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

A distributed stochastic first-order method for strongly concave-convex saddle point problems

Muhammad I. Qureshi and Usman A. Khan
Tufts University, Medford, MA, USA

***Abstract*— In this paper, we propose a distributed stochastic first-order method for saddle point problems over strongly connected graphs. Existing methods generally suffer from a steady-state error that arises due to the heterogeneous nature of data distribution (captured by the local versus global cost gaps) and the variance of the stochastic gradients. We propose `GT-SGDA`, a distributed stochastic gradient descent ascent method that uses network-level *gradient tracking* to eliminate the steady-state error component due to the local versus global cost gap. We show that `GT-SGDA` converges linearly to an error ball around the unique saddle point for sufficiently small constant step-sizes when the global cost is strongly concave-convex (a necessary condition for the existence of a unique saddle point). Moreover, we show that the size of this error ball depends on the variance of the stochastic gradients. We provide numerical experiments to illustrate the convergence properties of `GT-SGDA` for different applications and highlight the significance of gradient tracking. We also show the performance of `GT-SGDA` for training modern applications like distributed generative adversarial networks (GANs).**

***Index Terms*— Stochastic min-max optimization, first-order methods, saddle point problems, distributed algorithms**

I. Introduction and Related Work

Saddle point problems arise in many applications of control systems, signal processing, machine learning, and statistics [1]–[6]. Such problems (also known as min-max) are significant in the literature on optimization theory when the problem of interest lies in finding the point of inflection for a cost function F. Assuming that $F : \mathbb{R}^{p_x \times p_y} \rightarrow \mathbb{R}$ is convex in $\mathbf{x} \in \mathbb{R}^{p_x}$ and concave in $\mathbf{y} \in \mathbb{R}^{p_y}$, the saddle point $(\mathbf{x}^*, \mathbf{y}^*) \in \mathbb{R}^{p_x \times p_y}$ can be mathematically described as the point such that $\forall (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{p_x \times p_y}$,

$$F(\mathbf{x}^*, \mathbf{y}) \leq F(\mathbf{x}^*, \mathbf{y}^*) \leq F(\mathbf{x}, \mathbf{y}^*).$$

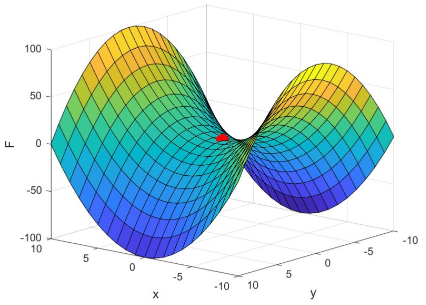

Fig. 1. Surface plot of two dimensional saddle point problem for $F(\mathbf{x}, \mathbf{y}) = \mathbf{x}^2 - \mathbf{y}^2$ with saddle point $(0, 0)$ highlighted in red.

Figure 1 shows the surface plot of a two-dimensional saddle point problem where $F(\mathbf{x}, \mathbf{y}) = \mathbf{x}^2 - \mathbf{y}^2$. The unique saddle point $(\mathbf{x}^*, \mathbf{y}^*) = (0, 0)$ is highlighted in red. To reach the saddle point, we minimize F with respect to $\mathbf{x}$ and maximize F with respect to $\mathbf{y}$, mathematically written as:

$$\min_{\mathbf{x} \in \mathbb{R}^{p_x}} \max_{\mathbf{y} \in \mathbb{R}^{p_y}} F(\mathbf{x}, \mathbf{y}).$$

A well-known technique to solve such problems using gradient-based approaches is the gradient descent ascent method [4]–[8]. Several variants of gradient descent ascent algorithms are studied extensively due to their applications in robust regression, image reconstruction, and generative adversarial networks (GANs) [2], [3], [9], [10].

Centralized methods, although effective, are not practically feasible in many large-scale applications when the data is divided among a network of geographically distributed nodes. In such distributed data scenarios, existing work has mainly focused on minimization methods [11]–[18]. Early work includes [11], [13], which requires the knowledge of a first-order oracle but the performance is compromised due to the inability to handle the dissimilarity between local and global cost functions. This gap leads to a steady-state error with constant step-size or sub-linear convergence rate to reach the optimal solution with decaying step-size. Gradient-tracking was proposed in [16], [19] to eliminate this error and establish linear convergence; see [18] for a detailed overview.

Towards distributed gradient descent ascent methods, of significance are [20], [21], which solve the saddle point problems in a deterministic fashion. These methods require every node to use all local data at each iteration to compute full-batch partial gradients for the $\mathbf{x}$ and $\mathbf{y}$ updates. Additionally, [20] assumes *similarity* conditions to tackle local versus global cost dissimilarity while [21] uses *gradient tracking* to address this gap. Several stochastic methods have also been proposed [22]–[25] but they use strong assumptions, i.e., [22] and [24] assume that the local costs are quadratic. Moreover, [25] assumes a federated setup and an upper bound to quantify the dissimilarity of local and global cost functions. Some related work can also be found in [26], [27] which solves for distributed constrained optimization problem by converting it to a distributed saddle point problem using Lagrangian multipliers.

In this paper, we propose **`GT-SGDA`**, a distributed optimization method to solve saddle point problems using a stochastic first-order oracle and gradient tracking. Unlike

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

existing methods, **GT-SGDA** is stochastic and is applicable to a wider class of problems under milder conditions. The main contributions are: (i) **GT-SGDA** addresses the data heterogeneity with the help of gradient tracking for both descent and ascent updates; (ii) For constant step-sizes, we show linear convergence of **GT-SGDA** to an error ball around the unique saddle point when the functions are strongly concave-convex; (iii) We show that the error ball is proportional to the variance of noisy partial gradients.

We now describe the rest of the paper. Section II discusses the problem formulation and Section III provides some useful applications to motivate distributed saddle point optimization. Section IV provides the algorithm development, while Section V provides the main results and the detailed convergence analysis. Section VI shows some numerical experiments to illustrate the performance of **GT-SGDA** with the help of distributed regression problems and distributed GANs. Finally, Section VII concludes the paper.

Basic notation: We use uppercase letters to denote matrices and lowercase bold letters to denote vectors. We define I_n as the $n \times n$ identity matrix and $\mathbf{0}_n$ as a column vector of n zeros. For a matrix $W \in \mathbb{R}^{n \times n}$, we denote $\rho(W)$ as its spectral radius. We define $\|\cdot\|$ as the vector two-norm and $|||\cdot|||$ as the matrix norm induced by this vector norm. For a function $F(\mathbf{x}, \mathbf{y})$, we denote $\nabla_x F$ as the partial derivative of F with respect to $\mathbf{x}$ and $\nabla_y F$ as the partial derivative of F with respect to $\mathbf{y}$. We also define $\mathbb{E}\left[F(\mathbf{x}, \mathbf{y}) | \mathbf{x}, \mathbf{y}\right]$ as the expected value of $F(\mathbf{x}, \mathbf{y})$ given the values of $\mathbf{x}$ and $\mathbf{y}$.

II. Problem Formulation

In this paper, we would like to solve a distributed min-max problem and find the unique saddle point when the strongly concave-convex local cost functions f_i's are geographically distributed over a network of n nodes. Each node communicates over a strongly connected network. The global cost function $F(\mathbf{x}, \mathbf{y})$ is the average of local cost functions $f_i(\mathbf{x}, \mathbf{y})$ where $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{p_x \times p_y}$. Mathematically, we have the following problem:

$$\mathbf{P}: \min_{\mathbf{x} \in \mathbb{R}^{p_x}} \max_{\mathbf{y} \in \mathbb{R}^{p_y}} F(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x} \in \mathbb{R}^{p_x}} \max_{\mathbf{y} \in \mathbb{R}^{p_y}} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}, \mathbf{y}),$$

where the local cost functions, at each node i, are defined as: $f_i(\mathbf{x}, \mathbf{y}) := g_i(\mathbf{x}) + \langle \mathbf{y}, P_i \mathbf{x} \rangle - h_i(\mathbf{y})$. Consequently, the global $F(\mathbf{x}, \mathbf{y}) := G(\mathbf{x}) + \langle \mathbf{y}, \overline{P} \mathbf{x} \rangle - H(\mathbf{y})$ such that $G(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} g_i(\mathbf{x})$, $H(\mathbf{y}) := \frac{1}{n} \sum_{i=1}^{n} h_i(\mathbf{y})$, and $\overline{P} = \frac{1}{n} \sum_{i=1}^{n} P_i \in \mathbb{R}^{p_y \times p_x}$. The convergence analysis of **GT-SGDA** will be derived in Section V under the following assumptions.

Assumption 1. *The nodes communicate over a weight-balanced, strongly connected graph.*

Assumption 2. *The coupling matrices $P_i, \forall i$ are full column rank, global G is convex, and global H is μ-strongly convex. Moreover, $\forall i, g_i$ is L_1-smooth and h_i is L_2-smooth.*

Assumption 3. *Each node i has access to the stochastic first-order oracle that returns $\nabla_{\mathbf{x}} \widehat{f}_i(\mathbf{x}_i^k, \mathbf{y}_i^k)$ and $\nabla_{\mathbf{y}} \widehat{f}_i(\mathbf{x}_i^k, \mathbf{y}_i^k)$, when queried by $(\mathbf{x}_k^i, \mathbf{y}_k^i)$, such that*

$$\mathbb{E}\left[\nabla_{\mathbf{x}} \widehat{f}_i(\mathbf{x}_i^k, \mathbf{y}_i^k) | \mathbf{x}_i^k, \mathbf{y}_i^k\right] = \nabla_{\mathbf{x}} f_i(\mathbf{x}_i^k, \mathbf{y}_i^k),$$
$$\mathbb{E}\left[\nabla_{\mathbf{y}} \widehat{f}_i(\mathbf{x}_i^k, \mathbf{y}_i^k) | \mathbf{x}_i^k, \mathbf{y}_i^k\right] = \nabla_{\mathbf{y}} f_i(\mathbf{x}_i^k, \mathbf{y}_i^k),$$
$$\mathbb{E}\left[\|\nabla_{\mathbf{x}} \widehat{f}_i(\mathbf{x}_i^k, \mathbf{y}_i^k) - \nabla_{\mathbf{x}} f_i(\mathbf{x}_i^k, \mathbf{y}_i^k)\|^2 | \mathbf{x}_i^k, \mathbf{y}_i^k\right] \leq \sigma_x^2,$$
$$\mathbb{E}\left[\|\nabla_{\mathbf{y}} \widehat{f}_i(\mathbf{x}_i^k, \mathbf{y}_i^k) - \nabla_{\mathbf{y}} f_i(\mathbf{x}_i^k, \mathbf{y}_i^k)\|^2 | \mathbf{x}_i^k, \mathbf{y}_i^k\right] \leq \sigma_y^2.$$

Assumption **1** is commonly used in the literature on distributed optimization and guarantees that the corresponding weight matrix $W := \{w_{i,r}\}$ is primitive and doubly stochastic. For such a W, we have that $\lambda := \rho(W - W^\infty) < 1$ from Perron Frobenius theorem [28], where $W^\infty := \lim_{k \to \infty} W^k$. Assumption **2** ensures that the global cost F is strongly concave in $\mathbf{y}$ and convex in $\mathbf{x}$. We note that this is the minimum requirement to ensure that F has a unique saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ [8]. Assumption **3** is typical in the literature on stochastic methods and requires that the sampled gradients have finite second moments. Here, we define $\sigma^2 := \max\{\sigma_x^2, \sigma_y^2\}$ and $L := \max\{L_1, L_2\}$ for the simplicity of analysis.

III. Motivation

In order to motivate **GT-SGDA**, we now discuss some useful applications that take the form of saddle point problems.

A. Constrained optimization

Most optimization problems require the minimization of a cost function given some realistic constraints. For example, in noise cancellation systems, we would like to find feedback filter coefficients that are *stable*. To ensure this, we can constrain the eigenvalues of the resulting closed-loop system. More generally, for equality constraints, we can formulate the problem as:

$$\min_{\mathbf{x}} g(\mathbf{x}), \qquad \text{such that} \qquad P\mathbf{x} = \mathbf{b};$$

where $g: \mathbb{R}^{p_x} \to \mathbb{R}$ is the cost function to be minimized and $P\mathbf{x} = \mathbf{b}$ are the constraints when $P \in \mathbb{R}^{p_y \times p_x}$ and $\mathbf{b} \in \mathbb{R}^{p_y}$. The saddle point equivalent form can be written using Lagrangian multipliers $\mathbf{y} \in \mathbb{R}^{p_y}$:

$$\min_{\mathbf{x}} \min_{\mathbf{y}} \left\{\mathcal{L}(\mathbf{x}, \mathbf{y}) := g(\mathbf{x}) + \langle \mathbf{y}, P\mathbf{x} \rangle - \langle \mathbf{y}, \mathbf{b} \rangle\right\}.$$

For very large-scale problems, the data is often geographically distributed and cannot be accessed at any single node. Hence, for a network of n nodes communicating over a strongly connected graph, we would like to solve for

$$\min_{\mathbf{x}} \min_{\mathbf{y}} \frac{1}{n} \sum_{i=1}^{n} \left\{\mathcal{L}_i(\mathbf{x}, \mathbf{y}) := g_i(\mathbf{x}) + \langle \mathbf{y}, P_i \mathbf{x} \rangle - \langle \mathbf{y}, \mathbf{b}_i \rangle\right\},$$

where each node i has $\mathcal{L}_i(\mathbf{x}, \mathbf{y})$ as the local cost function, provided that $\forall i$, $g_i: \mathbb{R}^{p_x} \to \mathbb{R}$, $P_i \in \mathbb{R}^{p_y \times p_x}$, $\mathbf{b}_i \in \mathbb{R}^{p_y}$, and the global cost is the average of the local cost functions.

B. Supervised learning

The methods used for supervised learning aim to train a predictor (using some data P) that minimizes the loss function $h(P\mathbf{x})$. To avoid overfitting, a regularizer term is often added $g(\mathbf{x})$. Thus, mathematically we can formulate the problem as: $\min_{\mathbf{x}}\left\{h(P\mathbf{x})+g(\mathbf{x})\right\}$. The saddle point equivalent form can be written as:

$$\min_{\mathbf{x}}\max_{\mathbf{y}}\left\{f(\mathbf{x},\mathbf{y}):=g(\mathbf{x})+\langle\mathbf{y},P\mathbf{x}\rangle-h(\mathbf{y})\right\}.$$

A large amount of data is usually required to train the models for best performance, which might not be possible due to computation, communication, or privacy constraints. Thus, the distributed implementation is useful, i.e., solving for

$$\min_{\mathbf{x}}\min_{\mathbf{y}}\frac{1}{n}\sum_{i=1}^{n}\left\{f_i(\mathbf{x},\mathbf{y}):=g_i(\mathbf{x})+\langle\mathbf{y},P_i\mathbf{x}\rangle-h_i(\mathbf{y})\right\}.$$

C. Game theory

The essence of the game theory lies in understanding the interaction between different groups. These groups conflict with each other until they reach an equilibrium point. The intuitive formulation for such problems is min-max optimization $\min_{\mathbf{x}}\max_{\mathbf{y}}F(\mathbf{x},\mathbf{y})$ and the point of equilibrium is essentially the saddle point $(\mathbf{x}^*,\mathbf{y}^*)$ of the problem,

$$\min_{\mathbf{x}}\max_{\mathbf{y}}F(\mathbf{x},\mathbf{y})=F(\mathbf{x}^*,\mathbf{y}^*).$$

Game theory traditionally has applications in economics and statistics but recently it has gained a lot of interest due to its applications in training generative adversarial networks (GANs). A GAN is composed of a generator $G(\mathbf{x})$ and a discriminator $D(\mathbf{y})$ where the generator tries to generate high-quality data samples and the discriminator tries to discriminate between actual and fake (generated) data. Mathematically, we can write:

$$\min_{G}\max_{D}\left\{F(G,D):=\log(D(\mathbf{x}))+\log(1-D(G(\mathbf{y})))\right\}.$$

We often compute the above problem in a stochastic manner such that $\mathbf{x}$ is sampled from the actual data given a probability $p_{data}(\mathbf{x})$ and $\mathbf{y}$ is sampled from a random probability distribution $p_{rand}(\mathbf{y})$. The generator then generates fake data $G(\mathbf{y})$. The goal is to learn a mapping from the random probability distribution to the data distribution $G(\cdot)$ such that the generator produces good (fake) data samples that are hard to distinguish from the real data by the discriminator. The generators and discriminators are often selected to be large neural networks. Although they are non-convex and non-concave, we show the performance of **GT-SGDA** for training distributed GANs in Section VI.

IV. Algorithm development

We first recap a well studied distributed method (**DGD** [11]) to find the unique minimizer $\mathbf{x}^*\in\mathbb{R}^{p_x}$ of a smooth and strongly convex global cost $G:=\sum_{i=1}^{n}g_i(\mathbf{x})$. For a positive step-size α, **DGD**, at each node i computes

$$\mathbf{x}_i^{k+1}=\sum_{r=1}^{n}w_{i,r}(\mathbf{x}_r^k-\alpha\nabla g_r^k),\qquad k\geq 0,$$

where $\mathbf{x}_i^k$ is node i's estimate of $\mathbf{x}^*$. We note that each $\mathbf{x}_i^k$ converges to a sub-optimal solution because $\nabla g_i\neq\nabla G$. The corresponding steady-state error can be eliminated with the help of gradient tracking by replacing ∇g_i^k with $\mathbf{q}_i^k$ such that

$$\mathbf{q}_i^{k+1}:=\sum_{r=1}^{n}w_{ir}(\mathbf{q}_r^k+\nabla g_r^{k+1}-\nabla g_r^k).$$

It can be shown that $\mathbf{q}_i^k\to\nabla G$ [16]. The gradient descent ascent version of [16] was recently proposed in [21], which uses gradient tracking. However, [21] works in a deterministic fashion. To eliminate this limitation, we propose a stochastic method when each node can only access the stochastic first-order oracle (see Assumption **3**).

```
Algorithm 1  GT-SGDA at each node $i$
Require: $\mathbf{x}_i^0\in\mathbb{R}^{p_x},\mathbf{y}_i^0\in\mathbb{R}^{p_y},P_i^0=P_i,\{w_{ir}\}_{r=1}^n,\alpha>0,$
         $\beta>0,\mathbf{q}_i^0=\nabla_x f_i(\mathbf{x}_i^0,\mathbf{y}_i^0),\mathbf{r}_i^0=\nabla_y f_i(\mathbf{x}_i^0,\mathbf{y}_i^0)$
for $k=0,1,2,\ldots,$ do,
    $P_i^{k+1}\leftarrow\sum_{r=1}^n w_{ir}P_r^k$
    $\mathbf{x}_i^{k+1}\leftarrow\sum_{r=1}^n w_{ir}(\mathbf{x}_r^k-\alpha\cdot\mathbf{q}_r^k)$
    $\mathbf{q}_i^{k+1}\leftarrow\sum_{r=1}^n w_{ir}(\mathbf{q}_r^k+\nabla_x\widehat{f}_r^{k+1}-\nabla_x\widehat{f}_r^k)$
    $\mathbf{y}_i^{k+1}\leftarrow\sum_{r=1}^n w_{ir}(\mathbf{y}_r^k+\beta\cdot\mathbf{r}_r^k)$
    $\mathbf{r}_i^{k+1}\leftarrow\sum_{r=1}^n w_{ir}(\mathbf{r}_r^k+\nabla_y\widehat{f}_r^{k+1}-\nabla_y\widehat{f}_r^k)$
end for
```

GT-SGDA is formally described in Algorithm 1. We note that we would like to find the saddle point of $F(\mathbf{x},\mathbf{y})$. The algorithm can be described in three main steps: (i) estimation of the global matrix $\overline{P}$; (ii) stochastic gradient descent for $\mathbf{x}_i^k$ updates and the corresponding gradient tracking $\mathbf{q}_i^k$; (iii) stochastic gradient ascent for $\mathbf{y}_i^k$ updates and the corresponding gradient tracking $\mathbf{r}_i^k$. At each node i, **GT-SGDA** randomly chooses the initial state vectors and some positive step-sizes α and β. At each iteration k, every node updates its local P_i^k to estimate the global $P_i^k\to\overline{P}$. We note that $\mathbf{q}_i^k$ and $\mathbf{r}_i^k$ are the partial gradient tracking terms such that $\mathbf{q}_i^k\to\frac{1}{n}\sum_i\nabla_x\widehat{f}_i$ and $\mathbf{r}_i^k\to\frac{1}{n}\sum_i\nabla_y\widehat{f}_i$. Moreover, the state variables $\mathbf{x}_i^k$ and $\mathbf{y}_i^k$ are evaluated by taking the steps in negative and positive directions of the partial gradient tracking terms $\mathbf{q}_i^k$ and $\mathbf{r}_i^k$, respectively. Next, we provide the main theorem describing the convergence properties of **GT-SGDA**.

V. Main results and convergence analysis

In this section, we discuss the main results and convergence properties of **GT-SGDA**.

Theorem 1. *Consider Problem* $\mathbf{P}$ *under Assumptions **1**, **2**, and **3**. **GT-SGDA** converges linearly to an error ball around the unique saddle point* $(\mathbf{x}^*,\mathbf{y}^*)$ *of* F*, the size of which depends on the variance of the stochastic partial gradients.*

To aid the convergence analysis, we first define four global state vectors $\mathbf{x}^k,\mathbf{q}^k\in\mathbb{R}^{np_x}$, $\mathbf{y}^k,\mathbf{r}^k\in\mathbb{R}^{np_y}$ that concatenate the local state vectors $\mathbf{x}_i^k,\mathbf{q}_i^k,\mathbf{y}_i^k$, and $\mathbf{r}_i^k$ for all i at each iteration k. We next describe some useful error terms that govern the dynamics of **GT-SGDA**: (i) Network agreement errors $\mathbb{E}\|\mathbf{x}^k - W_1^\infty \mathbf{x}^k\|^2$ and $\mathbb{E}\|\mathbf{y}^k - W_2^\infty \mathbf{y}^k\|^2$, where $W_1^\infty := W^\infty \otimes I_{p_x}$ and $W_2^\infty := W^\infty \otimes I_{p_y}$. These quantify how far the network is from the agreement; (ii) Optimality gaps $\mathbb{E}\|\overline{\mathbf{x}}^k - \mathbf{x}^*\|^2$ and $\mathbb{E}\|\overline{\mathbf{y}}^k - \nabla H^*(\overline{P}\overline{\mathbf{x}}^k)\|^2$, where $\overline{\mathbf{x}}^k := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^k$, $\overline{\mathbf{y}}^k := \frac{1}{n}\sum_{i=1}^n \mathbf{y}_i^k$, and H^* is the conjugate function of H. These evaluate how far the agreement is from the optimal solution; (iii) Gradient tracking errors $\mathbb{E}\|\mathbf{q}^k - W_1^\infty \mathbf{q}^k\|^2$ and $\mathbb{E}\|\mathbf{r}^k - W_2^\infty \mathbf{r}^k\|^2$. These measure the difference between global and local partial gradients. To establish the convergence results, we next provide a useful lemma that characterizes the time evolution of **GT-SGDA** in terms of the above-mentioned error terms.

Lemma 1. *Consider **GT-SGDA** described in Algorithm 1 under Assumptions **1**, **2** and **3**. We define $\mathbf{u}^k, \mathbf{s}^k, \mathbf{c} \in \mathbb{R}^6$ as*

$$\mathbf{u}^k := \begin{bmatrix} \mathbb{E}\|\mathbf{x}^k - W_1^\infty \mathbf{x}^k\|^2 \\ n\mathbb{E}\|\overline{\mathbf{x}}^k - \mathbf{x}^*\|^2 \\ L^{-2}\mathbb{E}\|\mathbf{q}^k - W_1^\infty \mathbf{q}^k\|^2 \\ \mathbb{E}\|\mathbf{y}^k - W_2^\infty \mathbf{y}^k\|^2 \\ n\mathbb{E}\|\overline{\mathbf{y}}^k - \nabla H^*(\overline{P}\overline{\mathbf{x}}^k)\|^2 \\ L^{-2}\mathbb{E}\|\mathbf{r}^k - W_2^\infty \mathbf{r}^k\|^2 \end{bmatrix}, \quad \mathbf{s}^k := \begin{bmatrix} \mathbb{E}\|\mathbf{x}^k\|^2 \\ \mathbb{E}\|\mathbf{y}^k\|^2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{c} := \begin{bmatrix} 0 \\ 2\alpha \frac{L_2}{\sigma_m^2} \\ \frac{16L^{-2}}{1-\lambda^2}\left[n + 6(\alpha^2 L_1^2 + \beta^2 \sigma_M^2)\right] \\ 0 \\ \frac{3\beta}{\mu} + \frac{9\alpha^2\sigma_M^2}{\beta\mu^3} \\ \frac{16L^{-2}}{1-\lambda^2}\left[n + 6(L_2^2\beta^2 + \sigma_M^2\alpha^2)\right] \end{bmatrix};$$

and let $N_{\alpha,\beta} \in \mathbb{R}^{6\times 6}$ be such that it has $\alpha\frac{8L_2}{\sigma_m^2}\tau$ and $\beta\frac{6}{\mu}\tau$ at the $(2,2)$ and $(5,1)$ locations, respectively, and zeros everywhere else. We define $\sigma_m \leq |||\, P_i \,||| \leq \sigma_M, \forall i$, for some positive σ_m, σ_M, and $\tau := \frac{1}{n}\sum_{i=1}^n |||\, P_i - \overline{P} \,|||^2$. Then, for all $k \geq 0$, $\alpha, \beta > 0$, $\alpha \leq \frac{\beta\mu^2}{6\Gamma\sigma_M^2}$, and $\Gamma > 2$, we have

$$\mathbf{u}^{k+1} \leq (M_0 + \beta M)\,\mathbf{u}^k + N_{\alpha,\beta}\mathbf{s}^k\lambda^{2k} + \mathbf{c}\sigma^2. \tag{1}$$

The matrix $M_0 \in \mathbb{R}^{6\times 6}$ takes the form:

$$M_0 := \begin{bmatrix} \frac{1+\lambda^2}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \times & 0 & \frac{1+\lambda^2}{2} & \times & 0 & 0 \\ 0 & 0 & 0 & \frac{1+\lambda^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \times & 0 & 0 & \times & 0 & \frac{1+\lambda^2}{2} \end{bmatrix};$$

where '$\times$' are the "don't care" terms (which are not necessary for further analysis) and $M \in \mathbb{R}^{6\times 6}$ takes the form:

$$M := \begin{bmatrix} 0 & 0 & \times & 0 & 0 & 0 \\ \times & -\frac{\sigma_m^2}{L_2\gamma} & 0 & 0 & \frac{4\mu^2 L_2}{3\Gamma\sigma_m^2} & 0 \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times \\ \times & \left(\frac{\mu L_1^2}{2\sigma_M^2\Gamma^2} + \frac{\sigma_M^2}{\mu\Gamma^2}\right) & 0 & \times & -\left(\mu - \frac{\mu}{\Gamma}\right) & 0 \\ \times & \times & \times & \times & \times & \times \end{bmatrix};$$

where γ is a positive constant such that $0 < \beta/\gamma \leq \alpha$.

In order to prove Theorem 1, we use Lemma 1 and the following result on matrix perturbation.

Lemma 2. *[29] Let an $n\times n$ matrix M_β of the form $M_0 + \beta M$ depends smoothly on a real parameter $\beta \geq 0$. Assume M_0 has $l < n$ equal eigenvalues, $\lambda_1 = \cdots = \lambda_l$, associated with right and left eigenvectors $\mathbf{b}_1, \cdots, \mathbf{b}_l$ and $\mathbf{a}_1, \cdots, \mathbf{a}_l$ such that*

$$\begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_l \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_l \end{bmatrix} = I_l.$$

Let $\lambda_i(\beta)$ denote the eigenvalues of M_β, as a function of β, corresponding to $\lambda_i, i \in \{1, \ldots, l\}$, and $M = dM_\beta/d\beta|_{\beta=0}$. Then, $d\lambda_i/d\beta|_{\beta=0}$ is the i-th eigenvalue of the following $l \times l$ matrix,

$$S := \begin{bmatrix} \mathbf{a}_1^\top M \mathbf{b}_1 & \cdots & \mathbf{a}_1^\top M \mathbf{b}_l \\ \vdots & \ddots & \vdots \\ \mathbf{a}_l^\top M \mathbf{b}_1 & \cdots & \mathbf{a}_l^\top M \mathbf{b}_l \end{bmatrix}.$$

Proof of Theorem 1: Equation (1) in Lemma 1 describes the LTI system that governs the error dynamics of **GT-SGDA**. To ensure linear convergence rate, we show that $\mathbf{u}^k$ linearly decays to a ball around $\mathbf{0}_6$ which is controlled by the variance σ^2. To this aim, we note that the second term in (1) decays exponentially, i.e., $N_{\alpha,\beta}\mathbf{s}^k\lambda^{2k} \to \mathbf{0}_6$. Next, we show that the spectral radius of $M_\beta := M_0 + \beta M$ is less than 1, for small enough step-size β, where $\rho(M_0) = 1$ and is governed by the two semi-simple eigenvalues of M_0. We use matrix perturbation analysis [29] of semi-simple eigenvalues to show that for a small increase in $\beta > 0$, we have that $\rho(M_0 + \beta M) < 1$.

With the help of Lemma 2, we show the perturbation effect on the semi-simple eigenvalues of M_β with a change in β. Denote $\mathbf{v}_1^\top = [0\ 1\ 0\ 0\ 0\ 0]$ and $\mathbf{v}_2^\top = [0\ 0\ 0\ 0\ 1\ 0]$, it can be verified that the right and left eigenvectors, corresponding to the two semi-simple eigenvalues of M_0, are the columns of $V^\top := [\mathbf{v}_1\ \mathbf{v}_2]$ and the rows of V, respectively. Using Lemma 2, we know that only four elements of M are significant to steer the semi-simple eigenvalues. Due to space limitation, we only state these four elements in a matrix $S := VMV^\top$ defined below:

$$S = \begin{bmatrix} -\frac{\sigma_m^2}{L_2\gamma} & \frac{4\mu^2 L_2}{3\Gamma\sigma_m^2} \\ \frac{\mu}{2\sigma_M^2\Gamma^2}\left(L_1^2 + \frac{2\sigma_M^4}{\mu^2}\right) & -\mu\left(1 - \frac{1}{\Gamma}\right) \end{bmatrix}.$$

The two semi-simple eigenvalues decrease as β increases (and move inside the unit circle) if S is negative definite [21]. To show this, we note that the trace of S and thus the sum of eigenvalues is negative since all constants in S are nonnegative, implying that at least one eigenvalue is negative. To ensure that S is negative definite, it is sufficient to show that the determinant (product of eigenvalues) is positive. For a 2×2 matrix, this product is positive when both eigenvalues have the same sign, i.e., negative in our case. We thus get

$$\frac{\sigma_m^2}{L_2\gamma}\cdot\mu\left(1-\frac{1}{\Gamma}\right)-\frac{4\mu^2L_2}{3\Gamma\sigma_m^2}\cdot\frac{\mu}{2\sigma_M^2\Gamma^2}\left(L_1^2+\frac{2\sigma_M^4}{\mu^2}\right)>0,$$
$$\Longleftarrow\ \Gamma^2>\frac{2\mu^2L_2^2\gamma}{3\sigma_m^4\sigma_M^2}\left(L_1^2+\frac{2\sigma_M^4}{\mu^2}\right).$$

We note that Γ appears in the bound on α (see Lemma 1) and γ is a positive constant such that $0<\beta/\gamma\leq\alpha$. Hence, any $\Gamma>2$ that satisfies the above bound can be chosen to make the two eigenvalues negative. Therefore, the asymptotic response $\limsup_{k\to\infty}\mathbf{u}^k$ depends on the last term of (1) and thus the size of the error ball depends on σ^2. □

VI. Numerical Experiments

In this section, we provide numerical experiments to demonstrate the convergence results of **GT-SGDA** and compare its performance with related methods (**D-GDA**, **GT-GDA**, and **D-SGDA**) for different problems. We consider two types of networks based on connectivity (see Figure 2), i.e., (i) directed exponential graph of $n=16$ nodes, representing a highly structured learning environment applicable to data centers; (ii) undirected geometric graph of $n=200$ nodes depicting large-scale ad hoc wireless training setups.

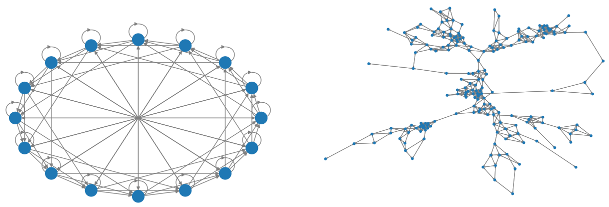

Fig. 2. (left) Directed exponential graph with $n=16$ nodes and (right) undirected geometric graph with $n=200$ nodes.

Regression problems: We first consider a distributed linear regression problem. In many cases, such problems are computationally expensive so we describe the saddle point equivalent form, i.e., for each node i,

$$f_i(\mathbf{x},\mathbf{y}):=\langle\mathbf{y},\mathbf{b}_i\rangle-\frac{1}{2}\|\mathbf{y}\|^2-\langle\mathbf{y},P_i\mathbf{x}\rangle+\lambda_R R_i(\mathbf{x}).$$

We note that every node has its private $P_i\in\mathbb{R}^{p_y\times p_x}$ matrix, a $\mathbf{b}_i\in\mathbb{R}^{p_y}$ vector, and a regularizer term $R_i(\mathbf{x}):\mathbb{R}^{p_x}\to\mathbb{R}$. Globally, we would like to evaluate $\min_{\mathbf{x}}\max_{\mathbf{y}}\frac{1}{n}\sum_{i=1}^{n}f_i(\mathbf{x},\mathbf{y})$.

We consider two types of regularizers:

1) strongly convex regularizer $R_i(\mathbf{x})=\|\mathbf{x}\|^2$; and
2) smooth approximation of *convex* regularizer $R_i(\mathbf{x})=\frac{1}{t_i}\sum_{j=1}^{p_x}\log(1+e^{t_ix_j})(1+e^{-t_ix_j})$. We note that for large t_i, $R_i(\mathbf{x})\approx\|\mathbf{x}\|_1$.

We evaluate the performance of **GT-SGDA** and compare it with related methods, i.e., **D-SGDA**, **GT-GDA**, and **D-GDA**, in terms of optimality gap, i.e., $\|\overline{\mathbf{x}}^k-\mathbf{x}^*\|^2+\|\overline{\mathbf{y}}^k-\mathbf{y}^*\|^2$, with respect to the number of epochs.

Figure 3 shows the performance comparison of **GT-SGDA** with strongly-convex regularizer $R_i(\mathbf{x})=\|\mathbf{x}\|^2$. When the data is distributed over a directed exponential graph with $n=16$ nodes, Figure 3 (left) shows linear convergence of **GT-SGDA** to an error ball around the unique saddle point. We note that its deterministic counterpart **GT-GDA** converges to the exact solution $(\mathbf{x}^*,\mathbf{y}^*)$ but at a much slower speed. The figure also shows that **D-SGDA** and **D-GDA** converge to an inexact solution because they do not use gradient tracking. Similar performance can be seen in Figure 3 (right) where the data is distributed over an undirected geometric graph of $n=200$ nodes. We note that due to the worse connectivity of the graph, all methods require more epochs for convergence.

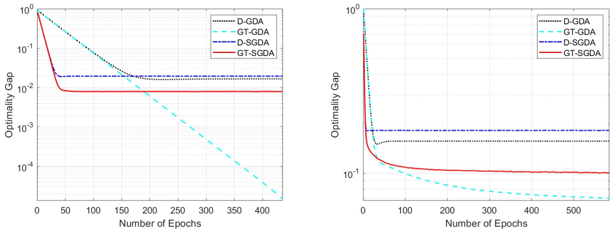

Fig. 3. Performance comparison of **GT-SGDA** with related methods trained over a network of $n=32$ nodes (left) and $n=200$ nodes (right) with strongly convex regularizer.

Next, we observe the performance of above-mentioned methods when the problem is strongly concave-convex, i.e., $R_i(\mathbf{x})=\frac{1}{t_i}\sum_{j=1}^{p_x}\log(1+e^{t_ix_j})(1+e^{-t_ix_j})$. Figure 3 shows the results when data is distributed over a directed exponential graph of $n=16$ nodes (Figure 3, left) and undirected geometric graphs of $n=200$ nodes (Figure 3, right). For both setups, **GT-SGDA** linearly converges to an error ball around the unique saddle point $(\mathbf{x}^*,\mathbf{y}^*)$, which is smaller than that of **D-SGDA**. We note that the stochastic methods are significantly faster than their deterministic counterparts. The slight fluctuations observed in the optimality gap of **D-SGDA** and **GT-SGDA** are due to the variance of stochastic gradients.

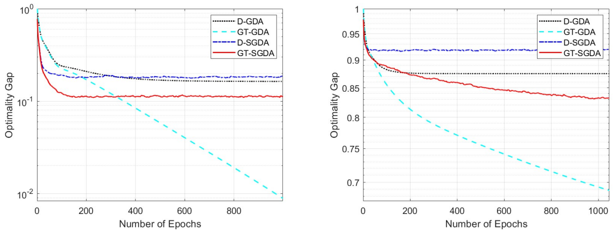

Fig. 4. Performance comparison **GT-SGDA** with related methods trained over a network of $n=32$ nodes (left) and $n=200$ nodes (right) with convex regularizer.

Distributed GANs: We next consider a directed exponential network of four nodes. Each node possesses a three layered fully connected neural network as the generator $\{2\times16\times32\times2\}$ and a four-layered fully connected neural network as the discriminator $\{2\times256\times128\times64\times1\}$. Every layer is linear, followed by a ReLU activation function (except the output layer of discriminator that uses the sigmoid function). The generator estimates the probability distribution of the real samples and the discriminator estimates if the given data is sampled from the real data or is generated by the generator. They compete with each other to solve this min-max problem. Our target is to generate a cycle of sine wave when we provide samples from a random distribution and 'snap-shots' of real data (sine wave) at each node as shown in Figure 5 (top left). We train three different configurations. The basic training setup allows each node to use its private data and train the local GAN. For distributed training, each node shares its local model parameters with its neighbors while performing the back-propagation step. For **D-SGDA**, only weights are shared whereas for **GT-SGDA**, the gradients are also shared with the neighbors. Individually, no node can predict the shape of data distribution due to heterogeneity; independent (local) training results are shown in Figure 5 (top left). The distributed variant **D-SGDA** (which does not use gradient tracking) gives an approximation of the sine wave but Figure 5 (bottom) clearly shows that **GT-SGDA** outperforms other methods using gradient tracking.

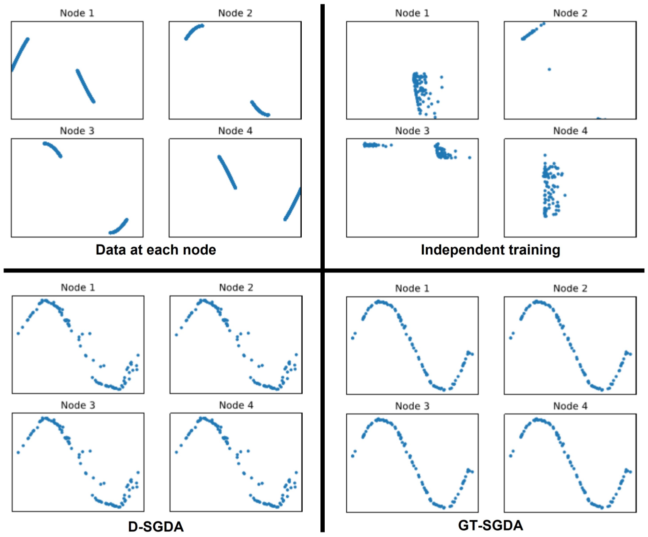

Fig. 5. Performance comparison of **GT-SGDA** with related methods while training distributed GANs.

VII. Conclusion

We propose a distributed stochastic first-order method that uses gradient tracking to solve saddle point problems over strongly connected networks. We show the linear convergence of **GT-SGDA** to an error ball around the unique saddle point when the global cost is strongly concave-convex. We also provide numerical experiments to illustrate the performance of **GT-SGDA** with related methods for distributed regression problems and show that the proposed method can also be used for training distributed GANs.

References

[1] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. 2014, vol. 27, Curran Associates, Inc.

[3] A. Sinha, H. Namkoong, and J. Duchi, "Certifiable distributional robustness with principled adversarial training," in *International Conference on Learning Representations*, 2018.

[4] T. Lin, C. Jin, and M. I. Jordan, "Near-optimal algorithms for minimax optimization," 2021, arXiv: 2002.02417.

[5] T. Lin, C. Jin, and M. I. Jordan, "On gradient descent ascent for nonconvex-concave minimax problems," 2021, arXiv: 1906.00331.

[6] T. Liang and J. Stokes, "Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks.," *CoRR*, vol. abs/1802.06132, 2018.

[7] S. S. Du, J. Chen, L. Li, L. Xiao, and D. Zhou, "Stochastic variance reduction methods for policy evaluation," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds. 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1049–1058, PMLR.

[8] S. S. Du and W. Hu, "Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity," 2019.

[9] A. Mokhtari, A. Ozdaglar, and S. Pattathil, "A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach," 2019, arXiv: 1901.08511.

[10] Y. Malitsky and M. K. Tam, "A forward-backward splitting method for monotone inclusions without cocoercivity," *SIAM Journal on Optimization*, vol. 30, no. 2, pp. 1451–1472, 2020.

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

[12] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[13] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[14] C. Xi and U. A. Khan, "DEXTRA: A fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, Oct. 2017.

[15] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "S-ADDOPT: Decentralized stochastic first-order optimization over directed graphs," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 953–958, Jul. 2021.

[16] P. D. 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.

[17] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *58th IEEE Conference on Decision and Control*, Dec. 2019, pp. 8353–8358.

[18] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.

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

[20] A. Beznosikov, G. Scutari, A. Rogozin, and A. Gasnikov, "Distributed saddle-point problems under similarity," 2021, arXiv: 2107.10706.

[21] M. I. Qureshi and U. A. Khan, "Distributed saddle point problems for strongly concave-convex functions," *arXiv:2202.05812*, 2022.

[22] H. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2018, NIPS'18, p. 9672–9683, Curran Associates Inc.

[23] D. Kovalev, A. Gasnikov, and P. Richtárik, "Accelerated primal-dual gradient method for smooth and convex-concave saddle-point problems with bilinear coupling," 2021, arXiv: 2112.15199.

[24] J. Ren, J. Haupt, and Z. Guo, "Communication-efficient hierarchical distributed optimization for multi-agent policy evaluation," *Journal of Computational Science*, vol. 49, pp. 101280, 2021.

[25] Y. Deng and M. Mahdavi, "Local stochastic gradient descent ascent: Convergence analysis and communication efficiency," 2021, arXiv: 2102.13152.

[26] M. Zhu and Sonia M., "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[27] P. C. Chen and P. P. Vaidyanathan, "Distributed algorithms for array signal processing," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4607–4622, 2021.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[29] A.P. Seyranian and A.A. Mailybaev, *Multiparameter Stability Theory With Mechanical Applications*, World Scientific Pub. Comp., 2003.