Distributed saddle point problems for strongly concave-convex functions

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

***Abstract*—In this paper, we propose `GT-GDA`, a distributed optimization method to solve saddle point problems of the form: $\min_{\mathbf{x}} \max_{\mathbf{y}} \{F(\mathbf{x},\mathbf{y}) := G(\mathbf{x}) + \langle \mathbf{y}, \overline{P}\mathbf{x} \rangle - H(\mathbf{y})\}$, where the functions $G(\cdot)$, $H(\cdot)$, and the coupling matrix $\overline{P}$ are distributed over a strongly connected network of nodes. `GT-GDA` is a first-order method that uses gradient tracking to eliminate the dissimilarity caused by heterogeneous data distribution among the nodes. In the most general form, `GT-GDA` includes a consensus over the local coupling matrices to achieve the optimal (unique) saddle point, however, at the expense of increased communication. To avoid this, we propose a more efficient variant `GT-GDA-Lite` that does not incur additional communication and analyze its convergence in various scenarios. We show that `GT-GDA` converges linearly to the unique saddle point solution when G is smooth and convex, H is smooth and strongly convex, and the global coupling matrix $\overline{P}$ has full column rank. We further characterize the regime under which `GT-GDA` exhibits a network topology-independent convergence behavior. We next show the linear convergence of `GT-GDA-Lite` to an error around the unique saddle point, which goes to zero when the coupling cost $\langle \mathbf{y}, \overline{P}\mathbf{x} \rangle$ is common to all nodes, or when G and H are quadratic. Numerical experiments illustrate the convergence properties and importance of `GT-GDA` and `GT-GDA-Lite` for several applications.**

***Index Terms*—Decentralized optimization, saddle point problems, constrained optimization, descent ascent methods.**

I. Introduction

Saddle point or min-max problems are of significant practical value in many signal processing and machine learning applications [1]–[9]. Applications of interest include but are not limited to constrained and robust optimization, beamforming, weighted linear regression, and reinforcement learning. In contrast to the traditional minimization problems where the goal is to find a global (or a local) minimum, the objective in saddle point problems is to find a point that maximizes the cost in one direction and minimizes it in the other. Consider for example Fig. 1 (left), where we show a simple function landscape ($F : \mathbb{R}^2 \rightarrow \mathbb{R}$) that increases in one direction and decreases in the other. Examples of such functions appear in constrained optimization where adding the constraints as a Lagrangian naturally leads to saddle point formulations.

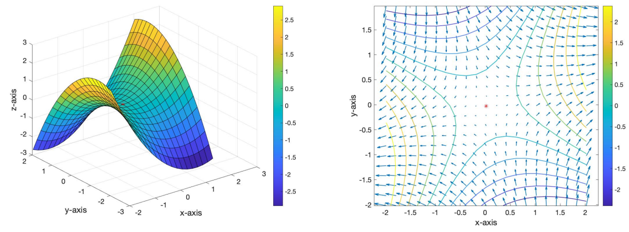

Fig. 1. Plot of two dimensional strongly concave-convex saddle point problem (left) and the corresponding gradient directions (right). The curves represent contours to show the value of the function and the red star is the point where partial gradients are all 0.

Gradient descent ascent (**`GDA`**) methods are popular approaches towards saddle point problems. To find a saddle point of the function in Fig. 1 (left), we would like to maximize F with respect to the corresponding variable, say $\mathbf{y}$, and minimize F in the direction, say $\mathbf{x}$. A natural way is to compute the partial gradients $\nabla_{\mathbf{y}} F$ and $\nabla_{\mathbf{x}} F$, shown in Fig. 1 (right). Then update the $\mathbf{y}$ estimate moving in the direction of $\nabla_{\mathbf{y}} F$ and the $\mathbf{x}$ estimate moving opposite to the direction of $\nabla_{\mathbf{x}} F$. The arrows, shown in Fig. 2, point towards the next step of **`GDA`** dynamics and the method converges to the unique saddle point (red star) under appropriate conditions on F. The extension of this method for convex and strongly concave, and strongly convex and concave objectives is straightforward as it is intuitive that the saddle point $(\mathbf{x}^*, \mathbf{y}^*) \in \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}$ is unique such that $\forall \mathbf{x} \in \mathbb{R}^{p_x}$ and $\forall \mathbf{y} \in \mathbb{R}^{p_y}$,

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

The traditional approaches mentioned above assume that the entire dataset is available at a central location. In many modern applications [10]–[13], however, data is often collected by a large number of geographically distributed devices or nodes and communicating/storing the entire dataset at a central location is practically infeasible. Distributed optimization methods are often preferred in such scenarios, which operate by keeping data local to each individual device and exploit

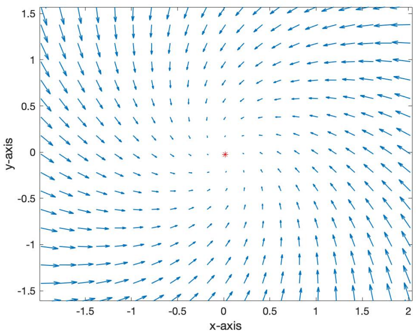

Fig. 2. The arrows point towards the next step of the gradient descent ascent dynamics. The unique saddle point is denoted by the red star.

The authors are with the Electrical and Computer Engineering Department at Tufts University; muhammad.qureshi@tufts.edu and khan@ece.tufts.edu. This work has been partially supported by NSF under awards #1903972, #1935555, and PIRE-2230630. UAK holds concurrent appointments as a Professor at Tufts University and as an Amazon Scholar with Amazon Robotics. This paper describes work performed at Tufts University and is not associated with Amazon.

local computation and communication to solve the underlying problem. Such methods often deploy two types of computational architectures: (i) master/worker networks – where the data is split among multiple workers and computations are coordinated by a master (or a parameter server); (ii) peer-to-peer mesh networks – where the nodes are able to communicate only with nearby nodes over a strongly connected network topology. The topology of mesh networks is more general as it is not limited to hierarchical master/worker architectures.

In this paper, we are interested in solving distributed saddle point optimization problems over peer-to-peer networks, where the corresponding data and cost functions are distributed among n nodes, communicating over a strongly connected weight-balanced directed graph. In this formulation, the networked nodes are tasked to find the saddle point of a sum of local cost functions $f_i(\mathbf{x},\mathbf{y})$, where $\mathbf{x} \in \mathbb{R}^{p_x}$ and $\mathbf{y} \in \mathbb{R}^{p_y}$. Mathematically, we consider 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 each local cost $f_i(\mathbf{x},\mathbf{y})$ is private to node i and takes the form as follows

$$f_i(\mathbf{x},\mathbf{y}) := g_i(\mathbf{x}) + \langle \mathbf{y}, P_i\mathbf{x}\rangle - h_i(\mathbf{y}).$$

We assume that $G(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} g_i(\mathbf{x})$ is convex and $H(\mathbf{y}) := \frac{1}{n}\sum_{i=1}^{n} h_i(\mathbf{y})$ is strongly convex[1], while the global coupling matrix $\overline{P} = \frac{1}{n}\sum_{i=1}^{n} P_i \in \mathbb{R}^{p_y \times p_x}$ has full column rank. Such problems arise naturally in many subfields of signal processing, machine learning, and statistics [14]–[17].

A. Related work

Theoretical studies on solutions for saddle point problems, in centralized scenarios, have attracted significant research [1], [2], [18], [19]. Recently, saddle point or min-max problems have become increasingly relevant because of their applications in constrained and robust optimization, supervised and unsupervised learning, image reconstruction, and reinforcement learning [5]–[7], [9]. Commonly studied sub-classes of saddle point problems of the form $\mathbf{P}$ are when $G(\cdot)$ and $H(\cdot)$ are assumed to be quadratic [14] or strongly convex [20]. In [20], the authors proposed a unified analysis technique for extra-gradient (**EG**) and optimistic gradient descent ascent (**OGDA**) methods assuming strongly concave-strongly convex saddle point problems. Furthermore, they discussed the convergence rates for the underlying problem classes and for bilinear objective functions. A more general approach was taken in [15], where $G(\cdot)$ was considered convex but not strongly convex. In [21], the authors established the convergence of **OGDA** only assuming the existence of saddle points. These are first-order methods with some modification of the vanilla gradient descent (**GD**). Apart from gradient-based methods, zeroth-order optimization techniques are proposed in [22]–[25]. Such methods are useful when gradient computation is not feasible either because the objective function is unknown and the partial derivatives cannot be evaluated for the whole search space, or the evaluation of partial gradients is too expensive. In such cases, Bayesian optimization [23] or genetic algorithms [24], [25] are used. These techniques are usually slower than gradient-based methods.

When the data is distributed over a network of nodes, existing work has mainly focused on *minimization* problems [26]–[32]. Of significant relevance are distributed methods that assume access to a first-order oracle where the early work includes [26], [33], [34]. The performance of these methods is however limited due to their inability to handle the dissimilarity between local and global cost functions, i.e., $\nabla f_i \neq \nabla F$. In other words, linear convergence is only guaranteed but to an inexact solution (with a constant stepsize). To avoid this inaccuracy while keeping linear convergence, recent work [29], [30], [35]–[38] propose a gradient tracking technique that allows each node to estimate the global gradient with only local communication; see also [28] for a related method.

On saddle point problems, there is not much progress made towards distributed solutions. Recent work in this regard includes [16], [17], [39]–[45]. Two primal-dual sub-gradient methods are proposed in [39] to solve distributed convex minimization problem under constrained sets. Some related work on solving distributed variational inequalities was proposed in [40]. Moreover, [41], [42] discuss extra-step and accelerated methods for distributed saddle point problems. However, majority of the work do not consider heterogeneous data distribution among different nodes. To deal with the dissimilarity between the local and global costs, [16] develops a function similarity metric. Similarly, [44] proposes local stochastic gradient descent ascent using similarity parameters but it is restricted to master/worker networks that are typical in federated learning scenarios. To get rid of the aforementioned similarity assumptions, [43] uses gradient tracking to eliminate this dissimilarity but assumes the functions $G(\cdot)$ and $H(\cdot)$ to be quadratic with a specific structure. Similarly, [45] extends [43] to directed graphs using the ideas from [46].

B. Main contributions

In this paper, we propose **GT-GDA** and **GT-GDA-Lite** to solve the underlying distributed saddle point problem $\mathbf{P}$. The **GT-GDA** algorithm performs a gradient descent in the $\mathbf{x}$ direction and a gradient ascent in the $\mathbf{y}$ direction, both of which are combined with a network consensus term along with the communication of coupling matrices P_i with neighbours. **GT-GDA-Lite** is a lighter (communication-efficient) version of **GT-GDA**, which does not require consensus over the coupling matrices and therefore reduces the communication complexity. To address the challenge that arises due to the dissimilarity between the local and global costs, the proposed methods use gradient tracking in both of the descend and ascend updates. To the best of our knowledge, there is no existing work for Problem $\mathbf{P}$ that shows linear convergence when $G(\cdot)$ is convex and $H(\cdot)$ is strongly convex. The main contributions of this paper are described next:

Novel Algorithm. We propose a novel algorithm that uses gradient tracking for distributed gradient descent ascent updates. Gradient tracking implements an extra consensus update where the networked nodes track the global gradients with the help of local information exchange among the nearby nodes.

[1] Note that the problem class $\mathbf{P}$ includes $-H$, which is strongly concave.

Weaker assumptions. We consider the problem class $\mathbf{P}$ such that g_i and h_i are smooth, G is convex, H is strongly convex, and the coupling matrix $\overline{P}$ has full column rank. We note that the constituent local functions, $g_i(\cdot)$ and $h_i(\cdot)$, can be non-convex as we only require convexity on their average. Earlier work [43] that shows linear convergence of distributed saddle point problems is only applicable to specific quadratic functions G and H, used in reinforcement learning, and does not provide explicit rates. It is noteworthy that the proposed problem $\mathbf{P}$ can be written in the primal form as follows:

$$\min_{\mathbf{x}} \theta(\mathbf{x}) = \min_{\mathbf{x}} \left\{ H^*(\overline{P}\mathbf{x}) + G(\mathbf{x}) \right\} \tag{1}$$

where $H^*(\cdot)$ is the conjugate function [47], see Definition 2, of $H(\cdot)$. We note that because $G(\cdot)$ is strongly convex, it is enough to ensure that $\overline{P}$ has full column rank to conclude that $\theta(\cdot)$ is strongly convex [48]. This results in significantly weaker assumptions as compared to the available literature.

Linear convergence and explicit rates. We show that **GT-GDA** converges linearly to the unique saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of Problem $\mathbf{P}$ under the assumptions described above. We note that all these assumptions are necessary for linear convergence even for the centralized case [15]. Furthermore, we evaluate explicit rates for gradient complexity per iteration and provide a regime in which the convergence of **GT-GDA** is network-independent. We also show linear convergence of **GT-GDA-Lite** in three different scenarios and establish that the rate is the same as **GT-GDA** (potentially with a steady-state error) with reduced communication complexity.

Exact analysis for quadratic problems. We provide exact analytic expressions to develop the convergence characteristics of **GT-GDA-Lite** when G and H are in general quadratic forms. With the help of matrix perturbation theory for semi-simple eigenvalues, we show that **GT-GDA-Lite** converges linearly to the unique saddle point of the underlying problem.

C. Notation and paper organization

We use lowercase letters to denote scalars, lowercase bold letters to denote vectors, and uppercase letters to denote matrices. We define $\mathbf{0}_n$ as vector of n zeros and I_n as the identity matrix of $n \times n$ dimensions. For a function $F(\mathbf{x}, \mathbf{y})$, $\nabla_{\mathbf{x}} F$ is the gradient of F with respect to $\mathbf{x}$, while $\nabla_{\mathbf{y}} F$ is the gradient of F with respect to $\mathbf{y}$. We denote the vector two-norm as $\|\cdot\|$ and the spectral norm of a matrix induced by this vector norm as $|||\cdot|||$. We denote the weighted vector norm of a vector $\mathbf{z}$ with respect to a matrix C as $\|\mathbf{z}\|_C := \mathbf{z}^\top C \mathbf{z}$ and the spectral radius of C as $\rho(C)$. We consider n nodes interacting over a potentially directed (balanced) graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} := \{1, \ldots, n\}$ is the set of node indices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a collection of ordered pairs (i, r) such that node r can send information to node i, i.e., $i \leftarrow r$.

The rest of the paper is organized as follows. Section II provides the motivation, with the help of several examples, and describes the algorithms **GT-GDA** and **GT-GDA-Lite**. We discuss our main results in Section III, provide simulations in Section IV, the convergence analysis in Section V, and conclude the paper with Section VI.

II. Motivation and Algorithm Description

In this section, we provide some motivating applications that take the form of convex-concave saddle point problems. For more applications, see e.g., [49], [50].

A. Some useful examples

Distributed constrained optimization. Minimizing an objective function under certain constraints is a fundamental requirement for several applications. For equality constraints, such problems can be written as:

$$\min_{\mathbf{x}} G(\mathbf{x}), \quad \text{subject to} \quad \overline{P}\mathbf{x} = \mathbf{b}, \tag{2}$$

which has a saddle point equivalent form written using the Lagrangian multipliers $\mathbf{y}$:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{y}) &= G(\mathbf{x}) + \mathbf{y}^\top (\overline{P}\mathbf{x} - \mathbf{b}) \\ &= G(\mathbf{x}) + \mathbf{y}^\top \overline{P}\mathbf{x} - \mathbf{y}^\top \mathbf{b}. \end{aligned}$$

Assuming zero duality gap, any solution of (2) is a saddle point of the Lagrangian. Hence, it is sufficient to solve for

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

For large-scale problems, the data is distributed heterogeneously and each node possesses its local $g_i(\cdot), P_i$ and $\mathbf{b}_i$. The network aims to solve (2) such that

$$G(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} g_i(\mathbf{x}), \quad \overline{P} := \frac{1}{n}\sum_{i=1}^{n} P_i, \quad \mathbf{b} := \frac{1}{n}\sum_{i=1}^{n} \mathbf{b}_i.$$

Then for $h_i(\mathbf{y}) := \langle \mathbf{b}_i, \mathbf{y} \rangle$ and $H(\mathbf{y}) := \frac{1}{n}\sum_{i=1}^{n} h_i(\mathbf{y})$, (2) takes the same form as Problem $\mathbf{P}$.

Distributed beamforming. Constrained optimization is widely used for array signal processing. When the signal is uncorrelated with the interference, the Capon beamformer [4] maximizes the SINR by solving the following problem:

$$\min_{\mathbf{x}} \mathbf{x}^H R_{\mathbf{xx}} \mathbf{x}, \quad \text{subject to} \quad \mathbf{s}^H \mathbf{x} = 1,$$

where $\mathbf{x}^H$ is Hermitian of vector $\mathbf{x}$, $R_{\mathbf{xx}} = \mathbb{E}[\mathbf{x}\mathbf{x}^H]$ and $\mathbf{s}$ is the steering vector [3]. Recently, the distributed Capon beamformer is proposed in [51], which essentially solves $\mathbf{P}$.

Distributed weighted linear regression and reinforcement learning. Most applications of weighted linear regression take the form:

$$\min_{\mathbf{x}} \|\overline{P}\mathbf{x} - \mathbf{b}\|^2_{C^{-1}}. \tag{3}$$

It can be shown [15] that the saddle point equivalent of (3) is

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \left\{ -\langle \mathbf{y}, \mathbf{b} \rangle - \frac{1}{2}\|\mathbf{y}\|^2_C + \langle \mathbf{y}, \overline{P}\mathbf{x} \rangle \right\}. \tag{4}$$

This signifies the importance of the saddle point formulation, which enables a solution of (3) without evaluating the inverse of the matrix C, thus decreasing the computational complexity. When the local data is distributed, i.e., $\overline{P} := \frac{1}{n}\sum_{i=1}^{n} P_i$, $h_i(\mathbf{y}) := \langle \mathbf{y}, \mathbf{b}_i \rangle - \frac{1}{2}\|\mathbf{y}\|^2_{C_i}$, and $H(\mathbf{y}) := \frac{1}{n}\sum_{i=1}^{n} h_i(\mathbf{y})$, the above optimization problem takes the form of Problem $\mathbf{P}$.

In several cases [14], [43], reinforcement leaning takes the same form as weighted linear regression. The main objective in reinforcement learning is policy evaluation that requires learning the value function $V^{\boldsymbol{\pi}}$, for any given joint policy $\boldsymbol{\pi}$. The data $\{s_k, s_{k+1}, r_k\}_{k=1}^N$ is generated by the policy $\boldsymbol{\pi}$, where s_k is the state and r_k is the reward at the k-th time step. With the help of a feature function $\phi(\cdot)$, which maps each state to a feature vector, we would like to estimate the model parameters $\mathbf{x}$ such that $V^{\boldsymbol{\pi}} \approx \langle \phi(s), \mathbf{x} \rangle$. A well known method for policy evaluation is to minimize the empirical mean squared projected Bellman error, which is essentially weighted linear regression: $\min_{\mathbf{x}} \|\overline{P}\mathbf{x} - \mathbf{b}\|_{C^{-1}}^2$, where $\overline{P} := \sum_{k=1}^N \langle \phi(s_k), \phi(s_k) - \gamma\phi(s_{k+1}) \rangle$, for some discount factor $\gamma \in (0,1)$, $C := \sum_{k=1}^N \|\phi(s_k)\|^2$, and $\mathbf{b} := \sum_{k=1}^N r_k \phi(s_k)$.

Supervised learning. Classical supervised learning problems are essentially empirical risk minimization. The aim is to learn a linear predictor $\mathbf{x}$ when $H(\cdot)$ is the loss function to be minimized using data matrix $\overline{P}$, and some regularizer $G(\cdot)$. The problem can be expressed as:

$$\min_{\mathbf{x}} \left\{ H(\overline{P}\mathbf{x}) + G(\mathbf{x}) \right\},$$

which has the following saddle point formulation: $\min_{\mathbf{x}} \max_{\mathbf{y}} \left\{ G(\mathbf{x}) + \langle \mathbf{y}, \overline{P}\mathbf{x} \rangle - H^*(\mathbf{y}) \right\}$. For large-scale systems, the data P_i is geographically distributed among different computational nodes and the local functions g_i's and h_i's are also private. Problem $\mathbf{P}$ can be obtained here by choosing $\overline{P} := \frac{1}{n}\sum_{i=1}^n P_i, G(\mathbf{x}) := \sum_{i=1}^n g_i(\mathbf{x})$, and $H^*(\mathbf{y}) := \sum_{i=1}^n h_i^*(\mathbf{y})$.

B. Algorithm development and description

In order to motivate the proposed algorithm, we first describe the canonical distributed minimization problem: $\min_{\mathbf{x}} G(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n g_i(\mathbf{x})$, where G is a smooth and strongly convex function. A well-known distributed solution is given by [30], [52]:

$$\mathbf{x}_i^{k+1} = \sum_{r=1}^n w_{ir}(\mathbf{x}_r^k - \alpha \cdot \nabla g_r(\mathbf{x}_r^k)), \tag{5}$$

where $\mathbf{x}_i^k$ is the estimate of the unique minimizer (denoted as $\mathbf{x}^*$ such that $\nabla G(\mathbf{x}^*) = \frac{1}{n}\sum_i \nabla g_i(\mathbf{x}^*) = \mathbf{0}_{p_x}$) of G at node i and time k, and w_{ir} are the network weights such that $w_{i,r} \neq 0$, if and only if $(i,r) \in \mathcal{E}$, and $W = \{w_{ir}\}$ is primitive and doubly stochastic. Consider for the sake of argument that each node at time k possesses the minimizer $\mathbf{x}^*$; it can be easily verified that $\mathbf{x}_i^{k+1} \neq \mathbf{x}^*$, because the local gradients are not zero at the minimizer, i.e., $\nabla g_i(\mathbf{x}^*) \neq \mathbf{0}_{p_x}$. To address this shortcoming of (5), recent work [29], [35]–[37], [46] uses a certain gradient tracking technique that updates an auxiliary variable $\mathbf{y}_i^k$ over the network such that $\mathbf{y}_i^k \to \frac{1}{n}\sum_i \nabla g_i(\mathbf{x}_k^i)$. The resulting algorithm:

$$\mathbf{x}_i^{k+1} = \sum_{r=1}^n w_{ir}(\mathbf{x}_r^k - \alpha \cdot \mathbf{y}_r^k), \tag{6}$$

$$\mathbf{y}_i^{k+1} = \sum_{r=1}^n w_{ir}(\mathbf{y}_r^k + \nabla g_r(\mathbf{x}_r^{k+1}) - \nabla g_r(\mathbf{x}_r^k)), \tag{7}$$

converges linearly to $\mathbf{x}^*$ thus removing the bias caused by the local ∇g_i versus global gradient ∇G dissimilarity.

To deal with data heterogeneity, the proposed method **GT-GDA**, formally described in Algorithm 1, uses gradient tracking in both the descend and ascend updates. In particular, there are three main components of the **GT-GDA** method: (i) gradient descent for $\mathbf{x}$ updates; (ii) gradient ascent for $\mathbf{y}$ updates; and (iii) gradient tracking. However, since the coupling matrices P_i's are not identical at the nodes, we add an intermediate step to implement consensus on P_i's (see Remark 3 for more details). Initially, **GT-GDA** requires random state vectors $\mathbf{x}_i^0$ and $\mathbf{y}_i^0$ at each node i, gradients evaluated with respect to $\mathbf{x}$ and $\mathbf{y}$ and some positive stepsizes α and β for descent and ascent updates, respectively. At each iteration k, every node computes gradient descent ascent type updates. The state vectors $\mathbf{x}_i^{k+1}$ (and $\mathbf{y}_i^{k+1}$) are evaluated by taking a step in the negative (positive) direction of the gradient of global problem, and then sharing them with the neighbouring nodes according to the network topology. It is important to note that $\mathbf{q}_i^k$ and $\mathbf{w}_i^k$ are the global gradient tracking vectors, i.e., $\mathbf{q}_i^k \to \nabla_{\mathbf{x}} F(\mathbf{x}, \mathbf{y})$ and $\mathbf{w}_i^k \to \nabla_{\mathbf{y}} F(\mathbf{x}, \mathbf{y})$.

Algorithm 1 **GT-GDA** at each node i

```
Require: $\mathbf{x}_i^0 \in \mathbb{R}^{p_x}, \mathbf{y}_i^0 \in \mathbb{R}^{p_y}, P_i^0 = P_i, \{w_{ir}\}_{r=1}^n, \alpha > 0,$
         $\beta > 0, \mathbf{q}_i^0 = \nabla_x f_i(\mathbf{x}_i^0, \mathbf{y}_i^0), \mathbf{w}_i^0 = \nabla_y f_i(\mathbf{x}_i^0, \mathbf{y}_i^0)$
1: for $k = 0, 1, 2, \ldots,$ do,
2:     $P_i^{k+1} \leftarrow \sum_{r=1}^n w_{ir} P_r^k$
3:     $\mathbf{x}_i^{k+1} \leftarrow \sum_{r=1}^n w_{ir}(\mathbf{x}_r^k - \alpha \cdot \mathbf{q}_r^k)$
4:     $\mathbf{q}_i^{k+1} \leftarrow \sum_{r=1}^n w_{ir}(\mathbf{q}_r^k + \nabla_x f_r^{k+1} - \nabla_x f_r^k)$
5:     $\mathbf{y}_i^{k+1} \leftarrow \sum_{r=1}^n w_{ir}(\mathbf{y}_r^k + \beta \cdot \mathbf{w}_r^k)$
6:     $\mathbf{w}_i^{k+1} \leftarrow \sum_{r=1}^n w_{ir}(\mathbf{w}_r^k + \nabla_y f_r^{k+1} - \nabla_y f_r^k)$
7: end for
```

GT-GDA-Lite: We note that **GT-GDA** implements consensus on the coupling matrices (Step 2), which can result in costly communication when the size of these matrices is large. We thus consider a special case of **GT-GDA** that does not implement consensus on the coupling matrices, namely **GT-GDA-Lite**[2] and characterize its convergence properties for the following three cases:

(i) strongly concave-convex problems with different coupling matrices P_i's at each node;
(ii) strongly concave-convex problems with identical P_i's;
(iii) quadratic problems with different P_i's at each node.

III. Main results

Next we provide some definitions followed by the assumptions required to establish the main results.

Definition 1 (Smoothness and convexity)**.** *A differentiable function $G : \mathbb{R}^p \to \mathbb{R}$ is L-smooth if* $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$,

$$\|\nabla G(\mathbf{x}) - \nabla G(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$$

and μ-strongly convex if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$,

$$G(\mathbf{y}) + \langle \nabla G(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2 \leq G(\mathbf{x}).$$

[2] We do not explicitly write **GT-GDA-Lite** as it is the same as Algorithm 1: **GT-GDA** but without the consensus (Step 2) on P_i's.

Algorithm	Problem class	Distributed	Computational complexity
GD	Strongly convex	No	$\mathcal{O}\left(\kappa \log \frac{1}{\epsilon}\right)$
GT-DGD [29]	Strongly convex	Yes	$\mathcal{O}\left(\frac{\kappa^2}{(1-\lambda)^2} \log \frac{1}{\epsilon}\right)$
GDA [15]	Strongly concave-convex	No	$\mathcal{O}\left(\left\{\kappa^4\gamma^2\frac{\mu^2}{\sigma_m^2} + \kappa^3\gamma^4\right\} \log \frac{1}{\epsilon}\right)$
GT-GDA	Strongly concave-convex	Yes	$\mathcal{O}\left(\max\left\{\frac{\gamma^6\kappa^3}{(1-\lambda)^4}, \frac{\sigma_m^2\sigma_M^2\kappa}{L^2\mu^2(1-\lambda)^4}, \gamma^2\kappa^5\right\} \log \frac{1}{\epsilon}\right)$

TABLE I
COMPUTATIONAL COMPLEXITIES OF OPTIMIZATION METHODS.

It is of significance to note that if $G(\cdot)$ is L smooth, then it is also $(L+\xi)$ smooth, $\forall \xi > 0$.

Definition 2 (Conjugate of a function)**.** *The conjugate of a function $H : \mathbb{R}^p \to \mathbb{R}$ is defined as*

$$H^*(\mathbf{y}) := \sup_{\mathbf{x}\in\mathbb{R}^p} \left\{\langle \mathbf{x}, \mathbf{y}\rangle - H(\mathbf{x})\right\}, \quad \forall \mathbf{y} \in \mathbb{R}^p.$$

Moreover, if $H(\cdot)$ is closed and convex, then $[H^*(\cdot)]^* = H(\cdot)$, and if $H(\cdot)$ is L-smooth and μ-strongly convex, then $H^*(\cdot)$ is $\frac{1}{\mu}$-smooth and $\frac{1}{L}$-strongly convex [53].

Next, we describe the assumptions under which the convergence results of **GT-GDA** will be developed; note that all of these assumptions may not be applicable at the same time.

Assumption 1 (Smoothness and convexity)**.** *Each local g_i is L_1-smooth and each h_i is L_2-smooth, where L_1, L_2 are arbitrary positive constants. Furthermore, the global G is convex and the global H is μ-strongly convex.*

Assumption 2 (Quadratic)**.** *The g_i's and h_i's are quadratic functions, i.e.,*

$$g_i(\mathbf{x}) := \mathbf{x}^\top Q_i \mathbf{x} + \mathbf{q}_i^\top \mathbf{x} + q_i,$$
$$h_i(\mathbf{y}) := \mathbf{y}^\top R_i \mathbf{y} + \mathbf{r}_i^\top \mathbf{y} + r_i,$$

such that $\mathbf{q}_i \in \mathbb{R}^{p_x}$, $\mathbf{r}_i \in \mathbb{R}^{p_y}$, $q_i, r_i \in \mathbb{R}$, $Q_i \in \mathbb{R}^{p_x \times p_x}$, and $R_i \in \mathbb{R}^{p_y \times p_y}$, $\forall i$. Moreover, for $\overline{Q} := \frac{1}{n}\sum_{i=1}^n Q_i$ and $\overline{R} := \frac{1}{n}\sum_{i=1}^n R_i$, we assume that $(\overline{Q} + \overline{Q}^\top)$ is positive definite and $(\overline{R} + \overline{R}^\top)$ is positive semi-definite.

Assumption 3 (Full ranked coupling matrix)**.** *The coupling matrix $\overline{P} := \frac{1}{n}\sum_i P_i$ has full column rank.*

Assumption 4 (Doubly stochastic weights)**.** *The weight matrices $W := \{w_{i,r}\}$ associated with the network are primitive and doubly stochastic, i.e., $W\mathbf{1}_n = \mathbf{1}_n$ and $\mathbf{1}_n^\top W = \mathbf{1}_n^\top$.*

We note that Assumption 1 does not require strong convexity of G while Assumptions 1 and 3 are necessary for linear convergence [15]. The primal problem $\min_{\mathbf{x}} \theta(\mathbf{x})$, defined in (1), requires poly (ϵ^{-1}) iterations to obtain an ϵ-optimal solution even in the centralized case if we ignore any of the above assumptions. It is important to note that Assumptions 1-3 are not applicable simultaneously; **GT-GDA** and **GT-GDA-Lite** are analyzed under different assumptions, clearly stated in each theorem. Next, we define some useful constants to explain the main results. Let $L := \max\{L_1, L_2\}$ and let the condition number of $H(\cdot)$ be L_2/μ. Furthermore, we denote $\kappa := L/\mu \geq L_2/\mu$. The maximum and minimum singular values of the coupling matrices P_i's for all i are defined as σ_M and σ_m, respectively. Moreover, the condition number for the global coupling matrix is denoted by $\gamma := \sigma_M/\sigma_m$.

A. Convergence results for *GT-GDA*

We now provide the main results on the convergence of **GT-GDA** and discuss their attributes.

Theorem 1. *Consider Problem* $\mathbf{P}$ *under Assumptions 1, 3, and 4. For a large enough positive constant $c > 0$, assume the stepsizes are such that*

$$\alpha = \overline{\alpha} := \overline{\beta}\frac{\mu^2}{c\sigma_M^2},$$
$$\beta = \overline{\beta} := \min\left\{\frac{\sigma_m^2(1-\lambda)^2}{192\sigma_M^2 L}, \frac{L(1-\lambda)^2}{48\sigma_M^2}, \frac{1}{382\kappa L}\right\}.$$

*Then **GT-GDA** achieves an ϵ-optimal solution in*

$$\mathcal{O}\left(\max\left\{\frac{\gamma^6\kappa^3}{(1-\lambda)^4}, \frac{\sigma_m^2\sigma_M^2\kappa}{L^2\mu^2(1-\lambda)^4}, \gamma^2\kappa^5\right\}\log\frac{1}{\epsilon}\right)$$

gradient computations (in parallel) at each node

Corollary 1. *Consider Problem* $\mathbf{P}$ *under Assumptions 1, 3, and 4, and **GT-GDA** with stepsizes $\alpha := \overline{\alpha}$, $\beta := \overline{\beta}$ and $\Gamma = \max\{\gamma^2, \sigma_m^2/L^2\}$. If*

$$\kappa \geq \frac{\Gamma}{(1-\lambda)^2},$$

*then **GT-GDA** achieves an ϵ-optimal solution linearly at a network-independent convergence rate of $\mathcal{O}\left(\gamma^2\kappa^5 \log\frac{1}{\epsilon}\right)$.*

Table I shows the computational complexities of gradient descent and gradient descent ascent methods along with their distributed counterparts. It can be seen that the computational complexity of **GDA** (centralized) is of the order $\mathcal{O}\left(\kappa^4\right)$ [15], when the objective function is strongly concave-convex (where we used $L = \max\{L_1, L_2\}$, $\kappa = L/\mu$ and $\gamma = \sigma_M/\sigma_m$). It is typical to lose one order of κ in making the algorithm distributed as can be observed in Table I where GD (centralized) has κ dependence but GT-DGD has κ^2 dependence. Similar behaviour is found for **GDA** and **GT-GDA**.

We now discuss these results in the following remarks.

Remark 1 (Linear convergence)**.** ***GT-GDA*** *eliminates the dissimilarity caused by heterogeneous data at each node using gradient tracking in both of the $\mathbf{x}_i^k$ and $\mathbf{y}_i^k$ updates. Theorem 1 provides an explicit linear rate at which **GT-GDA** converges to the unique saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of Problem* $\mathbf{P}$.

Remark 2 (Network-independence)**.** *Corollary 1 explicitly describes a regime in which the convergence rate of* **GT-GDA** *is independent of the network topology. We note that it signifies a relationship between the condition number and the spectral gap* $(1-\lambda)$*. For weakly connected graphs,* **GT-GDA** *requires a higher value of* κ *to attain network independent convergence.*

Remark 3 (Communication complexity)**.** *At each node,* **GT-GDA** *communicates two* p_x*-dimensional vectors, two* p_y*-dimensional vectors, and a* $p_y \times p_x$ *dimensional coupling matrix, per iteration. In ad hoc peer-to-peer networks, the node deployment may not be deterministic. Let* ω *be the expected degree of the underlying (possibly random) strongly connected communication graph. Then the expected communication complexity required for* **GT-GDA** *to achieve an* ϵ*-optimal solution is*

$$\mathcal{O}\left(\omega p_x p_y \max\left\{\frac{\gamma^6\kappa^3}{(1-\lambda)^4}, \frac{\sigma_m^2\sigma_M^2\kappa}{L^2\mu^2(1-\lambda)^4}, \gamma^2\kappa^5\right\}\log\frac{1}{\epsilon}\right)$$

scalars per node. We note that ω *is a function of underlying graph, e.g.,* $\omega = \mathcal{O}(1)$ *for random geometric graphs and* $\omega = \mathcal{O}(\log n)$ *for random exponential graphs.*

GT-GDA converges linearly to the unique saddle point but it requires each node to communicate the local coupling matrix P_i with its neighbors. This incurs additional communication cost for strongly concave-convex local objective functions. Gradient tracking does not account for the discrepancy between local and global coupling matrices as can be seen in Lemma 1. To reduce this communication cost, a finite-time consensus method may be used, see for example [54].

B. Convergence results for **GT-GDA-Lite**

We now discuss **GT-GDA-Lite** in the context of the aforementioned special cases below.

Theorem 2 (**GT-GDA-Lite** for Problem $\mathbf{P}$)**.** *Consider Problem* $\mathbf{P}$ *under Assumptions 1, 3, and 4. If the stepsizes* $\alpha \in (0, \overline{\alpha}]$ *and* $\beta \in (0, \overline{\beta}]$*, then* **GT-GDA-Lite** *converges linearly to an error ball around the unique saddle point.*

Remark 4 (Convergence to an inexact solution)**.** *We note that the speed of convergence for* **GT-GDA-Lite** *is of the same order as* **GT-GDA***, however,* **GT-GDA-Lite** *converges to an error ball around the unique saddle point, which depends on the size of* τ *(formally defined in Lemma 1). This error* τ *can be eliminated by using identical* P_i*'s at each node or by having consensus. The first possibility is considered in the next theorem and the second is explored in* **GT-GDA***.*

Theorem 3 (**GT-GDA-Lite** for Problem $\mathbf{P}$ with same P_i's)**.** *Consider Problem* $\mathbf{P}$ *under Assumptions 1, 3, and 4 and with identical* P_i*'s at each node. If the stepsizes* $\alpha = \overline{\alpha}$ *and* $\beta = \overline{\beta}$*, then the computational complexity of* **GT-GDA-Lite** *to achieve* ϵ*-optimal solution is the same as in Theorem 1, whereas the communication cost reduces by a factor of* $\mathcal{O}(\min(p_x, p_y))$*.*

Remark 5 (Reduced communication complexity)**.** *We note for* **GT-GDA-Lite***, each node communicates two* p_x *dimensional vectors and two* p_y *dimensional vectors per iteration. For large values of* p_x *and* p_y*, this is significantly less than what is required for* **GT-GDA***, i.e.,* $\mathcal{O}(p_x p_y)$*. This makes* **GT-GDA-Lite** *more convenient for applications where communication budget is low.*

Theorem 4 (**GT-GDA-Lite** for quadratic problems)**.** *Consider Problem* $\mathbf{P}$ *under Assumptions 2, 3, and 4 (with different* P_i*'s at the nodes). If the stepsizes* α *and* β *are small enough, then* **GT-GDA-Lite** *converges linearly to the unique saddle point* $(\mathbf{x}^*, \mathbf{y}^*)$ *without consensus on* P_i*'s.*

Remark 6 (Exact analysis)**.** *The convergence analysis we provide for the quadratic case is exact. In other words, we do not use the typical norm bounds and derive the error system of equations as an exact LTI system. Using the concepts from matrix perturbation theory for semi-simple eigenvalues, we show that* **GT-GDA-Lite** *linearly converges to the unique saddle point of* $\mathbf{P}$ *with quadratic cost functions.*

IV. Simulations

We now provide numerical experiments to compare the performance of distributed gradient descent ascent with (**GT-GDA**) and without gradient tracking (**D-GDA**) and verify the theoretical results. We would like to perform a preliminary empirical evaluation on a linear regression problem. We consider the problem of the form:

$$\min_{\mathbf{x}} \frac{1}{2n}\|\overline{P}\mathbf{x} - \mathbf{b}\|^2 + \lambda R(\mathbf{x}); \tag{8}$$

and the saddle point equivalent of above problem is

$$\min_{\mathbf{x}}\max_{\mathbf{y}} \left\{\langle \mathbf{y}, \overline{P}\mathbf{x}\rangle - \langle \mathbf{y}, \mathbf{b}\rangle - \frac{1}{2}\|\mathbf{y}\|^2 + \lambda R(\mathbf{x})\right\}. \tag{9}$$

Performance characterization using the saddle point form of (8) is common in the literature available on centralized gradient descent ascent [15], [20]. For large-scale problems, when data is available over geographically distributed nodes, decentralized implementation is often preferred. In this paper, we consider the network of nodes communicating over strongly connected networks of different sizes and connectivity to extensively evaluate the performance of **GT-GDA**. Figure 4 shows two directed exponential networks of $n = 8$ and $n = 32$ nodes. We note that although they are directed, their corresponding matrices W are weight-balanced. To highlight the significance of distributed processing for large-scale problems, we evaluate the simulation results with the networks shown in Fig. 4 and their extensions to $n = 100$ and $n = 200$ nodes.

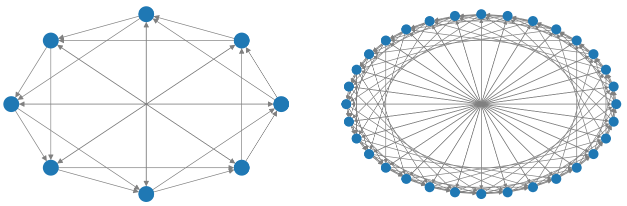

Fig. 4. Directed exponential graphs with $n = 8$ nodes (left) and $n = 32$ nodes (right).

Smooth and strongly convex regularizer: We first consider (9) with smooth and strongly convex regularizer $R(\mathbf{x}) := \|\mathbf{x}\|_C^2$. Therefore, the resulting problem is strongly-convex strongly-concave. For a peer-to-peer mesh network of n nodes, each node i has its private $\mathbf{b}_i \in \mathbb{R}^{p_x}$ and $C_i \in \mathbb{R}^{p_y \times p_x}$ such that the average $\mathbf{b} := \frac{1}{n}\sum_{i=1}^n \mathbf{b}_i$ and $C := \frac{1}{n}\sum_{i=1}^n C_i$, and $\overline{P} := \frac{1}{n}\sum_{i=1}^n P_i$ has full column rank.

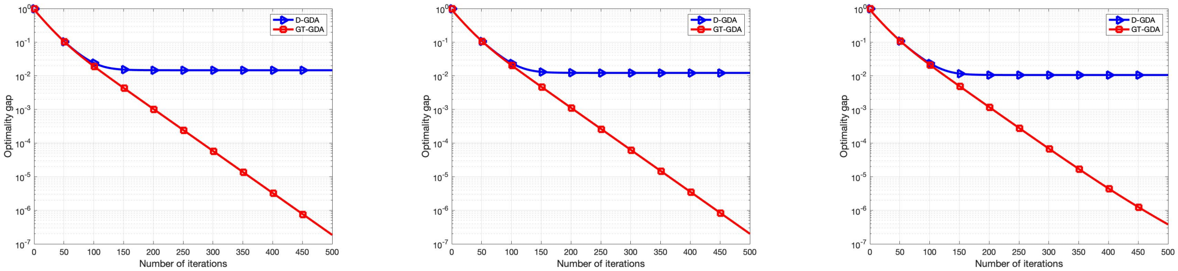

Fig. 3. Performance comparison of **D-GDA** and **GT-GDA** over a network of $n = 8$ nodes (left), $n = 32$ nodes (center) and $n = 100$ nodes (right).

We set the dimensions $p_x = 4$, $p_y = 10$, and evaluate the performance of **GT-GDA** for data generated by a random Gaussian distribution.

We characterize the performance by evaluating the optimality gaps: $\|\mathbf{x}^k - \mathbf{x}^*\| + \|\mathbf{y}^k - \mathbf{y}^*\|$. Fig. 3 represents the comparison of the simulation results of **D-GDA** and **GT-GDA** for different sizes of exponential networks ($n = 8, 32$ and 100); some shown in figure 4. The optimality gap reduces with the increase in the number of iterations. It can be observed that **D-GDA** (blue curve) converges to an inexact solution because it evaluates gradients with respect to its local data at each step; hence moves towards local optimal. On the contrary, the proposed method **GT-GDA** (red curve) uses gradient tracking and consistently converges to the unique saddle point of the global problem. We note that each iteration of **GT-GDA** requires an additional communication cost for exchanging the coupling matrix (see Remark 3 for the exact expression).

Smooth and convex regularizer: Next we use a smooth but *non* strongly convex regularizer [55]:

$$R(\mathbf{x}) := \sum_{i=1}^{n}\sum_{j=1}^{p_x}\left[\frac{1}{t_i}\left\{\log(1+e^{t_i x_j}) + \log(1+e^{-t_i x_j})\right\}\right].$$

Figure 5 shows the results for **GT-GDA** over a network of $n = 32$ and $n = 200$ nodes. It can be seen that **GT-GDA** converges linearly to the unique saddle point, as it's optimality gap decreases, meanwhile **D-GDA** exhibits a similar convergence rate but settles for an inexact solution due to heterogeneous nature of data at different nodes.

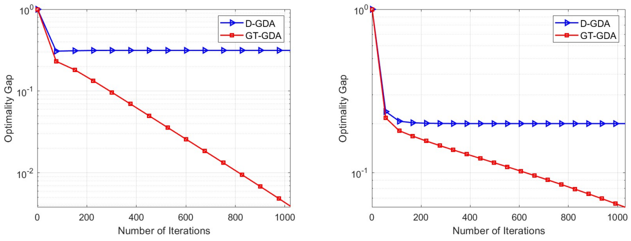

Fig. 5. Performance comparison of **D-GDA** and **GT-GDA** over a network of $n = 32$ nodes (left) and $n = 200$ nodes (right).

Convergence of GT-GDA-Lite: Next we show the performance of **GT-GDA-Lite** for (9) with smooth and strongly convex regularizer $R(\mathbf{x}) := \|\mathbf{x}\|_2^2$. Figure 6 (left) shows linear convergence of **GT-GDA-Lite** to the unique saddle point. Similarly, Figure 6 (right) shows the convergence properties of **GT-GDA-Lite** for the aforementioned smooth and convex regularizer with same and different coupling matrices. It can be observed that the proposed method converges linearly to an error ball around the unique saddle point when the nodes possess different coupling matrices, however, converges to exact solution for the same coupling matrices.

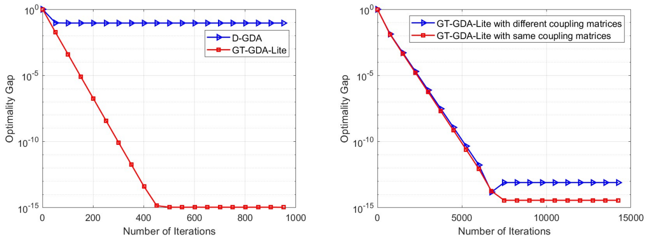

Fig. 6. (Left) Performance comparison of **D-GDA** and **GT-GDA-Lite** over a network of $n = 32$. (Right) Performance comparison of **GT-GDA-Lite** with different and same coupling matrices at each node.

Network independence and linear speedup: Now we analyze the convergence of **GT-GDA** considering three types of networks: (i) a circular graph (bad connectivity); (ii) an exponential graph; and (iii) a complete graph (best connectivity). For a fixed κ and varying $(1 - \lambda)$, Fig 7 (left) shows that the performance of **GT-GDA**. It can be verified that the convergence rate for circular graph is slow but is the same for an exponential graph and a complete graph, which shows network independent convergence rate as claimed in Corollary 1. Finally, we illustrate linear speed-up of **GT-GDA** as compared to its centralized counterpart. We plot the ratio of the number of iterations required to attain an optimality gap of 10^{-14} for **GT-GDA** as compared to the centralized **GDA** method in Fig. 7 (right). The results demonstrate that the performance improves linearly as the number of nodes increases ($n = 8, 16, 32, 100, 200$). We note that for n nodes, the centralized case has n times more data to work with at each iteration and thus has a slower convergence. In distributed setting, the processing is done in parallel which results in a faster overall performance. We emphasize that the implementation of **GT-GDA** requires each node i to communicate the coupling matrix P_i and the state variables with its neighbors. However, **GT-GDA-Lite** eliminates the requirement of communicating P_i.

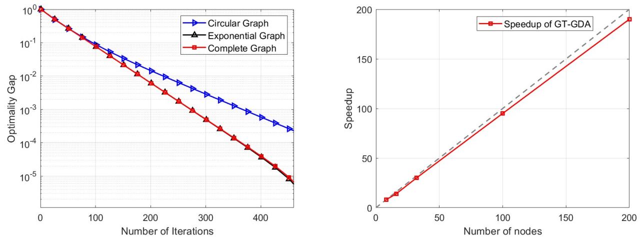

Fig. 7. (Left) Convergence of **GT-GDA** for networks with different connectivity. (Right) Linear speedup: Performance ratio of **GT-GDA** with its centralized counterpart to achieve optimality gap of 10^{-14}.

V. Convergence analysis

In this section, our aim is to establish linear convergence of the proposed algorithms to the unique saddle point (under given set of assumptions) for problem class $\mathbf{P}$. We first define four global state vectors $\mathbf{x}^k, \mathbf{q}^k \in \mathbb{R}^{np_x}$, $\mathbf{y}^k, \mathbf{w}^k \in \mathbb{R}^{np_y}$ that concatenate the local vectors $\mathbf{x}_i^k, \mathbf{q}_i^k, \mathbf{y}_i^k$, and $\mathbf{w}_i^k$ for all i. We next define the following error quantities with the goal of characterizing their time evolution in order to establish that the error decays to zero:

(i) Agreement errors, $\|\mathbf{x}^k - W_1^\infty \mathbf{x}^k\|$ and $\|\mathbf{y}^k - W_2^\infty \mathbf{y}^k\|$: Note that we define $W^\infty := \lim_{k\to\infty} W^k = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, $W_1 := W \otimes I_{p_x}$, $W_2 := W \otimes I_{p_y}$ (where $\otimes$ denotes the Kronecker product), and thus each error quantifies how far the network is from agreement;

(ii) Optimality gaps, $\|\overline{\mathbf{x}}^k - \mathbf{x}^*\|$ and $\|\overline{\mathbf{y}}^k - \mathbf{y}^*\|$ or $\|\overline{\mathbf{y}}^k - \nabla H^*(\overline{P}\overline{\mathbf{x}}^k)\|$: Note that $\overline{\mathbf{x}}^k := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^k$, $\overline{\mathbf{y}}^k := \frac{1}{n}\sum_{i=1}^n \mathbf{y}_i^k$, and thus each error quantifies the discrepancy between the network average and the unique saddle point $(\mathbf{x}^*, \mathbf{y}^*)$;

(iii) Gradient tracking errors, $\|\mathbf{q}^k - W_1^\infty \mathbf{q}^k\|^2$ and $\|\mathbf{w}^k - W_2^\infty \mathbf{w}^k\|^2$: Note that these errors quantify the difference between the local and global gradients.

A. Convergence of **GT-GDA**

The following lemma provides a relationship between the error quantities defined above with the help of an LTI system describing **GT-GDA**.

Lemma 1. *Consider **GT-GDA** described in Algorithm 1 under Assumptions 1, 3, and 4. We define* $\mathbf{u}^k, \mathbf{s}^k \in \mathbb{R}^6$ *as*

$$\mathbf{u}^k := \begin{bmatrix} \|\mathbf{x}^k - W_1^\infty \mathbf{x}^k\| \\ \sqrt{n}\|\overline{\mathbf{x}}^k - \mathbf{x}^*\| \\ L^{-1}\|\mathbf{q}^k - W_1^\infty \mathbf{q}^k\| \\ \|\mathbf{y}^k - W_2^\infty \mathbf{y}^k\| \\ \sqrt{n}\|\overline{\mathbf{y}}^k - \nabla H^*(\overline{P}\overline{\mathbf{x}}^k)\| \\ L^{-1}\|\mathbf{w}^k - W_2^\infty \mathbf{w}^k\| \end{bmatrix}, \qquad \mathbf{s}^k := \begin{bmatrix} \|\mathbf{x}^k\| \\ \|\mathbf{y}^k\| \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and let $N_{\alpha,\beta,k} \in \mathbb{R}^{6\times 6}$ *be such that it has* $\alpha\lambda^k\tau$ *and* $\beta\lambda^k\tau$ *at the* $(2,1)$ *and* $(1,5)$ *locations, respectively, and zeros everywhere else. We note that* $\tau := \left|\left|\left| P^0 - W_2^\infty P^0 \right|\right|\right|$ *where* P^0 *concatenates* P_i*'s initially available at each node. For all* $k \geq 0$, $\alpha, \beta > 0$, *and* $\alpha \leq \beta\frac{\mu^2}{c\sigma_M^2}$, *we have*

$$\mathbf{u}^{k+1} \leq M_{\alpha,\beta}\mathbf{u}^k + N_{\alpha,\beta,k}\mathbf{s}^k, \tag{10}$$

where the system matrix $M_{\alpha,\beta}$ *is defined in Appendix A.*

The proof of the above lemma is omitted from here due to space limitations and can be found in the technical report [56]. The main idea behind the proof is to establish bounds on error terms (elements of $\mathbf{u}^k$) for descent in $\mathbf{x}$ and ascent in $\mathbf{y}$ for a range of stepsizes α and β. It is noteworthy that the coupling between the ascent and descent equations gives rise to additional terms (see $M_{\alpha,\beta}$ in Appendix A) adding to the complexity of the analysis. Moreover, the analysis requires a careful manipulation of the two stepsizes, unlike the existing approaches. With the help of this lemma, our goal is to establish convergence of **GT-GDA** and further characterize its convergence rate. To this aim, we first show that the spectral radius $\rho(M_{\alpha,\beta})$ of the system matrix is less than 1 in the following lemma.

Lemma 2. *Consider **GT-GDA** under Assumptions 1, 3, and 4. For a large enough positive constant* $c > 0$, *assume the stepsizes are* $\alpha \in \left(0, \overline{\beta}\frac{\mu^2}{c\sigma_M^2}\right]$ *and* $\beta \in (0, \overline{\beta}]$, *such that*

$$\overline{\beta} := \min\left\{\frac{\sigma_m^2(1-\lambda)^2}{192\sigma_M^2 L}, \frac{L(1-\lambda)^2}{48\sigma_M^2}, \frac{1}{382\kappa L}\right\},$$

then $\rho(M_{\alpha,\beta}) \leq \eta < 1$, *where*

$$\eta := 1 - \mathcal{O}\left(\min\left\{\frac{(1-\lambda)^4}{c\gamma^6\kappa^3}, \frac{L^2\mu^2(1-\lambda)^4}{c\sigma_m^2\sigma_M^2\kappa}, \frac{1}{c\gamma^2\kappa^5}\right\}\right).$$

Proof. Recall that $M_{\alpha,\beta}$ is a non-negative matrix. From [57], we know that if there exists a positive vector $\boldsymbol{\delta}$ and a positive constant η such that $M_{\alpha,\beta}\boldsymbol{\delta} \leq \eta\boldsymbol{\delta}$, then $\rho(M_{\alpha,\beta}) \leq |||M_{\alpha,\beta}|||_\infty^{\boldsymbol{\delta}} \leq \eta$, where $|||\cdot|||_\infty^{\boldsymbol{\delta}}$ is the matrix norm induced by the weighted max-norm $\|\cdot\|_\infty^{\boldsymbol{\delta}}$, with respect to some positive vector $\boldsymbol{\delta}$. To this end, we first choose $\eta := \left(1 - \alpha\beta\frac{\sigma_m^2}{\kappa}\right)$, which is clearly less than 1. We next solve for a range of $\alpha, \beta > 0$ and for a positive vector $\boldsymbol{\delta} = [\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6]^\top$ such that the inequalities in $M_{\alpha,\beta}\boldsymbol{\delta} \leq \left(1 - \alpha\beta\frac{\sigma_m^2}{\kappa}\right)\boldsymbol{\delta}$ hold element-wise. With $M_{\alpha,\beta}$ in Appendix A, from the first and the fourth rows, we obtain

$$\alpha\beta\frac{\sigma_m^2}{\kappa} + \alpha L\frac{\delta_3}{\delta_1} \leq 1 - \lambda, \tag{11}$$

$$\alpha\beta\frac{\sigma_m^2}{\kappa} + \beta L\frac{\delta_6}{\delta_4} \leq 1 - \lambda. \tag{12}$$

Similarly, from the second and the fifth rows, we obtain

$$\beta\mu\delta_2 \leq \delta_2 - \frac{L}{\sigma_m^2}\left(L\delta_1 + \sigma_M\delta_5\right), \tag{13}$$

$$\begin{aligned} \alpha\frac{\sigma_m^2}{\kappa}\delta_5 \leq \mu\left(1 - \tfrac{1}{c}\right)\delta_5 - \frac{\mu L}{c\sigma_M}\delta_1 \\ - \frac{\mu^2}{c\sigma_M^2}m_3\delta_2 - \left(\tfrac{\mu}{c} + L\right)\delta_4. \end{aligned} \tag{14}$$

Finally, from the third and the sixth rows, we obtain

$$\begin{aligned} &\left(\alpha\lambda L + \beta\frac{\lambda\sigma_M^2}{L}\right)\delta_1 + (\alpha m_1 + \beta m_2)\,\delta_2 + \beta\lambda\sigma_M\delta_6 \\ &+ \left(\alpha\lambda L + \alpha\beta\frac{\sigma_m^2}{\kappa}\right)\delta_3 + (\alpha\lambda\sigma_M + \beta\lambda\sigma_M)\,\delta_4 \\ &+ (\alpha\lambda\sigma_M + \beta\lambda\sigma_M)\,\delta_5 \leq (1-\lambda)\,\delta_3 - \lambda\left(\delta_1 + \tfrac{\sigma_M}{L}\delta_4\right), \end{aligned} \tag{15}$$

$$\begin{aligned} &(\alpha\lambda\sigma_M + \beta\lambda\sigma_M)\,\delta_1 + (\alpha m_4 + \beta m_5)\,\delta_2 + \alpha\lambda\sigma_M\delta_3 \\ &+ \left(\alpha\frac{\lambda\sigma_M^2}{L} + \beta\lambda L\right)\delta_4 + \left(\alpha\frac{\lambda\sigma_M^2}{L} + \beta\lambda L\right)\delta_5 \\ &+ \left(\beta\lambda L + \alpha\beta\frac{\sigma_m^2}{\kappa}\right)\delta_6 \leq (1-\lambda)\,\delta_6 - \lambda\left(\tfrac{\sigma_M}{L}\delta_1 + \delta_4\right). \end{aligned} \tag{16}$$

We note that (13)–(16) hold true for some feasible range of α and β when their right hand sides are positive. Thus, we fix the elements of $\boldsymbol{\delta}$ (independent of stepsizes) as

$$\delta_1 = \frac{\sigma_M}{L}, \qquad \delta_2 = 4l_2\left[1 + l_3\right], \qquad \delta_3 = \frac{\lambda}{1-\lambda}\frac{2\sigma_M}{L},$$
$$\delta_4 = \frac{c-1}{2(1+c\kappa)}, \qquad \delta_5 = 2l_3\left[1 + 4l_1 l_2\left(1 + l_3\right)\right] + 1,$$
$$\delta_6 = \frac{\lambda}{1-\lambda}\sigma_M^2\left(\frac{1}{L^2} + \frac{1}{\sigma_m^2}\right),$$

where $c > \frac{2L^2}{\sigma_m^2} + \frac{2\sigma_M^2\kappa}{\sigma_m^2} + 1$ and

$$l_1 := \frac{L}{\sigma_M} + \frac{\sigma_M}{\mu}, \quad l_2 := \frac{\sigma_M L}{d\sigma_m^2}, \quad l_3 := \frac{1}{c-1}, \quad d = \frac{1}{1+2l_1 l_2 l_3}.$$

It can be verified that for the above choice of $\boldsymbol{\delta}$, the right hand sides of (13)–(16) are positive. Next we solve for the range of α and β. It can be verified that (11) and (12) are satisfied when $\alpha \leq \frac{(1-\lambda)^2}{4\lambda\sigma_M}$,

$$\beta \leq \tfrac{(1-\lambda)^2}{4\lambda\sigma_M^2}\left(\tfrac{c-1}{1+c\kappa}\right)\left(\tfrac{\sigma_m^2 L}{L^2+\sigma_m^2}\right) \qquad \text{and} \qquad \alpha\beta \leq \tfrac{\kappa(1-\lambda)}{2\sigma_m^2}.$$

Similarly, the relations (13) and (14) hold for

$$\alpha \leq \tfrac{\kappa\mu(c-1)}{2c\sigma_m^2} \qquad \text{and} \qquad \beta \leq \tfrac{1}{4\mu}.$$

Finally, it can be verified that (15) and (16) hold when

$$\alpha \leq \min\left\{\tfrac{d\sigma_m^2}{384L^3}, \tfrac{d\sigma_m^2}{384\sigma_M^2\kappa L}, \tfrac{d\mu}{384\sigma_M^3}, \tfrac{1-\lambda}{48L\boldsymbol{\delta}_5}, \tfrac{L(1-\lambda)}{24\sigma_M^2\boldsymbol{\delta}_5}\right\},$$

$$\beta \leq \min\left\{\tfrac{\sigma_m^2(1-\lambda)^2}{192\sigma_M^2 L}, \tfrac{L(1-\lambda)^2}{48\sigma_M^2}, \tfrac{d}{382\kappa L}, \tfrac{1}{24L\boldsymbol{\delta}_5}\right\},$$

and $\alpha\beta \leq \min\{\frac{(1-\lambda)\kappa}{48\sigma_m^2}, \frac{(1-\lambda)\kappa L^2}{48\sigma_m^4}\}$. The lemma follows by simplifying all the α and β bounds for some large enough c with $\eta = \left(1 - \alpha\beta\frac{\sigma_m^2}{\kappa}\right)$. □

The above lemma shows that the spectral radius of $M_{\alpha,\beta}$ is less than or equal to a positive constant $\eta < 1$ for appropriate stepsizes α and β. We emphasize that the proof of Lemma 2 does not follow the conventional strategies used in the literature on distributed optimization [57]–[59]. It requires careful selection of η and evaluation of appropriate bounds on both the stepsizes (α and β) to ensure convergence. Using the two lemmas above, we are now in a position to prove Theorem 1. It is noteworthy that $N_{\alpha,\beta,k}$ decays faster than $M_{\alpha,\beta}$ as it can be verified that $\lambda \leq \rho(M_{\alpha,\beta})$. We now show that $\|\mathbf{u}^k\| \to 0$ and prove Theorem 1.

1) Proof of Theorem 1: We first rewrite the LTI system dynamics described in Lemma 1 recursively as

$$\mathbf{u}^k \leq M_{\alpha,\beta}^k \mathbf{u}^0 + \textstyle\sum_{r=0}^{k-1} M_{\alpha,\beta}^{k-r-1} N_{\alpha,\beta,k}\mathbf{s}^r. \tag{17}$$

We now take the norm on both sides such that for some positive constants $\omega_1, \omega_2, \omega_3$ and ω_4, (17) can be written as

$$\begin{aligned}\|\mathbf{u}^k\| &\leq \|M_{\alpha,\beta}^k \mathbf{u}^0\| + \textstyle\sum_{r=0}^{k-1} \|M_{\alpha,\beta}^{k-r-1} N_{\alpha,\beta,k}\mathbf{s}^r\| \\ &\leq \omega_1\eta^k + \omega_2\eta^k \textstyle\sum_{r=0}^{k-1}\|\mathbf{s}^r\|,\end{aligned}$$

where $\|\mathbf{s}^r\| \leq \omega_3\|\mathbf{u}^k\| + \omega_4\|\mathbf{x}^*\| + \omega_5\|\mathbf{y}^*\|$, with the help of some arbitrary norm equivalence constants. It can be verified that for $a := \omega_4\|\mathbf{x}^*\| + \omega_5\|\mathbf{y}^*\|$,

$$\|\mathbf{u}^k\| \leq \left(\omega_1 + ka + \omega_2\omega_3 \textstyle\sum_{r=0}^{k-1}\|\mathbf{u}^r\|\right)\eta^k.$$

Let $b_k := \sum_{r=0}^{k-1}\|\mathbf{u}^r\|$, $c_k := (\omega_1 + ka)\eta^k$ and $d_k := \omega_2\omega_3\eta^k$. Then the above can be re-written as

$$\begin{aligned}\|\mathbf{u}^k\| = b_{k+1} - b_k &\leq (\omega_1 + ka + \omega_2\omega_3 b_k)\,\eta^k, \\ \iff b_{k+1} &\leq (1 + d_k)b_k + c_k.\end{aligned}$$

For non-negative sequences $\{b_k\}, \{c_k\}$ and $\{d_k\}$ related as $b_{k+1} \leq (1 + d_k)b_k + c_k, \forall k$, such that $\sum_{k=0}^{\infty} c_k < \infty$ and $\sum_{k=0}^{\infty} d_k < \infty$, we have that the sequence $\{b_k\}$ converges and is bounded [60]. Therefore, $\forall \nu \in (\eta, 1)$, we have

$$\lim_{k\to\infty}\frac{\|\mathbf{u}^k\|}{\nu^k} \leq \lim_{k\to\infty}\frac{(\omega_1 + ka + \omega_2\omega_3 b_k)\,\eta^k}{\nu^k} = 0,$$

and there exists a $\psi > 0$, such that $\|\mathbf{u}^k\| \leq \psi(\eta + \xi)^k$ for all k, where $\xi > 0$ is an arbitrarily small constant. To achieve an ϵ-accurate solution, we need

$$\begin{aligned}\|\mathbf{x}^k - \mathbf{1}_n \otimes \mathbf{x}^*\| + \|\mathbf{y}^k - \mathbf{1}_n \otimes \mathbf{y}^*\| \leq \epsilon, \\ \Longleftarrow \|\mathbf{u}^k\| \leq e^{-(1-(\eta+\xi))k}\theta \leq \epsilon,\end{aligned}$$

and the theorem follows. □

B. Convergence of ***GT-GDA-Lite***

We now establish the convergence of **GT-GDA-Lite** under the corresponding set of assumptions. It can be verified that the LTI system for **GT-GDA-Lite** is similar to the one described in Lemma 1 except that the non-zero elements in $N_{\alpha,\beta,k}$ are replaced by $\alpha\tau$ at the $(2,1)$ location and $\beta\tau$ at the $(1,5)$ location (named as $\widetilde{N}_{\alpha,\beta}$). In the following lemma, we consider the convergence of **GT-GDA-Lite** under least assumptions, i.e., when Pi's are not necessarily identical.

1) Proof of Theorem 2: Consider **GT-GDA-Lite** under Assumptions 1, 3, and 4. Given the stepsizes $\alpha \in (0, \overline{\alpha}]$ and $\beta \in (0, \overline{\beta}]$, with $\overline{\alpha}$ and $\overline{\beta}$ defined in Theorem 1, we have

$$\mathbf{u}^k \leq M_{\alpha,\beta}^k\mathbf{u}^0 + \sum_{r=0}^{k-1} M_{\alpha,\beta}^{k-r-1}\widetilde{N}_{\alpha,\beta}\mathbf{s}^k. \tag{18}$$

We have already established the fact that $\rho(M_{\alpha,\beta}) \leq \eta < 1$ (see Lemma 2). Therefore, the first term disappears exponentially, and the asymptotic response is

$$\limsup_{k\to\infty}\mathbf{u}^k \leq (I_6 - M_{\alpha,\beta})^{-1}\widetilde{N}_{\alpha,\beta}\mathbf{s}, \tag{19}$$

where $\mathbf{s} := \left[\sup_k\|\mathbf{x}^k\|, \sup_k\|\mathbf{y}^k\|, 0, 0, 0, 0\right]^\top$. The exact size of the error ball may be evaluated by calculating the norm $\|(I_6 - M_{\alpha,\beta})^{-1}\widetilde{N}_{\alpha,\beta}\mathbf{s}\|$. It is noteworthy that the only two nonzero terms in the vector $\widetilde{N}_{\alpha,\beta}\mathbf{s}$ are controllable by the stepsizes α or β and Theorem 2 follows. □

Next, we provide the proof of Theorem 3, which assumes that each node has the same coupling matrix, and establishes convergence of **GT-GDA-Lite**.

2) Proof of Theorem 3: Consider **GT-GDA-Lite** under Assumptions 1, 3, and 4, with identical P_i's. It can be verified that for stepsizes $\alpha \in (0, \overline{\alpha}]$, and $\beta \in (0, \overline{\beta}]$, the LTI system described in (10) reduces to

$$\mathbf{u}^{k+1} \leq M_{\alpha,\beta}\mathbf{u}^k, \tag{20}$$

because $\tau = 0$. The theorem thus follows since $\rho(M_{\alpha,\beta}) < 1$ from Lemma 2. □

C. ***GT-GDA-Lite*** *for quadratic problems*

We now consider **GT-GDA-Lite** for quadratic problems where the coupling matrices are not necessarily identical at each node. We show that **GT-GDA-Lite** converges to the unique saddle point without needing consensus. We now define the corresponding LTI system in the following lemma.

Lemma 3 (**GT-GDA-Lite** for quadratic problems)**.** *Consider Problem* $\mathbf{P}$ *under Assumptions 2, 3, and 4 (with different* P_i*'s at the nodes). Then the LTI dynamics governing* ***GT-GDA-Lite*** *are defined by the following*

$$\widetilde{\mathbf{u}}^{k+1} = \widetilde{M}_{\alpha,\beta}\widetilde{\mathbf{u}}^k, \tag{21}$$

where

$$\widetilde{\mathbf{u}}^k := \begin{bmatrix} \mathbf{x}^k - W_1^\infty \mathbf{x}^k \\ \overline{\mathbf{x}}^k - \mathbf{x}^* \\ \mathbf{q}^k - W_1^\infty \mathbf{q}^k \\ \mathbf{y}^k - W_2^\infty \mathbf{y}^k \\ \overline{\mathbf{y}}^k - \mathbf{y}^* \\ \mathbf{w}^k - W_2^\infty \mathbf{w}^k \end{bmatrix},$$

and the system matrix $\widetilde{M}_{\alpha,\beta}$ is defined in Appendix B.

Lemma 3 provides an exact analysis of **GT-GDA-Lite** for quadratic problems. The derivation of the above system follows similar arguments as provided in Lemma 1 without the use of norm inequalities; detailed analysis can be found in the technical report [56].

Our aim now is to establish linear convergence of **GT-GDA-Lite** to the exact saddle point. To proceed, we set $\beta = \alpha$, which makes $\widetilde{M}_{\alpha,\beta}$ as a function of α alone. Thus we define $\widetilde{M}_\alpha := \widetilde{M}_{\alpha,\beta=\alpha}$ and $\widetilde{M}_0 := \widetilde{M}_{\alpha=0}$ leading to

$$\widetilde{M}_0 := \begin{bmatrix} \overline{W}_1 & O & O & O & O & O \\ O & I_{p_x} & O & O & O & O \\ \times & O & \overline{W}_1 & \times & O & O \\ O & O & O & \overline{W}_2 & O & O \\ O & O & O & O & I_{p_y} & O \\ \times & O & O & \times & O & \overline{W}_2 \end{bmatrix},$$

where $\overline{W}_1 := W_1 - W_1^\infty$, $\overline{W}_2 := W_2 - W_2^\infty$, O are zero matrices of appropriate dimensions, and '$\times$' are "don't care terms" that do not affect further analysis. Using Schur's Lemma for determinant of block matrices, we note that, for the given structure of $\widetilde{M}_0$, the eigenvalues of $\widetilde{M}_0$ are the eigenvalues of the diagonal block matrices. Furthermore, we know that $\rho(W - W^\infty) < 1$, which implies that $\rho(\overline{W}_1) < 1$ and $\rho(\overline{W}_2) < 1$. Therefore, $\rho(\widetilde{M}_0) = 1$ and $\widetilde{M}_0$ has $p := p_x + p_y$ semi-simple eigenvalues. We would like to show that for sufficiently small positive stepsizes and $\beta = \alpha$, all eigenvalues of 1 decrease and thus the spectral radius of $\widetilde{M}_{\alpha,\beta}$ becomes less than 1. It is noteworthy that the analysis in the existing literature on distributed optimization [46], [61] is limited to a simple eigenvalue and thus cannot be directly extended to our case. We provide the following lemma to establish the change in the semi-simple eigenvalues with respect to α.

Lemma 4. *[62] Consider an $n \times n$ matrix M_α which depends smoothly on a real parameter $\alpha \geq 0$. Fix $l \in [1, n]$ and let $\lambda_1 = \cdots = \lambda_l$ be a semi-simple eigenvalue of M_0, with (linearly independent) right eigenvectors $\mathbf{y}_1, \cdots, \mathbf{y}_l$ and (linearly independent) left eigenvectors $\mathbf{z}_1, \cdots, \mathbf{z}_l$. Denote by $\lambda_i(\alpha)$ the eigenvalues of M_α corresponding to λ_i, $i \in [1, l]$, as a function of α. Then the derivatives $d\lambda_i(\alpha)/d\alpha$ exist, and $d\lambda_i(\alpha)/d\alpha|_{\alpha=0}$ is given by the eigenvalues $[\lambda_S]_i$ of the following $l \times l$ matrix*

$$S := \begin{bmatrix} \mathbf{z}_1^\top M' \mathbf{y}_1 & \cdots & \mathbf{z}_1^\top M' \mathbf{y}_l \\ \vdots & \ddots & \vdots \\ \mathbf{z}_l^\top M' \mathbf{y}_1 & \cdots & \mathbf{z}_l^\top M' \mathbf{y}_l \end{bmatrix},$$

where $M' := dM_\alpha/d\alpha|_{\alpha=0}$. Furthermore, the eigenvalue $\lambda_i(\alpha)$ under perturbation of the parameter α is given by

$$\lambda_i(\alpha) = \lambda_i(0) + \alpha[\lambda_S]_i + o(\alpha), \qquad \forall i \in [1, l].$$

With the help of Lemma 4, we now prove Theorem 4.

1) Proof of Theorem 4: We note that $\widetilde{M}_\alpha := \widetilde{M}_0 + \alpha M'$, where $M' := d\widetilde{M}_\alpha/d\alpha|_{\alpha=0}$. Furthermore, the left eigenvectors corresponding to p semi-simple eigenvalues are the rows of the matrix U and the right eigenvectors are the columns of the matrix V, as defined below:

$$U := \begin{bmatrix} O & I_{p_x} & O & O & O & O \\ O & O & O & O & I_{p_y} & O \end{bmatrix} \quad \text{and} \quad V := U^\top.$$

We would like to ensure that the semi-simple eigenvalues of $\widetilde{M}_0$ are forced to move inside the unit circle as α increases. Using Lemma 4, it can be established that the derivatives $d\lambda_i(\alpha)/d\alpha|_{\alpha=0}$ exist for all $i \in [1, p]$ and are the eigenvalues $[\lambda_S]_i$ of

$$S := UM'V = \begin{bmatrix} -\overline{Q} & -\overline{P}^\top \\ \overline{P} & -\overline{R} \end{bmatrix}.$$

Next we define $a_i := \mathrm{Re}\left([\lambda_S]_i\right)$ and $b_i := \mathrm{Im}\left([\lambda_S]_i\right)$, $\forall i \in [1, p]$. From Theorem 3.6 in [2], we know that $-S$ is positive stable if $(\overline{Q} + \overline{Q}^\top)$ is positive definite and $(\overline{R} + \overline{R}^\top)$ is positive semi-definite, i.e., $\forall i \in [1, p], a_i < 0$. Furthermore, we know from Lemma 4 that

$$\lambda_i(\alpha) = \lambda_i(0) + \alpha[\lambda_S]_i + o(\alpha), \tag{22}$$

see Theorem 2.7 in [62] for details. For sufficiently small stepsize $\alpha > 0$, the term $o(\alpha)$ can be made arbitrarily small as it contains higher order α terms. Therefore, we can rewrite (22) as

$$\lambda_i(\alpha) = 1 + \alpha[\lambda_S]_i = 1 + \alpha(a_i + jb_i), \qquad \forall i \in [1, p].$$

Since the real parts of $[\lambda_S]_i$ are $a_i < 0$, the semi-simple eigenvalues would move towards the direction of the secants of the unit circle for any b_i, see Fig. 8. We thus obtain that $\rho(\widetilde{M}_\alpha) < 1$ for sufficiently small stepsize $\alpha > 0$ and Theorem 4 follows. □

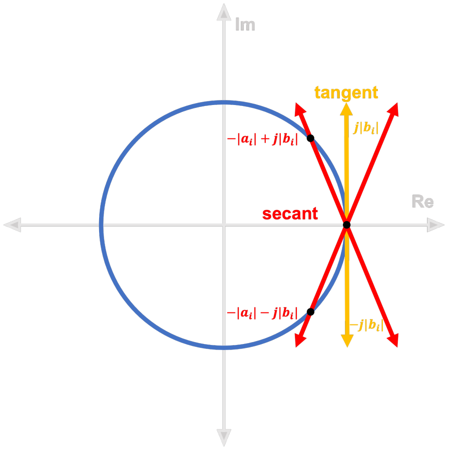

Fig. 8. Tangent lines intersect a circle at a single point. Secant lines intersect a circle at two points.

VI. Conclusion

In this paper, we describe first-order methods to solve distributed saddle point problems of the form: $\min_{\mathbf{x}} \max_{\mathbf{y}} \left\{G(\mathbf{x}) + \langle \mathbf{y}, \overline{P}\mathbf{x}\rangle - H(\mathbf{y})\right\}$, which has many practical applications. In particular, we assume that the underlying data is distributed over a strongly connected network of nodes 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 $\overline{P} := \frac{1}{n}\sum_{i=1}^{n} P_i$, where the constituent functions g_i, h_i, and local coupling matrices P_i are private to each node i. Under appropriate assumptions, we show that **GT-GDA** converges linearly to the unique saddle point of strongly concave-convex problems. We further provide explicit ϵ-complexities of the underlying algorithms and characterize a regime in which the convergence is network-independent. To reduce the communication complexity of **GT-GDA**, we propose a lighter (communication-efficient) version **GT-GDA-Lite** that does not require consensus on local P_i's and analyze **GT-GDA-Lite** under various relevant scenarios. Finally, we illustrate the convergence properties through numerical experiments.

References

[1] E. L. Hall, J. J. Hwang, and F. A. Sadjadi, "Computer Image Processing And Recognition," in *Optics in Metrology and Quality Assurance*, Harvey L. Kasdan, Ed. International Society for Optics and Photonics, 1980, vol. 0220, pp. 2 – 10, SPIE.

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

[3] C. Y. Chen and P. P. Vaidyanathan, "Quadratically constrained beamforming robust against direction-of-arrival mismatch," *IEEE Trans. on Sig. Processing*, vol. 55, no. 8, pp. 4139–4150, 2007.

[4] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[5] 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 NeurIPS*. 2014, vol. 27, Curran Associates, Inc.

[6] A. Sinha, H. Namkoong, and J. Duchi, "Certifiable distributional robustness with principled adversarial training," in *ICLR*, 2018.

[7] T. Lin, C. Jin, and M. I. Jordan, "Near-optimal algorithms for minimax optimization," in *Proceedings of 33rd Conf. on Learning Theory*. 09–12 Jul 2020, vol. 125, pp. 2738–2779, PMLR.

[8] T. Lin, C. Jin, and M. I. Jordan, "On gradient descent ascent for nonconvex-concave minimax problems," in *Proc. of the 37th Int. Conf. on Machine Learning*. 13–18 Jul 2020, vol. 119, pp. 6083–6093, PMLR.

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

[10] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. of Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.

[11] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "Distributed basis pursuit," *IEEE Trans. on Sig. Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

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

[13] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, 2019.

[14] S. S. Du, J. Chen, L. Li, L. Xiao, and D. Zhou, "Stochastic variance reduction methods for policy evaluation," in *ICML*, 2017.

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

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

[17] W. Xian, F. Huang, Y. Zhang, and H. Huang, "A faster decentralized algorithm for nonconvex minimax problems," in *Adv. in NeurIPS*, 2021.

[18] J. V. Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

[19] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory, 2nd Edition*, Society for Industrial and Applied Mathematics, 1998.

[20] A. Mokhtari, A. Ozdaglar, and S. Pattathil, "A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach," 2020.

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

[22] T. Xu, Z. Wang, Y. Liang, and H. V. Poor, "Gradient free minimax optimization: Variance reduction and faster convergence," 2021.

[23] I. Bogunovic, J. Scarlett, S. Jegelka, and V. Cevher, "Adversarially robust optimization with gaussian processes," in *NeurIPS*, 2018.

[24] J.W. Herrmann, "A genetic algorithm for minimax optimization problems," in *Proc. of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999, vol. 2, pp. 1099–1103 Vol. 2.

[25] E. Laskari, K. Parsopoulos, and M. Vrahatis, "Particle swarm optimization for minimax problems," 02 2002, vol. 2, pp. 1576–1581.

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

[27] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. on Auto. Control*, vol. 63, no. 5, pp. 1329–1339, 2017.

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

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

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

[31] M. Hong, Z. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. on Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[32] H. Wai, N. M. Freris, A. Nedić, and A. Scaglione, "SUCAG: stochastic unbiased curvature-aided gradient method for distributed optimization," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 1751–1756.

[33] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Trans. on Info. Th.*, vol. 58, no. 6, 2012.

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

[35] P. D. Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," *IEEE Trans. on Sig. and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[36] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *30th Advances in NeurIPS*, 2017, pp. 5330–5340.

[37] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D^2: Decentralized training over decentralized data," in *Proc. of the 35th Int. Conf. on Machine Learning*, Jul. 2018, vol. 80, pp. 4848–4856.

[38] R. Xin, D. Jakovetić, and U. A. Khan, "Distributed Nesterov gradient methods over arbitrary graphs," *IEEE Sig. Processing Letters*, vol. 26, no. 18, pp. 1247–1251, Jun. 2019.

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

[40] D. Kovalev, A. Beznosikov, A. Sadiev, M. Persiianov, P. Richtárik, and A. Gasnikov, "Optimal algorithms for decentralized stochastic variational inequalities," 2022, arXiv: 2202.02771.

[41] A. Beznosikov, V. Samokhin, and A. Gasnikov, "Distributed saddle-point problems: Lower bounds, near-optimal and robust algorithms," 2020, arXiv: 2010.13112.

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

[43] H. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," in *NeurIPS*, 2018, p. 9672–9683.

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

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

[46] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[47] R. T. Rockafellar, *Convex analysis*, Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.

[48] Y. Nesterov, *Lectures on convex optimization*, vol. 137, Springer, 2018.

[49] T. H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus admm," *IEEE Trans. on Sig. Proc.*, vol. 63, no. 2, pp. 482–497, 2015.

[50] Q. Lü, X. Liao, S. Deng, and H. Li, "A decentralized stochastic algorithm for coupled composite optimization with linear convergence," *IEEE Trans. on Sig. and Inf. Proc. over Networks*, vol. 8, 2022.

[51] P. C. Chen and P. P. Vaidyanathan, "Distributed algorithms for array sig. processing," *IEEE Trans. on Sig. Proc.*, vol. 69, pp. 4607–4622, 2021.

[52] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. on Autom. Con.*, vol. 54, pp. 48, 2009.

[53] S. M. Kakade and S. Shalev-Shwartz, "On the duality of strong convexity and strong smoothness : Learning applications and matrix regularization," 2009.

[54] A. I. Rikos, W. Jiang, T. Charalambous, and K. H. Johansson, "Finite-time distributed optimization with quantized gradient descent," 2022, arXiv: 2211.10855.

[55] M. W. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *ECML*, 2007.

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

[57] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 2nd edition, 2012.

[58] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "Push-saga: A decentralized stochastic algorithm with variance reduction over directed graphs," *IEEE Control Systems Letters*, vol. 6, pp. 1202–1207, 2022.

[59] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "Variance reduced stochastic optimization over directed graphs with row and column stochastic weights," 2022, arXiv: 2202.03346.

[60] B. Polyak, *Introduction to optimization*, Optimization Software, 1987.

[61] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "A decentralized variance-reduced method for stochastic optimization over directed graphs," in *ICASSP*, 2021, pp. 5030–5034.

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

Appendix

A. System Matrix for **GT-GDA**

To completely describe lemma 1, for $c > \frac{2L^2}{\sigma_m^2} + \frac{2\sigma_M^2\kappa}{\sigma_m^2} + 1$, we describe the system matrix as follows

$$M_{\alpha,\beta} := \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

$$M_{11} := \begin{bmatrix} \lambda & 0 & \alpha L \\ \alpha L_1 & 1 - \alpha\frac{\sigma_m^2}{L_2} & 0 \\ \lambda + \alpha\lambda L + \beta\frac{\lambda\sigma_M^2}{L} & \alpha m_1 + \beta m_2 & \lambda + \alpha\lambda L \end{bmatrix},$$

$$M_{12} := \begin{bmatrix} 0 & 0 & 0 \\ 0 & \alpha & 0 \\ \frac{\lambda}{L} + \alpha\lambda + \beta\lambda & \alpha\lambda + \beta\lambda & \beta\lambda \end{bmatrix}\sigma_M,$$

$$M_{21} := \begin{bmatrix} 0 & 0 & 0 \\ \alpha\frac{\sigma_M L_1}{\mu} & \alpha m_3 & 0 \\ \frac{\lambda\sigma_M}{L} + \alpha\lambda\sigma_M + \beta\lambda\sigma_M & \alpha m_4 + \beta m_5 & \alpha\lambda\sigma_M \end{bmatrix},$$

$$M_{22} := \begin{bmatrix} \lambda & 0 & \beta L \\ \alpha\frac{\sigma_M^2}{\mu} + \beta L_2 & 1 - \beta\mu\left(1 - \frac{1}{c}\right) & 0 \\ \lambda + \alpha\frac{\lambda\sigma_M^2}{L} + \beta\lambda L & \alpha\frac{\lambda\sigma_M^2}{L} + \beta\lambda L & \lambda + \beta\lambda L \end{bmatrix},$$

$$m_1 := \lambda m \qquad m_2 := \lambda\frac{\sigma_M^2}{L}\left(1 + \kappa\right), \qquad m_3 := \frac{m\sigma_M}{\mu},$$
$$m_4 := \frac{\lambda m\sigma_M}{L}, \quad m_5 := \lambda\sigma_M\left(1 + \kappa\right), \qquad m := L + \frac{\sigma_M^2}{\mu}.$$

B. System Matrix for **GT-GDA-Lite**: Quadratic case

Let $p := p_x + p_y$, and we define $\widetilde{M}_{\alpha,\beta} \in \mathbb{R}^{(2n+1)p\times(2n+1)p}$ as

$$\widetilde{M}_{\alpha,\beta} := \begin{bmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{bmatrix},$$

$$\widetilde{M}_{11} := \begin{bmatrix} \overline{W}_1 & 0 & -\alpha W_1 \\ -\alpha\frac{(\mathbf{1}_n^\top \otimes I_{p_x})\Lambda_Q}{n} & I_{p_x} - \alpha\overline{Q} & 0 \\ \widetilde{m}_1 & \widetilde{m}_2 & \widetilde{m}_3 \end{bmatrix},$$

$$\widetilde{M}_{12} := \begin{bmatrix} 0 & 0 & 0 \\ -\alpha\frac{(\mathbf{1}_n^\top \otimes I_{p_x})\Lambda_{P^\top}}{n} & -\alpha\overline{P}^\top & 0 \\ \widetilde{m}_4 & \widetilde{m}_5 & \widetilde{m}_6 \end{bmatrix},$$

$$\widetilde{M}_{21} := \begin{bmatrix} 0 & 0 & 0 \\ \beta\frac{(\mathbf{1}_n^\top \otimes I_{p_y})\Lambda_P}{n} & \beta\overline{P} & 0 \\ \overline{m}_1 & \overline{m}_2 & \overline{m}_3 \end{bmatrix},$$

$$\widetilde{M}_{22} := \begin{bmatrix} \overline{W} & 0 & \beta W_2 \\ -\beta\frac{(\mathbf{1}_n^\top \otimes I_{p_y})\Lambda_R}{n} & I_{p_y} - \beta\overline{R} & 0 \\ \overline{m}_4 & \overline{m}_5 & \overline{m}_6 \end{bmatrix},$$

such that for any collection of matrices $\{M_1, M_2, \cdots, M_n\}$, we define $\Lambda_M := \text{diag}([M_1, M_2, \cdots, M_n])$ as the block diagonal matrix where the i-th diagonal element is M_i. The rest of the terms used in $\widetilde{M}_{\alpha,\beta}$ are defined below:

$$\widetilde{m}_1 = \overline{W}_1\left[\Lambda_Q\left((W_1 - I_{np_x}) - \alpha W_1^\infty\Lambda_Q\right) + \beta\Lambda_{P^\top}W_2^\infty\Lambda_P\right],$$
$$\widetilde{m}_2 = \overline{W}_1\left[-\alpha\Lambda_Q W_1^\infty\Lambda_Q + \beta\Lambda_{P^\top}W_2^\infty\Lambda_P\right](\mathbf{1}_n \otimes I_{p_x}),$$
$$\widetilde{m}_4 = \overline{W}_1\left[-\alpha\Lambda_Q W_1^\infty\Lambda_{P^\top} + \Lambda_{P^\top}\left((W_2 - I_{np_y}) - \beta W_2^\infty\Lambda_R\right)\right],$$
$$\widetilde{m}_5 = \overline{W}_1\left[-\alpha\Lambda_Q W_1^\infty\Lambda_{P^\top} - \beta\Lambda_{P^\top}W_2^\infty\Lambda_R\right](\mathbf{1}_n \otimes I_{p_y}),$$
$$\widetilde{m}_3 = \left[\overline{W}_1 - \alpha\Lambda_Q W_1\right], \qquad \widetilde{m}_6 = \overline{W}_1\left[\beta\Lambda_{P^\top}W_2\right],$$
$$\overline{m}_1 = \overline{W}_2\left[-\beta\Lambda_R W_2^\infty\Lambda_P + \Lambda_P\left((W_1 - I_{np_x}) - \alpha W_1^\infty\Lambda_Q\right)\right],$$
$$\overline{m}_2 = \overline{W}_2\left[-\beta\Lambda_R W_2^\infty\Lambda_P - \alpha\Lambda_P W_1^\infty\Lambda_Q\right](\mathbf{1}_n \otimes I_{p_x}),$$
$$\overline{m}_4 = \overline{W}_2\left[-\Lambda_R\left((W_2 - I_{np_y}) - \beta W_2^\infty\Lambda_R\right) - \alpha\Lambda_P W_1^\infty\Lambda_{P^\top}\right],$$
$$\overline{m}_5 = \overline{W}_2\left[\beta\Lambda_R W_2^\infty\Lambda_R - \alpha\Lambda_P W_1^\infty\Lambda_{P^\top}\right](\mathbf{1}_n \otimes I_{p_y}),$$
$$\overline{m}_3 = \overline{W}_2\left[-\alpha\Lambda_P W_1\right], \qquad \overline{m}_6 = \left[\overline{W}_2 - \beta\Lambda_R W_2\right].$$