A One-Sample Decentralized Proximal Algorithm for Non-Convex Stochastic Composite Optimization

Tesi Xiao  Xuxing Chen  Krishnakumar Balasubramanian  Saeed Ghadimi

1Department of Statistics, University of California, Davis
2Department of Mathematics, University of California, Davis
3Department of Management Sciences, University of Waterloo

Abstract

We focus on decentralized stochastic non-convex optimization, where \( n \) agents work together to optimize a composite objective function which is a sum of a smooth term and a non-smooth convex term. To solve this problem, we propose two single-time scale algorithms: Prox-DASA and Prox-DASA-GT. These algorithms can find \( \epsilon \)-stationary points in \( O(n^{-1}\epsilon^{-2}) \) iterations using constant batch sizes (i.e., \( O(1) \)). Unlike prior work, our algorithms achieve a comparable complexity result without requiring large batch sizes, more complex per-iteration operations (such as double loops), or stronger assumptions. Our theoretical findings are supported by extensive numerical experiments, which demonstrate the superiority of our algorithms over previous approaches. Our code is available at [https://github.com/xuxingc/ProxDASA](https://github.com/xuxingc/ProxDASA).

1 INTRODUCTION

Decentralized optimization is a flexible paradigm for solving complex optimization problems in a distributed manner and has numerous applications in fields such as machine learning, robotics, and control systems. It has attracted increased attention due to the following benefits: (i) Robustness: Decentralized optimization is more robust than centralized optimization because each agent can operate independently, making the system more resilient to failures compared to a centralized system where a coordinator failure or overload can halt the entire system. (ii) Privacy: Decentralized optimization can provide greater privacy because each agent only has access to a limited subset of observations, which may help to protect sensitive information. (iii) Scalability: Decentralized optimization is highly scalable as it can handle large datasets in a distributed manner, thereby solving complex optimization problems that are difficult or even impossible to solve in a centralized setting.

Specifically, we consider the following decentralized composite optimization problems in which \( n \) agents collaborate to solve

\[
\min_{x \in \mathbb{R}^d} \Phi(x) := F(x) + \Psi(x), \quad F(x) := \frac{1}{n} \sum_{i=1}^{n} F_i(x), \quad (1)
\]

where each function \( F_i(x) \) is a smooth function only known to the agent \( i \); \( \Psi(x) \) is non-smooth, convex, and shared across all agents; \( \Phi(x) \) is bounded below by \( \Phi_* > -\infty \). We consider the stochastic setting where the exact function values and derivatives of \( F_i \)'s are unavailable. In particular, we assume that

\[
F_i(x) = \mathbb{E}_{\xi_i \sim D_i}[G_i(x, \xi_i)],
\]

where \( \xi_i \) is a random vector and \( D_i \) is the distribution used to generate samples for agent \( i \). The agents form a connected and undirected network and can communicate with their neighbors to cooperatively solve (1). The communication network can be represented with \( G = (V, W) \) where \( V = \{v_1, v_2, \ldots, v_n\} \) denotes all devices and \( W = [w_{ij}] \in \mathbb{R}^{n \times n} \) is the weighted adjacency matrix indicating how two agents are connected.

A majority of the existing decentralized stochastic algorithms for solving (1) require large batch sizes to achieve convergence. The few algorithms that operate with constant batch sizes mainly rely on complicated variance reduction techniques and require stronger assumptions to establish convergence results. To the best of our knowledge, the question of whether it is possible to develop decentralized stochastic optimization algorithms to solve (1) without the above mentioned limitations, remains unsolved.

To address this, we propose the two decentralized stochastic proximal algorithms, Prox-DASA and Prox-DASA-GT, for solving (1) and make the following contributions:

- We show that Prox-DASA is capable of achieving convergence in both homogeneous and bounded heterogeneous settings while Prox-DASA-GT works for general decentralized heterogeneous problems.
• We show that both algorithms find an $\epsilon$-stationary point in $\mathcal{O}(n^{-1}e^{-2})$ iterations using only $\mathcal{O}(1)$ stochastic gradient samples per agent and $m$ communication rounds at each iteration, where $m$ can be any positive integer. A topology-independent transient time can be achieved by setting $m = \lceil \frac{1}{\sqrt{\rho}} \rceil$, where $\rho$ is the second-largest eigenvalue of the communication matrix.

• Through extensive experiments, we demonstrate the superiority of our algorithms over prior works.

A summary of our results and comparison to prior work is provided in Table 1.

Related Works on Decentralized Composite Optimization. Motivated by wide applications in constrained optimization [Lee and Nedic [2013], Margellos et al. [2017] and non-smooth problems with a composite structure [1], arising in signal processing [Ling and Tian [2010], Mateos et al. [2010], Patterson et al. [2014] and machine learning [Facchinei et al. [2015], Hong et al. [2017], several works have studied the decentralized composite optimization problem in [1], a natural generalization of smooth optimization. For example, Shi et al. [2015], Li et al. [2019], Alghunaim et al. [2019], Ye et al. [2020], Xu et al. [2021], Li et al. [2021], Sun et al. [2022], Wu and Lu [2022] studied [1] in the convex setting. Furthermore, Facchinei et al. [2015], Di Lorenzo and Scutari [2016], Hong et al. [2017], Zeng and Yin [2018], Scutari and Sun [2019] studied [1] in the deterministic setting.

Although there has been a lot of research investigating decentralized composite optimization, the stochastic non-convex setting, which is more broadly applicable, still lacks a full understanding. Wang et al. [2021] proposes SPPDM, which uses a proximal primal-dual approach to achieve $\mathcal{O}(e^{-2})$ sample complexity. ProxGT-SA and ProxGT-SR-O [Xin et al. [2021a] incorporate stochastic gradient tracking and multi-consensus update in proximal gradient methods and obtain $\mathcal{O}(n^{-1}e^{-2})$ and $\mathcal{O}(n^{-1}e^{-1.5})$ sample complexity respectively, where the latter further uses a SARAH type variance reduction method [Pham et al. [2020] Wang et al. [2019]. A recent work [Mancino-ball et al. [2023] proposes DEEFSORM, which leverages the momentum-based variance reduction technique and gradient tracking to obtain $\mathcal{O}(n^{-1}e^{-1.5})$ and $\mathcal{O}(e^{-1.5})$ sample complexity under different stepsize choices. Nevertheless, existing works either require stronger assumptions [Mancino-ball et al. [2023] or increasing batch sizes [Wang et al. [2021], Xin et al. [2021a].

2 PRELIMINARIES

Notations. $\| \cdot \|$ denotes the $\ell_2$-norm for vectors and Frobenius norm for matrices. $\| \cdot \|_2$ denotes the spectral norm for matrices. $\mathbf{1}$ represents the all-one vector, and $\mathbf{I}$ is the identity matrix as a standard practice. We identify vectors at agent $i$ in the subscript and use the superscript for the algorithm step. For example, the optimization variable of agent $i$ at step $k$ is denoted as $\mathbf{x}_i^k$, and $\mathbf{z}_i^k$ is the corresponding dual variable. We use uppercase bold letters to represent the matrix that collects all the variables from nodes (corresponding lower case) as columns. We add an overbar to a letter to denote the average over all nodes. For example, we denote the optimization variables over all nodes at step $k$ as $\mathbf{X}_k = [\mathbf{x}_1^k, \ldots, \mathbf{x}_n^k]$. The corresponding average over all nodes can be thereby defined as

$$\bar{x}^k = \frac{1}{n} \sum_{i=1}^{n} x_i^k = \frac{1}{n} X_k \mathbf{1},$$

$$\mathbf{X}_k = [\bar{x}_1^k, \ldots, \bar{x}_n^k] = \bar{x}^k \mathbf{1}^\top = \frac{1}{n} X_k \mathbf{1} \mathbf{1}^\top.$$  

For an extended valued function $\Psi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$, its effective domain is written as $\text{dom}(\Psi) = \{x | \Psi(x) < +\infty\}$. A function $\Psi$ is said to be proper if $\text{dom}(\Psi)$ is non-empty. For any proper closed convex function $\Psi$, $x \in \mathbb{R}^d$, and scalar $\gamma > 0$, the proximal operator is defined as

$$\text{prox}_\gamma^\Psi(x) = \arg \min_{y \in \mathbb{R}^d} \left\{ \frac{1}{2\gamma} \|y - x\|^2 + \Psi(y) \right\}.$$  

For $x, z \in \mathbb{R}^d$ and $\gamma > 0$, the proximal gradient mapping of $z$ at $x$ is defined as

$$\mathcal{G}(x, z, \gamma) = \frac{1}{\gamma} (x - \text{prox}_\gamma^\Psi(x - \gamma z)).$$  

All random objects are properly defined in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and write $x \in H$ if $x$ is $H$-measurable given a sub-$\sigma$-algebra $H \subseteq \mathcal{F}$ and a random vector $x$. We use $\sigma(\cdot)$ to denote the $\sigma$-algebra generated by all the argument random vectors.  

Assumptions. Next, we list and discuss the assumptions made in this work.

Assumption 1. The weighted adjacency matrix $W = (w_{ij}) \in \mathbb{R}^{n \times n}$ is symmetric and doubly stochastic, i.e.,

$$W = W^\top, \quad W\mathbf{1}_n = \mathbf{1}_n, \quad w_{ij} \geq 0, \quad \forall i, j,$$

and its eigenvalues satisfy $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$ and $\rho := \max\{|\lambda_2|, |\lambda_n|\} < 1$.

Assumption 2. All functions $\{F_i\}_{1 \leq i \leq n}$ have Lipschitz continuous gradients with Lipschitz constants $L_{\nabla F_i}$, respectively. Therefore, $\nabla F$ is $L_{\nabla F}$-Lipschitz continuous with $L_{\nabla F} = \max_{1 \leq i \leq n}\{L_{\nabla F_i}\}$.

Assumption 3. The function $\Psi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ is a closed proper convex function.

For stochastic oracles, we assume that each node $i$ at every iteration $k$ is able to obtain a local random data vector $\xi_i^k$. The induced natural filtration is given by $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and

$$\mathcal{F}_k := \sigma(\xi_i^t | i = 1, \ldots, n, t = 1, \ldots, k), \forall k \geq 1.$$
We require that the stochastic gradient $\nabla G_i(\cdot, \xi_{ik+1})$ is unbiased conditioned on the filtration $\mathcal{F}_k$.

**Assumption 4 (Unbiasness).** For any $k \geq 0, x \in \mathcal{F}_k$, and $1 \leq i \leq n$, $E \left[ \| \nabla G_i(x, \xi_{ik+1}) - \nabla F_i(x) \| \bigg| \mathcal{F}_k \right] = 0$.

**Assumption 5 (Independence).** For any $k \geq 0, 1 \leq i, j \leq n, i \neq j$, $\xi_{ik+1}$ is independent of $\mathcal{F}_k$, and $\xi_{ik+1}$ is independent of $\xi_{jk+1}$.

In addition, we consider two standard assumptions on the variance and heterogeneity of stochastic gradients.

**Assumption 6 (Bounded variance).** For any $k \geq 0, x \in \mathcal{F}_k$, and $1 \leq i \leq n$,

$$E \left[ \| \nabla G_i(x, \xi_{ik+1}) - \nabla F_i(x) \|^2 \bigg| \mathcal{F}_k \right] \leq \sigma_i^2.$$

Let $\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \sigma_i^2$.

**Assumption 7 (Gradient heterogeneity).** There exists a constant $\nu \geq 0$ such that for all $1 \leq i \leq n, x \in \mathbb{R}^d$,

$$\| \nabla F_i(x) - \nabla F(x) \| \leq \nu.$$

**Remark (Bounded gradient heterogeneity).** The above assumption of gradient heterogeneity is standard [Lian et al., 2017] and less strict than the bounded second moment assumption on stochastic gradients which implies Lipschitzness of functions $\{F_i\}$. However, this assumption is only required for the convergence analysis of Prox-DASA and can be bypassed by employing a gradient tracking step.

**Remark (Smoothness and mean-squared smoothness).** Our theoretical results of the proposed methods are only built on the smoothness assumption on functions $\{F_i\}$ without further assuming mean-squared smoothness assumptions on $\{G_i, \xi\}$, which is required in all variance reduction based methods in the literature, such as ProxGT-SR-O [Xin et al., 2021a] and DEEPSTORM [Mancino-ball et al., 2023].

It is worth noting that a clear distinction in the lower bounds of sample complexity for solving stochastic optimization under two different sets of assumptions has been proven in [Arjevani et al., 2023]. Specifically, when considering the mean-squared smoothness assumption, the optimal sample complexity is $O(\epsilon^{-1.5})$, whereas under smoothness assumptions, it is $O(\epsilon^{-2})$. The proposed methods in this work achieve the optimal sample complexity under our weaker assumptions.

## 3 ALGORITHM

Several algorithms have been developed to solve Problem 1 in the stochastic setting; see Table 1. However, the most recent two types of algorithms have certain drawbacks: (i) increasing batch sizes: ProxGT-SA, ProxGT-SR-O, and DEEPSTORM with constant step sizes (Theorem 1 in [Mancino-ball et al., 2023]) require batches of stochastic gradients with batch sizes inversely proportional to tolerance $\epsilon$; (ii) algorithmic complexities: ProxGT-SR-O and DEEPSTORM are either double-looped or two-time-scale, and require stochastic gradients evaluated at different parameter values over the same sample, i.e., $\nabla G_i(x, \xi)$ and $\nabla G_j(x', \xi)$. These variance reduction techniques are unfavorable when gradient evaluations are computationally expensive such as forward-backward steps for deep neural networks. (iii) theoretical weakness: the convergence analyses of ProxGT-SR-O and DEEPSTORM are established under the stronger assumption of mean-squared lipschitzness of stochastic gradients. In addition, Theorem 2 in [Mancino-ball et al., 2023] fails to provide linear-speedup results for one-sample variant of DEEPSTORM with diminishing stepsizes.
Algorithm 1: Prox-DASA

Input: $x_0^i = z_0^i = 0, \gamma, \{\alpha_k\}_{k \geq 0}, m$

for $k = 0, 1, \ldots, K - 1$ do

  # Local Update
  for $i = 1, 2, \ldots, n$ (in parallel) do
  
  $y_k^i = \text{prox}_y(x_k^i - \gamma z_k^i)$
  $z_k^i + 1 = (1 - \alpha_k)z_k^i + \alpha_k y_k^i$

  # Compute stochastic gradient
  $v_k + 1 = \nabla G_i(x_k^i, z_k^i)$
  $z_k^i + 1 = (1 - \alpha_k)z_k^i + \alpha_k v_k^i$

  # Communication
  $x_k^{1i}, \ldots, x_k^{ni} = [z_k^{1i}, \ldots, z_k^{ni}] W_m$

end

Algorithm 2: Prox-DASA-GT

Input: $x_0^i = z_0^i = u_0^i = 0, \gamma, \{\alpha_k\}_{k \geq 0}, m$

for $k = 0, 1, \ldots, K$ do

  # Local Update
  for $i = 1, 2, \ldots, n$ (in parallel) do
  
  $y_k^i = \text{prox}_y(x_k^i - \gamma z_k^i)$
  $\hat{x}_k^{i+1} = (1 - \alpha_k)x_k^i + \alpha_k y_k^i$

  # Compute stochastic gradient
  $v_k + 1 = \nabla G_i(x_k^i, z_k^i)$
  $\hat{u}_k^{i+1} = u_k^i + v_k^{i+1} - \hat{x}_k^{i+1}$
  $\hat{z}_k^{i+1} = (1 - \alpha_k)z_k^i + \alpha_k \hat{u}_k^{i+1}$

  # Communication
  $x_k^{1i}, \ldots, x_k^{ni} = [\hat{x}_k^{1i}, \ldots, \hat{x}_k^{ni}] W_m$
  $u_k^{1i}, \ldots, u_k^{ni} = [\hat{u}_k^{1i}, \ldots, \hat{u}_k^{ni}] W_m$
  $z_k^{1i}, \ldots, z_k^{ni} = [\hat{z}_k^{1i}, \ldots, \hat{z}_k^{ni}] W_m$

end

3.1 DECENTRALIZED PROXIMAL AVERAGED STOCHASTIC APPROXIMATION

To address the above limitations, we propose Decentralized Proximal Averaged Stochastic Approximation (Prox-DASA) which leverages a common averaging technique in stochastic optimization [Ruszczyński 2008, Mokhtari et al. 2018, Ohadimi et al. 2020] to reduce the error of gradient estimation. In particular, the sequences of dual variables $Z^k = [z_1^k, \ldots, z_n^k]$ that aim to approximate gradients are defined in the following recursion:

$Z^{k+1} = \{(1 - \alpha_k)Z^k + \alpha_k V^{k+1}\} W^m$

$V^{k+1} = [v_1^{k+1}, \ldots, v_n^{k+1}]$

where each $v_i^{k+1}$ is the local stochastic gradient evaluated at the local variable $x_i^k$. For complete graphs where each pair of graph vertices is connected by a edge and there is no consensus error for optimization variables, i.e., $W = \frac{1}{n} 11^T$ and $x_i^k = x_j^k, \forall i, j$, the averaged dual variable over nodes $\bar{z}^k$ follows the same averaging rule as in centralized algorithms:

$\bar{z}^{k+1} = (1 - \alpha_k)\bar{z}^k + \alpha_k \bar{\bar{v}}^{k+1}$

$\mathbb{E}[\bar{v}^{k+1} | \mathcal{F}_k] = \nabla F(\bar{z}^k)$.

To further control the consensus errors, we employ a multiple consensus step for both primal and dual iterates $(x_i^k, z_i^k)$ which multiply the matrix of variables from all nodes by the weight matrix $m$ times. A pseudo code of Prox-DASA is given in Algorithm 1.

3.2 GRADIENT TRACKING

The constant $\nu$ defined in Assumption 7 measures the heterogeneity between local gradients and global gradients, and hence the variance of datasets of different agents. To remove $\nu$ in the complexity bound, [Liang et al. 2018] proposed the D2 algorithm, which modifies the $x$ update in D-PSGD [Lian et al. 2017]. However, it requires one additional assumption on the eigenvalues of the mixing matrix $W$. Here we adopt the gradient tracking technique, which was first introduced to deterministic distributed optimization to improve the convergence rate [Xu et al., 2015; Di Lorenzo and Scutari, 2016; Nedic et al., 2017; Qu and Li, 2017], and was later proved to be useful in removing the data variance (i.e., $\nu$) dependency in the stochastic case [Zhang and You, 2019; Lu et al., 2019; Pu and Nedic, 2021; Koloskova et al., 2021]. In the convergence analysis of Prox-DASA, an essential step is to control the heterogeneity of stochastic gradients, i.e., $\mathbb{E}[\|V^{k+1} - \bar{V}^{k+1}\|^2]$, which requires bounded heterogeneity of local gradients (Assumption 7). To bypass this assumption, we employ a gradient tracking step by replacing $V^{k+1}$ with pseudo stochastic gradients $\bar{U}^{k+1} = [\bar{u}_1^{k+1}, \ldots, \bar{u}_n^{k+1}]$, which is updated as follows:

$\bar{U}^{k+1} = (U^k + V^{k+1} - \bar{V}^k) W^m$.

Provided that $U^0 = V^0$ and $W1 = 1$, one can show that $\bar{U}^k = \bar{V}^k$ at each step $k$. In addition, with the consensus procedure over $\bar{U}^k$, the heterogeneity of pseudo stochastic gradients $\mathbb{E}[\|\bar{U}^{k+1} - \bar{U}^{k+1}\|^2]$ can be bounded above. The proposed algorithm, named as Prox-DASA with Gradient Tracking (Prox-DASA-GT), is presented in Algorithm 2.

3.3 CONSENSUS ALGORITHM

In practice, we can leverage accelerated consensus algorithms, e.g., [Liu and Morse, 2011; Olshevsky, 2017], to speed up the multiple consensus step $W^m$ to achieve improved communication complexities when $m > 1$. Specifically, we can replace $W^m$ by a Chebyshev-type polynomial of $W$ as described in Algorithm 3, which can improve the $\rho$-dependency of the communication complexity from a factor of $\frac{1}{1-\rho}$ to $\frac{1}{\sqrt{1-\rho}}$. 2327
Algorithm 3: Chebyshev Mixing Protocol

Input: Matrix $X$, mixing matrix $W$, rounds $m$
Set $A_0 = X$, $A_1 = XW$, $\rho = \max\{\|\lambda_2(W)\|, \lambda_n(W)\} < 1$, $\mu_0 = 1$, $\mu_1 = \frac{1}{\rho}$
for $t = 1, \ldots, m - 1$ do
\[ \begin{align*}
\mu_{t+1} & = \frac{1}{\rho_t} \mu_t - \mu_{t-1} \\
A_{t+1} & = \frac{2\mu_t}{\rho_t \mu_{t+1}} A_t W - \frac{\mu_{t-1}}{\mu_{t+1}} A_{t-1}
\end{align*} \]
end
Output: $A_m$

4 CONVERGENCE ANALYSIS

4.1 NOTION OF STATIONARITY

For centralized optimization problems with non-convex objective function $F(x)$, a standard measure of non-stationarity of a point $\bar{x}$ is the squared norm of proximal gradient mapping of $\nabla F(\bar{x})$ at $\bar{x}$, i.e.,
\[ \|G(\bar{x}, \nabla F(\bar{x}), \gamma)\|^2 = \left\| \frac{1}{\gamma} (x - \text{prox}_\gamma(\bar{x} - \gamma \nabla F(\bar{x}))) \right\|^2 . \]

For the smooth case where $\Psi(x) \equiv 0$, the above measure is reduced to $\|\nabla F(\bar{x})\|^2$.

In Theorem 1 for simplicity we assume $\gamma = \frac{1}{L_{F,W}}$, which relates our definition to [3]. Noting that by non-expansiveness of the proximal operator, we have $\|G(x_1, \nabla F(x_1), \gamma) - \nabla F(\bar{x})\| \leq \frac{2+\gamma L_{F,W}}{\gamma} \|x_1 - \bar{x}\|$, implying
\[ \frac{1}{n} \sum_{i=1}^{n} \|G(x_i, \nabla F(x_i), \gamma)\|^2 \lesssim \|\nabla F(\bar{x})\|^2 + \frac{1}{\gamma^2} \|X - \bar{X}\|^2 . \]

4.2 MAIN RESULTS

We present the complexity results of our algorithms below.

Theorem 1. Suppose Assumptions \[ \| \] hold and the total number of iterations $K \geq K_0$, where $K_0$ is a constant that only depends on constants $(n, L_{F,W}, \gamma(m), \sigma)$, where $\sigma(m) = \frac{(n^2 + \gamma^2)}{(1 - \gamma^2)^{2}}$. Let $C_0$ be some initialization-dependent constant and $R$ be a random integer uniformly distributed over $\{1, 2, \ldots, K\}$. Suppose we set $\alpha_K \asymp \frac{1}{\sqrt{nK}}$.

(Prox-DASA) Suppose in addition Assumption \[ \| \] also holds. The, for Algorithm 1 we have
\[ \mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^{n} \|G(\bar{x}_R, \nabla F(\bar{x}_R), \gamma)\|^2 \right] \lesssim \gamma^{-1} C_0 + \frac{\sigma^2 + n(\sigma^2 + \gamma^{-2} \nu^2) \sigma(m)}{\sqrt{nK}} , \]
\[ \mathbb{E} \left[ \left\| \bar{z}_R - \nabla F(\bar{x}_R) \right\|^2 \right] \lesssim \gamma^{-1} C_0 + \frac{\sigma^2 + n(\sigma^2 + \gamma^{-2} \nu^2) \sigma(m)}{\sqrt{nK}} , \]
\[ \mathbb{E} \left[ \frac{L_{F,W}^2}{n} \|X_R - \bar{X}_R\|^2 + \frac{1}{n} \|Z_R - \bar{Z}_R\|^2 \right] \lesssim \frac{n(\sigma^2 + \gamma^{-2} \nu^2) \sigma(m)}{K} . \]

(Prox-DASA-GT) For Algorithm 2 we have
\[ \mathbb{E} \left[ \frac{1}{n} \sum_{i=1}^{n} \|G(\bar{x}_R, \nabla F(\bar{x}_R), \gamma)\|^2 \right] \lesssim \gamma^{-1} C_0 + \frac{\sigma^2 + n(\sigma^2 + \gamma^{-2} \nu^2) \sigma(m)}{\sqrt{nK}} , \]
\[ \mathbb{E} \left[ \left\| \bar{z}_R - \nabla F(\bar{x}_R) \right\|^2 \right] \lesssim \gamma^{-1} C_0 + \frac{\sigma^2 + n(\sigma^2 + \gamma^{-2} \nu^2) \sigma(m)}{\sqrt{nK}} , \]
\[ \mathbb{E} \left[ \frac{L_{F,W}^2}{n} \|X_R - \bar{X}_R\|^2 + \frac{1}{n} \|Z_R - \bar{Z}_R\|^2 \right] \lesssim \frac{n(\sigma^2 + \gamma^{-2} \nu^2) \sigma(m)}{K} . \]

In Theorem 1 for simplicity we assume $\gamma \asymp \frac{1}{L_{F,W}}$, which can be relaxed to $\gamma > 0$. We have the following corollary characterizing the complexity of Algorithm 1 and 2 for finding $\epsilon$-stationary points. The proof is immediate.

Corollary 1. Under the same conditions of Theorem 1 provided that $K \geq n^3 \sigma(m)$, for any $\epsilon > 0$ the sample complexity per agent for finding $\epsilon$-stationary points in Algorithm 1 and 2 are $O(\max\{n^{-1}, n^2, K \})$ where the transient time $K_T \asymp \max\{K_0, n^3 \sigma(m)\}$.
Remark (Sample complexity). For a sufficiently small $\epsilon > 0$, Corollary 4 implies that the sample complexity of Algorithm 1 and 2 matches the optimal lower bound $\mathcal{O}(n^{-1}\epsilon^{-2})$ in decentralized smooth non-convex optimization [Lu and De Sa [2021]].

Remark (Transient time and communication complexity). Our algorithms can achieve convergence with a single communication round per iteration, i.e., $m = 1$, leading to a topology-independent $\mathcal{O}(n^{-1}\epsilon^{-2})$ communication complexity. In this case, however, the transient time $K_T$ still depends on $\rho$, as is also the case for smooth optimization problems [Xin et al. 2021b]. When considering multiple consensus steps per iteration with the communication complexity being $\mathcal{O}(mn^{-1}\epsilon^{-2})$, setting $m \approx \lceil 1 / \rho \rceil$ (or $m \approx \lceil 1 / \sqrt{\rho} \rceil$) for accelerated consensus algorithms) results in a topology-independent transient time given that $\rho(m) \approx 1$.

Remark (Dual convergence). An important aspect to emphasize is that in our proposed methods, the sequence of average dual variables $\bar{z}^k = \frac{1}{n} \sum_{i=1}^{n} z^k_i$ converges to $\nabla F(\bar{x}^k)$, while the consensus error of $\{z_1^k, \ldots, z_n^k\}$ decreases to zero. Our approach achieves this desirable property, which is commonly observed in modern variance reduction methods [Gower et al. [2020]], without the need for complex variance reduction operations in each iteration. As a result, it provides a reliable termination criterion in the stochastic setting without requiring large batch sizes.

### 4.3 PROOF SKETCH

Here, we present a sketch of our convergence analyses and defer details to Appendix. Our proof relies on the merit function below:

$$W(\bar{x}^k, \bar{z}^k) = \Phi(\bar{x}^k) - \Phi_0 + \Psi(\bar{x}^k) - \eta(\bar{x}^k, \bar{z}^k)$$

$$+ \lambda \left\| \nabla F(\bar{x}^k) - \bar{z}^k \right\|^2,$$

where $\eta(x, z) = \min_{y \in \mathbb{R}^n} \left\{ \langle z, y - x \rangle + \frac{1}{2\gamma} \|y - x\|^2 + \Psi(y) \right\}.$

Let $y^k = \text{prox}_{\Psi/\gamma}(\bar{x}^k - \gamma \bar{z}^k).$ Then, the proximal gradient mapping of $\bar{z}^k$ at $\bar{x}^k$ is $G(\bar{x}^k, \bar{z}^k, \gamma) = \frac{1}{\gamma}(\bar{x}^k - y^k).$ Since $y^k$ is the minimizer of a $1/\gamma$-strongly convex function, we have

$$\langle z^k, y^k - \bar{x}^k \rangle + \frac{1}{2\gamma} \|y^k - \bar{x}^k\|^2 + \Psi(y^k)$$

$$\leq \Psi(\bar{x}^k) - \frac{1}{2\gamma} \|y^k - \bar{x}^k\|^2,$$

implying the relation between $\Psi(\bar{x}^k) - \eta(\bar{x}^k, \bar{z}^k)$ and primal convergence:

$$\Psi(\bar{x}^k) - \eta(\bar{x}^k, \bar{z}^k) \geq \frac{\gamma}{2} \|G(\bar{x}^k, \bar{z}^k, \gamma)\|^2.$$

Following standard practices in optimization, we set $\gamma = \frac{1}{L\sqrt{v}}$ below for simplicity. However, our algorithms do not require any restriction on the choice of $\gamma$.

**Step 1:** Leveraging the merit function with $\lambda \propto \gamma$, we can first obtain an essential lemma (Lemma 11 in Appendix) in our analyses, which says that for sequences $\{x^k, z^k\}_{k=1}^{\infty}$ generated by $\text{Prox-DASA (GT)}$ (Algorithm 1 or 2) with $\alpha_k \lesssim \min\{1, (1+\gamma)^{-2}, \gamma^2(1+\gamma)^{-2}\}$, we have

$$W(\bar{x}^{k+1}, \bar{z}^{k+1}) - W(\bar{x}^k, \bar{z}^k)$$

$$\leq -\alpha_k \left\{ \Theta_k + \Upsilon_k + \alpha_k \Lambda_k + r^{k+1} \right\},$$

where $E[r^{k+1} | \mathcal{F}_k] = 0$, $\Theta_k \approx \gamma \|\Delta k+1\|^2$, $\Upsilon_k \approx \gamma \|z_k - \bar{z}_k\|^2 + \frac{1}{n} \|x_k - \bar{x}_k\|^2$, and $\Delta^{k+1} = \bar{v}^{k+1} - \frac{1}{n} \sum_{i=1}^{n} \nabla F_i(x^k_i) = \bar{v}^{k+1} - \frac{1}{n} \sum_{i=1}^{n} \nabla F_i(x^k_i)$ (for $\text{Prox-DASA (GT)}$). Thus, by telescoping and taking expectation with respect to $\mathcal{F}_0$, we have

$$\sum_{k=0}^{K} \alpha_k E\left[ \|\bar{x}^{k+1} - \bar{y}^k\|^2 + \gamma^2 \|\nabla F(\bar{x}^k) - \bar{z}^k\|^2 \right]$$

$$\lesssim \gamma W(\bar{x}^0, \bar{z}^0) + \gamma^2 \sigma^2 \sum_{k=0}^{K} \frac{\alpha_k}{n}$$

$$+ \sum_{k=0}^{K} \alpha_k \left\{ E\left[ \|x_k - \bar{x}_k\|^2 + \gamma^2 \|z_k - \bar{z}_k\|^2 \right] \right\}.$$ (4)

**Step 2:** We then analyze the consensus errors. Without loss of generality, we consider $X_0 = X_k = 0$, i.e., all nodes have the same initialization at 0. For $m \in \mathbb{N}_+$, define

$$\phi(m) = \frac{(1 + \rho^2 m)^2}{(1 - \rho^{2m})^2}.$$

Then, we have the following fact:

- $\phi(m)$ is monotonically decreasing with the maximum value being $\phi(1) = \frac{(1+\rho^2)^2}{(1-\rho^2)^2} := \phi_1$;
- $\phi(m) \leq 1$ if and only if $\rho^{2m} \leq \frac{1}{3}$.

With the definition of $\phi(m)$ and assuming $0 < \alpha_{k+1} \leq \alpha_k \leq 1$, we can show the consensus errors have the following upper bounds.

**Prox-DASA:** Let $\alpha_k \lesssim \phi(m)^{-1/2}$, we have

$$\sum_{k=0}^{K} \frac{\alpha_k}{n} E\left[ \|x_k - \bar{x}_k\|^2 \right] \leq \sum_{k=0}^{K} \gamma^2 \alpha_k \frac{1}{n} E\left[ \|z_k - \bar{z}_k\|^2 \right].$$
Moreover, noting that the consensus error of $Y_m$ be set as $\varrho$, we have
\[\sum_{k=0}^{K} \frac{\alpha_k}{n} \mathbb{E} \left[ \|X_k - \bar{X}_k\|^2 \right] \leq \sum_{k=0}^{K} \frac{\gamma^2 \alpha_k}{n} \mathbb{E} \left[ \|Z_k - \bar{Z}_k\|^2 \right]\]
\[\lesssim \varrho(m)^2 \sum_{k=0}^{K} \alpha_k^3 \{\gamma^2 + \alpha_k^2 \mathbb{E} \left[ \|\bar{x}_k - \bar{y}^k\|^2 \right] \} \quad \left(\text{6}\right)
\]

We can also see that to obtain a topology-independent iteration complexity, the number of communication rounds can be set as $m = \left\lceil \frac{\log 3}{2(1-\rho)} \right\rceil$, which implies $\varrho(m) \leq 1$.

In addition, we have the following fact that relates the consensus error of $Y$ to the consensus errors of $X$ and $Z$:
\[\left\|y^k - \bar{y}^k\right\|^2 + \frac{1}{n} \left\|Y_k - \bar{Y}_k\right\|^2 = \frac{1}{n} \sum_{i=1}^{n} \left\| y^k_i - \bar{y}^k_i \right\|^2 \leq \frac{2}{n} \left\{ \|X_k - \bar{X}_k\|^2 + \gamma^2 \|Z_k - \bar{Z}_k\|^2 \right\} \]

\textbf{Step 3:} Let $R$ be a random integer with
\[\Pr(R = k) = \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k}, \quad k = 1, 2, \ldots, K,\]
and dividing both sides of (5) by $\sum_{k=1}^{K} \alpha_k$, we can obtain that for Prox-DASA, the consensus error of $X_R$ satisfies
\[\mathbb{E} \left[ \frac{1}{n} \|X_R - \bar{X}_R\|^2 \right] \lesssim (\gamma^2 + \nu^2) \varrho(m) \sum_{k=0}^{K} \alpha_k^3 \sum_{k=1}^{K} \frac{\alpha_k}{\alpha_k} \quad \left(\text{5}\right)\]

Moreover, noting that
\[\|G(\bar{x}, \nabla F(\bar{x}), \gamma)\|^2 \lesssim \frac{1}{\gamma^2} \left\{ \|\bar{x}_k - \bar{y}^k\|^2 + \|y^k - \bar{y}^k\|^2 \right\} + \|\nabla F(\bar{x}_k) - \bar{z}^k\|^2 ,\]
and combining (4) with (5), we can get
\[\mathbb{E} \left[ \frac{\|G(\bar{x}_R, \nabla F(\bar{x}_R), \gamma)\|^2}{\gamma \sum_{k=1}^{K} \alpha_k} \right] \lesssim \frac{W(\bar{x}^0, \bar{z}^0)}{\gamma \sum_{k=1}^{K} \alpha_k} + \frac{\nu^2}{\gamma^2} \sum_{k=0}^{K} \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k} \quad \left(\text{5}\right)\]

Thus, setting $\alpha_k \approx \sqrt{\frac{1}{K}}$, we obtain the convergence results of Prox-DASA:
\[\mathbb{E} \left[ \frac{\|G(\bar{x}_R, \nabla F(\bar{x}_R), \gamma)\|^2}{\gamma \sum_{k=1}^{K} \alpha_k} \right] \lesssim \frac{W(\bar{x}^0, \bar{z}^0)}{\gamma \sum_{k=1}^{K} \alpha_k} + \frac{\nu^2}{\gamma^2} \sum_{k=0}^{K} \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k} \quad \left(\text{5}\right)\]

\[\text{5 EXPERIMENTS}\]

\textbf{5.1 SYNTHETIC DATA}\n
To demonstrate the effectiveness of our algorithms, we first evaluate our algorithms using synthetic data for solving sparse single index models [Alquier and Biau 2013] in the decentralized setting. We consider the homogeneous setting where the data sample at each node $\xi = (X, Y)$ is generated from the same single index model $Y = g(X \cdot \theta_k) + \varepsilon$, where $X, \theta \in \mathbb{R}^d$ and $\mathbb{E}[\varepsilon | X] = 0$. In this case, we solve the following $L_1$-regularized least square problems:
\[\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{(X, Y) \sim D} \left[ (Y - g(X \cdot \theta))^2 \right] + \lambda \|\theta\|_1\]

In particular, we set $\theta_k \in \mathbb{R}^{100}$ to be a sparse vector and $g(\cdot) = (\cdot)^2$ which corresponds to the sparse phase retrieval problem [Jaganathan et al. 2016]. We simulate streaming data samples with batch size $= 1$ for training and $10,000$ data samples per node for evaluations, where $X$ and $\varepsilon$ are sampled independently from two Gaussian distributions. We employ a ring topology for the network where self-weighting and neighbor weights are set to be $1/3$. We set the penalty parameter $\lambda = 0.01$, the total number of iterations $K = 10,000$, $\alpha_k = \sqrt{n/K}$, $\gamma = 0.01$, and the number of communication rounds per iteration $m = \lceil \frac{1}{1-\rho} \rceil$. We plot the test loss and the norm of proximal gradient mapping in the log scale against the number of iterations in Figure 4, which shows that our decentralized algorithms have an additional linear speed-up with respect to $n$. In other words, the algorithms become faster as more agents are added to the network.

\textbf{5.2 REAL-WORLD DATA}\n
Following [Mancino-ball et al. 2023], we consider solving the classification problem
\[\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|D_i|} \sum_{(x, y) \in D_i} \ell_i(f(x; \theta), y) + \lambda \|\theta\|_1, \quad \left(\text{7}\right)\]
on a9a and MNIST datasets. Here, $\ell_i$ denotes the cross-entropy loss, and $f$ represents a neural network parameterized by $\theta$ with $x$ being its input. $D_i$ is the training set only available to agent $i$. The $L_1$ regularization term is used to impose a sparsity structure on the neural network. We use the code in [Mancino-ball et al., 2023] for SPPDM, ProxGT-SR-O/E, DEEPSTORM, and then implement Prox-DASA and Prox-DASA-GT under their framework, which mainly utilizes PyTorch [Paszke et al., 2019] and mpi4py [Dalcin and Fang, 2021]. We use a 2-layer perception model on a9a and the LeNet architecture [LeCun et al., 2015] for the MNIST dataset. We have 8 agents ($n = 8$) which connect in the form of a ring for a9a and a random graph for MNIST. To demonstrate the performance of our algorithms in the constant batch size setting, the batch size is chosen to be 4 for a9a and 32 for MNIST for all algorithms. The learning rates provided in the code of [Mancino-ball et al., 2023] are adjusted accordingly, and we select the ones with the best performance. For Prox-DASA and Prox-DASA-GT we choose a diminishing stepsize sequence, namely, $\alpha_k = \min \{ \alpha, \frac{1}{\sqrt{k}} \}$ for all $k \geq 0$. Note that the same complexity (up to logarithmic factors) bounds can be obtained by directly plugging in the aforementioned expressions for $\alpha_k$ in Section 4.3. Then we tune $\gamma \in \{1, 3, 10\}$ and $\alpha \in \{0.3, 1.0, 3.0\}$. The penalty parameter $\lambda$ is chosen to be 0.0001 for all experiments. The number of communication rounds per iteration $m$ is set to be 1 for all algorithms. We evaluate the model performance periodically during training and then plot the results in Figure 2, from which we observe that both Prox-DASA and Prox-DASA-GT have considerably good performance with small variance in terms of test accuracy, training loss, and stationarity. In particular, it should be noted that although DEEPSTORM achieves better stationarity in Figure 2[1] and 2[0] training a neural network by using DEEPSTORM takes longer time than Prox-DASA and Prox-DASA-GT since it uses the momentum-based variance reduction technique, which requires two forward-backward passes (see, e.g., Eq. (10) and Algorithm 1 in [Mancino-ball et al., 2023]) to compute the gradients in one iteration per agent. In contrast, ours only require one, which saves a large amount of time (see Table 1 in Appendix). We include further details of our experiments in the Appendix.

6 CONCLUSION

In this work, we propose and analyze a class of single time-scale decentralized proximal algorithms (Prox-DASA-(GT)) for non-convex stochastic composite optimization in the form of Eq. (1). We show that our algorithms achieve linear speed-up with respect to the number of agents using an $O(1)$ batch size per iteration under mild assumptions. Furthermore, we demonstrate the efficiency and effectiveness of our algorithms through extensive experiments, in which our algorithms achieve relatively better results with less training time using a small batch size compared to existing methods. In future research, it would be intriguing to expand our work in the context of dependent and heavy-tailed stochastic gradient scenarios [Wai, 2020] and Li and Liu, 2022.

Author Contributions

TX and XC contributed equally to the paper. TX was responsible for conceptualizing the idea and writing the paper. TX and XC worked together to complete the proof. XC took charge of creating the code and conducting the experiments. The paper was further revised by KB and SG.

Acknowledgements

We thank the authors of [Mancino-ball et al., 2023] for kindly providing the code framework to support our experiments. The research of KB is supported by NSF grant DMS-2053918. The research of SG is partially supported by NSERC grant RGPIN-2021-02644.

References


Figure 2: Comparisons between SPPDM [Wang et al., 2021], ProxGT-SR-E [Xin et al., 2021a], DEEPSTORM [Mancino-ball et al., 2023], Prox-DASA [1] and Prox-DASA-GT [2]. The first two rows correspond to a9a and the last two rows correspond to MNIST. The results are averaged over 10 trials, and the shaded regions represent confidence intervals. The vertical axes in the third column are log-scale. It should be noted that ProxGT-SR-E maintains another hyperparameter $q$ (see, e.g., Algorithm 4 and Theorem 3 in [Xin et al., 2021a]) and computes gradients using a full batch every $q$ iterations. For simplicity, we do not include that amount of epochs when we plot this figure. In other words, the real number of epochs required to obtain a point on ProxGT-SR is larger than plotted in the figures in the second and fourth rows. We include the plots that take $q$ into account in Appendix.


