2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Fully Stochastic Distributed Convex Optimization on Time-Varying Graph with Compression

Chung-Yiu Yau and Hoi-To Wai[1]

Abstract— **This paper develops a fully stochastic proximal primal-dual (FSPPD) algorithm for distributed convex optimization. At each iteration, the distributed algorithm has agents communicating on a randomly drawn graph and applies random sparsification on the transmitted messages, while the agents only have access to a stochastic gradient oracle. To our best knowledge, this is the first compression-enabled distributed stochastic gradient algorithm on random graphs utilizing the primal-dual framework. With diminishing step size, we show that the FSPPD algorithm converges almost surely to an optimal solution of the strongly convex optimization problem. Numerical experiments are provided to verify our results.**

I. INTRODUCTION

Consider the optimization problem:

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}_i) \ \text{ s.t. } \ \mathbf{x}_i=\mathbf{x}_j,\ \forall\, i,j, \tag{1}$$

where each of the function $f_i:\mathbb{R}^d\to\mathbb{R}$ is strongly convex and continuously differentiable. Distributed optimization algorithms for (1) have become the major working horse behind many applications ranging from wireless sensor networks [1] to large-scale machine learning [2]. The move from traditional centralized to distributed computation has enabled these applications to scale up through utilizing the computation or data resources available at the agents. Particularly for algorithms that allow decentralized computation over a network/graph, they remove the reliance on a centralized server and allow the agents to participate dynamically in the joint optimization process.

Distributed algorithms that blend gradient descent methods with gossip (peer-to-peer) communication was first proposed in [3] through extending [4]. They have been actively developed. A number of works have focused on improving the theoretical properties of the algorithm: [5] extended the analysis for the algorithm to the stochastic optimization setting, also see [6] for a recent treatment on the topic for large-scale machine learning; [7], [8] considered incorporating gradient tracking to accelerate convergence; [9] studied an extension to directed graphs, also see [10], [11] for time varying graphs and stochastic optimization, etc.

As distributed algorithms are deployed to optimize *high-dimensional models* (e.g., weights of a deep neural network, linear models with large number of features), the delays caused by network overheads during the peer-to-peer communication steps can be a significant setback. Common strategies include balancing the frequency of communication steps with the optimization step [12] or adapting the algorithms to the constraints set by the network environment. The latter results in distributed algorithms that blend with communication using either compression [13] *or* time-varying/random topology [11], [14].

This paper aims at developing a communication efficient stochastic algorithm that adapts *simultaneously* to random topology and asynchronous computation at the agents, while respecting the limited bandwidth on network through compression. Recent works have proposed algorithms that partially enjoy the above features: [15] treated the peer-to-peer communication links as multi-layer graph for each coordinate of the model but requires multiple gossip steps per iteration, [16] used a two timescale updates scheme that may slow down convergence. Moreover, their algorithms are limited to exact gradient updates. We also remark that SwarmSGD [17] allows for compressed stochastic optimization on time-varying graph in a distributed setting, but it requires the local loss functions to satisfy a similarity condition.

We depart from the prior works by developing a *fully stochastic* primal-dual (FSPPD) algorithm solving a general stochastic optimization problem while relying on *random compression and communication graph*. Our development is based on the general primal-dual framework that naturally arises from (1), and has motivated the proximal primal-dual algorithms [18], [19] that enjoy good convergence properties in the deterministic computation setting. Notably this framework may also include the algorithms in [7], [8] as special cases. Our contributions can be summarized as:

- The FSPPD algorithm utilizes a new reformulation of the consensus optimization problem (1) as one with a *stochastic linear equality constraint*. This formulation enables us to develop FSPPD through utilizing the recent results from [20], [21] that study a class of forward-backward algorithms with stochastic operators.
- To our best knowledge, the FSPPD algorithm is the first algorithm that can simultaneously adapt to time varying (random) communication graph, compression with random sparsification, and stochastic gradient samples. Furthermore, it features a natural single-loop, asynchronous implementation where agents are not required to perform multiple gossiping steps nor to participate in distributed computation at each iteration.
- Under diminishing step sizes, we show that the FSPPD algorithm converges almost surely to the optimal solution of (1) and the agents local iterates attain consensus.

[1] Chung-Yiu Yau and Hoi-To Wai are with Department of SEEM, The Chinese University of Hong Kong, Shatin, Hong Kong SAR. Emails: cyyau@se.cuhk.edu.hk, htwai@se.cuhk.edu.hk. This research is supported in part by project #MMT-p5-23 of the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong.

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

We remark that the single-loop nature of the FSPPD algorithm gives a significant advantage over [15], [16] as our algorithm does not enforce near-consensus intentionally during the iterations. Instead, it relies on the stochastic constraint to slowly drive the iterates to consensus. Lastly, we provide numerical evidence to support our findings.

Notations. For matrix $\mathbf{A}$, denote $|\mathbf{A}|$ as the element-wise absolute value of $\mathbf{A}$, $\mathbf{A}_{i:j}$ as the row-block from ith row to jth row, $\mathrm{diag}(\mathbf{A})$ for the diagonal of $\mathbf{A}$ as a vector and $\mathrm{Diag}(\mathbf{A})$ for a diagonal matrix of $\mathbf{A}$. We use $\mathbf{1}$ as an all-one vector, $\mathbb{1}(\cdot) \in \{0, 1\}$ as indicator function and $\mathbf{I}_d$ as the d-dimensional identity matrix. Operators $\otimes$ and $\odot$ represent Kronecker product and Hadamard product respectively. We write $[n] := \{1, \ldots, n\}$ as the set of first n positive integers.

II. PROBLEM STATEMENT

We are concerned with the distributed optimization setting for (1) where each f_i is held by an agent in a network with n agents. The network is described by a connected, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = [n]$ represents the set of agents. The graph $\mathcal{G}$ is endowed with an incidence matrix $\mathbf{A} \in \mathbb{R}^{E \times n}$ where $E = |\mathcal{E}|$. To describe the matrix $\mathbf{A}$, we define an ordering of the set $\mathcal{E}$ and index map $\iota : \mathcal{E} \to [E]$ such that $(i, j) \in \mathcal{E}$ is the $\iota(i, j)$-th element of $\mathcal{E}$. For each $(i, j) \in \mathcal{E}, i < j$, the $\iota(i, j)$-th row of $\mathbf{A}$ is

$$\mathbf{A}_{\iota(i,j),:} = \boldsymbol{e}_i - \boldsymbol{e}_j, \tag{2}$$

where $\boldsymbol{e}_i$ is the ith canonical basis vector of $\mathbb{R}^n$.

Define the concatenated solution $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, we observe that as $\mathcal{G}$ is connected, (1) is equivalent to

$$\min_{\mathbf{x} \in \mathbb{R}^{nd}} \left[f(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}_i) \right] \quad \text{s.t.} \quad (\mathbf{A} \otimes \mathbf{I}_d)\mathbf{x} = \mathbf{0}, \tag{3}$$

where each agent holds a local solution $\mathbf{x}_i \in \mathbb{R}^d$. Notice that solution methods to (3) have been studied extensively. Existing approaches include primal-only algorithms that aim at mimicking the (centralized) gradient method applied to (1), this includes decentralized (stochastic) gradient descent [3], [5], EXTRA [8], DIGing [11], gradient tracking [7], etc. As an alternative, primal-dual algorithms that finds a saddle point to the Lagrangian function of (3) are also popular, e.g., ADMM [18], Prox-PDA [19].

This paper aims at handling two challenges in solving (3). First, we consider the case of *stochastic optimization* where exact evaluation of $f_i(\mathbf{x}_i)$ or its gradient is intractable; formally, we consider a probability space given by $(\Omega, \mathcal{F}, \mathbb{P})$ and describe the objective function as

$$f_i(\mathbf{x}_i) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[f_i(\mathbf{x}_i; \xi_i)] \tag{4}$$

where $\xi \in \Omega$. The algorithm only has access to a *stochastic* gradient given by $\nabla f_i(\mathbf{x}_i; \xi_i)$ that is available locally at agent i. Second, as practical communication networks are prone to unstable connections and low bandwidth, we seek for distributed algorithm that supports compression and asynchronous communication. We hence model an architecture such that the *instantaneous agent-to-agent communication* is given by a random subgraph of $\mathcal{G}$ and the edge-wise consensus requirement applies to a random subset of coordinates. In other words, the linear constraint in (3) can be interpreted as a *stochastic linear equality*:

$$\mathbb{E}_\xi[\widetilde{\mathbf{A}}(\xi)]\mathbf{x} := \mathbb{E}_\xi[\widetilde{\mathbf{I}}(\xi)(\mathbf{A} \otimes \mathbf{I}_d)]\mathbf{x} = \mathbf{0} \tag{5}$$

where $\widetilde{\mathbf{I}}(\xi)$ is a binary diagonal matrix controlling the instantaneous activation of the edges of $\mathcal{G}$ and the random choice of coordinates. Note that with a slight abuse of notation, we use the same state variable $\xi \in \Omega$ to represent the randomness in $\nabla f(\bar{\mathbf{x}}; \xi)$ and $\widetilde{\mathbf{A}}(\xi)$.

Eq. (4), (5) render (3) into a *stochastic equality constrained stochastic optimization* problem. In the sequel, we will develop a fully stochastic proximal primal-dual (FSPPD) algorithm that yields a distributed algorithm for the randomized computation and communication architecture.

III. ALGORITHM DEVELOPMENT

This section develops the FSPPD algorithm for solving (3) using a primal-dual formulation. To this end, let us observe the augmented Lagrangian function of (3). Denote the dual variable $\lambda = (\lambda_1, ..., \lambda_E) \in \mathbb{R}^{Ed}$, we have

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \langle \lambda \mid (\mathbf{A} \otimes \mathbf{I}_d)\mathbf{x} \rangle + \frac{1}{2} \|(\mathbf{A} \otimes \mathbf{I}_d)\mathbf{x}\|^2. \tag{6}$$

The aim of primal dual algorithms for (6) is to find the min-max saddle point solution to the following:

$$\min_{\mathbf{x} \in \mathbb{R}^{nd}} \max_{\lambda \in \mathbb{R}^{Ed}} \mathcal{L}(\mathbf{x}, \lambda) \tag{7}$$

by alternating between the updates of primal and dual variables. For example, Prox-GPDA [19] used a properly designed proximal primal-dual update that lead to a distributed algorithm. Notice that as discussed in [2], this primal-dual algorithm structure can be reduced to other popular distributed algorithms such as EXTRA [8].

The first ingredient of the FSPPD algorithm is to treat the stochastic objective function (4) and (consensus) constraint (5) via the following *stochastic linearized augmented Lagrangian* function:

$$\begin{aligned} \widetilde{\mathcal{L}}(\mathbf{x}, \lambda; \bar{\mathbf{x}}, \xi) = & \left\langle (\widetilde{\mathbf{C}}(\xi) \otimes \mathbf{I}_d) \nabla f(\bar{\mathbf{x}}; \xi) \;\middle|\; \mathbf{x} - \bar{\mathbf{x}} \right\rangle \\ & + \left\langle \lambda \;\middle|\; \widetilde{\mathbf{A}}(\xi)\mathbf{x} \right\rangle + \frac{\gamma}{2} \|\widetilde{\mathbf{A}}(\xi)\mathbf{x}\|^2, \end{aligned} \tag{8}$$

where the matrix $\widetilde{\mathbf{C}}(\xi) \in \mathbb{R}^{n \times n}$ is a diagonal matrix that controls the activation of agent i to be defined in (16), and $\gamma > 0$ stands for the consensus step size. The first term of (8) is a *linearized and sampled* objective $f(\mathbf{x}; \xi)$, see (4); while the last two terms pertain to *sampled* constraint (5).

The second ingredient of the FSPPD algorithm is a stochastically weighted proximal primal-dual update. Notice that applying plain proximal primal dual algorithms [22] on (8) may result in a non-distributed algorithm due to the quadratic coupling in the last term of (8). We are inspired by Prox-GPDA [19] to design the following update scheme that deploy a *stochastically weighted proximal term*. Let ξ^{t+1} be the global random state variable drawn at iteration $t+1$:

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^{nd}} \widetilde{\mathcal{L}}(\mathbf{x}, \lambda^t; \mathbf{x}^t, \xi^{t+1}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^t\|^2_{\mathbf{B}(\xi^{t+1})} \tag{9a}$$

$$\lambda^{t+1} = \lambda^t + \eta_{t+1}\nabla_\lambda \widetilde{\mathcal{L}}(\mathbf{x}^t, \lambda; \mathbf{x}^t, \xi^{t+1}) \tag{9b}$$

where $\mathbf{B}(\xi^{t+1}) := \gamma|\widetilde{\mathbf{L}}(\xi^{t+1})| + \eta_{t+1}^{-1}\mathbf{I}$ with the random graph's Laplacian $\widetilde{\mathbf{L}}(\xi) := \widetilde{\mathbf{A}}(\xi)^\top \widetilde{\mathbf{A}}(\xi)$, and $\eta_t > 0, t \in \mathbb{N}^+$ is a diminishing step size sequence. Importantly, the stochastic weight matrix $\mathbf{B}(\xi^{t+1})$ is chosen such that (9a) admits a distributed solution since

$$\widetilde{\mathbf{L}}(\xi^{t+1}) + |\widetilde{\mathbf{L}}(\xi^{t+1})| = 2 \ \mathrm{Diag}(\widetilde{\mathbf{L}}(\xi^{t+1})). \tag{10}$$

We remark another subtle difference between (9) and Prox-GPDA: (9) adopts a Jacobi type update that both primal and dual variables are computed simultaneously, while Prox-GPDA adopts a Gauss-Seidal type update.

Communication Efficient Implementation. We specify the diagonal binary matrix $\widetilde{\mathbf{I}}(\xi)$ in (5) and its implication on the actual communication protocol used by `FSPPD`. To this end, the diagonal elements of $\widetilde{\mathbf{I}}(\xi)$ are binary variables that control if consensus shall be enforced on a coordinate for the variable of the two nodes incident to an edge. We illustrate our design principle by incorporating *random sparsification* and *random graph*.

A simple strategy to improve communication efficiency of decentralized optimization is to apply random sparsification to the transmitted messages in the communication step. In particular, for each d-dimensional message $\mathbf{x}$, we sample a random subset $\mathcal{S}(\xi) \subseteq [d]$ and apply the compressor:

$$\mathcal{Q}(\mathbf{x}; \mathcal{S}(\xi)) = \mathbf{1}_{\mathcal{S}(\xi)} \odot \mathbf{x}, \tag{11}$$

where $[\mathbf{1}_{\mathcal{S}(\xi)}]_i = 1$ if $i \in \mathcal{S}(\xi)$, otherwise $[\mathbf{1}_{\mathcal{S}(\xi)}]_i = 0$. Observe that the output $\mathcal{Q}(\mathbf{x}; \mathcal{S}(\xi))$ is a $|\mathcal{S}(\xi)|$-sparse vector. We also allow the agents to communicate on a *random subgraph* of $\mathcal{G}$, similar to the model in [14]. The set of active edges, neighborhood of agent i are denoted as $\mathcal{E}(\xi), \mathcal{N}_i(\xi)$, respectively, such that $\mathcal{E}(\xi) \subseteq \mathcal{E}, \mathcal{N}_i(\xi) \subseteq \mathcal{N}_i$.

Importantly, both random sparsification and random graph topology can be incorporated *simultaneously* in `FSPPD` as the consensus constraint can be replaced by any stochastic linear equality satisfying (5). For the random diagonal matrix $\widetilde{\mathbf{I}}(\xi) \in \mathbb{R}^{Ed\times Ed}$, we set

$$\mathrm{diag}(\widetilde{\mathbf{I}}(\xi)) = \widetilde{\mathbf{s}}(\xi) \odot \widetilde{\boldsymbol{\alpha}}(\xi) \tag{12}$$

where for each $(i,j) \in \mathcal{E}$, the $\iota(i,j)$-th block for the binary vectors is given by

$$[\widetilde{\mathbf{s}}(\xi)]_{\iota(i,j)} = \mathbf{1}_{\mathcal{S}_{\iota(i,j)}(\xi)}, \quad [\widetilde{\boldsymbol{\alpha}}(\xi)]_{\iota(i,j)} = \widetilde{\alpha}_{\iota(i,j)}\mathbf{1}_{[d]} \tag{13}$$

such that $\mathcal{S}_{\iota(i,j)}(\xi) \subseteq [d]$ is a set of random coordinates sampled for the edge (i,j) and $\widetilde{\alpha}_{\iota(i,j)}$ is Bernoulli random variable such that

$$\mathbb{P}[k \in \mathcal{S}_{\iota(i,j)}(\xi)] = \omega_{\iota(i,j)}, \tag{14}$$

$$\mathbb{E}[\widetilde{\alpha}_{\iota(i,j)}] = \mathbb{P}[(i,j) \in \mathcal{E}(\xi)] = \alpha_{\iota(i,j)}, \tag{15}$$

for any $k \in [d], (i,j) \in \mathcal{E}$. Under (12), the random linear equation $\widetilde{\mathbf{I}}(\xi)(\mathbf{A} \otimes \mathbf{I}_d)\mathbf{x} = \mathbf{0}$ enforces consensus over the randomly picked coordinates and edges of $\mathcal{G}$. Define the i-th diagonal element of the control matrix in (8) as

$$[\widetilde{\mathbf{C}}(\xi)]_{i,i} = \frac{\mathbb{1}(|\mathcal{N}_i(\xi)| > 0)}{\mathbb{P}(|\mathcal{N}_i(\xi)| > 0)}, \ \ i \in [n]. \tag{16}$$

Agent i is *inactive* if none of its incident edges are selected.

Algorithm 1 `FSPPD` Algorithm

```
input: $\mathbf{x}_i^0$ for $i \in [n]$, step size sequence $\eta_t > 0$ and consensus step size $\gamma > 0$.
for $t = 0, 1, ..., T-1$ do
  Sample $\xi^{t+1}$ and compute $\nabla f(\mathbf{x}^t; \xi^{t+1})$.
  Agent $i$ determine $\mathcal{S}_{\iota(i,j)}(\xi^{t+1})$ if $(i,j) \in \mathcal{E}(\xi^{t+1})$.
  for each $(i,j) \in \mathcal{E}(\xi^{t+1})$ do
    Agent $i$ sends sparse index-value pair $\mathcal{S}_{\iota(i,j)}(\xi^{t+1}), (\mathbf{x}_{i,k}^t)_{k\in\mathcal{S}_{\iota(i,j)}(\xi^{t+1})}$ to agent $j$.
    Agent $j$ responds with values $(\mathbf{x}_{j,k}^t)_{k\in\mathcal{S}_{\iota(i,j)}(\xi^{t+1})}$.
  end for
  for $i = 1, ..., n$ do
    $\mathbf{x}_i^{t+\frac{1}{2},1} = \sum_{j\in\mathcal{N}_i(\xi^{t+1})} \mathrm{sign}(i<j) \cdot \mathcal{Q}_{\iota(i,j)}^{t+1}\left(\lambda_{\iota(i,j)}^t\right)$
    $\mathbf{x}_i^{t+\frac{1}{2},2} = \sum_{j\in\mathcal{N}_i(\xi^{t+1})} \mathcal{Q}_{\iota(i,j)}^{t+1}\left(\mathbf{x}_i^t\right) + \mathcal{Q}_{\iota(i,j)}^{t+1}\left(\mathbf{x}_j^t\right)$
    $\mathbf{x}_i^{t+\frac{1}{2}} = \mathbf{x}_i^t - \eta_{t+1}\big[[\widetilde{\mathbf{C}}(\xi^{t+1})]_{i,i}\nabla f_i(\mathbf{x}_i^t; \xi^{t+1}) + \mathbf{x}_i^{t+\frac{1}{2},1} - \gamma\mathbf{x}_i^{t+\frac{1}{2},2}\big]$
    $\mathbf{x}_{i,k}^{t+1} = \mathbf{x}_{i,k}^{t+\frac{1}{2}}/(1 + 2\gamma\eta_{t+1}\sum_{j\in\mathcal{N}_i(\xi^{t+1})} \mathbb{1}(k \in \mathcal{I}_{\iota(i,j)}(\xi^{t+1}))$ for $k = 1, ..., d$
  end for
  for each $(i,j) \in \mathcal{E}(\xi^{t+1})$ where $i < j$ do
    $\lambda_{\iota(i,j)}^{t+1} = \lambda_{\iota(i,j)}^t + \eta_{t+1}[\mathcal{Q}_{\iota(i,j)}^{t+1}(\mathbf{x}_i^t) - \mathcal{Q}_{\iota(i,j)}^{t+1}(\mathbf{x}_j^t)]$
  end for
end for
output: Weighted average $(\sum_{k=0}^T \eta_k)^{-1}\sum_{k=0}^T \eta_k\mathbf{x}^k$
```

We next derive closed form updates for (9) and illustrate that these updates can be implemented in a distributed manner. Solving the optimality condition of (9a) leads to

$$\begin{aligned}\mathbf{0} &= (\widetilde{\mathbf{C}}(\xi^{t+1}) \otimes \mathbf{I}_d)\nabla f(\mathbf{x}^t; \xi^{t+1}) + \widetilde{\mathbf{A}}(\xi^{t+1})^\top\lambda^t \qquad (17)\\ &\quad + \gamma\widetilde{\mathbf{L}}(\xi^{t+1})\mathbf{x}^{t+1} + (\gamma|\widetilde{\mathbf{L}}(\xi^{t+1})| + \eta_{t+1}^{-1}\mathbf{I})(\mathbf{x}^{t+1} - \mathbf{x}^t)\\ \iff &\ (\eta_{t+1}^{-1}\mathbf{I} + 2\gamma \ \mathrm{Diag}(\widetilde{\mathbf{L}}(\xi^{t+1})))\mathbf{x}^{t+1} \qquad (18)\\ &= (\gamma|\widetilde{\mathbf{L}}(\xi^{t+1})| + \eta_{t+1}^{-1}\mathbf{I})\mathbf{x}^t - (\widetilde{\mathbf{C}}(\xi^{t+1}) \otimes \mathbf{I}_d)\nabla f(\mathbf{x}^t; \xi^{t+1})\\ &\quad - \widetilde{\mathbf{A}}(\xi^{t+1})^\top\lambda^t\end{aligned}$$

where the equivalence is due to (10). Meanwhile (9b) can be easily computed in closed form. The `FSPPD` algorithm is thus equivalent to:

$$\begin{aligned}\mathbf{x}^{t+\frac{1}{2}} &= \mathbf{x}^t - \eta_{t+1}\Big((\widetilde{\mathbf{C}}(\xi^{t+1}) \otimes \mathbf{I}_d)\nabla f(\mathbf{x}^t; \xi^{t+1}) \qquad (19a)\\ &\qquad + \widetilde{\mathbf{A}}(\xi^{t+1})^\top\lambda^t - \gamma|\widetilde{\mathbf{L}}(\xi^{t+1})|\mathbf{x}^t\Big)\\ \mathbf{x}^{t+1} &= \Big[\mathbf{I} + 2\gamma\eta_{t+1} \ \mathrm{Diag}(\widetilde{\mathbf{L}}(\xi^{t+1}))\Big]^{-1}\mathbf{x}^{t+\frac{1}{2}} \qquad (19b)\\ \lambda^{t+1} &= \lambda^t + \eta_{t+1}\widetilde{\mathbf{A}}(\xi^{t+1})\mathbf{x}^t. \qquad (19c)\end{aligned}$$

At iteration t, agent i holds the following variables: $\mathbf{x}_i^t$, $\lambda_{\iota(i,j)}^t$ for $j \in \mathcal{N}_i$. Under this setting, we claim

Algorithm 2 Asynchronous Implementation of FSPPD (From Agent i's Perspective)

```
1:  Assume $\nabla f_i(\mathbf{x}_i;\xi)$ is always ready. Denote $\mathcal{B},\mathcal{U}$ as communication buffers.
2:  Set iteration counters $g_i = s_i = 0$.
3:  while not optimal do
4:    Agent $i$ wakes up and resets $\mathcal{U} = \emptyset$.
5:    Select a random subset $\widetilde{\mathcal{N}}_i \subseteq \mathcal{N}_i$.
6:    Initialize the job buffer $\mathcal{B} \leftarrow \{b_{\iota(i,j)} \mid j \in \widetilde{\mathcal{N}}_i\}$, where $b_{\iota(i,j)}$ represents a sparsified comm. job between $i, j$.
7:    while $|\mathcal{U}| < |\widetilde{\mathcal{N}}_i|$ do
8:      Execute $b_{\iota(i,j)} \in \mathcal{B}$ in random order and wait until the communication job is successfully executed on $i, j$.
9:      Update $\mathcal{U} \leftarrow \mathcal{U} \cup \{b_{\iota(i,j)}\}$, $\mathcal{B} \leftarrow \mathcal{B} \backslash \{b_{\iota(i,j)}\}$, $g_i \leftarrow \max\{g_i, g_j\}$.
10:   end while
11:   If $|\mathcal{U}| > 0$, apply $g_i \leftarrow g_i + 1, s_i \leftarrow s_i + 1$ and update state variables as $(\mathbf{x}_i^{g_i}, \lambda_{\iota(i,\cdot)}^{g_i})$ according to (19).
12: end while
```

that (19) can be implemented distributively over the network. In particular, to compute $\mathbf{x}_i^{t+1}$ from (19a), (19b), agent i only needs the information: $\mathbf{x}_i^t$, $\nabla f_i(\mathbf{x}_i^t;\xi^{t+1})$, $[\widetilde{\mathbf{A}}(\xi^{t+1})^\top]_{id:(i+1)d}\lambda^t$ and $|\widetilde{\mathbf{L}}(\xi^{t+1})|_{id:(i+1)d}\mathbf{x}^t$. With the notation $\mathcal{Q}_{\iota(i,j)}^{t+1}(\mathbf{x}) = \mathcal{Q}(\mathbf{x};\mathcal{S}_{\iota(i,j)}(\xi^{t+1}))$, we note that (i) $[\widetilde{\mathbf{A}}(\xi^{t+1})^\top]_{id:(i+1)d}\lambda^t$ is a linear combination of the *sparsified* local dual variables $\{\mathcal{Q}_{\iota(i,j)}^{t+1}(\lambda_{\iota(i,j)}^t) \mid j \in \mathcal{N}_i(\xi^{t+1})\}$, and (ii) $|\widetilde{\mathbf{L}}(\xi^{t+1})|_{id:(i+1)d}\mathbf{x}^t$ coincides with the sum of *sparsified* local decision variables $\{\mathcal{Q}_{\iota(i,j)}^{t+1}(\mathbf{x}_j^t) \mid j \in \mathcal{N}_i(\xi^{t+1})\}$. Similarly, in the update of (19c), the dual variables λ^{t+1} use the same information from the above sparsified communication. The details are summarized in Algorithm 1. We remark that the distributed computation architecture is due to the design of weighted proximal term in (9a), (10).

Asynchronous Implementation. FSPPD can be implemented in an asynchronous fashion where agents stay idle if they are not incident to any selected edges in the random graph. This can be achieved through assigning zeros to the diagonal matrix $\widetilde{\mathbf{C}}(\xi)$ in (16). An example implementation is given in Algorithm 2. As FSPPD requires synchronized dual variable $\lambda_{\iota(i,j)}$ on the adjacent agents, we enforce in line 9, 11 that a pair of agents must be consensual on applying the exchanged sparse parameters, i.e., the two copies of $\lambda_{\iota(i,j)}$ are consensual and received parameters from neighbors must be applied to local state variables through (19). Besides, computing $\widetilde{\mathbf{C}}(\xi)$ requires knowledge of the probability $\mathbb{P}(|\mathcal{N}_i(\xi)| > 0)$. Taking Algorithm 2 as an example, agent i can approximate the latter as

$$\mathbb{P}(|\mathcal{N}_i(\xi)| > 0) \approx s_i/g_i, \tag{20}$$

where s_i, g_i are the local and global iteration counters.

IV. CONVERGENCE ANALYSIS

In this section, we show that FSPPD converges asymptotically to an optimal solution of (1). Our analysis strategy follows that of [21] and utilizes [20, Corollary 3.1]. Observe the following assumptions:

Assumption IV.1 (Stochastic Gradient)**.** *There exists constants $\sigma_0, \sigma_1 \geq 0$ such that for any $i \in [n]$ and fixed $\mathbf{x}$*

$$\mathbb{E}_\xi[\nabla f_i(\mathbf{x};\xi)] = \nabla f_i(\mathbf{x}), \tag{21}$$

$$\sup_{\xi\in\Omega} \|\nabla f(\mathbf{x};\xi)\| \leq \sigma_0 + \sigma_1\|\mathbf{x}\|. \tag{22}$$

Assumption IV.2 (Strong Convexity)**.** *For each $i = 1,\ldots,n$, the local objective function $f_i(\mathbf{x})$ is μ_i-strongly convex. We define $\mu = \min_{i\in[n]} \mu_i > 0$.*

Note that Assumption IV.1 is a standard condition which states that each agent has access to an unbiased stochastic gradient oracle of the local objective function and the latter satisfies a growth condition. On the other hand, Assumption IV.2 is a standard strong convexity assumption.

Fixed Point of FSPPD. Our first task is to characterize the fixed point(s) found by the FSPPD algorithm (if the algorithm converges). Define the stochastic forward operator $\widetilde{F}_\xi$ and backward operator $\widetilde{B}_\xi$ by:

$$\widetilde{F}_\xi(\mathbf{x},\lambda) = \begin{bmatrix} (\widetilde{\mathbf{C}}(\xi)\otimes\mathbf{I}_d)\nabla f(\mathbf{x};\xi) + \widetilde{\mathbf{A}}(\xi)^\top\lambda - \gamma|\widetilde{\mathbf{L}}(\xi)|\mathbf{x} \\ -\widetilde{\mathbf{A}}(\xi)\mathbf{x} \end{bmatrix}$$

$$\widetilde{B}_\xi(\mathbf{x},\lambda) = \begin{bmatrix} 2\gamma\ \mathrm{Diag}(\widetilde{\mathbf{L}}(\xi))\mathbf{x} \\ \mathbf{0} \end{bmatrix}, \tag{23}$$

From (19), we observe that the FSPPD algorithm generates a sequence $(\mathbf{x}^t, \lambda^t)$ following

$$\begin{bmatrix} \mathbf{x}^{t+1} \\ \lambda^{t+1} \end{bmatrix} = (\mathrm{Id} + \eta_{t+1}\widetilde{B}_{\xi^{t+1}})^{-1} \left[\begin{bmatrix} \mathbf{x}^t \\ \lambda^t \end{bmatrix} - \eta_{t+1}\widetilde{F}_{\xi^{t+1}}(\mathbf{x}^t,\lambda^t) \right] \tag{24}$$

where Id is the identity operator.

When the operators in (24) are deterministic, e.g., they are replaced by $\mathsf{F}(\mathbf{x},\lambda), \mathsf{B}(\mathbf{x},\lambda)$ to facilitate our discussion, it is known that the *deterministic* forward backward algorithm (24) converges to a fixed point given by the zeros of $\mathbf{0} = \mathsf{F}(\mathbf{x},\lambda) + \mathsf{B}(\mathbf{x},\lambda)$ [22, Proposition 50]. Intuitively, the *stochastic* algorithm (24) should also converge to the following set of primal-dual solution:

$$\mathcal{Z}^\star := \left\{ (\mathbf{x}^\star,\lambda^\star) : \mathbb{E}\left[\widetilde{F}_\xi(\mathbf{x}^\star,\lambda^\star) + \widetilde{B}_\xi(\mathbf{x}^\star,\lambda^\star) \right] = \mathbf{0} \right\}. \tag{25}$$

This is the case as shown in Theorem IV.3. Before we discuss the convergence result, let us examine (25). We first notice that

$$\mathbb{E}[\widetilde{\mathbf{A}}(\xi^t)] = (\mathbf{R}\otimes\mathbf{I}_d)(\mathbf{A}\otimes\mathbf{I}_d), \tag{26}$$

where the rate matrix $\mathbf{R} \in \mathbb{R}^{E\times E}$ is diagonal with

$$\mathrm{diag}(\mathbf{R})_{\iota(i,j)} = \alpha_{\iota(i,j)}\omega_{\iota(i,j)}. \tag{27}$$

By (16), we have $\mathbb{E}[\widetilde{\mathbf{C}}(\xi)] = \mathbf{I}$ since $\mathbb{E}[\mathbb{1}(|\mathcal{N}_i(\xi)| > 0)] = \mathbb{P}(|\mathcal{N}_i(\xi)| > 0)$. Using (10), (26) and the fact

$$\widetilde{\mathbf{L}}(\xi^{t+1}) = \widetilde{\mathbf{A}}(\xi^{t+1})^\top\widetilde{\mathbf{A}}(\xi^{t+1}) = (\mathbf{A}\otimes\mathbf{I}_d)^\top\widetilde{\mathbf{A}}(\xi^{t+1}),$$

we observe that any $(\mathbf{x}^\star,\lambda^\star) \in \mathcal{Z}^\star$ satisfies

$$\begin{bmatrix} \nabla f(\mathbf{x}^\star) + ((\mathbf{R}\mathbf{A})\otimes\mathbf{I}_d)^\top\lambda^\star + \gamma((\mathbf{A}^\top\mathbf{R}\mathbf{A})\otimes\mathbf{I}_d)\mathbf{x}^\star \\ -((\mathbf{R}\mathbf{A})\otimes\mathbf{I}_d)\mathbf{x}^\star \end{bmatrix} = \mathbf{0},$$

which is equivalent to

$$\begin{cases}\nabla f(\mathbf{x}^\star) + ((\mathbf{R}\mathbf{A}) \otimes \mathbf{I}_d)^\top \lambda^\star = \mathbf{0} \\ \mathbf{x}_i^\star = \mathbf{x}_j^\star \ \forall (i,j) \in \mathcal{E} \end{cases} \tag{28}$$

$$\Longrightarrow \textstyle\sum_{i=1}^n \nabla f_i(\mathbf{x}_1^\star) = \mathbf{0},\ \mathbf{x}_j^\star = \mathbf{x}_1^\star,\ j = 1, ..., n,$$

where the last implication uses the fact that row sums of $\mathbf{A}$ are 0, thus $(\mathbf{1}_n \otimes \mathbf{I}_d)^\top((\mathbf{R}\mathbf{A}) \otimes \mathbf{I}_d)^\top = ((\mathbf{R}\mathbf{A}\mathbf{1}_n)^\top \otimes \mathbf{I}_d) = \mathbf{0}$. By convexity of f, $\mathbf{x}_j^\star$ is an optimal solution of (1).

Finally, we show that a weighted average iterate of `FSPPD` converges to an optimal solution of (3).

Theorem IV.3. *Assume $\mathcal{Z}^\star \neq \emptyset$, the step size conditions $\eta_{t+1}/\eta_t \to 1$, $\sum_{t=1}^\infty \eta_t \to \infty$ and $\sum_{t=1}^\infty \eta_t^2 < \infty$, and $\gamma \leq \mu / \lambda_{\max}(|\mathbf{A}|^\top \mathbf{R} |\mathbf{A}|)$. Then, for any initialization $(\mathbf{x}^0, \lambda^0)$, the weighted average iterate of Algorithm 1:*

$$\bar{\mathbf{z}}^t = (\bar{\mathbf{x}}^t, \bar{\lambda}^t) = (\textstyle\sum_{r=0}^t \eta_r)^{-1} \sum_{r=0}^t \eta_r (\mathbf{x}^r, \lambda^r) \tag{29}$$

converges almost surely to a point in $\mathcal{Z}^\star$ (25). In particular, the limit point of $\bar{\mathbf{x}}^t$ is a point satisfying (28).

Proof. It suffices to verify that the stochastic operators $\widetilde{F}_\xi, \widetilde{B}_\xi$ of `FSPPD` (24) satisfy the assumptions in Theorem 3.1 of [20], and subsequently applying Corollary 3.1 therein yields the proof. To fix ideas, we denote the expected operators $\mathsf{F} = \mathbb{E}[\widetilde{F}_\xi]$, $\mathsf{B} = \mathbb{E}[\widetilde{B}_\xi]$, with

$$\mathsf{F}(\mathbf{x}, \lambda) \overset{(i)}{=} \begin{bmatrix} \nabla f(\mathbf{x}) + ((\mathbf{R}\mathbf{A}) \otimes \mathbf{I}_d)^\top \lambda - \gamma((|\mathbf{A}|^\top \mathbf{R} |\mathbf{A}|) \otimes \mathbf{I}_d)\mathbf{x} \\ -((\mathbf{R}\mathbf{A}) \otimes \mathbf{I}_d)\mathbf{x} \end{bmatrix} \tag{30}$$

$$\mathsf{B}(\mathbf{x}, \lambda) = \begin{bmatrix} 2\gamma \ \mathrm{Diag}((\mathbf{A}^\top \mathbf{R}\mathbf{A}) \otimes \mathbf{I}_d)\mathbf{x} \\ \mathbf{0} \end{bmatrix} \tag{31}$$

where (i) uses that $|\widetilde{\mathbf{L}}(\xi)| = (|\mathbf{A}| \otimes \mathbf{I}_d)^\top \widetilde{\mathbf{I}}(\xi)(|\mathbf{A}| \otimes \mathbf{I}_d) \Rightarrow \mathbb{E}[|\widetilde{\mathbf{L}}(\xi)|] = (|\mathbf{A}|^\top \mathbf{R}|\mathbf{A}|) \otimes \mathbf{I}_d$. The following discussions verify Conditions 1) to 6) in Theorem 3.1 of [20]:

1) The operator F is monotone because for any $(\mathbf{x}, \lambda), (\mathbf{x}', \lambda') \in \mathbb{R}^{nd} \times \mathbb{R}^{Ed}$,

$$\begin{aligned}
&\langle \mathsf{F}(\mathbf{x}', \lambda') - \mathsf{F}(\mathbf{x}, \lambda) \mid (\mathbf{x}', \lambda') - (\mathbf{x}, \lambda) \rangle \\
&= \left\langle \left(\begin{bmatrix} -\gamma |\mathbf{A}|^\top \mathbf{R} |\mathbf{A}| & (\mathbf{R}\mathbf{A})^\top \\ -\mathbf{R}\mathbf{A} & \mathbf{0} \end{bmatrix} \otimes \mathbf{I}_d \right) \begin{bmatrix} \mathbf{x}' - \mathbf{x} \\ \lambda' - \lambda \end{bmatrix} \middle| \begin{bmatrix} \mathbf{x}' - \mathbf{x} \\ \lambda' - \lambda \end{bmatrix} \right\rangle \\
&\quad + \left\langle \begin{bmatrix} \nabla f(\mathbf{x}') - \nabla f(\mathbf{x}) \\ \mathbf{0} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{x}' - \mathbf{x} \\ \lambda' - \lambda \end{bmatrix} \right\rangle && (32) \\
&= \langle -\gamma((|\mathbf{A}|^\top \mathbf{R} |\mathbf{A}|) \otimes \mathbf{I}_d)(\mathbf{x}' - \mathbf{x}) \mid \mathbf{x}' - \mathbf{x} \rangle \\
&\quad + \langle \nabla f(\mathbf{x}') - \nabla f(\mathbf{x}) \mid \mathbf{x}' - \mathbf{x} \rangle && (33) \\
&\overset{(i)}{\geq} -\gamma \| ((\mathbf{R}^{1/2} |\mathbf{A}|) \otimes \mathbf{I}_d)(\mathbf{x}' - \mathbf{x}) \|^2 + \mu \| \mathbf{x}' - \mathbf{x} \|^2 && (34) \\
&\geq \left(\mu - \gamma \| \ (\mathbf{R}^{1/2} |\mathbf{A}|) \otimes \mathbf{I}_d \ \|_2^2 \right) \| \mathbf{x}' - \mathbf{x} \|^2 \overset{(ii)}{\geq} 0 && (35)
\end{aligned}$$

where (i) uses strong convexity of $f_i \ \forall i \in [n]$; (ii) utilizes the step size condition $\gamma \leq \mu / \lambda_{\max}(|\mathbf{A}|^\top \mathbf{R} |\mathbf{A}|)$.

The operator B is monotone because for any $(\mathbf{x}, \lambda), (\mathbf{x}', \lambda') \in \mathbb{R}^{nd} \times \mathbb{R}^{Ed}$,

$$\begin{aligned}
&\langle \mathsf{B}(\mathbf{x}', \lambda') - \mathsf{B}(\mathbf{x}, \lambda) \mid (\mathbf{x}', \lambda') - (\mathbf{x}, \lambda) \rangle \\
&= 2\gamma \| \mathrm{Diag}((\mathbf{A}^\top \mathbf{R}\mathbf{A}) \otimes \mathbf{I}_d)^{1/2} (\mathbf{x}' - \mathbf{x}) \|^2 \geq 0 && (36)
\end{aligned}$$

Furthermore, both operators F, B are maximal.

2) We verify that there exists $p \geq 1$, $(\mathbf{x}^\star, \lambda^\star) \in \mathcal{Z}^\star$ which admits a $2p$-integrable representation. First observe that for any $(\mathbf{x}^\star, \lambda^\star) \in \mathcal{Z}^\star$,

$$\int \widetilde{F}_\xi(\mathbf{x}^\star, \lambda^\star) \mathbb{P}(\mathrm{d}\xi) + \widetilde{B}_\xi(\mathbf{x}^\star, \lambda^\star) \mathbb{P}(\mathrm{d}\xi) = \mathbf{0}, \tag{37}$$

which holds using the definition of $\mathcal{Z}^\star$ in (25). Besides,

$$\begin{aligned}
\int \left\| \widetilde{B}_\xi \begin{bmatrix} \mathbf{x}^\star \\ \lambda^\star \end{bmatrix} \right\|^4 \mathbb{P}(\mathrm{d}\xi) &= \int \| 2\gamma \ \mathrm{Diag}(\widetilde{\mathbf{L}}(\xi)) \mathbf{x}^\star \|^4 \mathbb{P}(\mathrm{d}\xi) \\
&\leq (2\gamma n)^4 \| \mathbf{x}^\star \|^4 < \infty, && (38)
\end{aligned}$$

and similar conclusion for $\widetilde{F}_\xi$ is shown in (42). This verifies the condition with $p = 2$.

3) For compact $K \subseteq \mathbb{R}^{nd} \times \mathbb{R}^{Ed}$, take $\epsilon = 1$ and observe

$$\begin{aligned}
&\sup_{(\mathbf{x}, \lambda) \in K \cap (\mathbb{R}^{nd} \times \mathbb{R}^{Ed})} \int \| \widetilde{B}_\xi(\mathbf{x}, \lambda) \|^2 \mathbb{P}(\mathrm{d}\xi) \\
&\leq \sup_{(\mathbf{x}, \lambda) \in K \cap (\mathbb{R}^{nd} \times \mathbb{R}^{Ed})} (2\gamma n)^2 \int \| \mathbf{x} \|^2 \mathbb{P}(\mathrm{d}\xi) < \infty. && (39)
\end{aligned}$$

4) Since the domain of $\widetilde{B}_\xi$ is $\mathbb{R}^{nd} \times \mathbb{R}^{Ed}$, the distance between $(\mathbf{x}, \lambda)$ and the domain of $\widetilde{B}_\xi$ is zero.

5) For any $\eta > 0, (\mathbf{x}, \lambda) \in \mathbb{R}^{nd} \times \mathbb{R}^{Ed}$, we have

$$\begin{aligned}
&\frac{1}{\eta^4} \int \| (\mathbf{I} + \eta \widetilde{B}_\xi)^{-1}(\mathbf{x}, \lambda) - (\mathbf{x}, \lambda) \|^4 \mathbb{P}(\mathrm{d}\xi) \\
&= \frac{1}{\eta^4} \int \| (\mathbf{I} + 2\gamma \ \mathrm{Diag}(\widetilde{\mathbf{L}}(\xi)))^{-1}(\mathbf{x}) - \mathbf{x} \|^4 \mathbb{P}(\mathrm{d}\xi) \\
&\leq \frac{1}{\eta^4} \int \left\| \frac{1}{1 + 2\gamma n} \mathbf{x} - \mathbf{x} \right\|^4 \mathbb{P}(\mathrm{d}\xi) \\
&< \left(\frac{2\gamma n}{\eta(1 + 2\gamma n)} \right)^4 (1 + \| \mathbf{x} \|^4) && (40)
\end{aligned}$$

6) For all $(\mathbf{x}, \lambda) \in \mathbb{R}^{nd} \times \mathbb{R}^{Ed}$, we observe

$$\begin{aligned}
&\| \widetilde{F}_\xi(\mathbf{x}, \lambda) \| && (41) \\
&\leq \left\| \begin{bmatrix} -\gamma |\widetilde{\mathbf{L}}(\xi)| & \widetilde{\mathbf{A}}(\xi)^\top \\ -\widetilde{\mathbf{A}}(\xi) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} \right\| + \| \widetilde{\mathbf{C}}(\xi) \nabla f(\mathbf{x}; \xi) \| \\
&\overset{(22)}{\leq} \left\| \begin{bmatrix} -\gamma |\widetilde{\mathbf{L}}(\xi)| & \widetilde{\mathbf{A}}(\xi)^\top \\ -\widetilde{\mathbf{A}}(\xi) & \mathbf{0} \end{bmatrix} \right\|_F \| (\mathbf{x}, \lambda) \| + \sigma_0 + \sigma_1 \| \mathbf{x} \| \\
&\leq \left(\sqrt{4Ed + 2\gamma n^2 d} + \sigma_1 \right) \ \| (\mathbf{x}, \lambda) \| + \sigma_0
\end{aligned}$$

Obviously, the above implies

$$\begin{aligned}
&\int \| \widetilde{F}_\xi(\mathbf{x}, \lambda) \|^4 \mathbb{P}(\mathrm{d}\xi) && (42) \\
&\leq 8 \left[\left(\sqrt{4Ed + 2\gamma n^2 d} + \sigma_1 \right)^4 \| (\mathbf{x}, \lambda) \|^4 + \sigma_0^4 \right]
\end{aligned}$$

This concludes the proof of Theorem IV.3. □

Theorem IV.3 states that the consensus step size γ should be bounded according to the strong convexity modulus μ and the network structure. When the rate matrix $\mathbf{R}$ increases, we observe that γ has to be decreased for normalizing the effects of the mixing term on $\mathbf{x}$ in (19a).

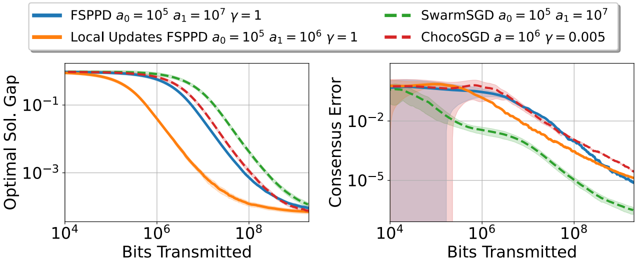

Fig. 1. Comparison under the stochastic gradient setting; cf. (A).

V. NUMERICAL EXPERIMENTS

We focus on a linear regression task and consider $\mathcal{G}$ as a fully connected graph of $n = 10$ agents. We first set $\mathbf{x}^{\rm tr} \in \mathbb{R}^{20}$ as a ground truth vector. For each $i = 1, \ldots, n$, each agent i holds a local objective function $f_i(\mathbf{x}, \xi_i) = |\mathbf{a}(\xi_i)^\top \mathbf{x} - y(\xi_i)|^2 + 10^{-4}\|\mathbf{x}\|_2^2$. The samples follow $\mathbf{a}(\xi_i) \sim \mathcal{N}(\mathbf{m}_i, 0.1\mathbf{I})$, $y(\xi_i) = \mathbf{a}(\xi_i)^\top \mathbf{x}^{\rm tr} + z(\xi_i)$ with $z(\xi_i) \sim \mathcal{N}(0, 0.1)$, where $\mathbf{m}_i$ is an agent-specfic mean. At iteration t, the agents communicate on random subgraphs $(\mathcal{V}, \mathcal{E}(\xi^{t+1}))$, where $\mathcal{E}(\xi^{t+1})$ is a singleton set as we only sample one edge from $\mathcal{G}$, thus $\mathbb{P}(|\mathcal{N}_i(\xi)| > 0) = 9/45$ for $i = 1, \ldots, n$. For $T < \infty$, the sequence of subgraphs $(\mathcal{V}, \mathcal{E}(\xi^t)), t = 1, \ldots, T$ satisfies the B-connectedness property in the time varying graph model [16]. Notice that FSPPD can be implemented with multiple local updates by choosing $\widetilde{\mathbf{C}}(\xi) = \mathbf{I}$ for all ξ.

We benchmark the performance of FSPPD against state-of-the-art distributed optimization algorithms in two different setups: (A) when only stochastic gradient is available: we compare with SwarmSGD [17] which works in a similar setting as FSPPD and ChocoSGD [13] which uses a composition of random sparsification and random gossip that samples 1 agent per iteration; (B) when exact gradient is available: we compare with Di-CS-GD [15] and DIMIX [16] which are two recent algorithms for this setting. The step sizes used to produce the experiment are shown in the legend, with $\eta_t = \frac{4}{10^{-4}(a+t)}$ for ChocoSGD and $\eta_t = \frac{a_0}{a_1+t}$ for others. For compressed communication, we used 8-bit quantizer in SwarmSGD, and random-1 coordinate sparsifier in other algorithms. In Fig. 1 and 2, we report the optimal solution gap $(\sum_{i=1}^n \|\bar{\mathbf{x}}_i^t - \mathbf{x}^\star\|_2^2)/(\sum_{i=1}^n \|\mathbf{x}_i^0 - \mathbf{x}^\star\|_2^2)$ and consensus error $\sum_{i=1}^n \|\bar{\mathbf{x}}_i^t - (\sum_{i=1}^n \bar{\mathbf{x}}_i^t/n)\|_2^2$ where $\bar{\mathbf{x}}_i^t = (\sum_{r=0}^t \eta_r \mathbf{x}_i^r)/(\sum_{r=0}^t \eta_r)$ for FSPPD and Di-CS-GD, $\bar{\mathbf{x}}_i^t = (\sum_{r=0}^t \mathbf{x}_i^r)/(t+1)$ for SwarmSGD and DIMIX and $\bar{\mathbf{x}}_i^t = (\sum_{r=0}^t (a+r)^2 \mathbf{x}_i^r)/(\sum_{r=0}^t (a+r)^2)$ for ChocoSGD. All figures report the average performance over 10 random-seeded simulations. We observe that FSPPD converges faster and suspect that this is due to implicit gradient tracking.

Conclusions. We have proposed an algorithm called FSPPD and showed that it solves (1) using peer-to-peer communication over a network of agents. The algorithm features communication efficient implementation via compressed message exchanges on time-varying (random) graph. Future works include analyzing FSPPD under more general settings, e.g., non-convex objective, nonlinear compression schemes, etc.

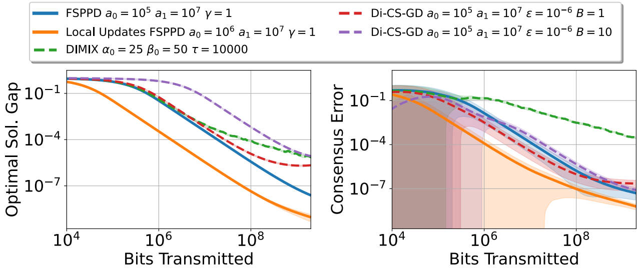

Fig. 2. Comparison under the exact gradient setting; cf. (B).

REFERENCES

[1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 20–27, 2004.

[2] T.-H. Chang, M. Hong, H.-T. Wai, X. Zhang, and S. Lu, "Distributed learning in the nonconvex world: From batch data to streaming and beyond," *IEEE Signal Processing Magazine*, 2020.

[3] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automat. Control.*, 2009.

[4] J. N. Tsitsiklis, "Problems in decentralized decision making and computation.," tech. rep., MIT LIDS, 1984.

[5] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, pp. 516–545, 2010.

[6] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *NeurIPS*, vol. 30, 2017.

[7] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2017.

[8] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[9] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *CDC*, 2012.

[10] F. Saadatniaki, R. Xin, and U. A. Khan, "Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices," *IEEE Trans. Automat. Control.*, 2020.

[11] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[12] Y. Lu and C. De Sa, "Optimal complexity in decentralized training," in *ICML*, pp. 7111–7123, 2021.

[13] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *ICML*, pp. 3478–3487, 2019.

[14] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Trans. Automat. Control.*, vol. 56, no. 6, pp. 1291–1306, 2010.

[15] Y. Chen, A. Hashemi, and H. Vikalo, "Decentralized optimization on time-varying directed graphs under communication constraints," in *ICASSP*, pp. 3670–3674, 2021.

[16] H. Reisizadeh, B. Touri, and S. Mohajer, "Dimix: Diminishing mixing for sloppy agents," *SIAM Journal on Optimization*, 2023.

[17] G. Nadiradze, A. Sabour, P. Davies, S. Li, and D. Alistarh, "Asynchronous decentralized sgd with quantized and local updates," *NeurIPS*, vol. 34, pp. 6829–6842, 2021.

[18] M. Hong and T.-H. Chang, "Stochastic proximal gradient consensus over random networks," *IEEE Trans. on Signal Process.*, vol. 65, no. 11, pp. 2933–2948, 2017.

[19] D. Hajinezhad and M. Hong, "Perturbed proximal primal–dual algorithm for nonconvex nonsmooth optimization," *Mathematical Programming*, vol. 176, no. 1-2, pp. 207–245, 2019.

[20] P. Bianchi and W. Hachem, "Dynamical behavior of a stochastic forward–backward algorithm using random monotone operators," *JOTA*, vol. 171, pp. 90–120, 2016.

[21] P. Bianchi, W. Hachem, and A. Salim, "A fully stochastic primal-dual algorithm," *Optimization Letters*, vol. 15, no. 2, pp. 701–710, 2021.

[22] P. L. Combettes and J.-C. Pesquet, "Fixed point strategies in data science," *IEEE Trans. on Signal Process.*, 2021.