Decentralized Optimization with Edge Sampling

Chi Zhang[1 2*], **Qianxiao Li**[2] and **Peilin Zhao**[3]
[1]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore
[2] IHPC, Agency for Science, Technology and Research, Singapore
[3]Tencent AI Lab, China
czhang024@e.ntu.edu.sg; liqix@ihpc.a-star.edu.sg; peilinzhao@hotmail.com

Abstract

In this paper, we propose a decentralized distributed algorithm with stochastic communication among nodes, building on a sampling method called "edge sampling". Such a sampling algorithm allows us to avoid the heavy peer-to-peer communication cost when combining neighboring weights on dense networks while still maintains a comparable convergence rate. In particular, we quantitatively analyze its theoretical convergence properties, as well as the optimal sampling rate over the underlying network. When compared with previous methods, our solution is shown to be unbiased, communication-efficient and suffers from lower sampling variances. These theoretical findings are validated by both numerical experiments on the mixing rates of Markov Chains and distributed machine learning problems.

1 Introduction

Decentralized optimization [Tsitsiklis, 1984; Tsitsiklis *et al.*, 1986] focuses on the development and analysis of solving optimization problems that are defined over networks. Different from the centralized distributed optimization where information from distributed nodes needs to be sent to a central unit (*e.g.*, the "Parameter-Server" framework in [Li *et al.*, 2014]), computation nodes in this framework only contact their immediate neighbors so the computation and communication are totally decentralized. Such a learning paradigm finds its applications in a variety of research areas, including sensor network estimation [Rabbat and Nowak, 2004], multi-agent coordination [Necoara, 2013; Cao *et al.*, 2013], distributed tracking [Olfati-Saber and Sandell, 2008] and source scheduling [Chunlin and Layuan, 2006]. In these cases, data may naturally be distributed and observed over the network and sending all data to a fusion center leads to extra transportation costs. This issue is compounded by privacy and security concerns, where it is favorable to compute models locally for political, privacy-sensitive and technological reasons.

The performance of decentralized optimization is known to be affected by the mixing ability of the underlying networks [Boyd *et al.*, 2004; Levin and Peres, 2017; Boyd *et al.*, 2006]. For dense networks, each node is well-connected to its neighbors and therefore guarantees sufficient information exchange on each round, leading to faster convergence rates. However, on the other side of the coin is the heavy computation and communication cost when transporting and computing parameters among nodes. As a canonical example, consider Online Gradient Descent (OGD) on a dense network called "n-complete network", where each node is connected to all the remaining nodes and the overall network obtains the best mixing ability. The combination step on each round requires a collection of neighboring parameters with $\mathcal{O}(nd)$ cost, dominating the following $\mathcal{O}(d)$ cost in the local online gradient descent phase when n is sufficiently large. This issue is further compounded by the scenarios when the communication is costly or each node has limited storage space to buffer the neighboring parameters [Iyengar *et al.*, 2004; Balcan *et al.*, 2012; Woodruff and Zhang, 2017]. On the other hand, sparse and poorly-connected networks require less neighboring communication costs and storage demands but suffer from slower convergence rates and poor adaptation ability to changes for networks. It seems the merits of achieving faster convergence rates while using less frequent neighboring communication for large networks cannot be obtained simultaneously in the decentralized optimization.

In this paper, we address this issue by proposing a decentralized optimization based on a sampling strategy named "edge sampling", allowing the dense networks to be dynamically sparse to reduce communication burden while maintaining a comparable convergence rate. More specifically, each node selects a subset of its connecting edges based on a sampling parameter, as well as the original combination weight, to generate an unbiased estimation when combining neighboring parameters. Such a strategy is shown to be unbiased, communication efficient and graph-dependent, which in fact allows us to use a relatively small sampling rate to avoid the redundant communication on dense networks while maintaining most connections on poorly-connected networks. These properties are further validated with both theoretical analysis where we show this sampling strategy allows the algorithm to converge significantly faster than the unsampled method when giving the same communication budget (*e.g.*,

*Contact Author

	Unbiased	Low Variance	Graph Dependent	Communication Efficient
DDA	✓	✓	✓	×
Syn Gossip [Boyd *et al.*, 2006]	×	×	×	✓
DDA-NS [Duchi *et al.*, 2012]	×	×	×	✓
This Paper	✓	✓	✓	✓

Table 1: Algorithm Comparison. DDA stands for an unsampled baseline and the following three act as sampling algorithms.

converge $\mathcal{O}(n)$ faster in the above mentioned n-complete networks) and experimental results on the mixing rates of Markov Chains and distributed machine learning problems.

Related Works: Our work directly follows the optimization paradigm by decomposing the learning into two phases, namely "combination phase" and "adaptation phase", in recent gradient-based decentralized optimization [Ram *et al.*, 2010; Duchi *et al.*, 2012; Sayed and others, 2014; Shi *et al.*, 2015; Yuan *et al.*, 2016]. The idea of replacing the deterministic combination phase with a stochastic counterpart is partly explored in previous distributed studies [Boyd *et al.*, 2006; Duchi *et al.*, 2012], but the these algorithms consider node-based sampling strategy and suffer from both biased estimation and large variances (see algorithm comparison in Table. 1). In the following parts, we consider the unsampled algorithm as a baseline and further these existing algorithms as competitors. Our sampling strategy is also partly related to "graph sparsification" techniques in [Spielman and Srivastava, 2011; Spielman and Teng, 2011], where the authors present algorithms to produce sparsifiers for dense networks, but their works mainly focus on maintaining the Laplacian matrix spectral gap while our method applies to the general doubly stochastic matrix. Finally, our work is also partly related to studies on time-varying graphs [Nedić and Olshevsky, 2015; Nedic *et al.*, 2017] as the underlying graph becomes dynamic after sampling, but our work differs from these works as it does not make extra graph connection assumptions only requires the original graph to be connected.

Notations: Lower-case letters $(\alpha, \beta, \cdots)$ denote as scalars and lower-case bold letters $(\mathbf{w}, \mathbf{v}, \cdots)$ are used as vectors. Upper-case letters (U, P, Q) denote matrices. For distributed algorithms, the element $a_{i,j}$ of a fixed matrix $A \in \mathbb{R}^{n\times n}$ denotes the combination weight from node j to node i. The singular values of A are denoted as $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_n(A)$. $\mathbf{1}$ refers to a $n \times 1$ vector with all elements equaling to 1. For a graph, D represents its degree matrix with d_{max} denoting its maximum degree, and J represents its adjacency matrix.

2 Decentralized Optimization with Edge Sampling

We formally propose our decentralized optimization algorithm with edge sampling in this part. Although such a learning paradigm can be generally adopted to most gradient-based decentralized algorithms, we focus on its application to the well-known DDA [Duchi *et al.*, 2012] algorithm in this paper.

2.1 Standard Distributed Dual Averaging (DDA)

We start from introducing the standard Distributed Dual Averaging (DDA) algorithm [Duchi *et al.*, 2012] that acts as our baseline. Consider a graph $G = (V, E)$ with vertex set $V = \{1, \cdots, n\}$ and edge set $E \subseteq V \times V$. Each node $i \in V$ stores its own parameters $\mathbf{w}_i \in \mathbb{R}^d$ and uses the associated local function $f_i : \mathbb{R}^d \to \mathbb{R}$ to evaluate the performance of its parameters. The communication between nodes is specified by the graph G: node i can only directly communicate with its immediate neighbors $N(i) = \{j \in V \mid (i, j) \in E\}$ through a combination matrix A. The overarching goal of decentralized optimization is minimizing a global objective defined by the average over the local functions:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}) \text{ subject to } \mathbf{w} \in \Omega. \tag{1}$$

In DDA algorithm, each node stores a primal parameter $\mathbf{w}_i^t$ and an auxiliary dual variable $\mathbf{z}_i^t$. On each round, it updates its parameters as follows:

$$\text{Combine: } \mathbf{z}_i^{t+1} = \sum_{j\in N(i)} a_{i,j}\mathbf{z}_j^t + \partial f_i(\mathbf{w}_i^t), \tag{2}$$

$$\begin{aligned}\text{Adapt: } \mathbf{w}_i^{t+1} &= \Pi_\Omega^\psi\left(\mathbf{z}_i^{t+1}, \eta^t\right) \\ &= \arg\min_{\mathbf{w}\in\Omega}\left\{\langle \mathbf{z}_i^{t+1}, \mathbf{w}\rangle + \frac{1}{\eta_t}\psi(\mathbf{w})\right\}. \end{aligned}\tag{3}$$

In particular, node i first computes its new dual parameter $\mathbf{z}_i^{t+1}$ from a weighted average of the neighboring parameters $\mathbf{z}_j^t$ and its own subgradient $\partial f_i(\mathbf{w}_i^t)$. The next local iterate $\mathbf{w}_i^{t+1}$ is chosen by minimizing an averaged first-order approximation to the function f_i with a proximal function ψ to ensure the primal parameters do not oscillate wildly. Typical examples of ψ include ℓ_1 regularization $\psi(\mathbf{w}) = \|\mathbf{w}\|_1$, ℓ_2 regularizations $\psi(\mathbf{w}) = \|\mathbf{w}\|_2^2$ and the entropy functions $\psi(\mathbf{w}) = \sum_{i=1}^d w_i \log(w_i) - w_i$.

The convergence rate of the standard DDA is well-established as $\mathcal{O}(1/\sqrt{T'})$ in the previous research [Duchi *et al.*, 2012] with the following assumptions.

Assumption 1. *Function Assumption: each function $f_i(\mathbf{w})$ is L-Lipschitz continuous w.r.t the same norm $\|\cdot\|$ on Ω, namely*

$$|f_i(\mathbf{x}) - f_i(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \textit{for } \mathbf{x}, \mathbf{y} \in \Omega.$$

Assumption 2. *Network Assumption: the network is connected and the corresponding combination matrix A is irreducible with non-negative elements. Further more, the combination matrix A is assumed to be doubly stochastic, namely*

$$\sum_{i=1}^{n} a_{i,j} = 1, \ \sum_{j=1}^{n} a_{i,j} = 1$$

is true for all $i \in \{1, \cdots, n\}$ *and* $j \in \{1, \cdots, n\}$.

Theorem 1. *Under the assumption 1 and 2 and setting the step size* $\eta^t \propto \frac{R\sqrt{1-\sigma_2(A)}}{4L\sqrt{t}}$*, the DDA algorithm converges as*

$$f(\hat{\mathbf{w}}_i^{T'}) - f(\mathbf{w}^*) \leq c' \frac{RL}{\sqrt{T'}} \frac{\log(T'\sqrt{n})}{\sqrt{1-\sigma_2(A)}} \quad \textit{for all } i, \quad (4)$$

with $\hat{\mathbf{w}}_i^{T'}$ *denoting the average weight* $(\mathbf{w}_i^1 + \cdots + \mathbf{w}_i^{T'})/T'$*,* c' *denoting a universal constant,* R *denoting another constant to upper bound* $\psi(\mathbf{w}^*) \leq R^2$ *and* $\sigma_2(A) = \max\{\lambda_2(A), |\lambda_n(A)|\}$.

2.2 Distributed Dual Averaging with Edge Sampling (DDA-ES)

Performance of the above DDA algorithm is known to be affected by the underlying combination matrix A and its corresponding component $a_{i,j}$. Dense combination matrix A allows the information to be well-transported among nodes while leads to heavy communication cost at the same time. To alleviate this issue while still maintain a comparable performance, we propose an edge sampling algorithm named DDA-ES as follows.

DDA-ES algorithm requires each existing edge $e_{i,j} \in E$ to appear with probability

$$p_{i,j} = \frac{1}{1 + \frac{z}{a_{i,j}}}, \quad z > 0 \quad (5)$$

on each round, with z denoting a graph-dependent parameter to control the sampling rate.

When an edge $(i, j) \in E$ is sampled on round t, the new combination weights $b_{i,j}^t$ and $b_{j,i}^t$ will be scaled by a factor $1/p_{i,j}$:

$$b_{i,j}^t = b_{j,i}^t = \begin{cases} \frac{1}{p_{i,j}} \cdot a_{i,j} & \text{if edge } e_{i,j} \text{ is sampled;} \\ 0 & \text{otherwise,} \end{cases}$$

to guarantee an *unbiased* estimation:

$$\mathbb{E}\left[b_{i,j}^t\right] = p_{i,j} \cdot \frac{1}{p_{i,j}} a_{i,j} = a_{i,j}. \quad (6)$$

A normalization step is finally performed on each node to preserve row-stochasticity for matrix $B(t)$:

$$b_{i,i}^t = 1 - \sum_{j=1\backslash i}^{n} b_{i,j}^t \quad \text{for } i \in [1, \cdots, n]. \quad (7)$$

It's worth mentioning that although $b_{i,i}^t$ also acts as an unbiased estimation for $a_{i,i}$, the new combination matrix $B(t)$ is not strictly "doubly stochastic" as $b_{i,i}^t$ can occasionally be negative. However, we shall show (S.M. B, Lemma 4) bounding z as

$$0 \leq z \leq \min_{a_{i,j} \neq 0} \left\{ \frac{\left[A^2\right]_{i,j}}{2a_{i,j}} \right\} \quad (8)$$

is sufficient to ensure the convergence of our proposed algorithm.

Algorithm 1 Distributed Dual Averaging with Edge Sampling (**DDA-ES**)

Input: Convex set Ω, combination matrix A, step-size $\{\eta^t\}$ for $\forall i \in \{1, \cdots, n\}$.
Initialize: $\mathbf{w}_i^0 = \mathbf{0}$ and $\mathbf{z}_i^0 = \mathbf{0}$ for $\forall i \in \{1, \cdots, m\}$.
Set $0 \leq z \leq \min_{a_{i,j} \neq 0} \left\{ \frac{\left[A^2\right]_{i,j}}{2a_{i,j}} \right\}$
for $t = 1, \cdots, T$ **do**
  Sample each edge with probability $p_{i,j} = \frac{1}{1+\frac{z}{a_{i,j}}}$.
  if edge $e_{i,j}$ appears **then**
    Assign weight $b_{i,j}^t = b_{j,i}^t = \frac{1}{p_{i,j}} a_{i,j}$
  else
    Assign weight $b_{i,j}^t = b_{j,i}^t = 0$
  end if
  Normalize: $b_{i,i}^t = 1 - \sum_{j=1\backslash i}^{n} b_{i,j}^t$ for $i \in [1, \cdots, n]$.
  for *Each learner* $i \in V$ **do**
    Combine: $\mathbf{z}_i^{t+1} = \sum_{j \in N(i)} b_{i,j}^t \mathbf{z}_j^t + \partial f_i(\mathbf{w}_i^t)$
    Adapt: $\mathbf{w}_i^{t+1} = \Pi_\Omega^\psi\left(\mathbf{z}_i^{t+1}, \eta^t\right)$
  end for
end for
Output: $\mathbf{w}_i^T$ for $i = 1, \cdots, n$

Remark: The sampling rate $p_{i,j}$ in (5) is designed with the following merits. (1) It allows the communication matrix $B(t)$ to be an unbiased i.i.d estimation of the previous dense matrix A. (2) $p_{i,j}$ is designed to be positively correlated to $a_{i,j}$, since the combination weight $a_{i,j}$ between node i and j often indicates the importance of a certain edge $e_{i,j} \in E$. For instance, if a certain edge acts as a bottleneck of the underlying graph, its weight $a_{i,j}$ in the original combination matrix A is generally larger to increase information exchange, and our edge sampling strategy is designed to sample this edge more frequently to boost communication between nodes. (3) Most importantly, this sampling strategy leads to the convenience in analyzing the convergence rate of DDA-ES algorithm, as shown in Sec. 3.

2.3 Node-Wise Implementation of DDA-ES

The above DDA-ES algorithm cannot be directly executed since most decentralized algorithms are implemented on nodes instead of the edges connecting them, so we provide its practical node-wise implementation as follows. For each existing edge $e_{i,j} \in E$, node i and node j can trigger the communication independently with probability

$$\hat{p}_{i,j} = \hat{p}_{j,i} = 1 - \sqrt{1 - p_{i,j}}.$$

The peer-to-peer communication is executed if at least one node triggers the communication, guaranteeing the final sampling rate to be $1 - (1 - \hat{p}_{i,j})(1 - \hat{p}_{j,i}) = p_{i,j}$.

3 Convergence Analysis

We establish the convergence analysis for DDA-ES algorithm in this part, followed by some discussions on the sampling strategy for different graphs.

3.1 Convergence Rate

Theorem 2. *Let $\{B(t)\}$ be an i.i.d. sequence of doubly stochastic matrices generated in Algorithm 1. By setting the step size $\eta^t \propto \frac{R\sqrt{1-\kappa(z)}}{L\sqrt{t}}$, DDA-ES converges as*

$$f(\hat{\mathbf{w}}_i^T) - f(\mathbf{w}^*) \leq c\frac{RL}{\sqrt{T}}\frac{\log(T\sqrt{n})}{\sqrt{1-\kappa(z)}} \tag{9}$$

with probability at least $1 - 1/(T^2 n)$ for any node $i \in \{1, \cdots, n\}$, with $\kappa(z) = \max\left\{\lambda_2^2(A) - 2z\lambda_2(A) + 2z, \lambda_n^2(A) - 2z\lambda_n(A) + 2z\right\}$.

Proof: See Supplementary Material B[1].

The above theorem shows that DDA-ES algorithm converges with a speed of $\mathcal{O}(1/\sqrt{T})$, similar to the rate established in Theorem 1 for standard DDA . It also numerically reveals how the sampling parameter z affects the spectral gap $1 - \kappa(z)$ and therefore the final convergence rate. A good sampling rate z will guarantee our gain in reducing the communication cost does not offset the loss in the convergence rate, and we shall establish the choices in the following part.

3.2 Convergence Comparison

For fair comparison, we adopt a scheme proposed in [Zhang *et al.*, 2013] by comparing the convergence rate of DDA-ES with the baseline under the same communication budget.

Clearly, when giving the sample communication budget, DDA-ES algorithm is able to update more iterations as it samples a subset of existing edges and bears less communication cost per iteration. A more precise description is established as

$$T = T'/\bar{p}(z), \tag{10}$$

with T and T' denoting the overall updating iterations for DDA-ES and DDA separately. Here $\bar{p}(z)$ denoting the averaging sampling rate for DDA-ES by summing $p_{i,j}$ over the total edge number $|E|$:

$$\bar{p}(z) := \frac{\sum_{i=1}^n \sum_{j=i+1}^n p_{i,j}}{|E|} = \frac{\sum_{i=1}^n \sum_{j=i+1}^n \frac{a_{i,j}}{a_{i,j}+z}}{|E|}.$$

Substitute T in (9) with $T'/\bar{p}(z)$, we have

$$f(\hat{\mathbf{w}}_i^{T'}) - f(\mathbf{w}^*) \leq c\frac{RL}{\sqrt{T'/\bar{p}(z)}}\frac{\log(T'\sqrt{n}/\bar{p}(z))}{\sqrt{1-\kappa(z)}}. \tag{11}$$

For simplicity of analysis, we ignore the effect of constant values c, c' and assume T' is sufficient large so that $\log(T'\sqrt{n}/\bar{p}(z)) = \log(T'\sqrt{n}) + \log(1/\bar{p}(z)) \approx \log(T'\sqrt{n})$.

Now the difference between the convergence rate of DDA-ES in (11) and the baseline of DDA in (4) can be summarized into a factor:

$$\Gamma(z) := \sqrt{\bar{p}(z)} \cdot \frac{\sqrt{1-\sigma_2(A)}}{\sqrt{1-\kappa(z)}}. \tag{12}$$

[1]Supplementary materials is available on the website: https://www.dropbox.com/s/v8bgjby9odexiqa/Sampling_SuppMaterial.pdf?dl=0.

Our goal now lies in minimizing the $\Gamma(z)$ by analyzing its two components: the lower sampling rate $\bar{p}(z) < 1$ in DDA-ES reduces $\Gamma(z)$ while the decreasing spectral gap $1 - \sigma_2(A) > 1 - \kappa(z)$ increases our target value. The optimal z^* corresponds to the best trade-off between these two components, namely

$$z^* = \arg\min_{z\in(8)} \Gamma(z) = \arg\min_{z\in(8)} \frac{\bar{p}(z)}{1-\kappa(z)}. \tag{13}$$

3.3 Acceleration of Edge Sampling with Optimal Sampling Parameter z^*

To show the benefits of edge sampling, we start from a simple example on the n-complete graph and establish its theoretical results in the following corollary.

Corollary 3. *For n-complete graphs with matrix $A = \frac{\mathbf{1}\cdot\mathbf{1}^T}{n}$, the optimal sampling parameter and its corresponding acceleration factor can be derived as*

$$z^* = \frac{4}{n+2} \text{ and } \Gamma(z) = \mathcal{O}(1/\sqrt{n}).$$

Proof: See Supplementary Material C.

To achieve $f(\hat{\mathbf{w}}_i(T')) - f(\mathbf{w}^*) \leq \epsilon$, the standard DDA algorithm needs approximately $\mathcal{O}(\frac{R^2L^2}{\epsilon^2})$ communication iterations while DDA-ES algorithm only needs approximately $\mathcal{O}(\frac{R^2L^2}{n\epsilon^2})$ communication rounds. In other words, the sampling algorithm is approximately n-times faster than the original DDA when given the same communication budget.

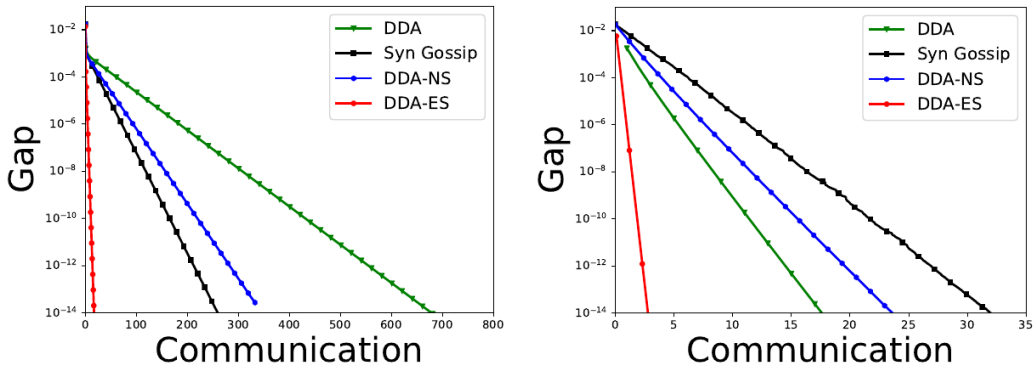
Replacing z in Eq (5) with the above corollory, we obtain the optimal sampling rate as $p_{i,j}^* = 4/(n+2)$. Namely, each node in DDA-ES only needs to contact its neighbors with $\mathcal{O}(4d)$ cost instead of $\mathcal{O}(nd)$ in standard DDA, which clearly reduces the communication burden and avoids the network overburden problem.

Eq (13) has a closed-form solution z^* for many standard graphs, and the acceleration of convergence rate under the same communication cost and reduction in peer-to-peer communication cost can be generally found on these graphs. In fact, the above findings can be extended to d-regular graphs where edge sampling technique allows the algorithm to converge $\mathcal{O}(d)$ times faster than the baseline with large d, and the reader is referred to Supplementary Material C for details.

As for the general case, Eq (13) may not have a closed-form solution for any arbitrary graph. In this case, we can simply use grid-search in $\left[0, \min_{a_{i,j}\neq 0}\left\{\frac{[A^2]_{i,j}}{2a_{i,j}}\right\}\right]$ to find a (sub)optimal z with $\mathcal{O}(n^2)$ computation cost, or simply use $\min_{a_{i,j}\neq 0}\left\{\frac{[A^2]_{i,j}}{2a_{i,j}}\right\}$ for convenience. A numerical computation of $\Gamma(z)$ leads to the conclusion whether edge sampling accelerates the original algorithm in this case.

4 Experiments

We have theoretically shown the DDA-ES algorithm achieves of goal of reducing the communication cost on each round



(a) Mixing rate of Markov chain on the k-connected cycle. n = 1000, k = 75, z = 0.25.

(b) Mixing rate of Markov chain on the random graph. n = 1000, c = 0.15, z = 0.05.

Figure 1: Mixing rate of Markov chains on different networks. The gap refers to $\|W^t - \mathbf{1}/n\|_2$.

while accelerating the overall convergence rates under the same communication budget, and we shall validate our findings with numerical experiments in this part.

4.1 Experimental Setup

Competing Algorithms: We compare our *DDA-ES* algorithm with the standard *DDA* algorithm, as well as two existing node sampling algorithms, namely *Syn Gossip* in [Boyd *et al.*, 2006] and *DDA-NS* in [Duchi *et al.*, 2012].

Graphs: : Experiments are conducted on two types of graphs: (1) *k-Connected Cycle* $C_{n,k}$ with each node connecting to its k neighbors on the left and right, representing a $(2k+1)$-regular graph with theoretical optimal z^*; (2) *Random Graph* $G_{n,c}$ (also known as "Erdos-Renyi graph" [Erdös and Rényi, 1960]) with any two nodes connected with probability c, representing a general graph without closed-form solutions for z.

Parameter Settings: For all algorithms, we set $\eta^t = \mathcal{O}(1/\sqrt{t})$ as suggested in the previous theoretical analysis. The communication matrix A is set by the 'Metropolis-Hasting' rule [Metropolis *et al.*, 1953; Hastings, 1970] to guarantee doubly stochasticity:

$$a_{i,j} = \begin{cases} \min\{1/d(i), 1/d(j)\} & \text{if } e_{i,j} \in E \text{ and } i \neq j; \\ \sum_{k \in N(i)} \max\{0, 1/d(i) - 1/d(k)\} & \text{if } i = j. \end{cases}$$

Measurements: One *natural iteration* (denoted as "Iteration" in figures) refers to a single execution of combination and adaption for one round, while one *communication round* (denoted as "Communication" in figures) refers to the communication cost for standard DDA algorithm in one natural iteration. Performance of each algorithm is measured by computing the gap between its status on round t and the optimal one.

4.2 Mixing Rate of Markov Chains

We first conduct experiments to show the mixing rates of Markov chain for all algorithms[2], which quantifies the diffusion ability of the networks [Levin and Peres, 2017] and also determines the convergence rates of distributed algorithms [Duchi *et al.*, 2012]. Each node is first given an initial value w_i^0 based on a predefined n-dimensional simplex, and then mixes its value with its neighbors as $w_i^{t+1} = \sum_{j \in N(i)}^{n} a_{i,j} w_j^t$. We measure the gap between the network's current status $W^t = (w_0^t, \cdots, w_n^t)$ and the final stable distribution $\mathbf{1}/n$ by calculating their gap, namely $\|W^t - \mathbf{1}/n\|_2$.

Figure 1(a) reports the mixing rate of Markov chain on k-connected cycle. Previous analysis in Sec 3.3 shows the edge sampling algorithm converges $\mathcal{O}(2k+1)$ times faster than the standard DDA, and this is validated in figure 1(a) as DDA-ES algorithm only needs a few rounds to reach its equilibrium distribution $\mathbf{1}/n$ and significantly outperforms its unsampled counterpart. Similar phenomenon can be observed in 1(b), where DDA-ES algorithms also converges faster than its competitors on the random graph. Sampling techniques including "Syn Gossip" and "DDA-NS" can boost the mixing rate of Markov chain on the k-connected graph but performs; worse than the baseline on the random graph, and their performance is consistently inferior to our proposed algorithm.

4.3 Distributed Logistic Regression

We now consider a distributed optimization problem for the *a9a* dataset[3]. Each node i on the network receives a subset of the dataset and a local loss function:

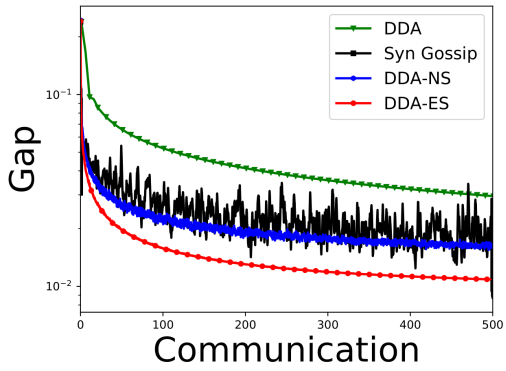
$$f_i(\mathbf{w}) = \log\left(1 + e^{-y_i \cdot \mathbf{w}^\top \mathbf{x}_i}\right)$$

to perform online logistic regression. The overall goal is minimizing the average of these local functions: $f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$, with proximal function in Eq (3) set as $\psi(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$ and $\lambda = 10^{-3}$. Since Theorem 1 and 2 hold for any node i, the performance of all algorithms is measured by calculating the gap $f(\hat{\mathbf{w}}_1^{T'}) - f(\mathbf{w}^*)$ with the weight $\hat{\mathbf{w}}_1^{T'}$ on the first node.

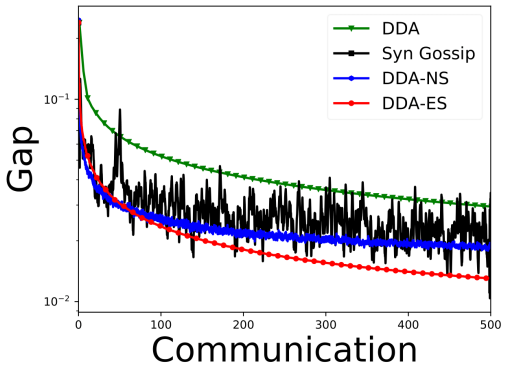
Figure 2(a) and 2(b) compare the communication-efficiency for all four algorithms. Consistent with previous

[2]The standard mixing rate of Markov chain is modified to better simulate the combination step in (2).

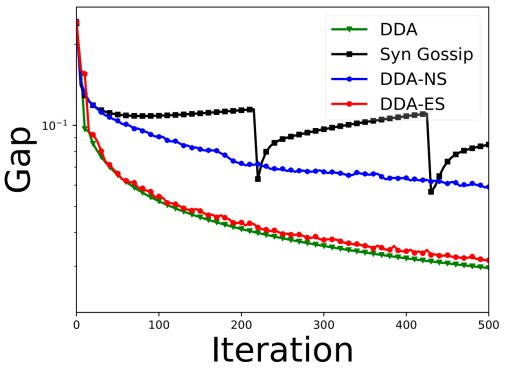
[3]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/



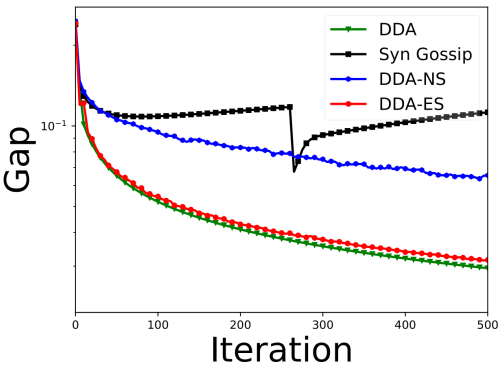
(a) Convergence rate of communication rounds. k-connected cycle, n = 1000, k= 75, z= 0.25.



(b) Convergence rate of communication rounds. Random graph, n = 1000, c= 0.15, z= 0.05.



(c) Convergence rate of natural iterations. k-connected cycle, n = 1000, k= 75, z= 0.25.



(d) Convergence rate of natural iterations. Random graph, n = 1000, c= 0.15, z= 0.05.

Figure 2: Distributed binary classification on different networks. The gap refers to $f(\hat{\mathbf{w}}_1^{T'}) - f(\mathbf{w}^*)$.

Markov chain experiments, DDA-ES enjoys the best communication utilization and achieves the fastest convergence rate when given the same communication budget. In the meanwhile, we also observe that DDA-ES algorithm suffers from a lower variance when compared with Syn Gossip and DDA-ES algorithms.

Figure 2(c) and 2(d) illustrate the convergence rate in terms of natural iterations. As can be observed from the figures, the standard DDA algorithm obtains the fastest convergence rate in this case since it combines information from all neighbors on each round. In the meanwhile, we can also observe that the DDA-ES algorithm converges almost exactly like the standard DDA even equipped with a relatively lower sampling rate and less communication on each round, indicating the sampling technique can efficiently reduce redundant communication and lead to better utilization of information exchange. Its competing sampling algorithms, Syn Gossip and DDA-NS, cannot match its performance in both graphs. In particular, due to the lowest sampling rate, $f(\hat{\mathbf{w}}_1^{T'}) - f(\mathbf{w}^*)$ for Syn Gossip algorithm only occasionally drops down when information exchange is executed, while for most of the time, it only minimizes towards its own local function and therefore performs poorly on the overall datasets.

5 Conclusion

In this paper, we propose and analyze a decentralized optimization algorithm based on a new sampling strategy named "edge sampling". This strategy provides an unbiased i.i.d. estimation of the initial combination matrix, while significantly reduce the communication cost on dense and well-connected networks. A comparable convergence rate is still preserved and even outperform the baseline algorithm when giving the same communication budget. When compared with existing node sampling algorithms, our strategy shows its superiorities for suffering from less sampling variance and being more communication-efficient, which are further validated by both theoretical analysis for its convergence rate and numerical experiments on the mixing rates of Markov chain and distributed machine learning problems.

Acknowledgements

This research is partially supported by The Joint NTU-WeBank Research Centre of Eco-Intelligent Applications (THEIA), Nanyang Technological University, Singapore.This research is also partially supported by the Singapore Government's Research, Innovation and Enterprise 2020 Plan, Advanced Manufacturing and Engineering domain (Programmatic Grant No. A18A1b0045).

References

[Balcan *et al.*, 2012] Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, pages 26–1, 2012.

[Boyd *et al.*, 2004] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM review*, 46(4):667–689, 2004.

[Boyd *et al.*, 2006] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

[Cao *et al.*, 2013] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2013.

[Chunlin and Layuan, 2006] Li Chunlin and Li Layuan. A distributed multiple dimensional qos constrained resource scheduling optimization policy in computational grid. *Journal of Computer and System Sciences*, 72(4):706–726, 2006.

[Duchi *et al.*, 2012] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

[Erdös and Rényi, 1960] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(17-61):43, 1960.

[Hastings, 1970] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[Iyengar *et al.*, 2004] S Sitharama Iyengar, Richard R Brooks, et al. *Distributed sensor networks*. CRC press, 2004.

[Levin and Peres, 2017] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[Li *et al.*, 2014] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *OSDI*, volume 14, pages 583–598, 2014.

[Metropolis *et al.*, 1953] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[Necoara, 2013] Ion Necoara. Random coordinate descent algorithms for multi-agent convex optimization over networks. *IEEE Transactions on Automatic Control*, 58(8):2001–2012, 2013.

[Nedić and Olshevsky, 2015] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

[Nedic *et al.*, 2017] Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[Olfati-Saber and Sandell, 2008] Reza Olfati-Saber and Nils F Sandell. Distributed tracking in sensor networks with limited sensing range. In *American Control Conference, 2008*, pages 3157–3162. IEEE, 2008.

[Rabbat and Nowak, 2004] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 20–27. ACM, 2004.

[Ram *et al.*, 2010] S Sundhar Ram, A Nedić, and Venugopal V Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147(3):516–545, 2010.

[Sayed and others, 2014] Ali H Sayed et al. Adaptation, learning, and optimization over networks. *Foundations and Trends® in Machine Learning*, 7(4-5):311–801, 2014.

[Shi *et al.*, 2015] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[Spielman and Srivastava, 2011] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[Spielman and Teng, 2011] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[Tsitsiklis *et al.*, 1986] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

[Tsitsiklis, 1984] John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, DTIC Document, 1984.

[Woodruff and Zhang, 2017] David P Woodruff and Qin Zhang. When distributed computation is communication expensive. *Distributed Computing*, 30(5):309–323, 2017.

[Yuan *et al.*, 2016] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[Zhang *et al.*, 2013] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems*, pages 2328–2336, 2013.