Accelerating Gradient Tracking with Periodic Global Averaging

Shujing Feng Xin Jiang

***Abstract*— Decentralized optimization algorithms have recently attracted increasing attention due to its wide applications in all areas of science and engineering. In these algorithms, a collection of agents collaborate to minimize the average of a set of heterogeneous cost functions in a decentralized manner. State-of-the-art decentralized algorithms like Gradient Tracking (GT) and Exact Diffusion (ED) involve communication at each iteration. Yet, communication between agents is often expensive, resource intensive, and can be very slow. To this end, several strategies have been developed to balance between communication overhead and convergence rate of decentralized methods. In this paper, we introduce GT-PGA, which incorporates GT with periodic global averaging. With the additional PGA, the influence of poor network connectivity in the GT algorithm can be compensated or controlled by a careful selection of the global averaging period. Under the stochastic, nonconvex setup, our analysis quantifies the crucial trade-off between the connectivity of network topology and the PGA period. Thus, with a suitable design of the PGA period, GT-PGA improves the convergence rate of vanilla GT. Numerical experiments are conducted to support our theory, and simulation results reveal that the proposed GT-PGA accelerates practical convergence, especially when the network is sparse.**

I. Introduction

In decentralized optimization, a group of n agents collaborate to solve the optimization problem

$$\text{minimize} \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where the optimization variable is $x \in \mathbb{R}^d$, and each component function $f_i(x)$ is smooth, potentially nonconvex, and held locally by agent $i \in [n]$. This problem formulation has been widely used in modeling various important applications throughout science and engineering, including optimal control, signal processing, resource allocation, and machine learning [1]–[4]. In particular, decentralized/distributed optimization is now prevalent in modern scenarios involving high-performance computing (HPC) resources [5].

Many decentralized methods have been proposed to solve the problem (1), including decentralized/distributed gradient descent (DGD) methods [6]–[8], EXTRA [9], Exact-Diffusion/D^2/NIDS (ED) [10]–[13], and Gradient Tracking (GT) methods [14]–[17]. Among them, DGD is arguably the conceptually simplest decentralized algorithms. At each iteration of DGD, each agent performs a local gradient step followed by a communication round. However, DGD fails to converge *exactly* with constant stepsizes when the local objective functions f_i are *heterogeneous* [18], [19] (*i.e.*, the minimizer of f is different from that of f_i.).

Due to the unsatisfactory convergence results of DGD, exact methods (a.k.a. bias-correction methods) have been extensively studied to account for the inherent heterogeneity in problem (1). Among them, the family of GT algorithms have each agent perform local gradient steps with an estimate of the global gradient called the tracking variable [14]–[17]. In these methods, the bias (or error) caused by problem/data heterogeneity observed in DGD is asymptotically removed.

In decentralized methods (including both DGD and exact methods), gossip communication over the network of agents is required at each iteration of the algorithm. Very often, communication is computationally expensive and resource intensive in practice [5], [20]. To this end, multiple local recursions (or local updates) have recently been studied in the literature. Among these methods include LocalGD [21], [22], Scaffold [23], S-Local-GD [24], FedLin [25], and Scaffnew [26]. LocalGD, which is based on DGD, still suffers from the bias caused by heterogeneity, and multiple local recursions cause agents to drift towards their local solution [27]. Other aforementioned methods combine bias-correction techniques with multiple local gradient updates; for example, GT with local updates (LU-GT) have recently been studied [28]–[31]. Nonetheless, existing analyses fail to establish any theoretical improvement in communication complexity when the number of local updates are deterministically prescribed [28]–[32]. To the best of our knowledge, the only existing analysis that theoretically establishes the benefit of local updates in LocalGD [26] considers the special case where the objective is strongly convex, the true gradients ∇f_i are always accessible, and the number of local updates is randomly selected during the optimization process.

Besides local updates, the *periodic global averaging* (PGA) technique has recently been developed [33] to balance the trade-off between convergence and communication in DGD. It is shown that PGA helps improve the transient stage of DGD with and without local updates [33]. In modern scenarios where high-performance data-center clusters are the computing resources, PGA is beneficial owing to efficient All-Reduce primitives [34]. In addition, the benefit of PGA in DGD is significant when the network is large and/or sparse. However, PGA does not remove the heterogeneity bias in DGD, so DGD with PGA still does not converge exactly with constant stepsizes.

In view of the potential benefits of PGA and the undesirable performance of DGD-PGA, in this work, we incorporate periodic global averaging (PGA) into GT and propose GT-PGA. On the one hand, we show that the incorporation

SF (shf321@lehigh.edu) is with Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA. XJ (xjiang@lehigh.edu) is with Department of Industrial and Systems Engineering, Lehigh Univeristy, Bethlehem, PA, USA.

of PGA accelerates the convergence rate compared with vanilla GT, especially on large and/or sparse networks. On the other hand, GT-PGA also extends LU-GT (with fully connected networks) via efficient gossip communication after local updates.

Despite the promising acceleration in practical convergence, the analysis of GT-PGA is not straightforward. Even though the main recursion of GT-PGA can be regarded as a special form of GT with time-varying topologies [15], its convergence guarantees and practical performance cannot be fully captured by existing analyses. In particular, existing convergence results for time-varying GT rely on the spectral gap of the least connected communication network [15]. Simply applying these results to GT-PGA does not fully explain the superiority of the PGA operation and lead to incomplete conclusions. Therefore, quantifying the benefits of PGA in GT and carefully balancing the trade-off between gossip communication and periodic global averaging require new analysis of the decentralized algorithm.

Overall, the contributions of this paper are as follows.

- We propose to incorporate periodic global averaging (PGA) into the Gradient Tracking (GT) algorithm and analyze the proposed GT-PGA under the stochastic, nonconvex setting.
- Theoretical results are established to guarantee convergence of GT-PGA, and in particular, to quantify the crucial trade-off between network connectivity and the global averaging period. We also discuss the connection and difference between the proposed GT-PGA, vanilla GT, and LU-GT (GT with local updates).
- Numerical experiments are conducted to verify the established theoretical results. In particular, the proposed GT-PGA accelerates practical convergence compared to vanilla GT, especially when the network is large and/or sparse.

The rest of the paper is organized as follows. Section II describes the proposed GT-PGA algorithm and presents the main convergence results. In Section III, we establish the convergence guarantees for GT-PGA, under the stochastic, nonconvex setting, and Section IV presents numerical evidence to support the theoretical results. Finally, Section V presents concluding remarks.

Notation. Lowercase letters define vectors or scalars, uppercase letters define matrices or scalars, and boldface letters represent augmented network quantities. Let $\mathrm{col}\{a_1,\dots,a_n\}$ denote the vector that concatenates the vectors/scalars a_i, and define $[n] := \{1,\dots,n\}$ for any positive integer $n \in \mathbb{N}_{\geq 1}$. The notation $\mathbb{1}$ represents the all-ones vector, of which the size will be clear from context. The inner product of two vectors x, y is denoted by $\langle x, y\rangle$. For any real $p \times q$ matrix A, denote its nullspace by $\mathrm{Null}(A) := \{x \in \mathbb{R}^q \mid Ax = 0\}$. Products of multiple matrices are defined as

$$\prod_{k=i}^{j} A_k := \begin{cases} A_i A_{i+1} \cdots A_j & \text{if } j \geq i \\ A_i A_{i-1} \cdots A_j & \text{if } j < i. \end{cases}$$

Note that we do not assume $j \geq i$. This definition will not cause any confusion as the value of i and j will be clear from context.

II. Gradient tracking with periodic global averaging

In this section, we present the proposed decentralized optimization algorithm for solving problem (1), state the assumptions needed in the analysis, and establish the main convergence result for the proposed algorithm.

In problem (1), the function $f_i\colon \mathbb{R}^d \to \mathbb{R}$ held locally by agent i is smooth, potentially nonconvex, and defined as the expected value with respect to some probability space $(\Omega_i, \mathcal{F}_i, \mathbb{P}_i)$; *i.e.*,

$$f_i(x) := \mathbb{E}_{\xi_i}[F_i(x;\xi_i)], \quad \text{for all } i \in [n].$$

The studied optimization algorithm only has access to stochastic gradient estimates of the true gradient of f_i (see upcoming Assumption 3), and solves the problem (1) in a decentralized manner. Its implementation involves a graph $\mathcal{G} = (\mathcal{V}, W, \mathcal{E})$ that models the connections between the group of n agents (*i.e.*, $|\mathcal{V}| = n$). Specifically, the element w_{ij} in the matrix W scales the information agent i receives from agent j, and $w_{ij} = 0$ if $j \notin \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of neighbors of agent i.

In this work, we incorporate periodic global averaging into the well-known Gradient Tracking (GT) algorithms [15], [16] and study its convergence results. Various forms of GT exist in the literature, and the particular variant of GT considered in this paper is called Semi-ATC-TV-GT [15]. The proposed algorithm, called Gradient Tracking with Periodic Global Averaging (GT-PGA), is listed in Algorithm 1. In the gossip (*i.e.*, decentralized communication) steps (Line 6 in Algorithm 1), every agent i collects information from all its *connected* neighbors, while for global averaging steps (Line 5 in Algorithm 1), agents synchronize their model parameters using, *e.g.*, the efficient All-Reduce primitives [34]. When the global averaging period $\tau \to \infty$, GT-PGA reduces to Gradient Tracking [15] with static topology; when $W = I$, GT-PGA reduces to GT with local updates and a fully-connected network [31].

The proposed periodic global averaging technique is efficient in situations where high-performance data-center clusters are the computing resources. In such a scenario, all GPUs are fully connected with high-bandwidth channels and the network topology can be fully controlled. Under this setting, PGA conducted with Ring All-Reduce has tolerable communication cost; see, *e.g.*, [33]. For scenarios where PGA is extremely expensive (*e.g.*, in wireless sensor networks), PGA can be approximated via multiple gossip steps, or may not be recommended.

To write Algorithm 1 in a more concise form, we introduce the network notation:

$$\begin{aligned} \boldsymbol{x}^{(k)} &:= \mathrm{col}\{x_1^{(k)},\dots,x_n^{(k)}\} \in \mathbb{R}^{nd}, \\ \boldsymbol{g}^{(k)} &:= \mathrm{col}\{g_1^{(k)},\dots,g_n^{(k)}\} \in \mathbb{R}^{nd}, \\ \nabla \boldsymbol{f}^{(k)} &:= \mathrm{col}\{\nabla f_1(x^{(k)}),\dots,\nabla f_n(x^{(k)})\} \in \mathbb{R}^{nd}, \end{aligned}$$

Algorithm 1 Gradient Tracking with Periodic Global Averaging (GT-PGA)

Agent i input: $x_i^{(0)} \in \mathbb{R}^d$, stepsize $\alpha \in \mathbb{R}_{>0}$, mixing matrix $W \in \mathbb{R}^{n\times n}$, averaging period $\tau \in \mathbb{N}_{\geq 1}$.
Initialize $g_i^{(0)} = \nabla F_i(x_i^{(0)}, \xi_i^{(0)}) \in \mathbb{R}^d$ for all $i \in [n]$.
for $k = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, n$ (in parallel) **do**
    if $\mathrm{mod}(k+1, \tau) = 0$ **then**

$$x_i^{(k+1)} = \frac{1}{n}\sum_{j=1}^n (x_j^{(k)} - \alpha g_j^{(k)})$$

$$g_i^{(k+1)} = \frac{1}{n}\sum_{j=1}^n g_j^{(k)} + \nabla F_i(x_i^{(k+1)}, \xi_i^{(k+1)}) - \nabla F_i(x_i^{(k)}, \xi_i^{(k)}).$$

    else

$$x_i^{(k+1)} = \sum_{j:(j,i)\in\mathcal{E}} w_{ij}(x_j^{(k)} - \alpha g_j^{(k)})$$

$$g_i^{(k+1)} = \sum_{j:(j,i)\in\mathcal{E}} w_{ij} g_j^{(k)} + \nabla F_i(x_i^{(k+1)}, \xi_i^{(k+1)}) - \nabla F_i(x_i^{(k)}, \xi_i^{(k)}).$$

    end if
  end for
end for

$$\begin{aligned}
\nabla \boldsymbol{F}^{(k)} &:= \mathrm{col}\{\nabla F_1(x_1^{(k)}; \xi_1^{(k)}), \ldots, \nabla F_n(x_n^{(k)}; \xi_n^{(k)})\},\\
\hat{\boldsymbol{x}}^{(k)} &:= \boldsymbol{x}^{(k)} - \mathbb{1}_n \otimes \bar{x}^{(k)} \in \mathbb{R}^{nd},\\
\boldsymbol{W} &:= W \otimes I_d, \quad \widehat{\boldsymbol{W}} := \boldsymbol{W} - \tfrac{1}{n}\mathbb{1}_n\mathbb{1}_n^{\mathsf{T}} \otimes I_d,\\
\boldsymbol{f}(\boldsymbol{x}^{(k)}) &:= \tfrac{1}{n}\sum_{i=1}^n f_i(x), \quad \overline{\nabla f}(\boldsymbol{x}^{(k)}) := \tfrac{1}{n}\sum_{i=1}^n \nabla f_i(x_i^{(k)}),\\
\bar{x}^{(k)} &:= \tfrac{1}{n}\sum_{i=1}^n x_i^{(k)}.
\end{aligned}$$

With the augmented notations, the main recursion of Algorithm 1 can be written concisely as:

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \boldsymbol{W}^{(k)}(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^k)\\
\boldsymbol{g}^{(k+1)} &= \boldsymbol{W}^{(k)}\boldsymbol{g}^{(k)} + \nabla \boldsymbol{F}(\boldsymbol{x}^{(k+1)}; \boldsymbol{\xi}^{(k+1)}) - \nabla \boldsymbol{F}(\boldsymbol{x}^{(k)}; \boldsymbol{\xi}^{(k)}),
\end{aligned}$$

where $\boldsymbol{W}^{(k)} := W^{(k)} \otimes I_d$ and

$$W^{(k)} = \begin{cases} \frac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}} & \text{if } \mathrm{mod}(k+1,\tau) = 0\\ W & \text{otherwise.}\end{cases}$$

Now, we list all the assumptions needed for the analysis.

Assumption 1 (Mixing matrix) *The network is strongly connected, and the mixing matrix $W \in \mathbb{R}^{n\times n}$ satisfies $W\mathbb{1} = \mathbb{1}$, $W^{\mathsf{T}}\mathbb{1} = \mathbb{1}$, and $\mathrm{Null}(I - W) = \mathbf{span}(\mathbb{1})$. In addition, denote*

$$\beta := \|W - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\|_2 \in (0, 1).$$

The quantity β indicates how well the network is connected. A smaller β indicates a better connected network while a larger one implies a worse connectivity.

The following two assumptions are made on the problem (1). In particular, convexity is not assumed, and the algorithm only has access to stochastic gradient estimates of each local function.

Assumption 2 (L-smoothness) *Each function $f_i\colon \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable with an L-Lipschitz continuous gradient; i.e., there exists a constant $L \in \mathbb{R}_{>0}$ such that*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|,$$

for all $(x, y) \in \mathbf{int\,dom}\, f_i \times \mathbf{int\,dom}\, f_i$ and for all $i \in [n]$. In addition, the objective function $f\colon \mathbb{R}^d \to \mathbb{R}$ is bounded below, and the optimal value of problem (1) is denoted by $f^\star \in \mathbb{R}$.

At iteration k of Algorithm 1, a stochastic gradient estimator of each component function f_i is computed, based on the random variable $\xi_i^{(k)} \in (\Omega_i, \mathcal{F}_i, \mathbb{P}_i)$. let $\mathcal{F}^{(0)}$ denote the σ-algebra corresponding to the initial conditions and, for all $k \in \mathbb{N}_{\geq 1}$, let $\mathcal{F}^{(k)}$ denote the σ-algebra defined by $\{\boldsymbol{x}^{(j)}\}_{j=0}^k$. The following assumption is made on the stochastic gradient estimator.

Assumption 3 (Stochastic noise) *For all $k \in \mathbb{N}$ and for all $i \in [n]$, the random variables $\xi_i^{(k)}$ are independent of each other. The stochastic gradient estimator satisfies*

$$\mathbb{E}[\nabla F_i(x_i^{(k)}; \xi_i^{(k)}) \mid \mathcal{F}^{(k)}] = \nabla f_i(x_i^{(k)}), \quad \text{for all } i \in [n].$$

In addition, there exists $\sigma \in \mathbb{R}_{>0}$ such that for all $k \in \mathbb{N}$ and for all $i \in [n]$, it holds that

$$\mathbb{E}[\|\nabla F_i(x_i^{(k)}; \xi_i^{(k)}) - \nabla f_i(x_i^{(k)})\|^2 \mid \mathcal{F}^{(k)}] \leq \sigma^2.$$

We now state the main result of this paper on the convergence guarantees of Algorithm 1.

Theorem 1 (Convergence of GT-PGA) *Let Assumptions 1 to 3 hold, let $\tau \in \mathbb{N}_{\geq 2}$, and let the stepsize satisfy $\alpha \leq \min\{\frac{1}{2L}, \frac{1}{4\sqrt{6}\beta\tau^2 L}\}$. Then, for any $K \in \mathbb{N}_{\geq \tau+1}$, the sequence $\{\boldsymbol{x}^{(k)}\}$ generated by Algorithm 1 satisfies*

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^K \Big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big)\\
&\leq \frac{\gamma_1 L^2}{nK} + \frac{\gamma_2 \beta\tau^2 L^2}{K} + \gamma_3\sigma^2\Big(\frac{1}{(1-\beta^2)\tau^2} + \frac{1}{\beta\tau^2 n}\Big) \quad (3)
\end{aligned}$$

with some constants $(\gamma_1, \gamma_2, \gamma_3) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$.

In general, GT-PGA exhibits an $O(1/K)$ convergence rate, consistent with the results for gradient tracking algorithms under the stochastic, nonconvex setup (see, *e.g.*, [15]). As is typical in the literature, the first term on the right-hand side of (3) is related to the number of agents and independent of the topology as well as the PGA period τ. The crucial trade-off between the connectivity of the communication network (β) and the PGA period (τ) is depicted in the second term in (3).

- When the network is large or sparse (*i.e.*, $\beta \to 1$), global averaging is more critical to drive consensus and a smaller τ is needed to compensate the negative effect of poor connectivity.
- When the network is small or dense, gossip communication is already helpful enough to achieve consensus and

a larger τ can be used. In the extreme case, GT-PGA with $\tau \to \infty$ reduces to vanilla GT.

- Recall that when $W = I$, GT-PGA reduces to GT with fully-connected graphs and with local updates (LU-GT). Thus, gossip communication in GT-PGA also contributes to consensus, and this property is critical to establish the superiority of GT-PGA with LU-GT.

Therefore, thanks to the periodic global averaging operation, GT-PGA enjoys promising convergence properties compared with vanilla GT and LU-GT, and our analysis supports the above discussion.

We end this section with an auxiliary convergence result with further tuning on the stepsize α. The same stepsize tuning strategy is common in the literature on decentralized optimization; see, *e.g.*, [21], [23], [27], [32].

Corollary 2 *Let Assumptions 1 to 3 hold, let $\tau \in \mathbb{N}_{\geq 2}$, and let the stepsize satisfy*

$$\alpha = \min\left\{\left(\frac{nL}{K\sigma^2}\right)^{\frac{1}{2}}, \left(\frac{1-\beta^2}{\beta^2\tau^2 K\sigma^2}\right)^{\frac{1}{2}}, \frac{1}{2L}, \frac{1}{4\sqrt{6}\beta\tau^2 L}\right\}. \tag{4}$$

In addition, suppose $n \gg 1/(\beta\tau)^2$ (e.g., the number of agents n is sufficiently large or the network is sufficiently sparse). Then, for any $K \in \mathbb{N}_{\geq \tau+1}$, the sequence $\{\boldsymbol{x}^{(k)}\}$ generated by Algorithm 1 satisfies

$$\frac{1}{K+1}\sum_{k=0}^{K}\Big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big) \leq \frac{\gamma_4\beta\tau^2L^3}{K} + \frac{\gamma_5 L^{\frac{3}{2}}\sigma}{(nK)^{\frac{1}{2}}} + \frac{\gamma_6\beta\tau L^2\sigma}{(1-\beta^2)^{\frac{1}{2}}K^{\frac{1}{2}}}$$

with some constants $(\gamma_4, \gamma_5, \gamma_6) \in \mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}$.

III. Algorithm Analysis

This section presents the theoretical analysis of Algorithm 1 stated in Theorem 1. As typical in the analyses of decentralized algorithms, the two important pillars are the *descent inequality* and the *consensus inequality*. The descent inequality establishes the convergence properties of the averaged iterates $\bar{x}^{(k)}$ to a first-order stationary point and is standard in the analyses of GT (see upcoming Lemma 3). The consensus inequality is different from existing analyses and characterizes the per-iteration behavior of the consensus error; see upcoming Lemma 4.

Lemma 3 (Descent inequality [35, Lemma 5.1]) *Let Assumptions 1 to 3 hold, and let the stepsize satisfy $\alpha \in \big(0, \frac{1}{2L}\big]$. Denote $\tilde{f} := f - f^\star$. Then, the sequence generated by Algorithm 1 satisfies*

$$\frac{1}{K+1}\sum_{k=0}^{K}\big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\big) \leq \frac{4}{\alpha(K+1)}\mathbb{E}\tilde{f}(\bar{x}^{(0)}) + \frac{2L^2}{n(K+1)}\sum_{k=0}^{K}\mathbb{E}\|\hat{\boldsymbol{x}}^{(k)}\|^2 + \frac{2\alpha L\sigma^2}{n},$$

for all $k \in \mathbb{N}$.

Note that this inequality does not involve the mixing matrix W, so it holds for gradient tracking with static topology as well as the proposed GT-PGA. Its derivation is standard in the literature and thus omitted here.

The second lemma studies the behavior of the consensus error and is used to establish that all agents' local variables converge to their average.

Lemma 4 (Consensus inequality) *Let Assumptions 1 to 3 hold, let $\tau \in \mathbb{N}_{\geq 2}$, and let the stepsize satisfy $\alpha \in \big(0, \frac{1}{4\sqrt{6}\beta\tau^2 L}\big]$. Then, for $K \in \mathbb{N}_{\geq \tau+1}$, the iterates generated by Algorithm 1 satisfy*

$$\begin{aligned}\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}\|\hat{\boldsymbol{x}}^{(k)}\|^2 \leq{}& \frac{2}{K+1}\sum_{k=0}^{\tau}\mathbb{E}[\|\hat{\boldsymbol{x}}^{(k)}\|^2] \\ &+ \frac{n}{192\beta^2\tau^4L^2(K+1)}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(\bar{x}^{(k)})\|^2] \\ &+ \Big(\frac{1}{768\beta^2\tau^4L^2} + \frac{n}{24\tau^4L^2} + \frac{n}{6(1-\beta^2)\tau^2L^2}\Big)\sigma^2. \end{aligned} \tag{5}$$

Due the space limitation, the proof of Lemma 4 can be found in the arxiv version [36, Lemma 4].

With Lemmas 3 and 4, we are ready to present the proof of the main result of this paper.

Proof of Theorem 1: Combining Lemma 3 with (5) yields

$$\begin{aligned}&\frac{1}{K+1}\sum_{k=0}^{K}\Big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2\Big) \\ &\leq \frac{4}{\alpha(K+1)}\mathbb{E}\tilde{f}(\bar{x}^{(0)}) + \frac{4L^2}{n(K+1)}\sum_{k=0}^{\tau}\mathbb{E}[\|\hat{\boldsymbol{x}}^{(k)}\|^2] \\ &\quad + \frac{96\alpha^4\beta^2\tau^4nL^4}{K+1}\sum_{k=0}^{K}\big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}[\|\nabla f(\bar{x}^{(k)})\|^2]\big) \\ &\quad + \Big(\frac{24\alpha^4\beta^2\tau^4L^4 + 2\alpha L}{n} + 8\alpha^2\beta^2L^2 + \frac{32\alpha^2\beta^2\tau^3L^2}{1-\beta^2}\Big)\sigma^2.\end{aligned}$$

Grouping similar terms on the left-hand side gives

$$\begin{aligned}&\frac{1-96\alpha^4\beta^2\tau^4nL^4}{K+1}\sum_{k=0}^{K}\big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}[\|\nabla f(\bar{x}^{(k)})\|^2]\big) \\ &\leq \frac{4}{\alpha(K+1)}\mathbb{E}\tilde{f}(\bar{x}^{(0)}) + \frac{4L^2}{n(K+1)}\sum_{k=0}^{\tau}\mathbb{E}[\|\hat{\boldsymbol{x}}^{(k)}\|^2] \\ &\quad + \Big(\frac{24\alpha^4\beta^2\tau^4L^4 + 2\alpha L}{n} + 8\alpha^2\beta^2L^2 + \frac{32\alpha^2\beta^2\tau^2L^2}{1-\beta^2}\Big)\sigma^2.\end{aligned}$$

The stepsize condition $\alpha \leq \frac{1}{4\sqrt{6}\beta\tau^2L}$ implies that

$$\begin{aligned}&\frac{1}{K+1}\sum_{k=0}^{K}\big(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \mathbb{E}[\|\nabla f(\bar{x}^{(k)})\|^2]\big) \\ &\leq \frac{8}{\alpha(K+1)}\mathbb{E}\tilde{f}(\bar{x}^{(0)}) + \frac{8L^2}{n(K+1)}\sum_{k=0}^{\tau}\mathbb{E}[\|\hat{\boldsymbol{x}}^{(k)}\|^2] \\ &\quad + \Big(\frac{48\alpha^4\beta^2\tau^4L^4 + 4\alpha L}{n} + 16\alpha^2\beta^2L^2 + \frac{64\alpha^2\beta^2\tau^2L^2}{1-\beta^2}\Big)\sigma^2 \\ &\leq \frac{32\sqrt{6}\beta\tau^2L}{K+1}\mathbb{E}\tilde{f}(\bar{x}^{(0)}) + \frac{8L^2}{n(K+1)}\sum_{k=0}^{\tau}\mathbb{E}[\|\hat{\boldsymbol{x}}^{(k)}\|^2]\end{aligned}$$

$$+\frac{\sigma^2}{192\beta^2\tau^4 n}+\frac{\sigma^2}{\sqrt{6}\beta\tau^2 n}+\frac{\sigma^2}{6\tau^4}+\frac{2\sigma^2}{3(1-\beta^2)\tau}, \tag{6}$$

where in the last step we plug in the stepsize condition. ■

Now we state the proof of Corollary 2. To improve the readability of the proof, from now on, we use the notation $\lesssim$ to hide irrelevant constants. The notation $a \lesssim b$ means that there exists a positive constant $\gamma \in \mathbb{R}_{>0}$ such that $a \leq \gamma b$. In our case, the important quantities that we keep are α, β, τ, n, L, and σ.

Proof of Corollary 2: From the stepsize condition $\alpha \leq \min\left\{\frac{1}{2L}, \frac{1}{4\sqrt{6}\beta\tau^2 L}\right\}$, the inequality (6) becomes

$$\begin{aligned}
&\frac{1}{K+1}\sum_{k=0}^{K}\left(\mathbb{E}\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2+\mathbb{E}[\|\nabla f(\bar{x}^{(k)})\|^2]\right)\\
&\lesssim \frac{L^2}{\alpha K}+\frac{\alpha L\sigma^2}{n}+\frac{\alpha^4\beta^2\tau^4L^4\sigma^2}{n}+\frac{\alpha^2\beta^2\tau^2L^2\sigma^2}{1-\beta^2}\\
&\lesssim \frac{L^2}{\alpha K}+\frac{\alpha L\sigma^2}{n}+\frac{\alpha^2\alpha^2\beta^2\tau^4L^4\sigma^2}{n}+\frac{\alpha^2\beta^2\tau^2L^2\sigma^2}{1-\beta^2}\\
&\lesssim \frac{L^2}{\alpha K}+\frac{\alpha L\sigma^2}{n}+\frac{\alpha^2L^2\sigma^2}{n}+\frac{\alpha^2\beta^2\tau^2L^2\sigma^2}{1-\beta^2}\\
&\lesssim \frac{L^2}{\alpha K}+\frac{\alpha L\sigma^2}{n}+\frac{\alpha^2\beta^2\tau^2L^2\sigma^2}{1-\beta^2},\\
&\lesssim \frac{c_1}{\alpha K}+c_2\alpha+c_3\alpha^2,
\end{aligned} \tag{7}$$

where $c_1 = L^2$, $c_2 = \frac{L\sigma^2}{n}$, and $c_3 = \frac{\beta^2\tau^2L^2\sigma^2}{1-\beta^2}$. In (7) we use the assumption that $\frac{1}{n} \ll \beta^2\tau^2$. Now we set the stepsize α as in (4). (Note that by definition, this choice of α satisfies the condition in Theorem 1.) We then discuss the following three cases.

1) If $\alpha = \min\left\{\frac{1}{2L}, \frac{1}{4\sqrt{6}\beta\tau^2 L}\right\}$, then

$$\frac{c_1}{\alpha K}+c_2\alpha+c_3\alpha^2 \lesssim \frac{c_1}{\alpha K}+\left(\frac{c_1c_2}{K}\right)^{\frac{1}{2}}+\left(\frac{c_1c_3}{K}\right)^{\frac{1}{2}}.$$

2) If $\alpha = \left(\frac{c_1}{c_2K}\right)^{\frac{1}{2}} \leq \left(\frac{c_1}{c_3K}\right)^{\frac{1}{2}}$, then

$$\frac{c_1}{\alpha K}+c_2\alpha+c_3\alpha^2 \lesssim \left(\frac{c_1c_2}{K}\right)^{\frac{1}{2}}+\left(\frac{c_1c_3}{K}\right)^{\frac{1}{2}}.$$

3) If $\alpha = \left(\frac{c_1}{c_3K}\right)^{\frac{1}{2}} \leq \left(\frac{c_1}{c_2K}\right)^{\frac{1}{2}}$, then

$$\frac{c_1}{\alpha K}+c_2\alpha+c_3\alpha^2 \lesssim \left(\frac{c_1c_2}{K}\right)^{\frac{1}{2}}+\left(\frac{c_1c_3}{K}\right)^{\frac{1}{2}}.$$

Combing all three cases yields

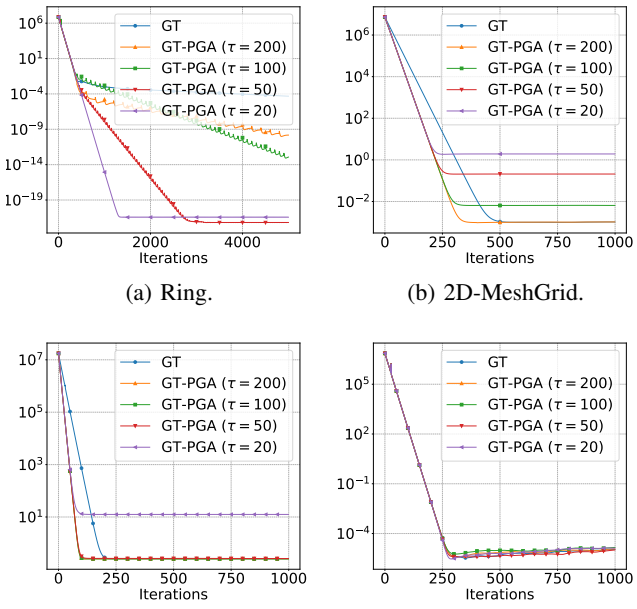
$$\begin{aligned}
\frac{c_1}{\alpha K}+c_2\alpha+c_3\alpha^2 &\lesssim \frac{c_1}{\alpha K}+\left(\frac{c_1c_2}{K}\right)^{\frac{1}{2}}+\left(\frac{c_1c_3}{K}\right)^{\frac{1}{2}}\\
&\lesssim \frac{\beta\tau^2L^3}{K}+\frac{L^{\frac{3}{2}}\sigma}{(nK)^{\frac{1}{2}}}+\frac{\beta\tau L^2\sigma}{(1-\beta^2)^{\frac{1}{2}}K^{\frac{1}{2}}}.
\end{aligned}$$

IV. Numerical Experiments

This numerical experiments presented in this section illustrate how GT-PGA accelerates practical convergence compared to vanilla GT. We apply GT-PGA to solve the least squares problem with a nonconvex regularization term:

$$\text{minimize} \quad \frac{1}{n}\sum_{i=1}^{n}\|A_ix-b_i\|_2^2+\lambda\sum_{j=1}^{d}\frac{x[j]}{1+x[j]}, \tag{8}$$

where the decision variable is $x \in \mathbb{R}^d$, $x[j]$ is the jth component of x, and $\{(A_i, b_i)\}_{i=1}^n \subset \mathbb{R}^{m_i\times d}\times\mathbb{R}^{m_i}$ are the local data held by agent i. In our simulation, we set $\lambda = 0.01$, $n = 64$, $d = 20$, $m_i = 500$ for all $i \in [n]$, and the entries of each A_i are drawn independently from standard Gaussian distribution. For all $i \in [n]$, we randomly generate $\tilde{x}_i \in \mathbb{R}^d$ and set $b_i = A_i\tilde{x}_i + z_i$, where $z_i \sim \mathcal{N}(0, 0.01)$ are drawn independently. We test Algorithm 1 on various topologies, including the ring graph, 2D-MeshGrid, star graph, and static hypercuboid [35].



(a) Ring. (b) 2D-MeshGrid.

(c) Star. (d) Static hypercuboid.

Fig. 1: Performance of GT-PGA for solving (8) with various topologies. The plots report $\|\overline{\nabla f}(\boldsymbol{x}^{(k)})\|^2 + \|\nabla f(\bar{x}^{(k)})\|^2$, and use the ring, 2D-MeshGrid, the star graph, and the static hypercuboid, respectively. Different curves on each figure use different PGA period, *i.e.*, $\tau = 20, 50, 100, 200$, and ∞ (equivalent to vanilla GT).

The simulation results depicted in Figure 1 show the superiority of the PGA operation and align with the theoretical insight from Theorem 1 and Corollary 2. For the sparse ring graph (Figure 1a), PGA helps reduce the stochastic noise caused by stochastic gradients, and GT-PGA converges to a more accurate solution compared to vanilla GT. For 2D-MeshGrid and the star graph, GT-PGA exhibits a better practical performance in terms of convergence rate. For static hyper-cuboids, the benefit of GT-PGA is marginal, potentially because of the desirable properties of static hypercuboids [35].

V. Conclusion

We incorporate periodic global averaging (PGA) into Gradient Tracking (GT) and propose a new decentralized algorithm GT-PGA. We establish convergence guarantees for GT-PGA under the stochastic, nonconvex setting and showcase the superiority of GT-PGA compared with vanilla GT. Numerical results validate the improvements in practical convergence due to the proposed periodic global averaging operation. While we focus on the nonconvex setting (due to space constraints), it is straightforward to extend our analysis to the convex setting.

In this work, we focus on a specific form of GT [15]. It is not clear whether PGA can be incorporated into other forms of GT and whether a unified analysis similar to, *e.g.*, [37], still holds. Moreover, the connection (or difference) between PGA and multi-consensus is still unknown, and further analysis is needed to quantify the trade-off between these two techniques.

References

[1] J. B. Predd, S. B. Kulkarni, and V. H. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[2] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[3] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2012.

[4] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[5] B. Ying, K. Yuan, H. Hu, Y. Chen, and W. Yin, "BlueFog: Make decentralized algorithms practical for optimization and deep learning," *arXiv e-prints*, vol. arXiv:2111.04287, 2021.

[6] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[8] 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, no. 3, pp. 516–545, 2010.

[9] 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.

[10] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—Part I: Algorithm development," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708–723, 2018.

[11] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D^2: Decentralized training over decentralized data," in *International Conference on Machine Learning*, 2018, pp. 4848–4856.

[12] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4494–4506, 2019.

[13] K. Yuan, S. A. Alghunaim, B. Ying, and A. H. Sayed, "On the influence of bias-correction on distributed stochastic optimization," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4352–4367, 2020.

[14] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *Proceedings of 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 2055–2060.

[15] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[16] A. Nedić, 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.

[17] 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, 2018.

[18] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, 2013.

[19] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[20] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[21] S. U. Stich, "Local SGD converges fast and communicates little," in *International Conference on Learning Representations*, 2019.

[22] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local SGD on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 4519–4529.

[23] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*, 2020, pp. 5132–5143.

[24] E. Gorbunov, F. Hanzely, and P. Richtárik, "Local SGD: Unified theory and new efficient methods," in *International Conference on Artificial Intelligence and Statistics*, vol. 130, 2021, pp. 3556–3564.

[25] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients," in *Advances in Neural Information Processing Systems*, 2021.

[26] K. Mishchenko, G. Malinovsky, S. U. Stich, and P. Richtárik, "ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally!" in *International Conference on Machine Learning*, 2022.

[27] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *International Conference on Machine Learning*, 2020, pp. 5381–5393.

[28] A. S. Berahas, R. Bollapragada, and S. Gupta, "Balancing communication and computation in gradient tracking algorithms for decentralized optimization," *arXiv e-prints arXiv:2303.14289*, 2023.

[29] S. Ge and T.-H. Chang, "Gradient tracking with multiple local SGD for decentralized non-convex learning," in *62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 133–138.

[30] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized gradient tracking with local steps," *arXiv e-prints arXiv:2301.01313*, 2023.

[31] E. D. H. Nguyen, S. A. Alghunaim, K. Yuan, and C. A. Uribe, "On the performance of gradient tracking with local updates," in *62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 4309–4313.

[32] S. A. Alghunaim, "Local exact-diffusion for decentralized optimization and learning," *arXiv e-prints arXiv:2302.00620*, 2023.

[33] Y. Chen, K. Yuan, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "Accelerating gossip SGD with periodic global averaging," in *International Conference on Machine Learning*, 2021, pp. 1791–1802.

[34] P. Patarasuk and X. Yuan, "Bandwidth optimal all-reduce algorithms for clusters of workstations," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 117–124, 2009.

[35] E. D. H. Nguyen, X. Jiang, B. Ying, and C. A. Uribe, "On graphs with finite-time consensus and their use in gradient tracking," *arXiv e-prints arXiv:2311.01317*, 2023.

[36] S. Feng and X. Jiang, "Accelerating gradient tracking with periodic global averaging," *arXiv e-prints*, vol. arXiv:2403.11293, 2024.

[37] S. A. Alghunaim and K. Yuan, "A unified and refined convergence analysis for non-convex decentralized learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3264–3279, 2022.