A Code-based Distributed Gradient Descent Scheme for Convex Optimization over Networks

Elie Atallah, Nazanin Rahnavard, and Qiyu Sun
Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL
Department of Mathematics, University of Central Florida, Orlando, FL
Emails: elieatallah@knights.ucf.edu, nazanin@ece.ucf.edu, Qiyu.Sun@ucf.edu

***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 also provide the consensus and convergence analysis for the CoDGraD algorithm and we demonstrate its performance via numerical simulations.**

***Index Terms*—distributed optimization, gradient coding, consensus, distributed networks**

I. 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 [1, 2, 3, 4, 5, 6, 7]. The objective functions f in such optimization problems,

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

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

The optimization problem

$$\bar{\mathbf{x}}=\arg\min_{\mathbf{x}\in\mathbb{R}^N}\sum_{l=1}^{m} f_l(\mathbf{x}), \tag{2}$$

associated with the objective function $f=\sum_{l=1}^m f_l$ has been well studied, see [8, 9] and references therein for various algorithms implementable in a strong fusion center or in local workers distributed over the network [1, 4, 10, 11, 12, 13, 14]. 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 (2) is the *gradient descent* algorithm,

$$\mathbf{x}(k+1)=\mathbf{x}(k)-\frac{\alpha_k}{m}\sum_{l=1}^{m}\nabla f_l(\mathbf{x}(k)),\ k\ge 0, \tag{3}$$

where $\{\alpha_k\}_{k=0}^\infty$ is a positive sequence chosen appropriately [15, 17, 18, 19, 20, 21, 22, 23]. Our illustrate examples of step sizes $\alpha_k, k\ge 0$, are

$$\alpha_k=(k+a)^{-\theta},\ k\ge 0, \tag{4}$$

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

Several distributed versions of the gradient descent algorithm (3) have been proposed, including the Adapt-Then-Combine algorithm (ATC) and the Combine-Then-Adapt algorithm (CTA) [11, 13], where implementation of the worker at region l is given by

$$\begin{cases}\mathbf{y}_l(k)=\mathbf{x}_l(k)-\alpha_k\nabla f_l(\mathbf{x}_l(k))\\ \mathbf{x}_l(k+1)=\sum_{j\in\mathcal{N}_l} w_{lj}(k)\mathbf{y}_j(k)\end{cases} \tag{5}$$

and

$$\begin{cases}\mathbf{y}_l(k)=\sum_{j\in\mathcal{N}_l} w_{lj}(k)\mathbf{x}_j(k)\\ \mathbf{x}_l(k+1)=\mathbf{y}_l(k)-\alpha_k\nabla f_l(\mathbf{y}_l(k))\end{cases} \tag{6}$$

respectively, where $\mathcal{N}_l$ contains all adjacent regions of the region l for data sharing, and $(w_{lj})_{1\le l,j\le m}$ is a consensus matrix. The above ATC and CTA algorithms may reach consensus over all nodes to the optimal solution $\bar{\mathbf{x}}$ in (2). They are essentially the same as the gradient descent algorithm (3) if the graph to describe topological structure of the network is complete and the consensus weights $w_{lj}=1/m, 1\le l,j\le m$. However, comparing with the gradient descent algorithm (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 $\nabla f_l, 1\le l\le m$, 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 [24, 25, 27]. To resolve the above problem, uncoded and coded local gradients have been proposed in [28, 29, 30] 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 $n \le m$ is the number of active nodes. In this paper, we follow the coded scheme in [28] and consider the paradigm that for every active node i, the global objective function can be recovered from coded objective functions g_j on non-stragglers relative to node i, i.e.,

$$f(\mathbf{x}) = \sum_{j=1}^{n} a(i,j) g_j(\mathbf{x}) \tag{7}$$

for some decoding matrix

$$\mathbf{A} = (a(i,j))_{1\le i,j\le n}.$$

The above requirement is met if the coded scheme for non-stragglers is given by

$$g_i(\mathbf{x}) = \sum_{l=1}^{m} b(i,l) f_l(\mathbf{x}),\ 1 \le i \le n, \tag{8}$$

and the coding matrix

$$\mathbf{B} = (b(i,l))_{1\le i\le n, 1\le l\le m}$$

satisfies

$$\mathbf{AB} = \mathbf{1}_{n\times m}, \tag{9}$$

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

The coded scheme (7) and (8) has been used in [28] to solve the global optimization problem (2), where the worker at each region evaluates the coded local gradients and sends them to the worker at the master node; then the worker at the master node takes weighted sum of coded local gradients to reach gradient of the global objective function, and applies a centralized gradient descent approach similar to (3) to update the approximation to the optimal solution $\bar{\mathbf{x}}$ in (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 coded scheme (7) and (8), we propose a *distributed* algorithm to solve the convex optimization problem (2).

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 $\mathbf{1}_{n\times m}$, and the column vector of size n with all entries taking value 1 by $\mathbf{1}_n$. 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,

The remainder of the paper is organized as follows. In Section II, we formulate our code-based distributed gradient descent algorithm (CoDGraD) to solve the optimization problem (2). Then in Sections III and IV, we consider the consensus and convergence properties of the proposed CoDGraD algorithm. Afterwards, we demonstrate the performance of CoDGraD through simulation in Section V. We conclude the paper in Section VI.

II. 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 coded scheme (7) and (8). Set decoding weights

$$w_i = \Big(\sum_{k\in\Gamma_i} |a(i,k)|\Big)^{-1},\ 1 \le i \le n, \tag{10}$$

where

$$\Gamma_i = \{j,\ a(i,j) \ne 0\}. \tag{11}$$

In [26, 28], all workers not in Γ_i are considered as "stragglers" for an active node i, since coded information at those workers are not used to evaluate the gradient ∇f of the global objective function f at the node i. We remark that in this paper active node are those workers over the network that don't witness delay or failure on computing or communicating while non-stragglers of an active node i are those active nodes that are in Γ_i.

To solve the optimization problem (2), we propose an iterative distributed algorithm with the implementation of the worker at the region i given by

$$\begin{cases} \mathbf{v}_i(k) = \nabla g_i(\mathbf{x}_i(k)), \\ \mathbf{y}_i^+(k) = \mathbf{x}_i(k) - \alpha_k \mathbf{v}_i(k), \\ \mathbf{y}_i^-(k) = \mathbf{x}_i(k) + \alpha_k \mathbf{v}_i(k), \\ \mathbf{x}_i(k+1) \ = \sum_{j\in\Gamma_i} w_i\big\{(a(i,j))_+ \mathbf{y}_j^+(k) \\ \qquad\qquad +(-a(i,j))_+ \mathbf{y}_j^-(k)\big\}, \quad k \ge 0, \end{cases} \tag{12}$$

where initials $\mathbf{x}_i(0)$ are chosen randomly or set zero initially, and step sizes $\alpha_k, k \ge 0$, [12, 16, 19] are so chosen that

$$\sum_{k=0}^{\infty} \alpha_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty. \tag{13}$$

We call this algorithm as a code-based distributed gradient descent algorithm and use CoDGraD for abbreviation. The implementation of the CoDGraD is described in Algorithm 1.

Algorithm 1 The CoDGraD Algorithm

```
Input: Set tolerance value $\epsilon$ 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 $i$.
while $e_i(k) \ge \epsilon$ do {Halting is done at each node independently with no coordination}
    At the worker $i$, use (12) to update the estimate $\mathbf{x}_i(k)$.
    Find error $e_i(k+1) = \|\mathbf{x}_i(k+1) - \mathbf{x}_i(k)\|$ for all $1 \le i \le n$
    $k = k + 1$
end while
Output: $\mathbf{x}_i(k)$ and $k$.
```

For our purpose, we consider the coded scheme (7) and (8) that matches our network topology. Here the network topology is described 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. Therefore the topological matching property of the coded scheme (7) and (8) 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 neighboring non-straggling node of any active node i 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 (11) and $\mathcal{N}_i = \{j : (i,j) \in E\} \cup \{i\}$.

For the above scenario of the coded scheme (7) and (8), the active worker at each region first evaluates the coded local gradient, then it shares the coded local gradient with non-straggling active workers at the adjacent regions, next it updates the approximation to the optimal solution $\bar{\mathbf{x}}$, and finally it shares the updated approximation to active workers at the adjacent region for the next iteration. Hence the CoDGradD algorithm is implementable in the network that does not have a fusion center at all.

For the decoding matrix $\mathbf{A}$ in (7), 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

$$\tilde{a}(i,j) = w_i a(i,j),\ 1 \le i,j \le n, \tag{14}$$

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

$$\mathbf{A}_{\rm sde} = \begin{pmatrix} \tilde{\mathbf{A}}_{+} & \tilde{\mathbf{A}}_{-} \\ \tilde{\mathbf{A}}_{+} & \tilde{\mathbf{A}}_{-} \end{pmatrix}, \tag{15}$$

where $w_i, 1 \le i \le n$, are decoding weights given in (10), and

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

are positive/negative parts of the normalized decoding matrix $\tilde{\mathbf{A}} = (\tilde{a}(i,j))_{1\le i,j\le n}$ respectively. In this paper, we consider the consensus and convergence properties of $\mathbf{x}_i(k), k \ge 0$, in the proposed CoDGraD algorithm (12) when the row stochastic matrix $\mathbf{A}_{\rm sde}$ has simple eigenvalue one and the global objective function f is strongly convex.

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

In this section, we consider the consensus property of $\mathbf{x}_i(k),\ k \ge 1$, in the CoDGraD algorithm (12).

Theorem 1. *Let $\mathbf{x}_i(k),\ k \ge 1$, be in the CoDGraD algorithm* (12). *If the row stochastic matrix $\mathbf{A}_{\rm sde}$ in* (15) *has simple eigenvalue one, the sequence $\{\alpha_k\}_{k=0}^{\infty}$ satisfies* (13), *and the local objective functions $g_i, 1 \le i \le n$, have bounded gradients, i.e., there exists a positive constant M such that*

$$\|\nabla g_i(\mathbf{x})\|_2 \le M,\ \mathbf{x} \in \mathbb{R}^N, \tag{16}$$

then

$$\lim_{k\to\infty} (\mathbf{x}_i(k) - \mathbf{x}_j(k)) = 0 \tag{17}$$

and

$$\lim_{k\to\infty} (f(\mathbf{x}_i(k)) - f(\mathbf{x}_j(k))) = 0 \tag{18}$$

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

Let the row stochastic matrix $\mathbf{A}_{\rm sde}$ in (15) have simple eigenvalue one and $\lambda_m(\mathbf{A}_{\rm sde}), 1 \le m \le 2n$, be its eigenvalues listed in the order that

$$1 = \lambda_1(\mathbf{A}_{\rm sde}) > |\lambda_2(\mathbf{A}_{\rm sde})| \ge \ldots \ge |\lambda_{2n}(\mathbf{A}_{\rm sde})|. \tag{19}$$

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

$$\mathbf{z}_i(k) = \mathbf{z}_{i+n}(k) = \mathbf{x}_i(k),\ k \ge 0. \tag{20}$$

Then the CoDGraD algorithm (12) can be rewritten as

$$\mathbf{z}_i(k+1) = \sum_{j=1}^{2n} q(i,j)\big(\mathbf{z}_j(k) - \alpha_k \mathbf{h}_j(k)\big),\ 1 \le i \le 2n, \tag{21}$$

where

$$\mathbf{h}_i(k) = \begin{cases} \nabla g_i(\mathbf{z}_i(k)) & \text{if } 1 \le i \le n \\ -\nabla g_{i-n}(\mathbf{z}_i(k)) & \text{if } n+1 \le i \le 2n. \end{cases} \tag{22}$$

Set

$$\mathbf{z}(k) := \big(\mathbf{z}_i(k)\big)_{1\le i\le 2n} \;\text{ and }\; \mathbf{h}(k) := \big(\mathbf{h}_i(k)\big)_{1\le i\le 2n} \tag{23}$$

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 (21) can be reformulated in a matrix form:

$$\mathbf{z}(k+1) = \mathbf{A}_{\rm sde}\mathbf{z}(k) - \alpha_k \mathbf{A}_{\rm sde}\mathbf{h}(k),\ k \ge 0. \tag{24}$$

Define

$$\tilde{\mathbf{z}}(k) = \mathbf{z}(k) - \mathbf{P}\mathbf{z}(k),\ k \ge 0, \tag{25}$$

where

$$\mathbf{P} = \mathbf{1}_{2n}(\mathbf{a}_{\rm sde})^T \tag{26}$$

and $\mathbf{a}_{\rm sde}$ is 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

$$\mathbf{a}_{\rm sde}^T \mathbf{A}_{\rm sde} = \mathbf{a}_{\rm sde}^T \;\text{ and }\; \mathbf{a}_{\rm sde}^T \mathbf{1}_{2n} = 1. \tag{27}$$

Then the consensus property (17) reduces to establishing

$$\lim_{k\to\infty} \|\tilde{\mathbf{z}}(k)\|_{2,\infty} = 0, \tag{28}$$

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 $\mathbf{z}_i \in \mathbb{R}^N, 1 \le i \le 2n$.

By (26) and (27), we have

$$\mathbf{P}\mathbf{A}_{\rm sde} = \mathbf{A}_{\rm sde}\mathbf{P} = \mathbf{P} \;\text{ and }\; \mathbf{P}^2 = \mathbf{P}. \tag{29}$$

This together with (24) implies that

$$\tilde{\mathbf{z}}(k+1) = (\mathbf{A}_{\rm sde} - \mathbf{P})\tilde{\mathbf{z}}(k) - \alpha_k (\mathbf{A}_{\rm sde} - \mathbf{P})\mathbf{h}(k). \tag{30}$$

Applying (30) repeatedly yields

$$\tilde{\mathbf{z}}(k) = (\mathbf{A}_{\rm sde} - \mathbf{P})^k \tilde{\mathbf{z}}(0) - \sum_{l=0}^{k-1} \alpha_l (\mathbf{A}_{\rm sde} - \mathbf{P})^{k-l}\mathbf{h}(l),\ k \ge 1. \tag{31}$$

Therefore, we have the following estimate for $\|\tilde{\mathbf{z}}(k)\|_{2,\infty}, k \ge 1$ in Proposition 1, see Appendix A for a detailed proof.

Proposition 1. *Let $\mathbf{A}_{\rm sde}$, $\lambda_2(\mathbf{A}_{\rm sde})$ and $\mathbf{P}$ be as in* (15), (19) *and* (26) *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* (16). *Then there exists a positive constant C_1 such that*

$$\begin{aligned}\|\tilde{\mathbf{z}}(k)\|_{2,\infty} \ \le\ & C_1 M \sum_{l=0}^{k-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l} \alpha_l \\ & + C_1 \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k} \|\tilde{\mathbf{z}}(0)\|_{2,\infty} \end{aligned} \quad (32)$$

hold for all $k \ge 1$.

By (26), we can write

$$\mathbf{P}\mathbf{z}(k) = \bar{\mathbf{x}}(k)\mathbf{1}_{2n}, \quad (33)$$

for some $\bar{\mathbf{x}}(k) \in \mathbb{R}^N$. 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 1 we obtain the following estimate about the consensus property of $\mathbf{x}_i(k)$ for different $1 \le i \le n$:

$$\begin{aligned} & \max_{1\le i\le n} \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|_2 \\ & \le C_1 \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k} \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2 \\ & \quad + C_1 M \sum_{l=0}^{k-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l} \alpha_l, \quad k \ge 1. \end{aligned} \quad (34)$$

Applying (34) to our illustrate example (4) of step sizes, we can find a postive constant C such that

$$\max_{1\le i\le n} \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|_2 \le C(k+a)^{-\theta}, \quad k \ge 0. \quad (35)$$

We finish this section with the proof of Theorem 1.

Proof of Theorem 1. By (19) and (32),

$$\begin{aligned} & \Big(\sum_{k=0}^{\infty} \max_{1\le i\le n} \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|_2^2\Big)^{1/2} \\ \le\ & C_1 M \Big(\sum_{k=0}^{\infty} \Big(\sum_{l=0}^{k-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l} \alpha_l\Big)^2\Big)^{1/2} \\ & + C_1 \Big(\sum_{k=0}^{\infty} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{2k}\Big)^{1/2} \\ & \quad \times \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2 \\ \le\ & \frac{2C_1 M}{1-|\lambda_2(\mathbf{A}_{\rm sde}))|} \Big(\sum_{k=0}^{\infty} \alpha_k^2\Big)^{1/2} \\ & + \frac{4C_1}{1-|\lambda_2(\mathbf{A}_{\rm sde}))|} \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2. \end{aligned} \quad (36)$$

Therefore the limit in (17) follows as the sequence $\{\alpha_k\}_{k=0}^{\infty}$ is square summable by (13).

By (7) and (10), we have

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

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

$$\begin{aligned} & |f(\mathbf{x}_i(k)) - f(\bar{\mathbf{x}}(k))| \\ & \le C_1 M W^{-1} \sum_{l=0}^{k-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k-l} \alpha_l \\ & \quad + C_1 W^{-1} \Big(\frac{1+|\lambda_2(\mathbf{A}_{\rm sde})|}{2}\Big)^{k} \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2 \\ & \to 0 \ \text{ as } k \to \infty \end{aligned} \quad (37)$$

for all $1 \le i \le n$. Then (18) follows. □

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

In this section, we consider the convergence of $\mathbf{x}_i(k), \ k \ge 1$, in the CoDGraD algorithm (12),

$$\lim_{k\to\infty} \mathbf{x}_i(k) = \bar{\mathbf{x}}, \ 1 \le i \le n, \quad (38)$$

where $\bar{\mathbf{x}}$ is the solution of the optimization problem (2). By (17), (33) and (36), it suffices to show that $\bar{\mathbf{x}}(k), k \ge 1$, converges to $\bar{\mathbf{x}}$.

Theorem 2. *Assume that the row stochastic matrix $\mathbf{A}_{\rm sde}$ in* (15) *has simple eigenvalue one, the sequence $\{\alpha_k\}_{k=0}^{\infty}$ satisfies* (13), *the local objective functions $g_j, 1 \le j \le n$, satisfy* (16) *and*

$$\|\nabla g_i(x) - \nabla g_i(y)\|_2 \le L\|x-y\|_2, \quad x, y \in \mathbb{R}^N, 1 \le i \le n, \quad (39)$$

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

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle_N \ge A\|\mathbf{x}-\mathbf{y}\|_2^2 \quad (40)$$

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

$$\begin{aligned} C_2 :=\ & \exp\Big(\sum_{j=0}^{\infty} \alpha_j^2\Big) \Big\{ \|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 \\ & + \frac{32 C_1^2 L^2}{(1-|\lambda_2(\mathbf{A}_{\rm sde}))|)^2} \max_{1\le i\le n} \|\mathbf{x}_i(0) - \bar{\mathbf{x}}(0)\|_2^2 \\ & + \frac{M^2(1+8C_1^2)}{(1-|\lambda_2(\mathbf{A}_{\rm sde}))|)^2} \Big(\sum_{j=0}^{\infty} \alpha_j^2\Big) \Big\}. \end{aligned}$$

Then $\bar{\mathbf{x}}(k), k \ge 1$, in (33) *converges to the solution $\bar{\mathbf{x}}$ of the optimization problem* (2),

$$\lim_{k\to\infty} \bar{\mathbf{x}}(k) = \bar{\mathbf{x}}. \quad (41)$$

To prove Theorem 2, we need a technical lemma, which follows the probability property for the vector $\mathbf{a}_{\rm sde}$ in (27). For the completeness of this paper, we include a detailed proof in Appendix B.

Lemma 1. *Let $\mathbf{a}_{\rm sde}$ and $w_i, 1 \le i \le n$, be as in* (10) *and* (27) *respectively. Set*

$$\mathbf{w} = (w_1, \ldots, w_n, w_1, \ldots, w_n)^T \quad (42)$$

and

$$\tilde{w} = \mathbf{a}_{\rm sde}^T \mathbf{w}. \tag{43}$$

Then

$$0 < \min_{1\le i\le n} w_i \le \tilde{w} \le \max_{1\le i\le n} w_i. \tag{44}$$

By (26) and (33), we have

$$\bar{\mathbf{x}}(k) = \mathbf{a}_{\rm sde}^T \mathbf{z}(k). \tag{45}$$

This together with (24) leads to the following iterative algorithm for $\bar{\mathbf{x}}(k), k \ge 0$:

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k) - \alpha_k \mathbf{a}_{\rm sde}^T \mathbf{h}(k),\ k \ge 0. \tag{46}$$

For local objective functions $g_i, 1 \le i \le n$, with Lipschitz gradients (39), 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 C for a detailed proof.

Proposition 2. *Let $\mathbf{A}_{\rm sde}$, $\lambda_2(\mathbf{A}_{\rm sde})$, $\mathbf{P}$ and $\tilde{w}$ be as in* (15)*,* (19)*,* (26) *and* (43) *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* (39)*. Then*

$$\|\mathbf{a}_{\rm sde}^T \mathbf{h}(k) - \tilde{w}\nabla f(\bar{\mathbf{x}}(k))\|_2 \le L\|\mathbf{a}_{\rm sde}\|_1 \|\tilde{\mathbf{z}}(k)\|_{2,\infty}. \tag{47}$$

By Proposition 2, the iterative algorithm (46) can be considered as the gradient descent algorithm (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 D for a detailed proof.

Proposition 3. *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 2. If the global objective function f is convex, then*

$$\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2 \le C_2 \ \text{ for all } \ k \ge 0, \tag{48}$$

where C_2 is the constant in Theorem 2.

The estimate in (48) can be improved if the objective function f has the strongly convex property (40), see Appendix D for a detailed proof.

.

Proposition 4. *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 2. Then*

$$\begin{aligned}\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 \ \le\ & \exp\Big(-\tilde{w}A\sum_{j=0}^{k-1}\alpha_j\Big)\Big\{\|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 \\ & + \sum_{j=0}^{k-1}\exp\Big(\tilde{w}A\sum_{l=0}^{j}\alpha_l\Big)\Big(M^2\alpha_j^2 \\ & + L^2\max_{1\le i\le n}\|\mathbf{x}_i(j) - \bar{\mathbf{x}}(j)\|_2^2\Big)\Big\}\end{aligned} \tag{49}$$

hold for all $k \ge 2$.

Applying (49) for the illustrative examples (4) of step sizes, and using (35), we obtain that

$$\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2 \le C(k+a)^{-\theta/2}, \quad k \ge 1, \tag{50}$$

hold for some positive constant C, cf. the convergence rate $O(\log k/\sqrt{k})$ for the uncoded incremental gradient methods [1, 8].

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 (39) and the row stochastic property for the matrix $\mathbf{A}_{\rm sde}$. This together with Proposition 4 proves that

$$\begin{aligned}f(\bar{\mathbf{x}}) \ \le\ f(\bar{\mathbf{x}}(k)) \le f(\bar{\mathbf{x}}) + LW^{-1}\exp\Big(-\tilde{w}A\sum_{j=0}^{k-1}\alpha_j\Big) \\ \times\Big\{\|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 + \sum_{j=0}^{k-1}\exp\Big(\tilde{w}A\sum_{l=0}^{j}\alpha_l\Big) \\ \times\Big(M^2\alpha_j^2 + L^2\max_{1\le i\le n}\|\mathbf{x}_i(j) - \bar{\mathbf{x}}(j)\|_2^2\Big)\Big\}\end{aligned} \tag{51}$$

hold for all $k \ge 2$.

We finish this section with the proof of Theorem 2 under the assumption that Proposition 4 holds.

Proof of Theorem 2. By (13) and (36), we have

$$\sum_{j=0}^{\infty} M^2\alpha_j^2 + L^2\max_{1\le i\le n}\|\mathbf{x}_i(j) - \bar{\mathbf{x}}(j)\|_2^2 < \infty, \tag{52}$$

and

$$\lim_{k\to\infty}\exp\Big(-\tilde{w}A\sum_{j=l}^{k}\alpha_j\Big) = 0 \tag{53}$$

for all $l \ge 0$. Combining (49), (52) and (53) proves the desired limit in (41) by the dominated convergence theorem. □

V. Numerical Simulations

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

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N}\|\mathbf{G}\mathbf{x} - \mathbf{y}\|_2^2, \tag{54}$$

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

$$\mathbf{y} = \mathbf{G}\mathbf{x}_o \in \mathbb{R}^M \tag{55}$$

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 (12) to solve the convex optimization problem (54) and also compare it with the performance of the the conventional distributed gradient descent algorithm (DGD) under the CTA prototype (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 (54) 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 $\mathbf{y}_i, 1 \le i \le n$ are the same. 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.

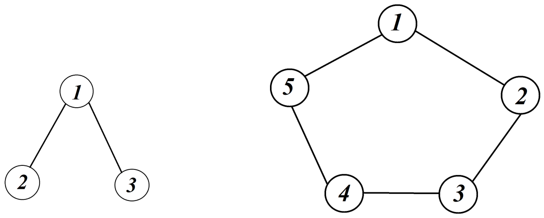

Fig. 1. Data exchanging structures with three/five-node network

In our simulations, we take $(M, N) = (225, 225)$ for Figures 2, 3, 4, 5 and $(M, N) = (250, 50)$ for Figures 6 and 7. 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 (12) and the CTA algorithm (6), where n is the number of active nodes in the network.

In the first simulation where there are 3 nodes with its topology described on the left of Figure 1, we take the coding matrix $\mathbf{B}$

$$\mathbf{B} = \begin{pmatrix} 1 & -5/4 & 0 \\ 0 & 1 & 4/9 \\ 18/10 & 0 & 1 \end{pmatrix} \tag{56}$$

and the decoding matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 10/18 \\ 1 & 9/4 & 0 \\ -4/5 & 0 & 1 \end{pmatrix}. \tag{57}$$

The above coding/decoding matrix pair satisfies (9) and the corresponding row stochastic matrix in (15) is

$$\mathbf{A}_{\text{sde}} = \begin{pmatrix} 0 & 1 & 10/18 & 0 & 0 & 0 \\ 1 & 9/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4/5 & 0 & 0 \\ 0 & 1 & 10/18 & 0 & 0 & 0 \\ 1 & 9/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4/5 & 0 & 0 \end{pmatrix}.$$

In Figures 2 and 3, we present the performance of the CoDGraD algorithm (12) and the CTA algorithm (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 $(k + 500)^{-0.95}, k \ge 0$, respectively.

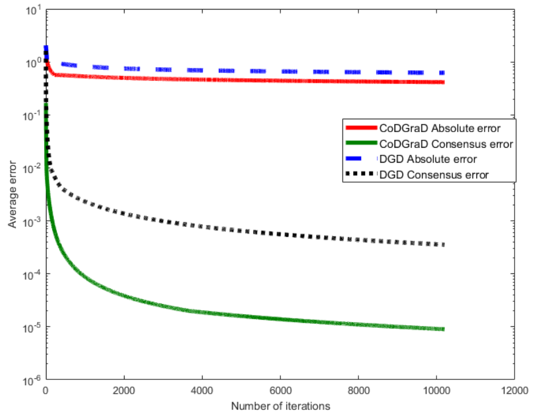

Fig. 2. Performance comparison of the CoDGraD algorithm (12) and the CTA algorithm (6) over a three active nodes network with $(M, N) = (225, 225)$ and step sizes $\alpha_k = (k + 300)^{-0.75}, k \ge 0$.

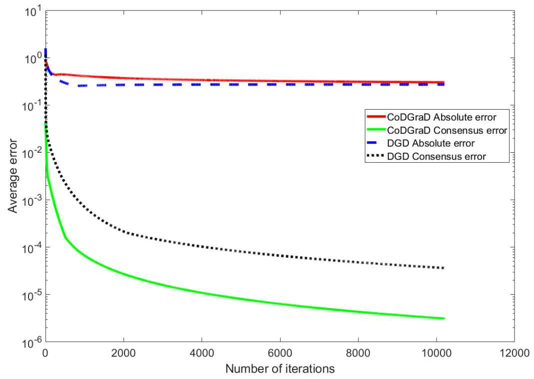

Fig. 3. Performance comparison of the CoDGraD algorithm (12) and the CTA algorithm (6) over a three active nodes network with $(M, N) = (225, 225)$ and step sizes $\alpha_k = (k + 500)^{-0.95}, k \ge 0$.

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

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1/2 & 0 & 0 \\ 0 & -1 & 3 & 4 & 0 \\ 0 & 0 & -5/2 & -3 & 1 \\ 1 & 0 & 0 & 1/5 & 13/5 \\ 2 & 1 & 0 & 0 & 4 \end{pmatrix} \tag{58}$$

and

$$\mathbf{A} = \begin{pmatrix} 1/2 & 1/4 & 0 & 0 & 1/4 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & -8/5 & 1 & 0 \\ 0 & 0 & -2/5 & -1 & 1 \\ 2 & 0 & 0 & 5 & -3 \end{pmatrix} \tag{59}$$

respectively. The above coding/decoding matrix pair satisfies (9) and the corresponding row stochastic matrix in (15) is

$$\mathbf{A}_{\rm sde} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{5}{18} & 0 & 0 & \frac{5}{18} & \frac{8}{18} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{12} & 0 & 0 & \frac{1}{6} & \frac{5}{12} & 0 \\ \frac{1}{5} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{3}{10} \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{5}{18} & 0 & 0 & \frac{5}{18} & \frac{2}{9} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{12} & 0 & 0 & \frac{1}{6} & \frac{5}{12} & 0 \\ \frac{1}{5} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{3}{10} \end{pmatrix}.$$

Shown in Figures 4 and 5 are the performance of the CoDGraD algorithm (12) and the CTA algorithm (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.

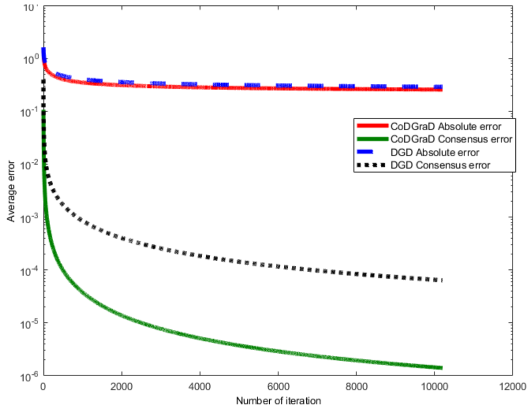

Fig. 4. Performance comparison of the CoDGraD algorithm (12) and the CTA algorithm (6) over a five node network with $(M, N) = (225, 225)$ and step sizes $\alpha_k = (k + 800)^{-0.9}, k \geq 0$.

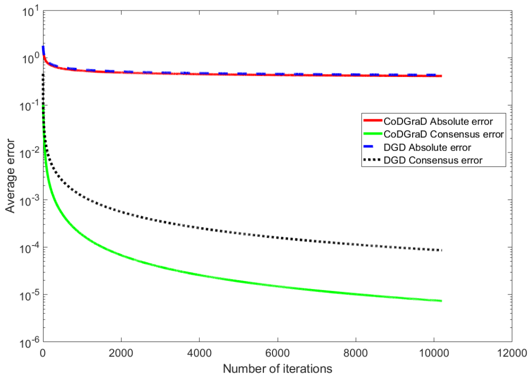

Fig. 5. Performance comparison of the CoDGraD algorithm (12) and the CTA algorithm (6) over a five node network with $(M, N) = (225, 225)$ and step sizes $\alpha_k = (k + 500)^{-0.95}, k \geq 0$.

From the above simulations, we observe that the CoDGraD algorithm (12) has much better performance than the CTA algorithm (6) in reaching consensus. Even though $\bar{\mathbf{x}}(k)$ satisfies a gradient descent algorithm (46) with an inexact global gradient, see Proposition 2, our simulations indicate that the CoDGraD algorithm (12) still has comparable performance in the absolute error with the CTA algorithm (6). Also we can conceive from the simulations that the CoDGraD algorithm (12) has faster convergence for a smaller exponent $\theta \in (1/2, 1]$, which confirms its convergence rate estimate in (35) and (50). On the other hand, our simulations also indicate that decreasing the exponent θ moves the CoDGraD algorithm (12) into the instability phase, which could directly be related to the sparsity of the network. It is worth mentioning that we can adequately calibrate this instability by increasing the value of a in our illustrate examples (4) for a fixed exponent θ. Thus we can anticipate in Figures 2 and 3 that the increase in the value to $\theta = 0.95$ degraded the convergence so that the CoDGraD algorithm became closer in performance to the CTA 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 Figures 6 and 7, we implement the CoDGraD algorithm (12) and the CTA algorithm (6) over the 5-node network with subsystems on nodes being overdetermined. It is observed that the absolute error decreases significantly in 2000 iterations to reach the machine precision of 10^{-16} using the step size with $\theta = 0.75$ and $a = 300$ or the algorithms have slower convergence with $\theta = 0.85$ and $a = 500$, cf. Figures 4 and 5. Our further simulations indicate that convergence behaviors of the CoDGraD algorithm (12) and the CTA algorithm (6) depends directly on maximal condition number of matrices $\mathbf{G}_i, 1 \leq i \leq n$, cf. (37) and (51) where A is closely related to the maximal condition number in the current setting.

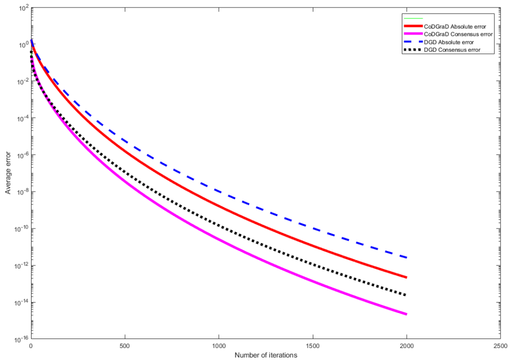

Fig. 6. Performance comparison of the CoDGraD algorithm (12) and the CTA algorithm (6) over a five node network with $(M, N) = (250, 50)$ and step sizes $\alpha_k = (k + 300)^{-0.75}, k \geq 0$.

VI. Conclusions

In this paper, we proposed the Code-Based Distributed Gradient Descent algorithm (12) to solve a convex optimization problem over a large network with some workers being stragglers due to the failure or heavy delay on computing

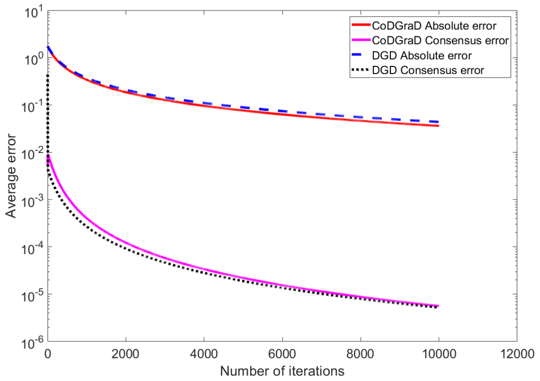

Fig. 7. Performance comparison of the CoDGraD algorithm (12) and the CTA algorithm (6) over a five node network with $(M, N) = (250, 50)$ and step sizes $\alpha_k = (k + 500)^{-0.85}, k \geq 0$.

or communicating. The proposed algorithm is a distributed version of gradient descent algorithm with inexact gradient updating, and it has better performance in reaching consensus as we apply the row stochastic matrix associated with the coding/decoding scheme. The convergence rate of the proposed CoDGraD algorithm depends on the topological structure of the network, the second largest eigenvalue of row stochastic matrix in magnitude, and the updating step sizes in the algorithm. Moreover, our coding scheme does not necessarily comply with the conventional paradigm of decomposing the global convex function onto a summand of local convex functions and hence our coding/decoding scheme may shed new light on distributed inexact (stochastic) gradient descent algorithms.

Appendix

A. Proof of Proposition 1

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

$$\sigma(\mathbf{A}_{\rm sde}) \subset \{1\} \cup \{z,\ |z| < 1\}, \tag{60}$$

and the eigenspace associated with eigenvalue one is given by

$$N(\mathbf{A}_{\rm sde} - \mathbf{I}) = {\rm span}\left\{\mathbf{1}_{2n}\right\}. \tag{61}$$

Combining (26), (29), (60) and (61), we obtain

$$\sigma(\mathbf{A}_{\rm sde} - \mathbf{P}) = \big(\sigma(\mathbf{A}_{\rm sde})\backslash\{1\}\big) \cup \{0\} \subset \{z,\ |z| \leq |\lambda_2(\mathbf{A}_{\rm sde})|\}. \tag{62}$$

Therefore there exists a positive constant C_1 such that

$$\|(\mathbf{A}_{\rm sde} - \mathbf{P})^k\|_{\mathcal{B}^\infty} \leq C_1\Big(\frac{1 + 2|\lambda_2(\mathbf{A}_{\rm sde})|}{3}\Big)^k,\ k \geq 1, \tag{63}$$

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

By (8), (16), (22) and (23), we have

$$\|\mathbf{h}(k)\|_{2,\infty} \leq \sup_{1 \leq i \leq n} \sup_{\mathbf{x} \in \mathbb{R}^N} \|\nabla g_i(\mathbf{x})\|_2 \leq M. \tag{64}$$

Then combining (31), (63) and (64) completes the proof.

B. Proof of Lemma 1

By (29), we have

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

Therefore

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

by (63). This, together with the observation that all entries of $\mathbf{A}_{\rm sde}^k, k \geq 1$ are nonnegative. Hence the required estimate implies that all entries of $\mathbf{a}_{\rm sde}$ are nonnegative and the desired estimate (44) follows.

C. Proof of Proposition 2

By (22), (33) and (39), we have

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

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

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

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

$$\|\mathbf{h}(k) - \tilde{\mathbf{h}}(k)\|_{2,\infty} \leq L\|\tilde{\mathbf{z}}(k)\|_{2,\infty}, \tag{65}$$

where

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

Therefore

$$\begin{aligned}&\|\mathbf{a}_{\rm sde}^T\mathbf{h}(k) - \mathbf{a}_{\rm sde}^T\tilde{\mathbf{h}}(k)\|_2 \\ &\leq \|\mathbf{a}_{\rm sde}\|_1\|\mathbf{h}(k) - \tilde{\mathbf{h}}(k)\|_{2,\infty} \leq L\|\mathbf{a}_{\rm sde}\|_1\|\tilde{\mathbf{z}}(k)\|_{2,\infty},\end{aligned} \tag{66}$$

where the second estimate follows from (65).

Observe from (9) that

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

which together with (27) implies that

$$\mathbf{a}_{\rm sde}^T\tilde{\mathbf{h}}(k) = \mathbf{a}_{\rm sde}^T\mathbf{A}_{\rm sde}\tilde{\mathbf{h}}(k) = \tilde{w}\nabla f(\bar{\mathbf{x}}(k)). \tag{67}$$

Combining (66) and (67) proves the desired estimate (47).

D. Proof of Proposition 3

Set

$$\beta_k = M^2\alpha_k^2 + L^2\|\tilde{\mathbf{z}}(k)\|_{2,\infty}^2,\ k \geq 0. \tag{68}$$

By (22), (16), (46) and (47), we obtain

$$\begin{aligned}\|\bar{\mathbf{x}}(k+1) - \bar{\mathbf{x}}\|_2^2 &= \|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 + \alpha_k^2\|\mathbf{a}_{\rm sde}^T\mathbf{h}(k)\|_2^2 \\ &\quad -2\alpha_k\langle\mathbf{a}_{\rm sde}^T\mathbf{h}(k), \bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\rangle_N \\ &\leq \|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 + M^2\|\mathbf{a}_{\rm sde}\|_1^2\alpha_k^2 \\ &\quad +2L\|\mathbf{a}_{\rm sde}\|_1\alpha_k\|\tilde{\mathbf{z}}(k)\|_{2,\infty}\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2 \\ &= \|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 + M^2\alpha_k^2 \\ &\quad +2L\alpha_k\|\tilde{\mathbf{z}}(k)\|_{2,\infty}\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2 \\ &\leq (1 + \alpha_k^2)\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 + \beta_k,\end{aligned} \tag{69}$$

where we also use the positivity of $\tilde{w}$ in (44), $\|\mathbf{a}_{\rm sde}\|_1 = 1$ and the convexity of the objective function f,

$$\langle \nabla f(\mathbf{x}), \mathbf{x} - \bar{\mathbf{x}} \rangle_N \ge 0, \ \mathbf{x} \in \mathbb{R}^N. \tag{70}$$

Applying (69) repeatedly, we get

$$\begin{aligned} & \|\bar{\mathbf{x}}(k+1) - \bar{\mathbf{x}}\|_2^2 \\ \le \ & \prod_{j=0}^{k} (1+\alpha_j^2) \|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 + \beta_k + \sum_{j=0}^{k-1} \beta_j \prod_{j'=j+1}^{k} (1+\alpha_{j'}^2) \\ \le \ & \exp\Big(\sum_{j=0}^{k} \alpha_j^2\Big) \Big(\|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 + \sum_{j=0}^{k} \beta_j \Big) \\ \le \ & \exp\Big(\sum_{j=0}^{\infty} \alpha_j^2\Big) \Big(\|\bar{\mathbf{x}}(0) - \bar{\mathbf{x}}\|_2^2 + \sum_{j=0}^{\infty} \beta_j \Big), \quad k \ge 1. \end{aligned} \tag{71}$$

This together with (13) and (36) proves the desired bound (48) for the sequence $\bar{\mathbf{x}}(k), k \ge 0$.

E. Proof of Proposition 4

By (13), 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 (69) with the convexity (70) replaced by the strong converxity (40), we obtain

$$\|\bar{\mathbf{x}}(k) - \bar{\mathbf{x}}\|_2^2 \le (1 - \tilde{w} A \alpha_{k-1}) \|\bar{\mathbf{x}}(k-1) - \bar{\mathbf{x}}\|_2^2 + \beta_{k-1}, \tag{72}$$

cf. (69). Applying the above estimate repeatedly leads to

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

where $\beta_j, j \ge 0$, is given in (68). This proves the desired estimate (49).

References

[1] A. Nedic and D. Bertsekas (2001). Convergence rate of incremental subgradient algorithms. In *Stochastic Optimization: Algorithms and Applications*, pp. 223–264. Springer.

[2] C. R. Da Silva, B. Choi, and K. Kim (2007). Distributed spectrum sensing for cognitive radio systems. In *2007 Information Theory and Applications Workshop*, pp. 120–123. IEEE.

[3] B. Johansson (2008). On distributed optimization in networked systems. PhD. thesis, KTH.

[4] A. Nedic, A. Ozdaglar, and P. A. Parrilo (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, **55**(4), 922–938.

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

[6] C. Cheng, Y. Jiang, and Q. Sun (2017). Spatially distributed sampling and reconstruction. *Applied and Computational Harmonic Analysis*, In press. https://doi.org/10.1016/j.acha.2017.07.007

[7] J. Jiang, C. Cheng, and Q. Sun (2017). Nonsubsampled graph filter banks: theory and distributed algorithms. arXiv preprint arXiv:1709.04107

[8] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo (2015). Convergence rate of incremental gradient and incremental newton methods. arXiv preprint arXiv:1510.08562.

[9] A. S. Bedi and K. Rajawat (2018). Asynchronous incremental stochastic dual descent algorithm for network resource allocation. *IEEE Transactions on Signal Processing*, **66**(9), 2229–2244.

[10] N. Takahashi, I. Yamada, and A. H. Sayed (2010). Diffusion least-mean squares with adaptive combiners: formulation and performance analysis. *IEEE Transactions on Signal Processing*, **58**(9), 4795–4810.

[11] F. S. Cattivelli and A. H. Sayed (2010). Diffusion LMS strategies for distributed estimation. *IEEE Transactions on Signal Processing*, **58**(3), 1035–1048.

[12] D. P. Bertsekas (2011). Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. In *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S. Wright, Eds., pp. 1-38. MIT Press.

[13] A. H. Sayed, S. Barbarossa, S. Theodoridis, and I. Yamada (2013). Adaptation and learning over complex networks. *IEEE Signal Processing Magazine*, **30**(3), 14–15.

[14] D. Needell, R. Ward, and N. Srebro (2014). Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algrithm. In *Advances in Neural Information Processing Systems*, pp. 1017–1025.

[15] V. J. Mathews, and Z. Xie (1993). A stochastic gradient adaptive filter with gradient adaptive step size. *IEEE Transactions on Signal Processing*, **41**(6), 2075–2087.

[16] H. Robbins, and S. Monro (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, **22**(3), 400–407.

[17] N. N. Schraudolph (1999). Local gain adaptation in stochastic gradient descent. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99*, pp. 569–574. IEEE.

[18] J. H. Friedman (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, **38**(4), 367–378.

[19] L. Bottou (2012). Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, K.-R. Muller eds, pp 421–436.

[20] M. D. Zeiler (2012). Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

[21] D. P. Kingma and J. Ba (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[22] M. Hardt, B. Recht, and Y. Singer (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33 rd International Conference on Machine Learning*, pp. 1225–1234, 2016.

[23] C. Tan, S. Ma, Y.-H. Dai, and Y. Qian (2016). Barzilai-borwein step size for stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 685–693.

[24] 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 (2012). Large scale distributed deep networks. In *Proceeding NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 1223–1231.

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

[26] E. Atallah, N. Rahnavard (2018). A Code-Based Distributed Gradient Descent Method. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton) IEEE*. pp 951 –958.

[27] M. Li, D. G. Andersen, A. J. Smola, and K. Yu (2014). Communication efficient distributed machine learning with the parameter server. In *Proceeding NIPS'14 Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 19–27.

[28] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis (2017). In *Proceedings of the 34th International Conference on Machine Learning, (PMLR)*, **70**, pp. 3368–3376.

[29] W. Halbawi, N. Azizan-Ruhi, F. Salehi, and B. Hassibi (2017). Improving distributed gradient descent using reed-solomon codes. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2027–2031, IEEE.

[30] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis (2018). Gradient coding from cyclic mds codes and expander graphs. In *Proceedings of the 35 th International Conference on Machine Learning*, Stockholm, Sweden.