpha_l,\ \ k\ge 1.\end{aligned} \tag{3.10}$$

Set $\gamma = (1 + |\lambda_2(\mathbf{A}_{\rm sde})|)/2 \in (1/2, 1)$. Applying (3.10) to our illustrative example (1.4) of step sizes, we can find a positive constant C such that

$$\begin{aligned}\max_{1\le i\le n} \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|_2 \le\ & C_1 M \sum_{l=0}^{k-1} \gamma^{k-l}(k-l+a)^{\theta}(k+a)^{-\theta} \\ & + C_1 \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2 \gamma^k \le C(k+a)^{-\theta} \end{aligned} \tag{3.11}$$

hold for all $k \ge 0$, where the first inequality follows from (3.10) and the observation that $a \ge 1$ and $(l+a)^{-\theta} \le (k+a)^{-\theta}(k-l+a)^{\theta}, 0 \le l \le k$, and the second estimate holds by the boundness of the sequence $\gamma^k(k+a)^{\theta}, k \ge 0$, and the convergence of the series $\sum_{m=0}^{\infty} \gamma^m (m+a)^{\theta}$. Therefore we have the following consensus property.

Corollary 3.3. Let $\mathbf{x}_i(k)$, $k \ge 1$, be in the CoDGraD algorithm (2.5) with $\alpha_k = (k+1)^{-\theta}, k \ge 0$ for some $1/2 < \theta \le 1$, and the row stochastic matrix $\mathbf{A}_{\rm sde}$ in (2.4) and the local objective functions $g_i, 1 \le i \le n$, be as in Theorem 3.1. There there exists a positive constant C such that

$$|\mathbf{x}_i(k) - \mathbf{x}_j(k)| \le C(k+a)^{-\theta} \tag{3.12}$$

hold for all $1 \le i, j \le n$ and $k \ge 0$.

Remark 3.4. For the case that the step sizes have exponential decay, i.e., $\alpha_k = r^k, k \ge 0$ for some $r \in (0,1)$, (the first condition in (2.6) is not satisfied in this case), we obtain from Proposition 3.2 that $\mathbf{x}_i(k), k \ge 0$ in the CoDGrad algorithm (2.5) have the exponential consensus property, i.e., there exists a positive constant C such that

$$|\mathbf{x}_i(k) - \mathbf{x}_j(k)| \le C\Big(\max\Big(r, \frac{1 + |\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)\Big)^k,\ k \ge 0.$$

However, we cannot obtain the convergence of $\mathbf{x}_i(k), k \ge 0$ to the solution $\bar{\mathbf{x}}$ of the optimization problem (1.2) from Proposition 4.6, cf. Theorem 4.2.

We finish this section with the proof of Theorem 3.1.

Proof of Theorem 3.1. By (3.7) and (3.8),

$$\begin{aligned} & \Big(\sum_{k=0}^{\infty} \max_{1\le i\le n} \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|_2^2\Big)^{1/2} \\ \le\ & C_1 M \Big(\sum_{k=0}^{\infty}\Big(\sum_{l=0}^{k-1}\Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l}\alpha_l\Big)^2\Big)^{1/2} \\ & + C_1\Big(\sum_{k=0}^{\infty}\Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{2k}\Big)^{1/2} \max_{1\le i\le n}\|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2 \\ \le\ & \frac{2C_1 M}{1-|\lambda_2(\mathbf{A}_{\rm sde})|}\Big(\sum_{k=0}^{\infty}\alpha_k^2\Big)^{1/2} + \frac{2C_1}{1-|\lambda_2(\mathbf{A}_{\rm sde})|}\max_{1\le i\le n}\|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2. \end{aligned} \tag{3.13}$$

Therefore the limit in (3.2) follows.

By (1.8) and (2.1), we have

$$\|\nabla f(\mathbf{x})\|_2 \le W^{-1} \sup_{1\le j\le n} \|\nabla g_j(\mathbf{x})\|_2 \le MW^{-1},$$

where $W = \max_{1\le i\le n} w_i$. This together with Proposition 3.2 implies that

$$\begin{aligned}
&|f(\mathbf{x}_i(k)) - f(\bar{\mathbf{x}}(k))| \le |\langle \nabla f(\bar{\mathbf{x}}(k)), \mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\rangle| \\
&\le C_1 M^2 W^{-1} \sum_{l=0}^{k-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l} \alpha_l \\
&\quad + C_1 M W^{-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k} \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2 \\
&\to 0 \ \text{ as } k\to\infty
\end{aligned} \tag{3.14}$$

for all $1 \le i \le n$. Then (3.3) follows. □

4. Convergence property of the Code-based Distributed Gradient Descent Algorithm

In this section, we establish the convergence of $\mathbf{x}_i(k)$, $k \ge 1$, in the CoDGraD algorithm (2.5) to the solution $\bar{\mathbf{x}}$ of the optimization problem (1.2). By (3.2), (3.9) and (3.13), it suffices to show that $\bar{\mathbf{x}}(k), k \ge 1$, in (2.13) converges to $\bar{\mathbf{x}}$.

Theorem 4.1. *Assume that the row stochastic matrix $\mathbf{A}_{\rm sde}$ in (2.4) has simple eigenvalue one and all other eigenvalues contained in the open unit complex disk centered at the origin, the sequence $\{\alpha_k\}_{k=0}^{\infty}$ satisfies (2.6), the local objective functions $g_j, 1 \le j \le n$, satisfy (3.1) and*

$$\|\nabla g_i(x) - \nabla g_i(y)\|_2 \le L\|x-y\|_2, \quad x, y \in \mathbb{R}^N, 1 \le i \le n, \tag{4.1}$$

for some positive constant L, and the global objective function f is strongly convex in the sense that

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle_N \ge A\|\mathbf{x}-\mathbf{y}\|_2^2 \tag{4.2}$$

for all $\mathbf{x}, \mathbf{y} \in B(\bar{\mathbf{x}}, C_2)$, where $\langle\cdot,\cdot\rangle_N$ is the inner product on $\mathbb{R}^N$, $B(\bar{\mathbf{x}}, C_2)$ is the ball with center $\bar{\mathbf{x}}$ and radius C_2, and

$$\begin{aligned}
C_2 :&= \exp\Big(\sum_{j=0}^{\infty} \alpha_j^2\Big)\Big\{\frac{8C_1^2L^2}{(1-|\lambda_2(\mathbf{A}_{\rm sde}))|)^2} \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2^2 \\
&\quad + \frac{M^2(1+8C_1^2)}{(1-|\lambda_2(\mathbf{A}_{\rm sde})|)^2}\Big(\sum_{j=0}^{\infty}\alpha_j^2\Big) + \|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2\Big\}.
\end{aligned}$$

Then $\bar{\mathbf{x}}(k), k \ge 1$, in (2.13) converges to the solution $\bar{\mathbf{x}}$ of the optimization problem (1.2),

$$\lim_{k\to\infty} \bar{\mathbf{x}}(k) = \bar{\mathbf{x}}. \tag{4.3}$$

Combining Theorems 3.1 and 4.1, we have the following result on the convergence of the proposed CoDGraD algorithm.

Theorem 4.2. *Let the decoding matrix $\mathbf{A}_{\rm sde}$ and the objective function f satisfy Assumption 2.1, and step size $\{\alpha_k\}_{k=0}^{\infty}$ in the CoDGrad algorithm satisfy (2.6). Then the sequences $\{\mathbf{x}_i(k)\}_{k=1}^{\infty}, 1 \le i \le n$ in the CoDGrad algorithm converge to the optimal point $\bar{\mathbf{x}}$, i.e.,*

$$\lim_{k\to\infty} \mathbf{x}_i(k) = \bar{\mathbf{x}}, \ 1 \le i \le n. \tag{4.4}$$

To prove Theorem 4.1, we need a technical lemma, which follows the probability property for the vector $\mathbf{a}_{\rm sde}$ in (2.12). For the completeness of this paper, we include a detailed proof in Appendix A.5.

Lemma 4.3. *Let* $\mathbf{a}_{\rm sde}$ *and* $w_i, 1 \le i \le n$, *be as in* (2.12) *and* (2.1) *respectively. Set*

$$\mathbf{w} = (w_1, \ldots, w_n, w_1, \ldots, w_n)^T \tag{4.5}$$

and

$$\tilde{w} = \mathbf{a}_{\rm sde}^T \mathbf{w}. \tag{4.6}$$

Then

$$0 < \min_{1\le i\le n} w_i \le \tilde{w} \le \max_{1\le i\le n} w_i. \tag{4.7}$$

Let $\bar{\mathbf{x}}(k), k \ge 0$ be as in (2.13). By (2.11), the weighted average $\bar{\mathbf{x}}(k), k \ge 0$ can be defined by the following iterative algorithm,

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k) - \alpha_k \mathbf{a}_{\rm sde}^T \mathbf{h}(k),\ k \ge 0. \tag{4.8}$$

For local objective functions $g_i, 1 \le i \le n$, with Lipschitz gradients (4.1), we observe that $\mathbf{a}_{\rm sde}^T \mathbf{h}(k)$ is an inexact estimate of the scaled global gradient $\tilde{w}\nabla f(\bar{\mathbf{x}}(k))$, see Appendix A.6 for a detailed proof.

Proposition 4.4. Let $\mathbf{A}_{\rm sde}$, $\lambda_2(\mathbf{A}_{\rm sde})$, $\mathbf{P}$ and $\tilde{w}$ be as in (2.4), (3.7), (3.5) and (4.6) respectively. Assume that the row stochastic matrix $\mathbf{A}_{\rm sde}$ has simple eigenvalue one, and the local objective functions $g_j, 1 \le j \le n$, satisfy (4.1). Then

$$\|\mathbf{a}_{\rm sde}^T \mathbf{h}(k) - \tilde{w}\nabla f(\bar{\mathbf{x}}(k))\|_2 \le L \|\mathbf{a}_{\rm sde}\|_1 \|\tilde{\mathbf{z}}(k)\|_{2,\infty}. \tag{4.9}$$

By Proposition 4.4, the iterative algorithm (4.8) can be considered as the gradient descent algorithm (1.3) with inexact gradient update. Then following a standard argument, we have the boundedness of $\bar{\mathbf{x}}(k), k \ge 1$, when the objective function f is convex, see Appendix A.7 for a detailed proof.

Proposition 4.5. Let the matrix $\mathbf{A}_{\rm sde}$, the sequence $\{\alpha_k\}_{k=0}^{\infty}$ and the local objective functions $g_j, 1 \le j \le n$, be as in Theorem 4.1. If the global objective function f is convex, then

$$\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 \le C_2 \quad \text{for all} \quad k \ge 0, \tag{4.10}$$

where C_2 is the constant in Theorem 4.1.

The estimate in (4.10) can be improved if the objective function f has the strongly convex property (4.2), see Appendix A.8 for a detailed proof.

Proposition 4.6. Let the matrix $\mathbf{A}_{\rm sde}$, the sequence $\{\alpha_k\}_{k=0}^{\infty}$, the local objective functions $g_j, 1 \le j \le n$, and the global objective function f be as in Theorem 4.1. Then

$$\begin{aligned}\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 \le\ & \exp\Big(-\tilde{w}A\sum_{j=0}^{k-1}\alpha_j\Big)\Big\{\|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 + \sum_{j=0}^{k-1}\exp\Big(\tilde{w}A\sum_{l=0}^{j}\alpha_l\Big) \\ & \times\Big(M^2\alpha_j^2 + L^2 \max_{1\le i\le n}\|\mathbf{x}_i(j) - \bar{\mathbf{x}}(j)\|_2^2\Big)\Big\}\end{aligned} \tag{4.11}$$

hold for all $k \ge 2$.

Set $W = \max_{1\le i\le n} w_i$. Observe that

$$\|\nabla f(\mathbf{x})\|_2 \le \|\nabla f(\mathbf{x}) - \nabla f(\bar{\mathbf{x}})\|_2 \le LW^{-1}\|\mathbf{x} - \bar{\mathbf{x}}\|_2,$$

where the second inequality follows from (4.1) and the row stochastic property for the matrix $\mathbf{A}_{\rm sde}$. This together with Proposition 4.6 proves that

$$\begin{aligned} f(\bar{\mathbf{x}}) \le f(\bar{\mathbf{x}}(k)) \;\le\; f(\bar{\mathbf{x}}) + LW^{-1}\exp\Big(-\tilde{w}A\sum_{j=0}^{k-1}\alpha_j\Big) \\ \times\Big\{\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2 + \sum_{j=0}^{k-1}\exp\Big(\tilde{w}A\sum_{l=0}^{j}\alpha_l\Big) \\ \times\Big(M^2\alpha_j^2 + L^2\max_{1\le i\le n}\|\mathbf{x}_i(j)-\bar{\mathbf{x}}(j)\|_2^2\Big)\Big\} \end{aligned} \tag{4.12}$$

hold for all $k \ge 2$.

Applying (4.11) to the illustrative examples (1.4) of step sizes, we obtain the following convergence result for the proposed CoDGraD algorithm, cf. the convergence rate $O(\log k/\sqrt{k})$ for the uncoded incremental gradient methods [14, 24].

Corollary 4.7. Let $\mathbf{x}_i(k)$, $k \ge 1$, be in the CoDGraD algorithm (2.5) with $\alpha_k = (k+1)^{-\theta}, k \ge 0$ for some $1/2 < \theta \le 1$, and the row stochastic matrix $\mathbf{A}_{\rm sde}$ in (2.4), the local objective functions $g_i, 1 \le i \le n$ and the global objective function f be as in Theorem 4.1. Then there exists a positive constant C such that

$$\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2 \le C(k+a)^{-\theta/2},\ k \ge 0, \tag{4.13}$$

where $\bar{\mathbf{x}}$ is the solution of the optimization problem (1.2),

The detailed proof of Corollary 4.7 will be given in Appendix A.9.

We finish this section with the proof of Theorem 4.1 under the assumption that Proposition 4.6 holds.

Proof of Theorem 4.1. By (2.6) and (3.13), we have

$$\sum_{j=0}^{\infty} M^2\alpha_j^2 + L^2\max_{1\le i\le n}\|\mathbf{x}_i(j)-\bar{\mathbf{x}}(j)\|_2^2 < \infty, \tag{4.14}$$

and

$$\lim_{k\to\infty}\exp\Big(-\tilde{w}A\sum_{j=l}^{k}\alpha_j\Big) = 0 \tag{4.15}$$

for all $l \ge 0$. Combining (4.11), (4.14) and (4.15) proves the desired limit in (4.3) by the dominated convergence theorem. □

5. Numerical Simulations

In this section, we consider the following unconstrained convex optimization problem on a network

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N}\|\mathbf{G}\mathbf{x} - \mathbf{y}\|_2^2, \tag{5.1}$$

where the network contains n regions with each region of the partition equipped with a worker, $\mathbf{G}$ is a random matrix of size $Q \times N$ whose entries are independent and identically distributed standard normal random variables, and

$$\mathbf{y} = \mathbf{G}\mathbf{x}_o \in \mathbb{R}^Q \tag{5.2}$$

has entries of $\mathbf{x}_o$ being identically independent random variables sampled from the uniform bounded random distribution between -1 and 1. The solution $\bar{\mathbf{x}}$ of the above optimization problem is the least squares solution of the overdetermined system $\mathbf{y} = \mathbf{G}\mathbf{x}_o, \mathbf{x}_o \in \mathbb{R}^N$. In this section, we demonstrate the performance of the CoDGraD algorithm (2.5) to solve the convex optimization problem (5.1) and also compare it with the performance of the the conventional distributed gradient descent algorithm (DGD) under the CTA prototype (1.6).

Assume that the network has n active nodes. Then we can repartition the network into n regions around those n nodes, and accordingly, the random measurement matrix $\mathbf{G}$, the measurement data $\mathbf{y}$, and the objective function $f(\mathbf{x}) := \|\mathbf{G}\mathbf{x} - \mathbf{y}\|_2^2$ in (5.1) as follows,

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}) := \sum_{i=1}^{n} \|\mathbf{G}_i\mathbf{x} - \mathbf{y}_i\|_2^2.$$

In our simulations, we assume that the repartitioned regions have the same size, i.e., the number of rows in $\mathbf{G}_i$ and lengths of vectors $\mathbf{y}_i$, for all i where $1 \le i \le n$ are all equal. Shown in Figure 1 are two undirected graphs to describe data exchanging structure for active nodes of a 3-node and 5-node network respectively.

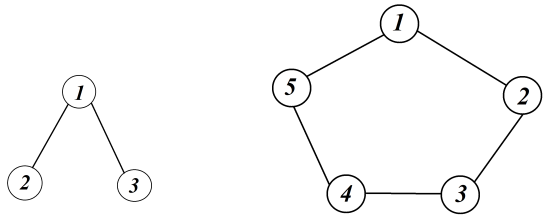

FIGURE 1. Data exchanging structures with three/five-node network

In our simulations, we take $(Q, N) = (225, 75)$ for Figures 2, 3 and $(M, N) = (250, 50)$ for Figures 4 and 5. We use absolute error

$$\text{AE} := \max_{1 \le i \le n} \frac{\|\mathbf{x}_i(k) - \mathbf{x}_o\|_2}{\|\mathbf{x}_0\|_2}$$

and consensus error

$$\text{CE} := \max_{1 \le i \le n} \frac{\|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|_2}{\|\mathbf{x}_o\|_2}$$

to measure the performance of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6), where n is the number of active nodes in the network.

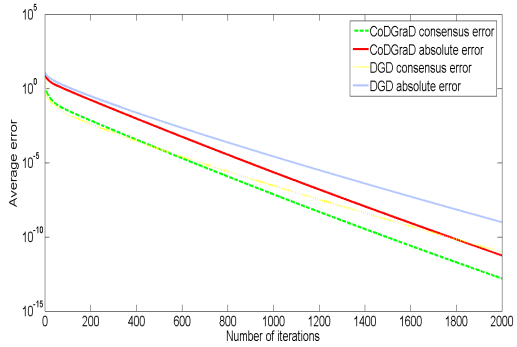

FIGURE 2. Performance comparison of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) over a three active nodes network with $(Q, N) = (225, 225)$ and step sizes $\alpha_k = (k + 300)^{-0.75}, k \geq 0$.

In the first simulation where there are 3 nodes with its topology described on the left of Figure 1, we take the coding matrix

$$\mathbf{B} = \begin{pmatrix} 1 & -5/4 & 0 \\ 0 & 1 & 4/9 \\ 18/10 & 0 & 1 \end{pmatrix} \tag{5.3}$$

and the decoding matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 10/18 \\ 1 & 9/4 & 0 \\ -4/5 & 0 & 1 \end{pmatrix}. \tag{5.4}$$

The above coding/decoding matrix pair satisfies (1.9) and the corresponding row stochastic matrix in (2.4) is

$$\mathbf{A}_{\text{sde}} = \begin{pmatrix} 0 & 18/28 & 10/28 & 0 & 0 & 0 \\ 4/13 & 9/13 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5/9 & 4/9 & 0 & 0 \\ 0 & 18/28 & 10/28 & 0 & 0 & 0 \\ 4/13 & 9/13 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5/9 & 4/9 & 0 & 0 \end{pmatrix}.$$

In Figures 2 and 3, we consider three non-straggler network with the coding/decoding matrices given in (5.3) and (5.4) respectively, and present the performance of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) with absolute and consensus metric being the average of the corresponding metrics over 100 trials, where random measurement matrix $\mathbf{G}$ has independent and identically distributed standard normal random variables as its entries, the original vector $\mathbf{x}_o$ is identically independent random variables uniform distributed in $[-1, 1]$, and step sizes are $\alpha_k = (k + 300)^{-0.75}$ and $(k + 500)^{-0.85}, k \geq 0$, respectively.

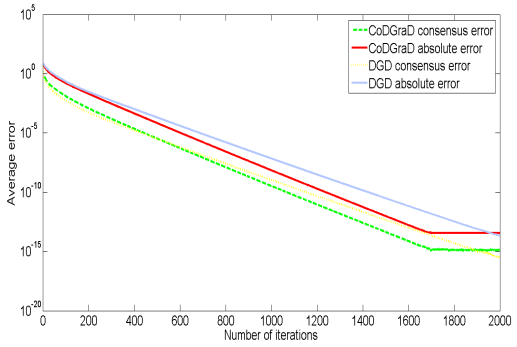

FIGURE 3. Performance comparison of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) over a three active nodes network with $(Q, N) = (225, 225)$ and step sizes $\alpha_k = (k + 500)^{-0.85}, k \geq 0$.

In the second simulation, the network has 5 active nodes with data exchanging structure described on the right of Figure 1. In that simulation, the coding/decoding matrices are given by

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1/2 & 0 & 0 \\ 0 & -1 & 3 & 4 & 0 \\ 0 & 0 & -5/2 & -3 & 1 \\ 1 & 0 & 0 & 1/5 & 13/5 \\ 2 & 1 & 0 & 0 & 4 \end{pmatrix} \tag{5.5}$$

and

$$\mathbf{A} = \begin{pmatrix} 1/2 & 1/4 & 0 & 0 & 1/4 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & -8/5 & 1 & 0 \\ 0 & 0 & -2/5 & -1 & 1 \\ 2 & 0 & 0 & 5 & -3 \end{pmatrix} \tag{5.6}$$

respectively. The above coding/decoding matrix pair satisfies (1.9) and the corresponding row stochastic matrix in (2.4) is

$$\mathbf{A}_{\text{sde}} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{5}{18} & 0 & 0 & \frac{5}{18} & \frac{8}{18} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{12} & 0 & 0 & \frac{1}{6} & \frac{5}{12} & 0 \\ \frac{1}{5} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{3}{10} \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{5}{18} & 0 & 0 & \frac{5}{18} & \frac{8}{18} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{12} & 0 & 0 & \frac{1}{6} & \frac{5}{12} & 0 \\ \frac{1}{5} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{3}{10} \end{pmatrix}.$$

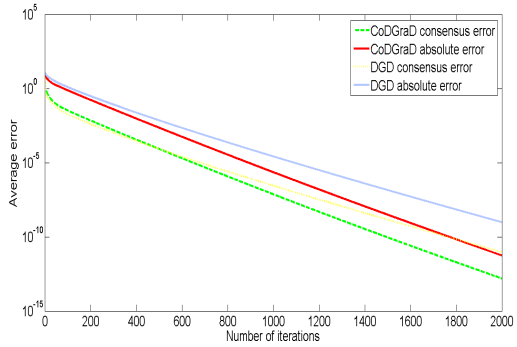

FIGURE 4. Performance comparison of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) over a five node network with $(Q, N) = (225, 225)$ and step sizes $\alpha_k = (k + 800)^{-0.9}, k \geq 0$.

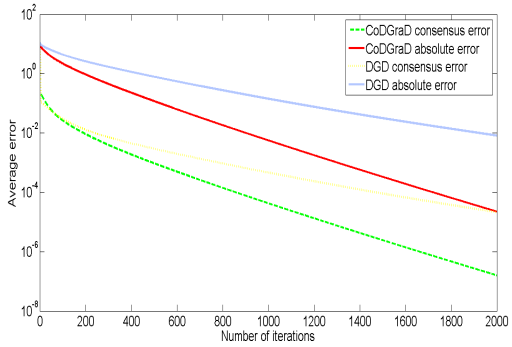

FIGURE 5. Performance comparison of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) over a five node network with $(Q, N) = (225, 225)$ and step sizes $\alpha_k = (k + 500)^{-0.95}, k \geq 0$.

Shown in Figures 4 and 5 are the performance of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6), where the absolute metric and consensus metric are the average of the corresponding metrics over 100 trials with random measurement matrix $\mathbf{G}$ and the original vector $\mathbf{x}_o$ being selected as in the first simulation, and step sizes being $\alpha_k = (k + 800)^{-0.90}$ and $(k + 500)^{-0.95}, k \geq 0$, respectively.

From the above simulations, we observe that the CoDGraD algorithm (2.5) has much better performance than the CTA algorithm (1.6) in reaching consensus. Even though $\bar{\mathbf{x}}(k)$ satisfies a gradient descent algorithm (4.8) with an inexact global gradient, see Proposition 4.4, our simulations indicate that the CoDGraD algorithm (2.5) still has comparable performance in the absolute error with the DGD algorithm under the CTA prototype (1.6). Also we can conceive from the simulations that the CoDGraD algorithm (2.5) has faster convergence for a smaller exponent $\theta \in (1/2, 1]$, which confirms its convergence rate estimate in (3.11) and (A.26). On the other hand, our simulations

also indicate that decreasing the exponent θ moves the CoDGraD algorithm (2.5) into the instability phase, which could directly be related to the sparsity of the network (i.e., graph degree $\delta(G)$ and $\deg(i)$ distribution). It is worth mentioning that we can adequately calibrate this instability by increasing the value of a in our illustrative examples of step sizes (1.4) for a fixed exponent θ. Thus we can anticipate in Figures 2 and 3 that the increase in the value to $\theta = 0.85$ degraded the convergence so that the CoDGraD algorithm became closer in performance to the DGD algorithm. While in Figures 4 and 5 we realize that the lower value of $\theta = 0.75$ is impermissible since the CoGraD algorithm will considerably enter the instability region while a higher value of $\theta = 0.9$ favors a better convergence rate and the highest value of $\theta = 0.95$ degraded the convergence again.

In these simulations, we have compared the performance of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) over the described 3-node and 5-node networks with subsystems on nodes being overdetermined. Therefore, a least squares solutions on each node will correspond to the unique solution of a strongly convex function and will consequently correspond to the least squared solution of the whole network, that is, the unique solution of the strongly convex global function f. It is observed that the errors decrease significantly in 2000 iterations where CoDGraD outperforms DGD in reaching the unique minimizer (i.e., absolute error) and in reaching consensus. While for both algorithms the consensus error decreases at a higher rate than the absolute error meaning that the workers become closer in their estimates while they all drift towards the unique solution. Our further simulations indicate that convergence behaviors of the CoDGraD algorithm (2.5) and the DGD algorithm (1.6) depends directly on maximal condition number of matrices $\mathbf{G}_i, 1 \leq i \leq n$, cf. (3.14) and (4.12) where A is closely related to the maximal condition number in the current setting.

6. Conclusions

In this paper, we propose the Code-Based Distributed Gradient Descent algorithm (2.5) to solve a convex optimization problem over a network. The proposed algorithm is a distributed version of gradient descent algorithm which employs coding to leverage the convergence rate performance and allow resilience to stragglers in conjunction with information privacy. To implement the proposed algorithm, we employ a centralized coordinated coding strategy. This work can be considered as the first step for a future investigation of exploiting distributed coding on time-varying networks under the effect of computation and communication failures or delays.

Acknowledgements

This work was partially supported by the National Science Foundation under Grants No. CCF-1718195 and ECCS-181256. The authors would like to thank the reviewer for the constructive comments.

Appendix A

In the appendix, we provide the proofs of Lemma 4.3, Corollary 4.7, and Propositions 2.3, 2.5, 2.7, 3.2, 4.4, 4.5 and 4.6.

A.1. Proof of Proposition 2.3. Observe that $\tilde{\mathbf{A}}_+ + \tilde{\mathbf{A}}_- = |\tilde{\mathbf{A}}|$. Then, in order to establish the equivalence in the proposition, it suffices to to prove that for any positive integer $k \geq 1$ and nonzero complex number $\lambda \neq 0$, the null space of $\left(\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix} - \lambda \mathbf{I}_{2n}\right)^k$ and the one of $(\mathbf{A} + \mathbf{B} - \lambda \mathbf{I}_n)^k$, to be denoted by $\mathcal{N}_k^1$ and $\mathcal{N}_k^2$ respectively, have the same dimension, where $\mathbf{A}$ and $\mathbf{B}$ are two square matrices of size $n \times n$. In particular, we will show that

$$\mathcal{N}_k^1 = \left\{ \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix},\ \mathbf{u} \in \mathcal{N}_k^2 \right\}. \tag{A.1}$$

Set $\mathbf{C} = \mathbf{A} + \mathbf{B}$. By induction on $j \geq 1$, we can show that

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix}^j = \begin{pmatrix} \mathbf{C}^{j-1} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{C}^{j-1} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix}. \tag{A.2}$$

Therefore any $\mathbf{u} \in \mathcal{N}_k^2$, i.e., $(\mathbf{C} - \lambda \mathbf{I}_n)^k \mathbf{u} = \mathbf{0}_{n\times 1}$, we have

$$\begin{aligned} &\left(\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix} - \lambda \mathbf{I}_{2n}\right)^k \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix} \\ &= \sum_{j=1}^{k} \binom{k}{j} (-\lambda)^{k-j} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix}^j \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix} + (-\lambda)^k \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix} \\ &= \begin{pmatrix} (\mathbf{C} - \lambda \mathbf{I}_n)^k \mathbf{u} \\ (\mathbf{C} - \lambda \mathbf{I}_n)^k \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \end{pmatrix}, \end{aligned}$$

where the second equality follows from (A.2). This proves that

$$\mathcal{N}_k^1 \supset \left\{ \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix},\ \mathbf{u} \in \mathcal{N}_k^2 \right\}. \tag{A.3}$$

On the other hand, for any $\mathbf{u}, \mathbf{w} \in \mathbb{C}^n$ satisfying

$$\left(\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix} - \lambda \mathbf{I}_{2n}\right)^k \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \end{pmatrix},$$

we obtain from (A.2) that

$$\begin{aligned} (-\lambda)^k \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} &= -\sum_{j=1}^{k} \binom{k}{j} (-\lambda)^{k-j} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{A} & \mathbf{B} \end{pmatrix}^j \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} \\ &= -\sum_{j=1}^{k} \binom{k}{j} (-\lambda)^{k-j} \begin{pmatrix} \mathbf{C}^{j-1}(\mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{w}) \\ \mathbf{C}^{j-1}(\mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{w}) \end{pmatrix}. \end{aligned}$$

This implies that $\mathbf{w} = \mathbf{u}$. Substituting the above equality back, we get

$$(-\lambda)^k \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix} = -\sum_{j=1}^{k} \binom{k}{j} (-\lambda)^{k-j} \begin{pmatrix} \mathbf{C}^j \mathbf{u} \\ \mathbf{C}^j \mathbf{u} \end{pmatrix}$$

which implies that $(\mathbf{C} - \lambda \mathbf{I}_n)^k \mathbf{u} = \mathbf{0}_{n\times 1}$. Hence

$$\mathcal{N}_k^1 \subset \left\{ \begin{pmatrix} \mathbf{u} \\ \mathbf{u} \end{pmatrix},\ \mathbf{u} \in \mathcal{N}_k^2 \right\}. \tag{A.4}$$

Combining (A.3) and (A.4) proves (A.1), and hence completes the proof.

A.2. Proof of Proposition 2.5. Set $\mathbf{B} = |\tilde{\mathbf{A}}|$, and write $\mathbf{B} = (b(i,j))_{1\le i,j\le n}$ and $\mathbf{B}^k = (b_k(i,j))_{1\le i,j\le n}, k \ge 1$. Then $\mathbf{B}$ is a row stochastic matrix. By Perron-Frobenius theorem and Proposition 2.3, it suffices to prove that $\mathbf{B}$ is primitive, i.e., for any $1 \le i, j \le n$, there exists $k \ge 1$ such that

$$b_k(i,j) \neq 0. \tag{A.5}$$

Set $C_1 = \{l, b(i,l) \neq 0\}$ and $C_2 = \{l, b(l,j) \neq 0\}$. By the assumption on the decoding matrix $\mathbf{A}$, they contain at least $\delta(\mathcal{G}) > n/2$ elements and hence there exists $l_0 \in C_1 \cap C_2$. Hence

$$b_2(i,j) = \sum_{l=1}^{n} b(i,l)b(l,j) \ge b(i,l_0)b(l_0,j) > 0.$$

This proves (A.5) with $k = 2$. But since $\mathbf{B}$ is row stochastic then eigenvalue one is the unique eigenvalue on the unit circle centered at the origin and this completes the proof.

A.3. Proof of Proposition 2.7. As in the proof of Proposition 2.5, it suffices to prove that $\mathbf{B}$ is primitive, i.e., for any $1 \le i, j \le n$, there exists $k \ge 1$ such that

$$b_k(i,j) \neq 0. \tag{A.6}$$

By the assumption on the decoding matrix $\mathbf{A}$, we have

$$b(i,j) \neq 0 \text{ if } i+1 \le j \le i+n-s \text{ and if } j+s \le i. \tag{A.7}$$

Observe that for any $1 \le i, j \le n$, we have

$$\begin{aligned} b_{k+1}(i,j) \;\; = \;\; & \sum_{l=1}^{n} b(i,l)b_k(l,j) \ge \sum_{l=i+1}^{\min(i+n-s,n)} b(i,l)b_k(i,j) \\ & + \sum_{l=1}^{i-s} b(i,l)b_k(l,j), \;\; k \ge 1. \end{aligned} \tag{A.8}$$

By (A.7) and (A.8), we can prove by induction on $k \ge 1$ that

$$b_k(i,j) \neq 0 \tag{A.9}$$

for all (i,j) satisfying either $i+1 \le j \le \min(i+k(n-s), n)$ or $j+ks \le i$. This proves (A.6) for all $k \ge n/s$. But since $\mathbf{B}$ is row stochastic then eigenvalue one is the unique eigenvalue on the unit circle centered at the origin and this completes the proof.

A.4. Proof of Proposition 3.2. By (2.12) and (3.5), we have

$$\mathbf{P}\mathbf{A}_{\text{sde}} = \mathbf{A}_{\text{sde}}\mathbf{P} = \mathbf{P} \;\text{ and }\; \mathbf{P}^2 = \mathbf{P}. \tag{A.10}$$

This together with (2.11) implies that

$$\tilde{\mathbf{z}}(k+1) = (\mathbf{A}_{\text{sde}} - \mathbf{P})\tilde{\mathbf{z}}(k) - \alpha_k(\mathbf{A}_{\text{sde}} - \mathbf{P})\mathbf{h}(k). \tag{A.11}$$

Applying (A.11) repeatedly yields

$$\tilde{\mathbf{z}}(k) = (\mathbf{A}_{\text{sde}} - \mathbf{P})^k\tilde{\mathbf{z}}(0) - \sum_{l=0}^{k-1} \alpha_l(\mathbf{A}_{\text{sde}} - \mathbf{P})^{k-l}\mathbf{h}(l), \;\; k \ge 1. \tag{A.12}$$

Denote the spectrum of a square matrix $\mathbf{A}$ by $\sigma(\mathbf{A})$. By the assumption on the matrix $\mathbf{A}_{\rm sde}$,

$$\sigma(\mathbf{A}_{\rm sde}) \subset \{1\} \cup \{z,\ |z| < 1\}, \tag{A.13}$$

and the eigenspace associated with eigenvalue one is given by

$$N(\mathbf{A}_{\rm sde} - \mathbf{I}) = \operatorname{span}\{\mathbf{1}_{2n}\}. \tag{A.14}$$

Combining (3.5), (A.10), (A.13) and (A.14), we obtain

$$\sigma(\mathbf{A}_{\rm sde} - \mathbf{P}) = \big(\sigma(\mathbf{A}_{\rm sde})\backslash\{1\}\big) \cup \{0\} \subset \{z,\ |z| \le |\lambda_2(\mathbf{A}_{\rm sde})|\}. \tag{A.15}$$

Therefore there exists a positive constant C_1 such that

$$\|(\mathbf{A}_{\rm sde} - \mathbf{P})^k\|_{\mathcal{B}^\infty} \le C_1\Big(\frac{1 + 2|\lambda_2(\mathbf{A}_{\rm sde})|}{3}\Big)^k,\ k \ge 1, \tag{A.16}$$

where $\|\mathbf{A}\|_{\mathcal{B}^\infty} = \sup_{\|\mathbf{x}\|_\infty = 1} \|\mathbf{A}\mathbf{x}\|_\infty$.

By (1.7), (2.9), (2.10) and (3.1), we have

$$\|\mathbf{h}(k)\|_{2,\infty} \le \sup_{1 \le i \le n} \sup_{\mathbf{x} \in \mathbb{R}^N} \|\nabla g_i(\mathbf{x})\|_2 \le M. \tag{A.17}$$

Then combining (A.12), (A.16) and (A.17) completes the proof.

A.5. Proof of Lemma 4.3. By (A.10), we have

$$(\mathbf{A}_{\rm sde} - \mathbf{P})^k = \mathbf{A}_{\rm sde}^k - \mathbf{P},\ k \ge 1.$$

Therefore

$$\lim_{k \to \infty} \mathbf{A}_{\rm sde}^k = \mathbf{P}$$

by (A.16). This, together with the observation that all entries of $\mathbf{A}_{\rm sde}^k, k \ge 1$ are nonnegative. Hence the required estimate implies that all entries of $\mathbf{a}_{\rm sde}$ are nonnegative and the desired estimate (4.7) follows.

A.6. Proof of Proposition 4.4. By (2.9), (3.9) and (4.1), we have

$$\begin{aligned} \|\mathbf{h}_i(k) - \nabla g_i(\bar{\mathbf{x}}(k))\|_2 &= \|\nabla g_i(\mathbf{z}_i(k)) - \nabla g_i(\bar{\mathbf{x}}(k))\|_2 \\ &\le L\|\mathbf{z}_i(k) - \bar{\mathbf{x}}(k)\|_2 \le L\|\tilde{\mathbf{z}}(k)\|_{2,\infty} \end{aligned}$$

for $1 \le i \le n$, and

$$\|\mathbf{h}_i(k) + \nabla g_{i-n}(\bar{\mathbf{x}}(k))\|_2 \le L\|\tilde{\mathbf{z}}(k)\|_{2,\infty}$$

for $n + 1 \le i \le 2n$. Therefore

$$\|\mathbf{h}(k) - \tilde{\mathbf{h}}(k)\|_{2,\infty} \le L\|\tilde{\mathbf{z}}(k)\|_{2,\infty}, \tag{A.18}$$

where

$$\tilde{\mathbf{h}}(k) = \begin{pmatrix} \nabla G(\bar{\mathbf{x}}(k)) \\ -\nabla G(\bar{\mathbf{x}}(k)) \end{pmatrix} \quad \text{and} \quad \nabla G(\mathbf{x}) = \begin{pmatrix} \nabla g_1(\mathbf{x}) \\ \vdots \\ \nabla g_n(\mathbf{x}) \end{pmatrix}.$$

Therefore

$$\|\mathbf{a}_{\rm sde}^T\mathbf{h}(k) - \mathbf{a}_{\rm sde}^T\tilde{\mathbf{h}}(k)\|_2 \le \|\mathbf{a}_{\rm sde}\|_1\|\mathbf{h}(k) - \tilde{\mathbf{h}}(k)\|_{2,\infty} \le L\|\mathbf{a}_{\rm sde}\|_1\|\tilde{\mathbf{z}}(k)\|_{2,\infty}, \tag{A.19}$$

where the second estimate follows from (A.18).

Observe from (1.9) that

$$\mathbf{A}_{\rm sde}\tilde{\mathbf{h}}(k) = \nabla f(\bar{\mathbf{x}}(k))\mathbf{w},$$

which together with (2.12) implies that

$$\mathbf{a}_{\rm sde}^T\tilde{\mathbf{h}}(k)=\mathbf{a}_{\rm sde}^T\mathbf{A}_{\rm sde}\tilde{\mathbf{h}}(k)=\tilde{w}\nabla f(\bar{\mathbf{x}}(k)). \tag{A.20}$$

Combining (A.19) and (A.20) proves the desired estimate (4.9).

A.7. Proof of Proposition 4.5. Set

$$\beta_k=M^2\alpha_k^2+L^2\|\tilde{\mathbf{z}}(k)\|_{2,\infty}^2,\ k\ge 0. \tag{A.21}$$

By (2.9), (3.1), (4.8) and (4.9), we obtain

$$\begin{aligned}\|\bar{\mathbf{x}}(k+1)-\bar{\mathbf{x}}\|_2^2 &= \|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2+\alpha_k^2\|\mathbf{a}_{\rm sde}^T\mathbf{h}(k)\|_2^2-2\alpha_k\langle\mathbf{a}_{\rm sde}^T\mathbf{h}(k),\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\rangle_N\\ &= \|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2+\alpha_k^2\|\mathbf{a}_{\rm sde}^T\mathbf{h}(k)\|_2^2-2\alpha_k\langle\mathbf{a}_{\rm sde}^T\tilde{\mathbf{h}}(k),\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\rangle_N\\ &\quad -2\alpha_k\langle\mathbf{a}_{\rm sde}^T\mathbf{h}(k)-\mathbf{a}_{\rm sde}^T\tilde{\mathbf{h}}(k),\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\rangle_N\\ &\le \|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2+M^2\|\mathbf{a}_{\rm sde}\|_1^2\alpha_k^2\\ &\quad +2L\|\mathbf{a}_{\rm sde}\|_1\alpha_k\|\tilde{\mathbf{z}}(k)\|_{2,\infty}\|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2\\ &= \|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2+M^2\alpha_k^2+2L\alpha_k\|\tilde{\mathbf{z}}(k)\|_{2,\infty}\|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2\\ &\le (1+\alpha_k^2)\|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2+\beta_k,\end{aligned} \tag{A.22}$$

where we also use the positivity of $\tilde{w}$ in (4.7), $\|\mathbf{a}_{\rm sde}\|_1=1$ and the convexity of the objective function f,

$$\langle\nabla f(\mathbf{x}),\mathbf{x}-\bar{\mathbf{x}}\rangle_N\ge 0,\ \mathbf{x}\in\mathbb{R}^N. \tag{A.23}$$

Applying (A.22) repeatedly, we get

$$\begin{aligned}\|\bar{\mathbf{x}}(k+1)-\bar{\mathbf{x}}\|_2^2 &\le \prod_{j=0}^k(1+\alpha_j^2)\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2+\beta_k+\sum_{j=0}^{k-1}\beta_j\prod_{j'=j+1}^k(1+\alpha_{j'}^2)\\ &\le \exp\Big(\sum_{j=0}^k\alpha_j^2\Big)\Big(\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2+\sum_{j=0}^k\beta_j\Big)\\ &\le \exp\Big(\sum_{j=0}^\infty\alpha_j^2\Big)\Big(\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2+\sum_{j=0}^\infty\beta_j\Big),\quad k\ge 1.\end{aligned} \tag{A.24}$$

This together with (2.6) and (3.13) proves the desired bound (4.10) for the sequence $\bar{\mathbf{x}}(k), k\ge 0$.

A.8. Proof of Proposition 4.6. By (2.6), without a loss of generality, we assume that

$$0\le\alpha_k\le\tilde{w}A\ \text{ for all }\ k\ge 0.$$

Then for $k\ge 1$, following the argument in (A.22) with the convexity (A.23) replaced by the strong converxity (4.2), we obtain

$$\|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2\le(1-\tilde{w}A\alpha_{k-1})\|\bar{\mathbf{x}}(k-1)-\bar{\mathbf{x}}\|_2^2+\beta_{k-1}, \tag{A.25}$$

cf. (A.22). Applying the above estimate repeatedly leads to

$$\|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2 \le \prod_{j=0}^{k-1}(1-\tilde{w}A\alpha_j)\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2+\beta_{k-1}+\sum_{j=0}^{k-2}\beta_j\prod_{j'=j+1}^{k-1}(1-\tilde{w}A\alpha_{j'})$$
$$\le \exp\Big(-\tilde{w}A\sum_{j=0}^{k-1}\alpha_j\Big)\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2+\beta_{k-1}+\sum_{j=0}^{k-2}\exp\Big(-\tilde{w}A\sum_{j=j+1}^{k-1}\alpha_j\Big)\beta_j,$$

where $\beta_j, j\ge 0$, is given in (A.21). This proves the desired estimate (4.11).

A.9. Proof of Corollary 4.7. Applying (4.11) for the illustrative examples (1.4) of step sizes, and using (3.11), we can find a positive constant $\tilde{C}$ such that

$$\begin{aligned}\|\bar{\mathbf{x}}(k)-\bar{\mathbf{x}}\|_2^2 &\le \exp\Big(-\tilde{w}A\sum_{l=0}^{k-1}(l+a)^{-\theta}\Big)\Big\{\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2\\ &\quad+(M^2+L^2C^2)\sum_{j=0}^{k-1}\exp\Big(\tilde{w}A\sum_{l=0}^{j}(l+a)^{-\theta}\Big)(j+a)^{-2\theta}\Big\}\\ &\le \exp\Big(-\frac{\tilde{w}A}{1-\theta}\big((k+a)^{1-\theta}-a^{1-\theta}\big)\Big)\|\bar{\mathbf{x}}(0)-\bar{\mathbf{x}}\|_2^2\\ &\quad+(M^2+L^2C^2)\exp\Big(-\frac{\tilde{w}A}{1-\theta}(k+a)^{1-\theta}\Big)\\ &\quad\times\sum_{j=0}^{k-1}\exp\Big(\frac{\tilde{w}A}{1-\theta}(j+1+a)^{1-\theta}\Big)(j+a)^{-2\theta}\le\tilde{C}(k+a)^{-\theta}\end{aligned}\tag{A.26}$$

hold for all $k\ge 1$, where the first estimate holds by (4.11), the second one follows from the observation

$$\sum_{l=m}^{k-1}(l+a)^{-\theta}\ge\sum_{l=m}^{k-1}\int_l^{l+1}(x+a)^{-\theta}dx=\frac{(k+a)^{1-\theta}-(m+a)^{1-\theta}}{1-\theta}$$

for $0\le m\le k-1$, and the third one is obtained from the boundness of the sequences $(k+a)^\theta\exp(-\frac{\tilde{w}A}{1-\theta}(k+a)^{1-\theta})$ and $(k+a+1)^{1-\theta}-(k+a)^{1-\theta}, k\ge 0$, and the intergrability

of $\int_0^\infty \exp(-\frac{\tilde{w}A}{1-\theta}t)(t+1)^{\theta/(1-\theta)}dt$, and the following estimate

$$\begin{aligned}
&\sum_{j=0}^{k-1} \exp\Big(\frac{\tilde{w}A}{1-\theta}(j+1+a)^{1-\theta}\Big)(j+a)^{-2\theta} \\
&\le 3^{2\theta}\int_0^k \exp\Big(\frac{\tilde{w}A}{1-\theta}(x+a+1)^{1-\theta}\Big)(x+a+1)^{-2\theta}dx \\
&\le 3^{2\theta}(1-\theta)^{-1}\int_0^{a_k} \exp\Big(\frac{\tilde{w}A}{1-\theta}\big((k+a+1)^{1-\theta}-t\big)\Big) \\
&\quad\times\big((k+a+1)^{1-\theta}-t\big)^{-\theta/(1-\theta)}dt \\
&\le 3^{2\theta}(1-\theta)^{-1}\exp\Big(\frac{\tilde{w}A}{1-\theta}(k+a+1)^{1-\theta}\Big)(k+a+1)^{-\theta} \\
&\quad\times\int_0^{a_k} \exp\Big(-\frac{\tilde{w}A}{1-\theta}t\Big)(t+1)^{\theta/(1-\theta)}dt \\
&\le 3^{2\theta}(1-\theta)^{-1}\exp\Big(\frac{\tilde{w}A}{1-\theta}(k+a+1)^{1-\theta}\Big)(k+a)^{-\theta} \\
&\quad\times\int_0^{\infty} \exp\Big(-\frac{\tilde{w}A}{1-\theta}t\Big)(t+1)^{\theta/(1-\theta)}dt,
\end{aligned}$$

where $a_k = (k+1+a)^{1-\theta} - (a+1)^{1-\theta} \le (k+1+a)^{1-\theta} - 1$. Then the desired conclusion (4.13) follows by taking square roots of both sides of the above estimate (A.26).

References

[1] E. Atallah and N. Rahnavard, A code-based distributed gradient descent method, In: Proceedings of the 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 02–05 October 2018, Monticello, IL, USA: IEEE, pp. 951–958.

[2] A. S. Bedi and K. Rajawat, Asynchronous incremental stochastic dual descent algorithm for network resource allocation, IEEE Trans. Signal Process., 66 (2018), 2229–2244.

[3] D. P. Bertsekas, *Incremental gradient, subgradient, and proximal methods for convex optimization: a survey*, Optimization for Machine Learning, S. Sra, S. Nowozin, and S. J. Wright, eds., 85–119, Neural Information Processing Series, MIT Press, Cambridge, MA, 2012.

[4] L. Bottou, *Stochastic gradient descent tricks*, Neural Networks: Tricks of the Trade, G. Montavon, G. B. Orr, K.-R. Muller eds., Second edition, 421–436, Springer Berlin, Heidelberg, 2012.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2011), 1–122.

[6] F. S. Cattivelli and A. H. Sayed, Diffusion LMS strategies for distributed estimation, IEEE Trans. Signal Process., 58 (2010), 1035–1048.

[7] C. Cheng, Y. Jiang and Q. Sun, Spatially distributed sampling and reconstruction, Appl. Comput. Harmon. Anal., 47 (2019), 109–148.

[8] C. R. Da Silva, B. Choi and K. Kim, Distributed spectrum sensing for cognitive radio systems, In: Proceedings of the 2007 Information Theory and Applications Workshop, 29 January–02 February 2007, La Jolla, CA, USA: IEEE, pp. 120–123.

[9] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng,

Large scale distributed deep networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), 03–06 December 2012, Lake Tahoe Nevada, pp. 1223–1231.

[10] N. Emirov, G. Song and Q. Sun, A divide-and-conquer algorithm for distributed optimization on networks, Appl. Comput. Harmon. Anal., 70 (2024), Paper No. 101623, 19 pp.

[11] J. H. Friedman, Stochastic gradient boosting, Comput. Stat. Data Anal., 38 (2002), 367–378.

[12] D. Fu, L. Han, L. Liu, Q. Gao, and Z. Feng, An efficient centralized algorithm for connected dominating set on wireless networks, Procedia Comput. Sci., 56 (2015), 162–167.

[13] M. Glasgow and M. Wootters, Approximate gradient coding with optimal decoding, IEEE J. Sel. Topics Inf. Theory, 2 (2021), 855–866.

[14] M. Gürbüzbalaban, A. Ozdaglar and P. Parrilo, Convergence rate of incremental gradient and incremental newton methods, SIAM J. Optim., 29 (2019), 2542–2565.

[15] W. Halbawi, N. Azizan-Ruhi, F. Salehi, and B. Hassibi, Improving distributed gradient descent using reed-solomon codes, In: Proceedings of the 2018 IEEE International Symposium on Information Theory (ISIT), 17-22 June 2018, Vail, CO, USA: IEEE, pp. 2027–2031.

[16] M. Hardt, B. Recht and Y. Singer, Train faster, generalize better: Stability of stochastic gradient descent, In: Proceedings of the 33rd International Conference on Machine Learning, 19–24 June 2016, New York, USA, pp. 1225–1234.

[17] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, and E. P. Xing, More effective distributed ml via a stale synchronous parallel parameter server, In: Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13), 05–10 December 2013, Lake Tahoe Nevada, pp. 1223–1231.

[18] J. Jiang, C. Cheng and Q. Sun, Nonsubsampled graph filter banks: theory and distributed algorithms, IEEE Trans. Signal Process., 67 (2019), 3938–3953.

[19] B. Johansson, On distributed optimization in networked systems, PhD. thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2008.

[20] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, In: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 07–09 May 2015, San Diego, 15 pp.

[21] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, Communication efficient distributed machine learning with the parameter server, In: Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14), 08–13 December 2014, Montreal, Canada, pp. 19–27.

[22] V. J. Mathews, and Z. Xie, A stochastic gradient adaptive filter with gradient adaptive step size, IEEE Trans. Signal Process., 41 (1993), 2075–2087.

[23] A. Nedić, A. Ozdaglar and P. A. Parrilo, Constrained consensus and optimization in multi-agent networks, IEEE Trans. Automat. Control, 55 (2010), no. 4, 922–938.

[24] A. Nedić and D. Bertsekas, *Convergence rate of incremental subgradient algorithms*, Stochastic optimization: algorithms and applications, (Gainesville, FL, 2000), 223–264, Appl. Optim., 54, Kluwer Academic Publishers, Dordrecht, 2001.

[25] A. Nedić, Convergence rate of distributed averaging dynamics and optimization in networks, Found. Trends Syst. Control., 2 (2015), 1–100.

[26] D. Needell, R. Ward and N. Srebro, Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm, In: Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14), 08–13 December 2014, Montreal, Canada, pp. 1017–1025.

[27] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis, Gradient coding from cyclic mds codes and expander graphs. In: Proceedings of the 35th International Conference on Machine Learning (ICML 2018), 10–15 July 2018, Stockholm, Sweden, pp. 4305–4313.
[28] H. Robbins, and S. Monro, A stochastic approximation method, Ann. Math. Statist., 22 (1951), 400–407.
[29] A. H. Sayed, Adaptation, learning, and optimization over networks, Found. Trends Mach. Learn., 7 (2014), 311–801.
[30] A. H. Sayed, S. Barbarossa, S. Theodoridis, and I. Yamada, Adaptation and learning over complex networks, IEEE Signal Process. Mag., 30 (2013), 14–15.
[31] N. N. Schraudolph, Local gain adaptation in stochastic gradient descent, In: Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks (ICANN 99), 07–10 September 1999, Edinburgh, UK, pp. 569–574.
[32] N. Takahashi, I. Yamada and A. H. Sayed, Diffusion least-mean squares with adaptive combiners: formulation and performance analysis, IEEE Trans. Signal Process., 58 (2010), 4795–4810.
[33] C. Tan, S. Ma, Y.-H. Dai, and Y. Qian, Barzilai-borwein step size for stochastic gradient descent, In: Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS'16), Barcelona, Spain, 05–10 December 2016, pp. 685–693.
[34] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, Gradient coding: avoiding stragglers in distributed learning, In: Proceedings of the 34th International Conference on Machine Learning (ICML'17)-Volume 70, 06–11 August 2017, Sydney, NSW, Australia, pp. 3368–3376.
[35] K. Yuan, Q. Ling and W. Yin, On the convergence of decentralized gradient descent, SIAM J. Optim., 26 (2016), 1835–1854.
[36] M. D. Zeiler, ADADELTA: an adaptive learning rate method, eprint arXiv:1212.5701v1 [cs.LG], Dec 2012.

(Elie Atallah) Department of Electrical and Computer Engineering, University of Central Florida, Orlando, Florda
E-mail address: elieatallah@knights.ucf.edu

(Nazanin Rahnavard) Department of Electrical and Computer Engineering, University of Central Florida, Orlando, Florda
E-mail address: nazanin@ece.ucf.edu

(Qiyu Sun) Department of Mathematics, University of Central Florida, Orlando, Florda
E-mail address: qiyu.sun@ucf.edu