2021 American Control Conference (ACC)
New Orleans, USA, May 25-28, 2021

Bilevel Distributed Optimization in Directed Networks

Farzad Yousefian[1]

***Abstract*— Motivated by emerging applications in wireless sensor networks and large-scale data processing, we consider distributed optimization over directed networks where the agents communicate their information locally to their neighbors to cooperatively minimize a global cost function. We introduce a new unifying distributed constrained optimization model that is characterized as a bilevel optimization problem. This model captures a wide range of existing problems over directed networks including: (i) Distributed optimization with linear constraints; (ii) Distributed unconstrained nonstrongly convex optimization over directed networks. Employing a novel regularization-based relaxation approach and gradient-tracking schemes, we develop an iteratively regularized push-pull gradient algorithm. We establish the consensus and derive new convergence rate statements for suboptimality and infeasibility of the generated iterates for solving the bilevel model. The proposed algorithm and the complexity analysis obtained in this work appear to be new for addressing the bilevel model and also for the two sub-classes of problems. The numerical performance of the proposed algorithm is presented.**

I. Introduction

We consider a class of bilevel distributed optimization problems in directed networks given as follows:

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^m f_i(x) \quad \text{s.t. } x\in \operatorname*{argmin}_{x\in\mathbb{R}^n} \sum_{i=1}^m g_i(x), \qquad (1)$$

where we make the following assumptions:

Assumption 1: (a) Functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are μ_f–strongly convex and L_f–smooth for all i. (b) Functions $g_i : \mathbb{R}^n \to \mathbb{R}$ are convex and L_g–smooth for all i. (c) The set $\operatorname{argmin}_{x\in\mathbb{R}^n} \sum_{i=1}^m g_i(x)$ is nonempty.

Here, m agents cooperatively seek to find among the optimal solutions to the problem $\min_{x\in\mathbb{R}^n} \sum_{i=1}^m g_i(x)$, one that minimizes a secondary metric, i.e., $\sum_{i=1}^m f_i(x)$. Here, functions f_i and g_i are known locally only by agent i and the cooperation among the agents occurs over a directed network. Given a set of nodes $\mathcal{N}$, a directed graph (digraph) is denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of ordered pairs of vertices. For any edge $(i, j) \in \mathcal{E}$, i and j are called parent node and child node, respectively. Graph $\mathcal{G}$ is called *strongly connected* if there is a path between the pair of any two different vertices. The digraph induced by a given nonnegative matrix $\mathbf{B} \in \mathbb{R}^{m\times m}$ is denoted by $\mathcal{G}_{\mathbf{B}} \triangleq (\mathcal{N}_{\mathbf{B}}, \mathcal{E}_{\mathbf{B}})$, where $\mathcal{N}_{\mathbf{B}} \triangleq [m]$ and $(j, i) \in \mathcal{E}_{\mathbf{B}}$ if and only if $B_{ij} > 0$. We let $\mathcal{N}^{\text{in}}_{\mathbf{B}}(i)$ and $\mathcal{N}^{\text{out}}_{\mathbf{B}}(i)$ denote the set of parents (in-neighbors) and the set of children (out-neighbors) of vertex i, respectively. Also, $\mathcal{R}_{\mathbf{B}}$ denotes the set of roots of all possible spanning trees in $\mathcal{G}_{\mathbf{B}}$.

Problem formulation (1) is often referred to as the "selection problem" and is considered in addressing ill-conditioned optimization problems where $min_{x\in\mathbb{R}^n} \sum_{i=1}^m g_i(x)$ is sensitive to data perturbations. In addition, it can be employed in addressing the following classes of problems.

Special cases of the proposed model: Problem (1) provides a unifying mathematical framework capturing several existing problems in the distributed optimization literature. From these, we present two important cases below:

(i) Distributed linearly constrained optimization in directed networks: Consider the model given as:

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^m f_i(x) \quad \text{s.t.} \quad \begin{array}{l} A_i x = b_i \quad \text{for all } i \in [m], \\ x_j \geq 0 \quad \text{for } j \in \mathcal{J} \subseteq [n], \end{array} \qquad (2)$$

where $A_i \in \mathbb{R}^{m_i\times n}$ and $b_i \in \mathbb{R}^{m_i}$ are known parameters. Let problem (2) be feasible. Then, by defining for $i \in [m]$,

$$g_i(x) := \tfrac{1}{2}\|A_i x - b_i\|_2^2 + \tfrac{1}{2m} \textstyle\sum_{j\in\mathcal{J}} \max\{0, -x_j\}^2, \quad (3)$$

problem (2) is equivalent to (1) (cf. proof of Corollary 1).

(ii) Distributed unconstrained optimization in the absence of strong convexity: Let us define $f_i(x) := \|x\|_2^2/m$. Then, problem (1) is equivalent to finding the least ℓ_2-norm solution of the following canonical distributed unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^n} \textstyle\sum_{i=1}^m g_i(x), \qquad (4)$$

where g_i's are all smooth merely convex (cf. Corollary 2).

Existing theory on distributed optimization in networks: The classical mathematical models, tools, and algorithms for *consensus-based optimization* were introduced and studied as early as the '70s [12] and '80s [34], [35], [5]. Of these, in the seminal work of Tsitsiklis [34], it was assumed the agents share a *global (smooth) objective* while their decision component vectors are distributed *locally* over the network. In the past two decades, in light of the unprecedented growth in data and its imperative role in several broad fields such as social networks, biology, and medicine, the theory of distributed and parallel optimization over networks has much advanced. The distributed optimization problems with *local objective functions* were first studied in [19], [25]. In this framework, the agents communicate their local information with their neighbors in the network at discrete times to cooperatively minimize the global cost function. Without characterizing the communication rules explicitly, this model can be formulated as $\sum_{i=1}^m f_i(x)$ subject to $x \in \mathcal{X}$. Here, the

[1]Assistant Professor in the School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078, USA (email: farzad.yousefian@okstate.edu)

The author gratefully acknowledges the support of the National Science Foundation through CAREER grant ECCS-1944500.

local function f_i is known only to the agent i and $\mathcal{X}$ denotes the system constraint set. This modeling framework captures a wide spectrum of decentralized applications in the areas of statistical learning, signal processing, sensor networks, control, and robotics [9]. Because of this, in the past decade, there has been a flurry of research focused on the design and analysis of fast and scalable computational methods to address applications in networks. Among these, average-based consensus methods are one of the most studied approaches. Here, the network is characterized with a *stochastic matrix* that is possibly time-varying. The underlying idea is that at a given time, each agent uses this matrix and obtains a weighted-average of its neighbors' local variables. Then, the update is completed by performing a standard subgradient step for the agent. Random projection distributed scheme were developed for both synchronous and asynchronous cases, assuming $\mathcal{X} \triangleq \cap_{i=1}^m \mathcal{X}_i$ [16], [32]. For the constrained model where $\mathcal{X}$ is easy-to-project, a dual averaging scheme was developed in [10]. The algorithm EXTRA [30] and its proximal variant were developed addressing $\mathcal{X} = \mathbb{R}^n$. EXTRA is a synchronous and time-invariant scheme and achieves a sublinear and a linear rate of convergence for smooth merely convex and strongly convex problems, respectively. Among many other works such as [18], [13], is the DIGing algorithm [24] which was the first work achieving a linear convergence rate for unconstrained optimization over a time-varying network. When the graph is directed, a key shortcoming in the weighted-based schemes lies in that the double stochasticity requirement of the weight matrix is impractical. Push-sum protocols were first leveraged in [22], [23], [26] to weaken this requirement. Recently, the Push-Pull algorithm equipped with a linear convergence rate was developed in [27] for unconstrained strongly convex problems. Extensions of push-sum algorithms to nonconvex regimes have been developed more recently [29], [20], [33]. Other popular distributed optimization schemes are the dual-based methods, such as ADMM-type methods studied in [6], [37], [36], [17], [31], [3]. Most of these algorithms can address only static and undirected graphs. Moreover, there are only a few works in the literature that can cope with constraints employing primal-dual methods [8], [21], [7], [2].

Research gap and contributions: Despite much advances, the existing models and algorithms for in-network optimization have some shortcomings. For example, the problem is often assumed to be unconstrained, e.g., in Push-DIGing [24] and Push-Pull [27] algorithms that have been recently developed. Further, the complexity analysis in those algorithms is done under the assumption that the objective function is strongly convex. In this work, we aim at addressing these shortcomings through considering the bilevel framework (1). Utilizing a novel regularization-based relaxation approach, we develop a new push-pull gradient algorithm where at each iteration, the information of iteratively regularized gradients is pushed to the neighbors, while the information about the decision variable is pulled from the neighbors. We establish the consensus and derive new convergence rate statements for suboptimality and infeasibility of the generated iterates for solving the bilevel model. The proposed algorithm extends [27] to address a class of bilevel problems. The complexity analysis obtained in this work appears to be new and addresses the aforementioned shortcomings.

Notation: For an integer m, the set $\{1, \ldots, m\}$ is denoted as $[m]$. A vector x is assumed to be a column vector (unless otherwise noted) and x^T denotes its transpose. We use $\|x\|_2$ to denote the Euclidean vector norm of x. A continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be μ_f–strongly convex if and only if its gradient mapping is μ_f–strongly monotone, i.e., $(\nabla f(x) - \nabla f(y))^T (x-y) \geq \mu_f \|x-y\|_2^2$ for any $x, y \in \mathbb{R}^n$. Also, it is said to be L_f–smooth if its gradient mapping is Lipschitz continuous with parameter $L_f > 0$, i.e., for any $x, y \in \mathbb{R}^n$, we have $\|\nabla f(x) - \nabla f(y)\|_2 \leq L_f \|x - y\|_2$. We use the following definitions:

$$\mathbf{x} \triangleq [x_1,\ \ldots,\ x_m]^T, \quad \mathbf{y} \triangleq [y_1,\ \ldots,\ y_m]^T \in \mathbb{R}^{m\times n}$$
$$f(x) \triangleq \textstyle\sum_{i=1}^m f_i(x), \quad \mathbf{f}(\mathbf{x}) \triangleq \textstyle\sum_{i=1}^m f_i(x_i),$$
$$\nabla \mathbf{f}(\mathbf{x}) \triangleq [\nabla f_1(x_1),\ \ldots,\ \nabla f_m(x_m)]^T \in \mathbb{R}^{m\times n}.$$

Analogous definitions apply to functions g and $\mathbf{g}$, and mapping $\nabla \mathbf{g}$. Here, x_i denotes the local copy of the decision vector for agent i and $\mathbf{x}$ includes the local copies of all agents. Vector y_i denotes the auxiliary variable for agent i to track the average of regularized gradient mappings. Throughout, we use the following definition of a matrix norm: Given an arbitrary vector norm $\|\cdot\|$, the induced norm of a matrix $W \in \mathbb{R}^{m\times n}$ is defined as $\|\mathbf{W}\| \triangleq \|[\|\mathbf{W}_{\bullet 1}\|, \ldots, \|\mathbf{W}_{\bullet n}\|]\|_2$.

Remark 1: Under the above definition of matrix norm, it can be seen we have $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$ for any $\mathbf{A} \in \mathbb{R}^{m\times m}$ and $\mathbf{x} \in \mathbb{R}^{m\times p}$. Also, for any $a \in \mathbb{R}^{m\times 1}$ and $x \in \mathbb{R}^{1\times n}$, we have $\|ax\| = \|a\|\|x\|_2$.

II. Algorithm outline

To solve the model (1) in directed networks, due to the presence of the inner-level optimization constraints, Lagrangian duality does not seem applicable. Overcoming this challenge calls for new relaxation rules that can tackle the inner-level constraints. We consider a regularization-based relaxation rule. To this end, motivated by the recent success of so-called *iteratively regularized (IR)* algorithms in centralized regimes [40], [39], [14], [1], [15], we develop Algorithm 1. Core to the IR framework is the philosophy that the regularization parameter λ_k is updated after every step within the algorithm. Here, each agent holds a local copy of the global variable x, denoted by $x_{i,k}$, and an auxiliary variable $y_{i,k}$ is used to track the average of a regularized gradient. At each iteration, each agent i uses the ith row of two matrices $\mathbf{R} = [R_{ij}] \in \mathbb{R}^{m\times m}$ and $\mathbf{C} = [C_{ij}] \in \mathbb{R}^{m\times m}$ to update vectors $x_{i,k}$ and $y_{i,k}$, respectively. Below, we state the main assumptions on the these two *weight mixing* matrices.

Assumption 2: (a) The matrix $\mathbf{R}$ is nonnegative, with a strictly positive diagonal, and is row-stochastic, i.e., $\mathbf{R}\mathbf{1} = \mathbf{1}$. (b) The matrix $\mathbf{C}$ is nonnegative, with a strictly positive diagonal, and is column-stochastic, i.e., $\mathbf{1}^T\mathbf{C} = \mathbf{1}^T$. (c) The induced digraphs $\mathcal{G}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{C}^T}$ satisfy $\mathcal{R}_{\mathbf{R}} \cap \mathcal{R}_{\mathbf{C}^T} \neq \emptyset$.

Assumption 2 does not require the strong condition of a doubly stochastic matrix for communication in a directed

Algorithm 1 Iteratively Regularized Push-Pull

1: **Input:** For all $i \in [m]$, agent i sets step-size $\gamma_{i,0} \geq 0$, pulling weights $R_{ij} \geq 0$ for all $j \in \mathcal{N}_{\mathbf{R}}^{\text{in}}(i)$, pushing weights $C_{ij} \geq 0$ for all $j \in \mathcal{N}_{\mathbf{C}}^{\text{out}}(i)$, an arbitrary initial point $x_{i,0} \in \mathbb{R}^n$ and $y_{i,0} := \nabla g_i(x_{i,0}) + \lambda_0 \nabla f_i(x_{i,0})$;

2: **for** $k = 0, 1, \ldots$ **do**

3: For all $i \in [m]$, agent i receives (pulls) the vector $x_{j,k} - \gamma_{j,k} y_{j,k}$ from each agent $j \in \mathcal{N}_{\mathbf{R}}^{\text{in}}(i)$, sends (pushes) $C_{\ell i} y_{i,k}$ to each agent $\ell \in \mathcal{N}_{\mathbf{C}}^{\text{out}}(i)$, and does the following updates:
$x_{i,k+1} := \sum_{j=1}^{m} R_{ij} \left(x_{j,k} - \gamma_{j,k} y_{j,k}\right)$
$y_{i,k+1} := \sum_{j=1}^{m} C_{ij} y_{j,k} + \nabla g_i(x_{i,k+1})$
$+ \lambda_{k+1} \nabla f_i(x_{i,k+1}) - \nabla g_i(x_{i,k}) - \lambda_k \nabla f_i(x_{i,k})$;

4: **end for**

network. In turn, utilizing a push-pull protocol and in a similar fashion to the recent work [27], it only entails a row stochastic $\mathbf{R}$ and a column stochastic matrix $\mathbf{C}$. An example is as follows where agent i chooses scalars $r_i, c_i > 0$ and sets $R_{i,j} := 1/\left(\left|\mathcal{N}_{\mathbf{R}}^{\text{in}}(i)\right| + r_i\right)$ for $j \in \mathcal{N}_{\mathbf{R}}^{\text{in}}(i)$, $R_{i,i} := r_i/\left(\left|\mathcal{N}_{\mathbf{R}}^{\text{in}}(i)\right| + r_i\right)$, $C_{\ell,i} := 1/\left(|\mathcal{N}_{\mathbf{C}}^{\text{out}}(i)| + c_i\right)$ for $\ell \in \mathcal{N}_{\mathbf{C}}^{\text{out}}(i)$, $C_{i,i} := c_i/\left(|\mathcal{N}_{\mathbf{C}}^{\text{out}}(i)| + c_i\right)$, and 0 otherwise. Note that Assumption 2(c) is weaker than imposing strong connectivity on $\mathcal{G}_{\mathbf{R}}$ and $\mathcal{G}_{\mathbf{C}}$. The update rules in Algorithm 1 can be compactly represented as the following:

$$\mathbf{x}_{k+1} := \mathbf{R}\left(\mathbf{x}_k - \boldsymbol{\gamma}_k \mathbf{y}_k\right), \tag{5}$$

$$\mathbf{y}_{k+1} := \mathbf{C}\mathbf{y}_k + \nabla \mathbf{g}(\mathbf{x}_{k+1}) + \lambda_{k+1} \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{g}(\mathbf{x}_k) - \lambda_k \nabla \mathbf{f}(\mathbf{x}_k), \tag{6}$$

where $\boldsymbol{\gamma}_k \geq 0$ is defined as $\boldsymbol{\gamma}_k \triangleq \text{diag}\left(\gamma_{1,k}, \ldots, \gamma_{m,k}\right)$.

III. Preliminaries of convergence analysis

Under Assumption 2, there exists a unique nonnegative left eigenvector $u \in \mathbb{R}^m$ such that $u^T \mathbf{R} = u^T$ and $u^T \mathbf{1} = m$. Similarly, there exists a unique nonnegative right eigenvector $v \in \mathbb{R}^m$ such that $\mathbf{C}v = v$ and $\mathbf{1}^T v = m$ (cf. Lemma 1 in [27]). Throughout, we use the following definitions

Definition 1: For $k \geq 0$ and the regularization parameter $\lambda_k > 0$, let $x^* \triangleq \text{argmin}_{x \in \text{argmin}\, g(x)}\{f(x)\} \in \mathbb{R}^{1\times n}$, $x_{\lambda_k}^* \triangleq \text{argmin}_{x \in \mathbb{R}^n}\left\{g(x) + \lambda_k f(x)\right\} \in \mathbb{R}^{1\times n}$. We define the mapping $\mathbf{G}_k(\mathbf{x}) \triangleq \nabla \mathbf{g}\left(\mathbf{x}\right) + \lambda_k \nabla \mathbf{f}\left(\mathbf{x}\right) \in \mathbb{R}^{m\times n}$, and functions $G_k(\mathbf{x}) \triangleq \frac{1}{m}\mathbf{1}^T \mathbf{G}_k(\mathbf{x}) \in \mathbb{R}^{1\times n}$, $\mathcal{G}_k(x) \triangleq G_k\left(\mathbf{1}x^T\right) \in \mathbb{R}^{1\times n}$, $\bar{g}_k \triangleq \mathcal{G}_k(\bar{x}_k) \in \mathbb{R}^{1\times n}$ where $\bar{x}_k \triangleq \frac{1}{m} u^T \mathbf{x}_k \in \mathbb{R}^{1\times n}$. We let $L_k \triangleq L_g + \lambda_k L_f$ and $\bar{y}_k \triangleq \frac{1}{m}\mathbf{1}^T \mathbf{y}_k \in \mathbb{R}^{1\times n}$. Lastly, we define $\Lambda_k \triangleq \left|1 - \frac{\lambda_{k+1}}{\lambda_k}\right|$.

Here, x^* denotes the optimal solution of problem (1) and x_λ^* is defined as the optimal solution to a regularized problem. Note that the strong convexity of $g(x) + \lambda_k f(x)$ implies that $x_{\lambda_k}^*$ exists and is a unique vector (cf. Proposition 1.1.2 in [4]). Also, under Assumption 1, the set $\text{argmin}\, g(x)$ is closed and convex. As such, from the strong convexity of f and invoking Proposition 1.1.2 in [4] again, we conclude that x^* also exists and is a unique vector. The sequence $\{x_{\lambda_k}^*\}$ is the so-called *Tikhonov trajectory* and plays a key role in the convergence analysis (cf. [11]). The mapping $\mathbf{G}_k(\mathbf{x})$ denotes the regularized gradient matrix. The vector $\bar{x}_k$ holds a weighted average of the local copies of the agent's iterates. Next, we consider a family of update rules for the sequences of the step-size and the regularization parameter under which the convergence and rate analysis can be performed.

Assumption 3 (Update rules): Assume the step-size $\boldsymbol{\gamma}_k$ and the regularization parameter λ_k are updated satisfying: $\hat{\gamma}_k := \frac{\hat{\gamma}_0}{(k+1)^a}$ and $\lambda_k := \frac{\lambda_0}{(k+1)^b}$ where $\hat{\gamma}_k \triangleq \max_{j\in[m]} \gamma_{j,k}$ for $k \geq 0$, and a and b satisfy the following conditions: $0 < b < a < 1$ and $a + b < 1$. Also, let $\alpha_k \geq \theta \hat{\gamma}_k$ for $k \geq 0$ for some $\theta > 0$, where $\alpha_k \triangleq \frac{1}{m} u^T \boldsymbol{\gamma}_k \nu$.

The constant θ in Assumption 3 measures the size of the range within which the agents in $\mathcal{R}_{\mathbf{R}} \cap \mathcal{R}_{\mathbf{C}^T}$ select their stepsizes. The condition $\alpha_k \geq \theta \hat{\gamma}_k$ is satisfied in many cases including the case where all the agents choose strictly positive stepsizes (see Remark 4 in [27] for more details). In the following lemma, we list some of the main properties of the update rules in Assumption 3 that will be used in the analysis.

Lemma 1 (Properties of the update rules): Under Assumption 3, we have: $\{\lambda_k\}_{k=0}^{\infty}$ is a decreasing strictly positive sequence satisfying $\lambda_k \to 0$, $\frac{\Lambda_k}{\lambda_k} \to 0$, $\Lambda_{k+1} \leq \Lambda_k$ for all $k \geq 0$, $\Lambda_{k-1} \leq \frac{1}{k+1}$ for $k \geq 1$, where Λ_k is given by Def. 1. Also, $\{\hat{\gamma}_k\}_{k=0}^{\infty}$ is a decreasing strictly positive sequence such that $\hat{\gamma}_k \to 0$ and $\frac{\hat{\gamma}_k}{\lambda_k} \to 0$. Moreover, for any scalar $\tau > 0$, there exists an integer K_τ such that $\frac{(k+1)\hat{\gamma}_k \lambda_k}{k\hat{\gamma}_{k-1}\lambda_{k-1}} \leq 1 + \tau \hat{\gamma}_k \lambda_k$ for all $k \geq K_\tau$.

Remark 2: The proofs of Lemmas 1, 2, and 3 are omitted to utilize the space. However, for completeness, these proofs are provided in an extended version of this paper in [38].

Next, we present some key properties of the regularized sequence $\{x_{\lambda_k}^*\}$ that will be used in the rate analysis.

Lemma 2 (Properties of Tikhonov trajectory): Let Assumptions 1 and 3 hold and $x_{\lambda_k}^*$ be given by Def. 1. Then, we have: (i) The sequence $\{x_{\lambda_k}^*\}$ converges to the unique solution of problem (1), i.e., x^*. (ii) There exists a scalar $M > 0$ such that for any $k \geq 1$, we have $\left\|x_{\lambda_k}^* - x_{\lambda_{k-1}}^*\right\|_2 \leq \frac{M}{\mu_f}\Lambda_{k-1}$.

In the following, we state the properties of the regularized maps to be used in finding error bounds in the next section.

Lemma 3: Consider Algorithm 1. Let Assumptions 1 and 2 hold. For any $k \geq 0$, mappings G_k, $\mathcal{G}_k$, and $\bar{g}_k$ given by Def. 1 satisfy the following relations: (i) We have that $\bar{y}_k = G_k(\mathbf{x}_k)$. (ii) We have $\mathcal{G}_k\left(x_{\lambda_k}^*\right) = 0$. (iii) The mapping $\mathcal{G}_k(x)$ is $(\lambda_k \mu_f)$-strongly monotone and Lipschitz continuous with parameter L_k. (iv) We have $\|\bar{y}_k - \bar{g}_k\|_2 \leq \frac{L_k}{\sqrt{m}}\|\mathbf{x}_k - \mathbf{1}\bar{x}_k\|_2$ and $\|\bar{g}_k\|_2 \leq L_k \|\bar{x}_k - x_{\lambda_k}^*\|_2$.

We state the following result from [27] introducing two matrix norms induced by matrices $\mathbf{R}$ and $\mathbf{C}$.

Lemma 4 (cf. Lemma 4 and Lemma 6 in [27]): Let Assumption 2 hold. Then: (i) There exist matrix norms $\|\cdot\|_{\mathbf{R}}$ and $\|\cdot\|_{\mathbf{C}}$ such that for $\sigma_{\mathbf{R}} \triangleq \left\|\mathbf{R} - \frac{\mathbf{1}u^T}{m}\right\|_{\mathbf{R}}$ and $\sigma_{\mathbf{C}} \triangleq \left\|\mathbf{C} - \frac{\mathbf{1}v^T}{m}\right\|_{\mathbf{C}}$ we have that $\sigma_{\mathbf{R}} < 1$ and $\sigma_{\mathbf{C}} < 1$. (ii) There exist scalars $\delta_{\mathbf{R},2}, \delta_{\mathbf{C},2}, \delta_{\mathbf{R},\mathbf{C}}, \delta_{\mathbf{C},\mathbf{R}} > 0$ such that for any $W \in \mathbb{R}^{m\times n}$, we have $\|W\|_{\mathbf{R}} \leq \delta_{\mathbf{R},2}\|W\|_2$,

$\|W\|_{\mathbf{C}} \leq \delta_{\mathbf{C},2}\|W\|_2$, $\|W\|_{\mathbf{R}} \leq \delta_{\mathbf{R},\mathbf{C}}\|W\|_{\mathbf{C}}$, $\|W\|_{\mathbf{C}} \leq \delta_{\mathbf{C},\mathbf{R}}\|W\|_{\mathbf{R}}$, $\|W\|_2 \leq \|W\|_{\mathbf{R}}$, and $\|W\|_2 \leq \|W\|_{\mathbf{C}}$.

IV. Convergence and rate analysis

We analyze the convergence of Algorithm 1 by introducing the errors metrics $\left\|\bar{x}_{k+1} - x^*_{\lambda_k}\right\|_2$, $\|\mathbf{x}_{k+1} - \mathbf{1}\bar{x}_{k+1}\|_{\mathbf{R}}$, $\|\mathbf{y}_{k+1} - \nu\bar{y}_{k+1}\|_{\mathbf{C}}$. Of these, the first term relates the averaged iterate with the Tikhonov trajectory, the second term measures the consensus violation for the decision matrix, and the third term measures the consensus violation for the matrix of the regularized gradients. For $k \geq 1$, let us define Δ_k as $\Delta_k \triangleq \left[\|\bar{x}_k - x^*_{\lambda_{k-1}}\|_2, \|\mathbf{x}_k - \mathbf{1}\bar{x}_k\|_{\mathbf{R}}, \|\mathbf{y}_k - \nu\bar{y}_k\|_{\mathbf{C}}\right]^T$. We begin by deriving a system of recursive relations for the three error terms provided below. The proof is provided in the extended version of this paper [38].

Proposition 1: Consider Algorithm 1 under Assumptions 1, 2, and 3. Let α_k and $\hat{\gamma}_k$ be given by Assumption 3, and $c_0 \triangleq \delta_{\mathbf{C},2}\left\|\mathbf{I} - \frac{1}{m}\nu\mathbf{1}^T\right\|_{\mathbf{C}}$. Then, there exist scalars $M > 0$, $B_{\mathbf{g}} > 0$, and an integer K such that for any $k \geq K$, we have $\Delta_{k+1} \leq H_k\Delta_k + h_k$ where $H_k = [H_{ij,k}]_{3\times 3}$ and $h_k = [h_{i,k}]_{3\times 1}$ are given as follows:

$$H_{11,k} := 1 - \mu_f\alpha_k\lambda_k,\ H_{12,k} := \tfrac{\alpha_k L_k}{\sqrt{m}},\ H_{13,k} := \tfrac{\hat{\gamma}_k\|u\|_2}{m},$$
$$H_{21,k} := \sigma_{\mathbf{R}}\hat{\gamma}_k L_k\|\nu\|_{\mathbf{R}},\ H_{22,k} := \sigma_{\mathbf{R}}\left(1 + \hat{\gamma}_k\|\nu\|_{\mathbf{R}}\tfrac{L_k}{\sqrt{m}}\right),$$
$$H_{23,k} := \sigma_{\mathbf{R}}\hat{\gamma}_k\delta_{\mathbf{R},\mathbf{C}},\ H_{33,k} := \sigma_{\mathbf{C}} + c_0 L_k\hat{\gamma}_k\|\mathbf{R}\|_2,$$
$$H_{31,k} := c_0 L_k\left(\hat{\gamma}_k\|\mathbf{R}\|_2\|\nu\|_2 L_k + 2\sqrt{m}\Lambda_k\right),$$
$$H_{32,k} := c_0 L_k\left(\|\mathbf{R} - \mathbf{I}\|_2 + \hat{\gamma}_k\|\mathbf{R}\|\|\nu\|_2\tfrac{L_k}{\sqrt{m}} + 2\Lambda_k\right),$$
$$h_{1,k} := \tfrac{M\Lambda_{k-1}}{\mu_f},\ h_{2,k} := \tfrac{M\sigma_{\mathbf{R}}\hat{\gamma}_k L_k\|\nu\|_{\mathbf{R}}}{\mu_f}\Lambda_{k-1},$$
$$h_{3,k} := c_0 L_k\left(\hat{\gamma}_k\|\mathbf{R}\|_2\|\nu\|_2 L_k + \sqrt{m}\Lambda_k + \tfrac{\mu_f c_0 B_{\mathbf{g}}}{M}\right)\tfrac{M\Lambda_{k-1}}{\mu_f}.$$

Next, we derive a unifying recursive bound for the three error terms introduced earlier. The proof is provided in the extended version of this paper in [38].

Proposition 2: Consider Algorithm 1. Let Assumptions 1, 2, and 3 hold. Then, there exists an integer $\mathcal{K} \geq 1$ such that for any $k \geq \mathcal{K}$, the following holds:

(a) $\|\Delta_{k+1}\|_2 \leq (1 - 0.5\mu_f\alpha_k\lambda_k)\|\Delta_k\|_2 + \Theta\Lambda_{k-1}$, where

$$\Theta \triangleq \max\{1, \sigma_{\mathbf{R}}\hat{\gamma}_0 L_0\|\nu\|_{\mathbf{R}}, c_0 L_0\left(\hat{\gamma}_0\|\mathbf{R}\|_2\|\nu\|_2 L_0 + \sqrt{m}\Lambda_0 + \mu_f c_0 B_{\mathbf{g}}/M\right)\}\sqrt{3}M/\mu_f.$$

(b) There exists a scalar $\mathcal{B} > 0$ such that $\|\Delta_k\|_2 \leq \frac{\mathcal{B}}{k^{1-a-b}}$.

Our first main result is provided below where we derive a family of convergence rates for the bilevel formulation (1).

Theorem 1 (Rate statements for the bilevel model): Consider problem (1) and Algorithm 1. Let Assumptions 1, 2, and 3 hold. Then, we have the following results:

(a) We have $\lim_{k\to\infty}\bar{x}_k = x^*$. Also, the consensus violation of $\mathbf{x}_k$ and $\mathbf{y}_k$ characterized by $\|\mathbf{x}_{k+1} - \mathbf{1}\bar{x}_{k+1}\|_{\mathbf{R}}$ and $\|\mathbf{y}_{k+1} - \nu\bar{y}_{k+1}\|_{\mathbf{C}}$, respectively, are both bounded by $\mathcal{O}\left(1/k^{1-a-b}\right)$ for any sufficiently large k.

(b) We have $f(\bar{x}_k) - f(x^*) \leq \frac{\mathcal{Q}_1(L_g+\lambda_0 L_f)}{2}\frac{1}{k^{2-2a-3b}}$ for some $\mathcal{Q}_1 > 0$ and any sufficiently large k.

(c) $g(\bar{x}_k) - g(x^*) \leq \frac{\mathcal{Q}_2(L_g+\lambda_0 L_f)}{2}\frac{1}{k^{2-2a-2b}} + \frac{\lambda_0\mathcal{Q}_3}{k^b}$ for $\mathcal{Q}_2, \mathcal{Q}_3 > 0$ and any sufficiently large k.

Proof: **(a)** From Lemma 2(a), we have that $\{x^*_{\lambda_k}\}$ converges to x^*. Moreover, from Proposition 2(b), we have that $\|\bar{x}_k - x^*_{\lambda_{k-1}}\|_2$ converges to zero. Therefore, we have $\lim_{k\to\infty}\bar{x}_k = x^*$. To derive the bounds for $\|\mathbf{x}_k - \mathbf{1}\bar{x}_k\|_{\mathbf{R}}$ and $\|\mathbf{y}_k - \nu\bar{y}_k\|_{\mathbf{C}}$, from the definition of Δ_k in Proposition 2, we can write: $\|\mathbf{x}_k - \mathbf{1}\bar{x}_k\|_{\mathbf{R}} \leq \|\Delta_k\|_2 = \mathcal{O}\left(k^{1-a-b}\right)$. Similarly, we obtain $\|\mathbf{y}_k - \nu\bar{y}_k\|_{\mathbf{C}} = \mathcal{O}\left(k^{1-a-b}\right)$.

(b) Consider the regularized function $g(x) + \lambda_k f(x)$. Note that it is L_k-smooth, where $L_k \triangleq L_g + \lambda_k L_f$. Since $x^*_{\lambda_k}$ is the minimizer of $g(x) + \lambda_k f(x)$, we have $x \in \mathbb{R}^n$:

$$g(x) + \lambda_k f(x) - g\left(x^*_{\lambda_k}\right) - \lambda_k f\left(x^*_{\lambda_k}\right) \leq \tfrac{L_k}{2}\left\|x - x^*_{\lambda_k}\right\|_2^2.$$

Also, we can write that $g\left(x^*_{\lambda_k}\right) + \lambda_k f\left(x^*_{\lambda_k}\right) \leq g(x^*) + \lambda_k f(x^*)$. Combining the preceding two relations and substituting x by $\bar{x}_{k+1}$, we obtain $g(\bar{x}_{k+1}) - g(x^*) + \lambda_k\left(f(\bar{x}_{k+1}) - f(x^*)\right) \leq \frac{L_k}{2}\left\|\bar{x}_{k+1} - x^*_{\lambda_k}\right\|_2^2$. Applying the bound from Proposition 2(b), we obtain:

$$g(\bar{x}_{k+1}) - g(x^*) + \lambda_k\left(f(\bar{x}_{k+1}) - f(x^*)\right) \leq \tfrac{L_k\mathcal{B}^2}{2(k+1)^{2-2a-2b}} \quad \text{for all } k \geq \mathcal{K}. \tag{7}$$

Note that from the definition of x^* in Def. 1, we have $g(\bar{x}_{k+1}) - g(x^*) \geq 0$. This implies that for all $k \geq \mathcal{K}$:

$$f(\bar{x}_{k+1}) - f(x^*) \leq \left(\tfrac{L_0\mathcal{B}^2}{2\lambda_0}\right)\tfrac{1}{(k+1)^{2-2a-3b}}.$$

Therefore, the desired relation holds for $\mathcal{Q}_1 \triangleq \frac{\mathcal{B}^2}{\lambda_0}$.

(c) From part (a), we know that $\{\bar{x}_k\}$ converges to x^*. This result and that f is a continuous function imply that there exists a scalar $\mathcal{Q}_3 > 0$ such that $|f(\bar{x}_{k+1}) - f(x^*)| \leq \mathcal{Q}_3$. Thus, from the inequality (7) and the update rule for λ_k:

$$g(\bar{x}_{k+1}) - g(x^*) \leq \left(\tfrac{L_0\mathcal{B}^2}{2}\right)\tfrac{1}{(k+1)^{2-2a-2b}} + \tfrac{\mathcal{Q}_3\lambda_0}{(k+1)^b},$$

for all $k \geq \mathcal{K}$. Therefore, the desired relation holds. ■

In the following, we present the implications of the results of Theorem 1 in solving the constrained problem (2).

Corollary 1 (Rates for the linearly constrained model): Consider problem (2) and Algorithm 1 where $g_i(x)$ is defined by (3). Let the feasible set be nonempty and Assumption 1(a) and Assumption 2 hold. Suppose Assumption 3 holds with $a := 0.2$ and $b := 0.2 - \epsilon/3$ where $\epsilon > 0$ is a sufficiently small scalar. Then, we have $\lim_{k\to\infty}\bar{x}_k = x^*$ and for any sufficiently large k:

(a) We have $\|\mathbf{x}_{k+1} - \mathbf{1}\bar{x}_{k+1}\|_{\mathbf{R}} = \mathcal{O}\left(1/k^{0.6+\epsilon/3}\right)$, and $\|\mathbf{y}_{k+1} - \nu\bar{y}_{k+1}\|_{\mathbf{C}} = \mathcal{O}\left(1/k^{0.6+\epsilon/3}\right)$.

(b) We have $f(\bar{x}_k) - f(x^*) = \mathcal{O}\left(1/k^{1-\epsilon}\right)$.

(c) We have $\|\mathbf{A}\bar{x}_k - \mathbf{b}\|_2^2 = \mathcal{O}\left(1/k^{0.2-\epsilon/3}\right)$ where $\mathbf{A} \triangleq \left[A_1^T, \ldots, A_m^T\right]^T$ and $\mathbf{b} \triangleq \left[b_1^T, \ldots, b_m^T\right]$.

Proof: First, we show that problem (2) is equivalent to problem (1). Let X_1 and X_2 denote the feasible set of problem (1) and (2), respectively. Suppose $\hat{x} \in X_1$ is an arbitrary vector. Thus, we have $\hat{x} \in \operatorname{argmin}_{x\in\mathbb{R}^n}\frac{1}{2}\sum_{i=1}^m\|A_i x - b_i\|_2^2 + \frac{1}{2m}\sum_{j\in\mathcal{J}}\max\{0, -x_j\}^2$. From the assumption $X_2 \neq \emptyset$, there exists a point $\bar{x}$ satisfying $\mathbf{A}\bar{x} = \mathbf{b}$ and $\bar{x}_j \geq 0$ for all $j \in \mathcal{J}$. This implies that the minimum of the

function $\frac{1}{2}\sum_{i=1}^m \|A_i x - b_i\|_2^2 + \frac{1}{2m}\sum_{j\in\mathcal{J}} \max\{0,-x_j\}^2$ is zero. Therefore, $\hat{x}$ must satisfy $\mathbf{A}x = \mathbf{b}$ and $x_j \geq 0$ for all $j \in \mathcal{J}$, implying that $\hat{x} \in X_2$. Next, suppose $\hat{x} \in X_2$ is an arbitrary vector. Thus, we have $\frac{1}{2}\sum_{i=1}^m \|A_i\hat{x} - b_i\|_2^2 + \frac{1}{2m}\sum_{j\in\mathcal{J}} \max\{0,-\hat{x}_j\}^2 = 0$ implying that $\hat{x}$ is a minimizer of the $\frac{1}{2}\sum_{i=1}^m \|A_i x - b_i\|_2^2 + \frac{1}{2m}\sum_{j\in\mathcal{J}} \max\{0,-x_j\}^2$. Therefore, we have $\hat{x} \in X_1$. We conclude that $X_1 = X_2$ and thus problems (1) and (2) are equivalent. Next, we show that Assumption 1(b) is satisfied. From the definition of function g_i by (3), it is not hard to show that $\nabla g_i(x)$ indeed exists and $\nabla g_i(x) = A_i^T(A_i x - b_i) - \frac{1}{m}\sum_{j\in\mathcal{J}} \max\{0,-x_j\}\mathbf{e}_j$. Note that the mapping $A_i^T(A_i x - b_i)$ is Lipschitz with parameter $\rho\left(A_i^T A_i\right)$ denoting the spectral norm of $A_i^T A_i$. Also, it can be shown that the mapping $-\frac{1}{m}\sum_{j\in\mathcal{J}} \max\{0,-x_j\}\mathbf{e}_j$ is Lipschitz with parameter $\frac{1}{\sqrt{m}}$ (proof omitted). Thus, we conclude that Assumption 1(b) is met for $L_g \triangleq \max_{i\in[m]} \rho\left(A_i^T A_i\right) + \frac{1}{\sqrt{m}}$. Therefore, all conditions of Theorem 1 hold. To obtain the rate results in part (a), (b), (c), it suffices to substitute a by 0.2 and b by $0.2 - \frac{\epsilon}{3}$ in the corresponding parts in Theorem 1. ■

Lastly, we present the implications of the results of Theorem 1 in addressing the absence of strong convexity. The proof is in [38].

Corollary 2 (Rates for problem (4)*):* Consider problem (4) and Algorithm 1 where we set $f_i(x) := \|x\|_2^2/m$. Let Assumption 1(b), 1(c) and Assumption 2 hold. Suppose Assumption 3 holds with $a := 0.4$ and $b := 0.4 - \epsilon$ where $\epsilon > 0$ is a sufficiently small scalar. Let $x^*_{\ell_2}$ denote the least ℓ_2-norm optimal solution of problem (4). Then, for any sufficiently large k:

(a) We have $\|\mathbf{x}_{k+1} - \mathbf{1}\bar{x}_{k+1}\|_{\mathbf{R}} = \mathcal{O}\left(1/k^{0.2+\epsilon}\right)$ and $\|\mathbf{y}_{k+1} - \nu\bar{y}_{k+1}\|_{\mathbf{C}} = \mathcal{O}\left(1/k^{0.2+\epsilon}\right)$.

(b) We have $g(\bar{x}_k) - g\left(x^*_{\ell_2}\right) = \mathcal{O}\left(1/k^{0.4-\epsilon}\right)$ and that $\|\bar{x}_k - x^*_{\ell_2}\|_2^2 = \mathcal{O}\left(1/k^{3\epsilon}\right)$.

V. Numerical Results

(1) Distributed sensor network problems: We first compare Algorithm 1 with the Push-Pull algorithm [27] in a sensor network example. We consider the unconstrained ill-posed problem $\min_{x\in\mathbb{R}^n}\sum_{k=1}^m \|z_i - H_i x\|_2^2$, where $H_i \in \mathbb{R}^{d\times n}$ and $z_i \in \mathbb{R}^d$ denote the measurement matrix and the noisy observation of the i^{th} sensor. Due to the challenges raised by ill-conditioning and also the lack of convergence and rate guarantees, Push-Pull algorithm needs to be applied to a regularized variant of the problem. To this end, in the implementation of the Push-Pull scheme, we use an ℓ_2 regularizer with a parameter 0.1. Accordingly, in Algorithm 1, we set $\lambda_0 := 0.1$. We employ the tuning rules according to Corollary 2, while a constant step-size is used for the Push-Pull method. We generate H_i and z_i randomly and choose $m = 10$, $n = 20$, and $d = 1$. We generate matrices $\mathbf{R}$ and $\mathbf{C}$ from the same underlying graph with two different directed graphs (see Figure 1). We use $\mathbf{R} = \mathbf{I} - \frac{1}{2d_{in}^{\max}}\mathbf{L}_{\mathbf{R}}$ where $\mathbf{L}_{\mathbf{R}}$ denotes the Laplacian matrix and $d_{in}^{\max}$ denotes the maximum in-degree. We use the same formula for $\mathbf{C}$ using maximum out-degree.

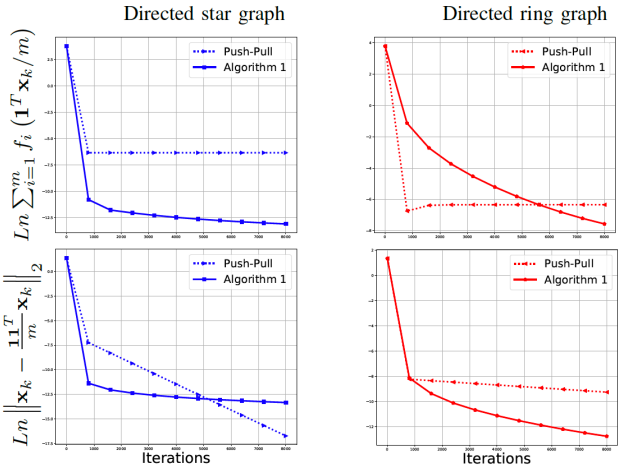

Fig. 1: Algorithm 1 vs. regularized Push-Pull algorithm under different choices of $\mathbf{R}$ and $\mathbf{C}$.

Insights: Figure 1 shows the comparison of the two schemes. We compare objective function values and consensus violations. For the latter, we use the term $\left\|\mathbf{x}_k - \frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{x}_k\right\|_2$. In terms of the objective function value, Algorithm 1 performs significantly better both cases.

(2) Distributed ill-conditioned linear inverse problems: Here $g(x) := \sum_{i=1}^m \|A_i x - b_i\|_2^2$ and $f(x) := \frac{1}{2}\|x\|_2^2$, where $A_i \in \mathbb{R}^{d\times n}$ and $b_i \in \mathbb{R}^d$ denote the locally known i^{th} block of the Toeplitz blurring operator and the given blurred image, respectively. Figure 2 shows the progress of deblurring across the 9 agents over a directed ring graph.

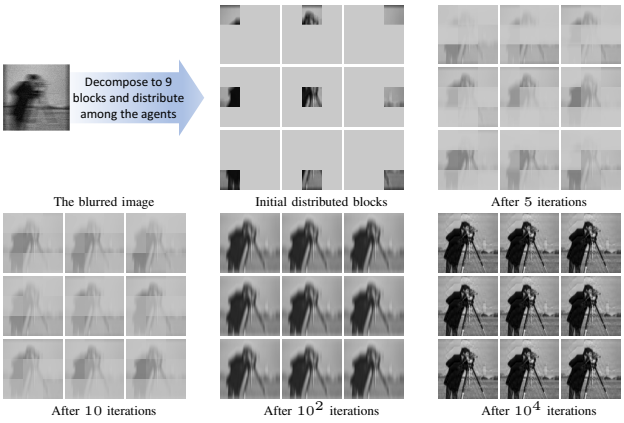

Fig. 2: Performance of IR-PushPull in distributed image deblurring using 9 agents over a ring digraph

(3) Distributed linear SVM: Consider a linear SVM where $\mathcal{D} \triangleq \{(u_\ell, v_\ell) \in \mathbb{R}^n \times \{-1,+1\} \mid \ell \in \mathcal{S}\}$ denotes the data set and $\mathcal{S} \triangleq \{1,\ldots,s\}$ denotes the index set. Let $\mathcal{S}$ be partitioned into $\mathcal{S}_{\text{train}}$ and $\mathcal{S}_{\text{test}}$ randomly. Let $\mathcal{S}_i$ denote the data locally known by agent i where $\cup_{i=1}^m \mathcal{S}_i = \mathcal{S}_{\text{train}}$. Consider the following primal SVM model:

$$\min_{x,b,z} \sum_{i=1}^m \left(\frac{\eta}{2m}\|x\|_2^2 + \sum_{\ell\in\mathcal{S}_i} z_\ell\right) \text{ s.t.} \begin{array}{l} v_\ell\left(x^T u_\ell + b\right) \geq 1 - z_\ell, \\ z_\ell \geq 0,\ \forall \ell \in \mathcal{S}_i,\ \forall i \in [m], \end{array} \quad (8)$$

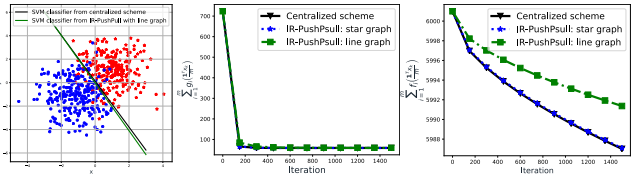

Fig. 3: IR-PushPull vs. centralized scheme for SVM: 10 agents, 300 training sample size, and 500 testing sample size

where $x \in \mathbb{R}^n, b \in \mathbb{R}, z \in \mathbb{R}^{|\mathcal{S}_{\text{train}}|}$, $\eta > 0$. Figure 3 shows the implementation of IR-PushPull on directed line and star graphs with $m := 10$ and $\eta := 0.05$.

Insights: IR-PushPull performs very well compared to the centralized variant. This is examined both in terms of suboptimality and infeasibility metrics in different network topology settings.

REFERENCES

[1] M. Amini and F. Yousefian. An iterative regularized incremental projected subgradient method for a class of bilevel optimization problems. *Proceedings of the American Control Conference (ACC)*, pages 4069–4074, 2019.

[2] N. S. Aybat and E. Y. Hamedani. A primal-dual method for conic constrained distributed optimization problems. *Advances in Neural Information Processing Systems*, pages 5050–5058, 2016.

[3] N. S. Aybat, Z. Wang, T. Lin, and S. Ma. Distributed linearized alternating direction method of multipliers for composite convex consensus optimization. *IEEE Transactions on Automatic Control*, 63(1):5–20, 2017.

[4] D. P. Bertsekas. *Nonlinear Programming: 3rd Edition*. Athena Scientific, Bellmont, MA, 2016.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, 1989.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[7] T. H. Chang. A proximal dual consensus ADMM method for multi-agent constrained optimization. *IEEE Transactions on Signal Processing*, 64(14):3719–3734, 2016.

[8] T. H. Chang, A. Nedić, and A. Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automatic Control*, 59(6):1524–1538, 2014.

[9] M. F. Duarte and Y. H. Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2014.

[10] J. C. Duchi, P. L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization (SIOPT)*, 22(2):674–701, 2012.

[11] F. Facchinei and J.-S. Pang. *Finite-dimensional Variational Inequalities and Complementarity Problems. Vols. I,II*. Springer Series in Operations Research. Springer-Verlag, New York, 2003.

[12] M. H. De Groot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[13] D. Jakovetić, J. Xavier, and J. M. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[14] H. Kaushik and F. Yousefian. A randomized block coordinate iterative regularized subgradient method for high-dimensional ill-posed convex optimization. *Proceedings of the American Control Conference (ACC)*, pages 3420–3425, 2019.

[15] H. D. Kaushik and F. Yousefian. A method with convergence rates for optimization problems with variational inequality constraints. 2021. arXiv preprint: https://arxiv.org/pdf/2007.15845.pdf.

[16] S. Lee and A. Nedić. Asynchronous gossip-based random projection algorithms over networks. *IEEE Transactions on Automatic Control*, 61(4):953–958, 2016.

[17] Q. Ling and A. Ribeiro. Decentralized dynamic optimization through the alternating direction method of multiplier. *IEEE Transactions on Signal Processing*, 62(5):1185–1197, 2014.

[18] I. Lobel, A. Ozdaglar, and D. Feijer. Distributed multi-agent optimization with state-dependent communication. *Mathematical Programming*, 129(2):255—-284, 2011.

[19] C. G. Lopes and A. H. Sayed. Distributed processing over adaptive networks. *In 2007 9th International Symposium on Signal Processing and Its Applications*, pages 1–3, 2007.

[20] P. Di Lorenzo and G. Scutari. Distributed nonconvex optimization over time-varying networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[21] D. Mateos-Nunez and J. Cortes. Distributed subgradient methods for saddle-point problems. *IEEE Conference on Decision and Control (CDC)*, 2015.

[22] A. Nedić and A. Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

[23] A. Nedić and A. Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Transactions on Automatic Control*, 61(12):3936–3947, 2016.

[24] A. Nedić, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[25] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, 2009.

[26] S. Pu and A. Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 2020.

[27] S. Pu, W. Shi, J. Xu, and A. Nedić. Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 2020.

[28] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018.

[29] G. Scutari and Y. Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176:497–544, 2019.

[30] W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: an exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):5944–966, 2015.

[31] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

[32] K. Srivastava and A. Nedić. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):772—-90, 2011.

[33] T. Tatarenki and B. Touri. Non-convex distributed optimization. *IEEE Transactions on Automatic Control*, 62(8):3744–3757, 2017.

[34] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachussetts Institute of Technology, 1984.

[35] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803—-812, 1986.

[36] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. *IEEE Conference on Decision and Control (CDC)*, pages 5445–5450, 2012.

[37] E. Wei and A. Ozdaglar. On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers. *In 2013 IEEE Global Conference on Signal and Information Processing*, pages 551–554, 2013.

[38] F. Yousefian. Bilevel distributed optimization in directed networks. 2021. arXiv preprint: https://arxiv.org/pdf/2006.07564v3.pdf.

[39] F. Yousefian, A. Nedić, and U. V. Shanbhag. On smoothing, regularization, and averaging in stochastic approximation methods for stochastic variational inequality problems. *Mathematical Programming*, 165(1):391–431, 2017.

[40] F. Yousefian, A. Nedić, and U. V. Shanbhag. On stochastic and deterministic quasi-Newton methods for nonstrongly convex optimization: Asymptotic convergence and rate analysis. *SIAM Journal on Optimization*, 30(2):1144–1172, 2020.