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

On the Performance of Gradient Tracking with Local Updates

Edward Duc Hien Nguyen, Sulaiman A. Alghunaim, Kun Yuan and César A. Uribe

***Abstract*— We study the decentralized optimization problem where a network of n agents seeks to minimize the average of a set of heterogeneous non-convex cost functions distributedly. State-of-the-art decentralized algorithms like Exact Diffusion and Gradient Tracking (GT) involve communicating every iteration. However, communication is expensive, resource intensive, and slow. This work analyzes a locally updated GT method (LU-GT), where agents perform local recursions before interacting with their neighbors. While local updates have been shown to reduce communication overhead in practice, their theoretical influence has not been fully characterized. We show LU-GT has the same communication complexity as the Federated Learning setting but allows decentralized (symmetric) network topologies. In addition, we prove that the number of local updates does not degrade the quality of the solution achieved by LU-GT. Numerical results reveal that local updates may lead to lower communication costs in specific regimes (e.g., well-connected graphs).**

I. Introduction

We study the distributed multi-agent optimization problem

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \quad f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x), \qquad (1)$$

where $f_i(\cdot) : \mathbb{R}^m \to \mathbb{R}$ is a smooth, *non-convex* function held privately by agent $i \in \{1, \ldots, n\}$. The agents collaborate to find a consensual solution $x*$ of (1) with communication constrained by some network topology.

Many decentralized methods have been proposed to solve (1). Among the most prolific include decentralized/distributed gradient descent (DGD) [1], [2], EXTRA [3], Exact-Diffusion/D^2/NIDS (ED) [4]–[7], and Gradient Tracking (GT) [8]–[11]. DGD is an algorithm wherein agents perform a local gradient step followed by a communication round. However, DGD has been shown not optimal for constant stepsizes when agents' local objective functions are heterogeneous, i.e., the minimizer of functions $f_i(\cdot)$ differs from the minimizer of $f(\cdot)$. This shortcoming has been analyzed in [12], [13] where the heterogeneity causes the rate of DGD to incur an additional bias term with a magnitude directly proportional to the level of heterogeneity. Moreover, this bias term is inversely influenced by the connectivity of the network (becomes larger for sparse networks) [6], [14].

EXTRA, ED, and GT employ bias-correction techniques to account for heterogeneity. EXTRA and ED use local updates with memory. GT methods have each agent perform the local update with an estimate of the global gradient called the tracking variable. In these techniques, the bias term proportional to the heterogeneity found in DGD is removed [15], [16]. However, they require communication over the network at every iteration.

Communication is expensive, resource intensive, and slow in practice [17]. Centralized methods in which agents communicate with a central coordinator (i.e., server) have been developed to solve (1) with an explicit focus on reducing the communication cost. This has been achieved empirically by requiring agents to perform local recursions before communicating. Among these methods include LocalGD [18]–[22], Scaffold [23], S-Local-GD [24], FedLin [25], and Scaffnew [26]. Analysis on LocalGD revealed that local recursions cause agents to drift towards their local solution [14], [19], [27]. Scaffold, S-Local-GD, FedLin, and Scaffnew address this issue by introducing bias-correction techniques. However, besides [26], analysis of these methods has failed to show communication complexity improvements. The work [26] has shown that for μ-strongly-convex, L-smooth, and deterministic functions, the communication complexity of Scaffnew can be improved from $O(\kappa)$ to $O(\sqrt{\kappa})$ if one performs $\sqrt{\kappa}$ local recursions with $\kappa \triangleq L/\mu$.

Local recursions in *decentralized methods* have been much less studied. DGD with local recursions has been studied in [14], but the convergence rates still have bias terms due to heterogeneity. Additionally, the magnitude of the bias term is proportional to the number of local recursions taken. Scaffnew [26] has been studied under the decentralized case but for the strongly convex and smooth function class. In [26], for sufficiently connected graphs, an improvement to a communication complexity of $O(\sqrt{\kappa/(1-\lambda)})$ where λ is the mixing rate of the matrix is shown. Several works studied GT under time-varying graphs such as [9], [11], [28]–[30], among these only the works [9], [28], [31] considered nonconvex setting. Different from [9], [28], [31], we provide explicit expressions that characterize the convergence rate in terms of the problem parameters (e.g., network topology).

In this work, we propose and study LU-GT, a locally updated decentralized algorithm based on the bias-corrected method GT. Our contributions are as follows:

- We analyze LU-GT under the deterministic, non-convex regime. As a byproduct, we provide an alternative and simpler analysis for GT, which extends the techniques from [15].
- We show LU-GT has a communication complexity match-

EDHN and CAU (*{en18,cauribe}@rice.edu*) are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA. SAA (*sulaiman.alghunaim@ku.edu.kw*) is with the Department of Electrical Engineering, Kuwait University, Kuwait City, Kuwait. KY (*kun.yuan@alibaba-inc.com*) is with Alibaba DAMO Academy, Hangzhou, Zhejiang, China.

Edward Nguyen is supported by a training fellowship from the Gulf Coast Consortia, on the NLM Training Program in Biomedical Informatics & Data Science (T15LM007093). The work of CAU and EDHN was partially supported by the National Science Foundation under Grants No. 2211815 and No. 2213568.

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

ing locally updated variants of federated algorithms.

- We demonstrate that LU-GT retains the bias-correction properties of GT irrespective of the number of local recursions and that the number of local recursions does not affect the quality of the solution.
- Numerical analysis shows that local recursions can reduce the communication overhead in certain regimes, e.g., well-connected graphs.

This paper is organized as follows. Section II defines relevant notation, states the assumptions used in our analysis, introduces LU-GT, and states our main result on the convergence rate. In Section III, we prove the convergence rate of LU-GT. Section IV shows evidence that the local recursions of LU-GT can reduce communication costs in certain regimes. We have two additional appendix sections available on the arXiv version in which we provide intuition into how the direction of our analysis can show that following LU-GT, agents reach a consensus that is also a first-order stationary point. We also cover relevant lemmas needed in the analysis of LU-GT in the appendix.

Notation: Lowercase letters define vectors or scalars, while uppercase letters define matrices. We let $\mathrm{col}\{a_1,...,a_n\}$ or $\mathrm{col}\{a_i\}_{i=1}^n$ denote the vector that concatenates the vectors/scalars a_i. We let $\mathrm{diag}\{d_1,...,d_n\}$ or $\mathrm{diag}\{d_i\}_{i=1}^n$ denote the matrix with diagonal elements d_i. Similarly, $\mathrm{blkdiag}\{D_1,D_2,...,D_n\}$ or $\mathrm{blkdiag}\{D_i\}_{i=1}^n$ represents the block diagonal matrix with matrices D_i along the diagonal. The notation $\mathbf{1}$ represents the one vector of size that should be inferred while $\mathbf{1}_n$ represents the one vector of size n. The inner product of two vectors a,b is defined as $\langle a,b\rangle$. $\otimes$ represents the Kronecker product. Boldface variables such as $(\mathbf{x},\mathbf{W})$ represent augmented network quantities.

II. Algorithm, Assumptions, and Main Result

The original gradient tracking method has the form [8]:

$$x_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j^k - \bar{\eta} g_j^k)$$

$$g_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}\big(g_j^k + \nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k)\big),$$

with $g_i^0 = \nabla f(x_i^0)$. Here, x_i^k is agent i's current parameter estimate at iteration k, and $g_i^k \in \mathbb{R}^n$ is an additional parameter held by agent i that tracks the average of the gradient. Here, w_{ij} is a scalar weight that scales the information agent i receives from agent j, and $\mathcal{N}_i$ is the set of neighbors of agent i. We set $w_{ij}=0$ if $j\notin\mathcal{N}_i$.

In this work, we study a locally updated variant of gradient tracking listed in Alg. 1 where instead of agents communicating every iteration, they communicate every T_{o} iterations. The proposed method LU-GT is detailed in Algorithm 1 where α and η are step-size parameters, and T_{o} is the number of local recursions before a round of communication. The intuition behind the algorithm is to have agents perform a descent step using a staling estimate of the global gradient for T_{o} iterations. Afterwards, agents perform a weighted average of their parameters with their neighbors and update their tracking variable.

Algorithm 1 LU-GT for each agent i

```
Input: $x_i^0 = 0 \in \mathbb{R}^m$, $y_i^0 = \alpha\nabla f_i(x_i^0)$, $\alpha > 0$, $\eta > 0$
    $T_{\mathrm{o}} \in \mathbb{Z}_{\geq 0}$, $K \in \mathbb{Z}_+$
Define: $\tau = \{0, T_{\mathrm{o}}, 2T_{\mathrm{o}}, 3T_{\mathrm{o}}...\}$
for $k = 0, ..., K-1$ do
    if $k \in \tau$ then
        $x_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j^k - \eta y_j^k)$
        $y_i^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij}(y_j^k + \alpha\nabla f_j(x_j^{k+1}) - \alpha\nabla f_j(x_j^k))$
    else
        $x_i^{k+1} = x_i^k - \eta y_i^k$
        $y_i^{k+1} = y_i^k + \alpha\nabla f_i(x_i^{k+1}) - \alpha\nabla f_i(x_i^k)$
    end if
end for
```

Remark 1 *For $T_{\mathrm{o}} = 1$, Algorithm 1 becomes equivalent to the original ATC-GT [8] with stepsize $\bar{\eta} = \eta\alpha$. This can be seen by introducing the change of variable $g_i^k = (1/\alpha)y_i^k$. Thus, our analysis also covers the original GT method.*

For analysis reasons, we will rewrite algorithm 1 using network notation. To do so,we define $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ as the mixing matrix for an undirected graph that models the connections of a group of n agents. We also introduce the network notations:

$$\mathbf{W} = W \otimes I_d \in \mathbb{R}^{mn\times mn}$$

$$\mathbf{x}^k = \mathrm{col}\{x_1^k,\ldots,x_n^k\}, \quad \mathbf{y}^k = \mathrm{col}\{y_1^k,\ldots,y_n^k\}$$

$$\mathbf{f}(\mathbf{x}) {=} \sum_{i=1}^n f_i(x_i), \quad \nabla\mathbf{f}(\mathbf{x}) {=} \mathrm{col}\{\nabla f_1(x_1),\ldots,\nabla f_n(x_n)\}.$$

To analyze Algorithm 1, we first introduce the following time-varying matrix:

$$\mathbf{W}_k \triangleq \begin{cases} \mathbf{W} & \text{when } k \in \tau, \\ \mathbf{I} & \text{otherwise.} \end{cases}$$

Thus, we can succinctly rewrite Algorithm 1 as follows

$$\mathbf{x}^{k+1} = \mathbf{W}_k(\mathbf{x}^k - \eta\mathbf{y}^k) \tag{4a}$$

$$\mathbf{y}^{k+1} = \mathbf{W}_k(\mathbf{y}^k + \alpha\nabla\mathbf{f}(\mathbf{x}^{k+1}) - \alpha\nabla\mathbf{f}(\mathbf{x}^k)). \tag{4b}$$

We now list the assumptions used in our analysis.

Assumption 1 (Mixing matrix) *The mixing matrix W is doubly stochastic and symmetric.*

The Metropolis-Hastings algorithm [32] can be used to construct mixing matrices from an undirected graph satisfying Assumption 1. Moreover, from Assumption 1, the mixing matrix W has a singular, maximum eigenvalue denoted as $\lambda_1 = 1$. All other eigenvalues are defined as $\{\lambda_i\}_{i=2}^n$. We define the mixing rate as $\lambda := \max_{i\in\{2,\ldots,n\}}\{|\lambda_i|\}$.

Assumption 2 (L-smoothness) *Each function $f_i : \mathbb{R}^m \to \mathbb{R}$ is L-smooth for $i \in \mathcal{V}$, i.e., $\|\nabla f_i(y) - \nabla f_i(z)\| \leq L\|y-z\|$, $\forall\ y,z \in \mathbb{R}^m$ for some $L > 0$. We assume there exists a $f^* \in \mathbb{R}$ such that $f(x) \geq f^*$.*

We are now ready to state the main result of this paper on the convergence analysis of LU-GT.

Theorem 1 (Convergence of LU-GT) *Let Assumptions 1 and 2 hold, and let, $T_{\mathrm{o}} \in \mathbb{Z}_{\geq 0}$, $\eta > 0$, and $\alpha > 0$ with $\eta < O(1/T_{\mathrm{o}})$, and $\alpha < O((1-\lambda)/L)$ (Exact bounds found in (6), (7), and (9)). Then, for any $K \geq 1$, the output $\mathbf{x}^K$, of Algorithm 1 (LU-GT) with $\mathbf{x}^0 = (\mathbf{1} \otimes x^0)$ for any $x^0 \in \mathbb{R}^m$ has the following property:*

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right) + \frac{L^2}{Kn}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq$$

$$+\frac{8}{\eta\alpha K}\tilde{f}(\bar{x}^0) + \frac{3\alpha^2 L^2 T_{\mathrm{o}}\zeta_0}{nK(1-\bar{\lambda})^2}, \tag{5}$$

where $\bar{x}^k = \frac{1}{n}\sum_{i=1}^n x_i^k$, $\overline{\nabla \mathbf{f}}(\mathbf{x}^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i^k)$, $\bar{\lambda} = (1+\lambda)/2$, $\tilde{f}(\bar{x}^0) = f(\bar{x}^0) - f^*$, $\|\mathbf{\Phi}^k\|^2 = \|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + \|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2$, *and* $\zeta_0 = \|\nabla \mathbf{f}(\bar{\mathbf{x}}^0) - \mathbf{1} \otimes \overline{\nabla \mathbf{f}}(\bar{\mathbf{x}}^0)\|^2$.

Note that the left-hand side of (5) has three main components. The first two indicate the asymptotic convergence to a stationary point, while the third term $\|\mathbf{\Phi}^k\|^2$ guarantees asymptotic consensus. If in Theorem 1, we consider a sufficiently well-connected graph where $1 \geq 2\sqrt{\lambda}$ and set $\alpha \propto (1-\lambda)/L$, and $\eta \propto 1/T_{\mathrm{o}}$, then we obtain the convergence rate,

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right) + \frac{L^2}{Kn}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq$$

$$O\left(\frac{T_{\mathrm{o}}\tilde{f}(\bar{x}^0)}{K} + \frac{T_{\mathrm{o}}\zeta_0}{nK}\right).$$

The communication complexity of LU-GT is obtained by dividing the number of iterations K by T_{o} to find the number of communication rounds, i.e., $R = K/T_{\mathrm{o}}$. Theorem 1 implies that LU-GT matches the same communication complexity ($R = O(1/\epsilon)$ for a desired accuracy $\epsilon > 0$) as [23] for distributed (federated) setups. However, LU-GT allows arbitrary symmetric undirected network topologies (Assumptions 1).

III. Convergence Analysis of LU-GT

Here we provide the convergence analysis of LU-GT of which the results are given in Theorem 1. The proof of Lemma 2 and other technical lemmas and rigorous proofs needed to prove Theorem 1 are shown in Appendices included in the version on arXiv. We first state a crucial lemma that characterizes the asymptotic convergence of the consensus error, which is used to prove Theorem 1.

Lemma 2 (Consensus Inequality) *Let Assumptions 1 and 2 hold and*

$$\eta < \min\left\{1, (1-\sqrt{\lambda})/(\sqrt{\lambda}(T_{\mathrm{o}}))\right\}, \tag{6}$$

$$\alpha \leq \min\left\{\frac{1}{2L}, \sqrt{\frac{(1-\lambda)(1-\theta)}{16L^2\lambda}}, \sqrt{\frac{(\bar{\lambda}-\bar{\lambda}^2)(1-\theta)}{8L^2\eta^2 T_{\mathrm{o}}^2}}\right\}, \tag{7}$$

hold. Define $\theta = \lambda(1+\eta T_{\mathrm{o}})^2 < 1$. Then, the output of Algorithm (1) satisfies the following inequality

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq \frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{1-\bar{\lambda}-e_1 T_{\mathrm{o}}}\|\mathbf{\Phi}^0\|^2 \tag{8}$$

$$+\left(\frac{e_2 T_{\mathrm{o}}}{K(1-\bar{\lambda}-e_1 T_{\mathrm{o}})}\right)\sum_{k=0}^{K-1}\left(\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 + \|\nabla f(\bar{x}^k)\|^2\right),$$

where $\|\mathbf{\Phi}^k\|^2 = \|\mathbf{x}^k - \bar{\mathbf{x}}^k\|^2 + \|\mathbf{y}^k - \bar{\mathbf{y}}^k\|^2$, $r_k \triangleq \lfloor k/T_{\mathrm{o}} \rfloor$, $e_1 \triangleq \frac{8L^2\eta^2\alpha^2 T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2}{(1-\theta)}$, *and* $e_2 \triangleq \frac{8nL^2\eta^2\alpha^4 T_{\mathrm{o}}(1+\eta T_{\mathrm{o}})^2}{(1-\theta)}$.

Now we are ready to state the proof of Theorem 1.

Proof: [**Proof of Theorem 1**] Following similar arguments as in [15, Lemma 3] and imposing $\alpha \leq \frac{1}{2L}$, we have the following inequality

$$f(\bar{x}^{k+1}) \leq f(\bar{x}^k) - \frac{\eta\alpha}{2}\|\nabla f(\bar{x}^k)\|^2 - \frac{\eta\alpha}{4}\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2$$

$$+\frac{\eta\alpha L^2}{2n}\|\mathbf{\Phi}^k\|^2.$$

Reorganize and lower bound the left-hand side to find

$$\frac{\eta\alpha}{4}(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2) \leq f(\bar{x}^k) - f(\bar{x}^{k+1})$$

$$+\frac{\eta\alpha L^2\|\mathbf{\Phi}^k\|^2}{2n}.$$

Next, subtract and add f^* and set $\tilde{f}(\bar{x}^k) = f(\bar{x}^k) - f^*$, then

$$\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 \leq \frac{4}{\eta\alpha}(f(\bar{x}^k) - f(\bar{x}^{k+1}))$$

$$+\frac{2L^2}{n}\|\mathbf{\Phi}^k\|^2.$$

Sum both sides from $k = 0, ..., K-1$ and divide by K

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right) \leq$$

$$\frac{4}{\eta\alpha K}\sum_{k=0}^{K-1}(\tilde{f}(\bar{x}^k) - \tilde{f}(\bar{x}^{k+1})) + \frac{2L^2}{nK}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2.$$

Multiplying (8) from Lemma 2 by c, a constant to be defined later, and adding it to the above equation, we then have the following

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right) + c\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq$$

$$\frac{4}{\eta\alpha K}\sum_{k=0}^{K-1}(\tilde{f}(\bar{x}^k) - \tilde{f}(\bar{x}^{k+1})) + \frac{2L^2}{nK}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2$$

$$+c\frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{1-\bar{\lambda}-e_1 T_{\mathrm{o}}}\|\mathbf{\Phi}^0\|^2$$

$$+c\left(\frac{e_2 T_{\mathrm{o}}}{K(1-\bar{\lambda}-e_1 T_{\mathrm{o}})}\right)\sum_{k=0}^{K-1}\left(\|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2 + \|\nabla f(\bar{x}^k)\|^2\right)$$

Rearranging and setting $c = \frac{3L^2}{n}$ we find

$$\left(1 - \frac{3L^2 e_2 T_{\mathrm{o}}}{n(1-\bar{\lambda}-e_1 T_{\mathrm{o}})}\right)\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2 + \|\overline{\nabla \mathbf{f}}(\mathbf{x}^k)\|^2\right)$$

$$+\left(\frac{L^2}{n}\right)\frac{1}{K}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2 \leq \frac{4}{\eta\alpha K}\sum_{k=0}^{K-1}(\tilde{f}(\bar{x}^k)-\tilde{f}(\bar{x}^{k+1}))$$
$$+\,c\frac{(1-\bar{\lambda})(\frac{1}{K}\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{1-\bar{\lambda}-e_1T_{\rm o}}\|\mathbf{\Phi}^0\|^2$$

Require

$$\frac{1}{2}\leq\left(1-\frac{3L^2e_2T_{\rm o}}{n(1-\bar{\lambda}-e_1T_{\rm o})}\right)\Rightarrow\alpha\leq\sqrt[4]{\frac{(1-\bar{\lambda})^2(1-\theta)}{48L^4\eta^2T_{\rm o}^2}}. \tag{9}$$

Then, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(\|\nabla f(\bar{x}^k)\|^2+\|\overline{\nabla\mathbf{f}}(\mathbf{x}^k)\|^2\right)+\frac{L^2}{Kn}\sum_{k=0}^{K-1}\|\mathbf{\Phi}^k\|^2\leq$$
$$+\frac{8}{\eta\alpha K}\tilde{f}(\bar{x}^0)+\frac{6L^2(1-\bar{\lambda})(\sum_{k=0}^{K-1}\bar{\lambda}^{r_k})}{nK(1-\bar{\lambda}-e_1T_{\rm o})}\|\mathbf{\Phi}^0\|^2.$$

Assume that the initialization for $x_1, x_2, ..., x_n$ is identical. Then $\mathbf{x}^0 = \mathbf{1}\otimes x^0$ (for some $x^0\in\mathbb{R}^d$). As a result, $\mathbf{x}^0=\bar{\mathbf{x}}^0$ meaning $\|\hat{\mathbf{Q}}^T\mathbf{x}^0\|^2=0$. Then,

$$\|\mathbf{\Phi}^0\|^2=\|\hat{\mathbf{Q}}^T\mathbf{y}^0\|^2=\left\|\alpha\hat{\mathbf{Q}}^T\nabla\mathbf{f}(\mathbf{x}^0)\right\|^2$$
$$=\alpha^2\|\nabla\mathbf{f}(\bar{\mathbf{x}}^0)-\mathbf{1}\otimes\overline{\nabla\mathbf{f}}(\bar{\mathbf{x}}^0)\|^2.$$

Define $\zeta_0=\|\nabla\mathbf{f}(\bar{\mathbf{x}}^0)-\mathbf{1}\otimes\overline{\nabla\mathbf{f}}(\bar{\mathbf{x}}^0)\|^2$. We also upper bound $\sum_{k=0}^{K-1}\bar{\lambda}^{r_k}$ with $T_{\rm o}/(1-\bar{\lambda})$, a repeating geometric sequence and impose the following condition on alpha

$$\frac{1-\bar{\lambda}-e_1T_{\rm o}}{1-\bar{\lambda}}\geq 1-\bar{\lambda}\Rightarrow\alpha\leq\sqrt{\frac{(\bar{\lambda}-\bar{\lambda}^2)(1-\theta)}{8L^2\eta^2T_{\rm o}^2}}.$$

Thus, the desired relation follows. ■

IV. Numerical Results

We simulate the performance of Algorithm 1 for the following least squares problem with a non-convex regularization term:

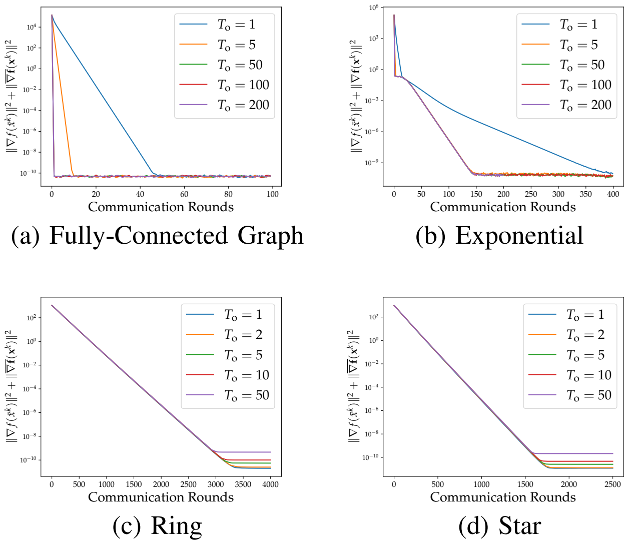
$$\min_x \frac{1}{n}\sum_{i=1}^{n}\|A_ix-b_i\|^2+\rho\sum_{j=1}^{m}\frac{x(j)^2}{1+x(j)^2}, \tag{10}$$

where $\{A_i, b_i\}$ is the local data held by agent i and $x(j)$ is the $j-th$ component of the parameter x. We consider two cases: 1) close to homogeneous, where local stationary points are different but sufficiently close; 2) heterogeneous, where no assumptions are made on the similarity of local stationary points. We generate $A_i\in\mathbb{R}^{p\times m}$ where $p = 500, m = 20$ with values drawn from $\mathcal{N}(0,1)$, a parameter vector $x_i^*\in\mathbb{R}^m$ with values drawn from $\mathcal{N}(0,1)$, and $b_i\in\mathbb{R}^p = A_ix_0^i+\gamma\times z_i$ where $z_i\in\mathbb{R}^p$ is drawn from $\mathcal{N}(0,1)$. This is a heterogeneous case. The difference for the close to homogeneous is that we draw A_i once such that $A_i = A_j, \forall i,j$. For the close to homogeneous case, we examine exponential and fully-connected graphs, while for the heterogeneous case, we examine star and ring graphs, all with 16 nodes. We set $\rho = 0.01, \gamma = 150$.

Table I lists the manually optimized $\eta\alpha$ for each graph and $T_{\rm o}$ combination. Our simulation results in Figure 1 reveal that for (sufficiently well-connected) graphs, LU-GT reduces communication costs up to a certain $T_{\rm o}$. In addition, for the exponential graph, the benefits saturate much faster. For sparse networks, the hyperparameter tuning of $\eta\alpha$ matches the suggested inversely proportional relation with $T_{\rm o}$ predicted by the theory. In this scenario, communication costs are equivalent to no local updates, matching the analysis.

TABLE I: Manually optimized $\eta\alpha$ used for each graph and $T_{\rm o}$ combination.

	$T_o = 1$	$T_o = 5$	$T_o = 50$	$T_o = 100$	$T_o = 200$
Complete	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}
Exponential	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}
	$T_o = 1$	$T_o = 2$	$T_o = 5$	$T_o = 10$	$T_o = 50$
Ring	2×10^{-5}	1×10^{-5}	0.4×10^{-5}	$.2\times10^{-5}$	0.04×10^{-5}
Star	$.4\times10^{-4}$	$.2\times10^{-4}$	$.08\times10^{-4}$	$.04\times10^{-4}$	$.008\times10^{-4}$



(a) Fully-Connected Graph

(b) Exponential

(c) Ring

(d) Star

Fig. 1: Performance of LU-GT to solve (10) with varying $T_{\rm o}$, $\alpha\eta$, and topologies.

V. Conclusions

We propose the algorithm LU-GT that incorporates local recursions into Gradient Tracking. Our analysis shows that LU-GT matches the same communication complexity as the Federated Learning setting but allows arbitrary network topologies. In addition, regardless of the number of local recursions, LU-GT incurs no additional bias term in the rate. We show reduced communication complexity in simulation for well-connected graphs. However, further refinement of the analysis is necessary to quantify the precise effect of local recursions on Gradient Tracking. It is still unclear under what regimes local updates reduce the communication cost and what the upper bound is on these local updates. Numerical results suggest that local updates might not benefit sparsely connected networks. Such explicit relations between network topologies and local updates are left for future work. While we focus on the non-convex setting in this work due to space constraints, we can extend our work to the convex setting.

References

[1] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[2] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process*, vol. 58, no. 3, p. 1035, 2010.

[3] 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.

[4] 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, 2019.

[5] 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, Sept. 2019.

[6] 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.

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

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

[9] 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.

[10] 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, Sept. 2018.

[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] J. Chen and A. H. Sayed, "Distributed pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, April 2013.

[13] 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.

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

[15] 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, June 2022.

[16] A. Koloskova, T. Lin, and S. U. Stich, "An improved analysis of gradient tracking for decentralized machine learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 422–11 435, 2021.

[17] B. Ying, K. Yuan, H. Hu, Y. Chen, and W. Yin, "Bluefog: Make decentralized algorithms practical for optimization and deep learning," 2021.

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

[19] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local GD on heterogeneous data," *CoRR*, vol. abs/1909.04715, 2019.

[20] A. Khaled, K. Mishchenko, and P. Richtarik, "Tighter theory for local SGD on identical and heterogeneous data," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 108. PMLR, 26–28 Aug 2020, pp. 4519–4529.

[21] J. Zhang, C. De Sa, I. Mitliagkas, and C. Ré, "Parallel SGD: When does averaging help?" *arXiv preprint arXiv:1606.07365*, 06 2016.

[22] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local SGD," in *International Conference on Learning Representations*, 2020.

[23] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5132–5143.

[24] E. Gorbunov, F. Hanzely, and P. Richtarik, "Local SGD: Unified theory and new efficient methods," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, 13–15 Apr 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*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

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

[27] 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*. PMLR, 2020, pp. 4519–4529.

[28] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Mathematical Programming*, vol. 176, no. 1-2, pp. 497–544, 2019.

[29] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM Journal on Optimization*, vol. 32, no. 2, pp. 354–385, 2022.

[30] F. Saadatniaki, R. Xin, and U. A. Khan, "Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4769–4780, 2020.

[31] S. Lu and C. W. Wu, "Decentralized stochastic non-convex optimization over weakly connected time-varying digraphs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5770–5774.

[32] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.