Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs

3 authors, including:

Angelia Nedic
Arizona State University
282 PUBLICATIONS   **21,905** CITATIONS

Alex Olshevsky
Boston University
107 PUBLICATIONS   **8,902** CITATIONS

ACHIEVING GEOMETRIC CONVERGENCE FOR DISTRIBUTED OPTIMIZATION OVER TIME-VARYING GRAPHS[*]

ANGELIA NEDIĆ, ALEX OLSHEVSKY, AND WEI SHI[†]

Abstract. This paper considers the problem of distributed optimization over time-varying graphs. For the case of undirected graphs, we introduce a distributed algorithm, referred to as DIGing, based on a combination of a distributed inexact gradient method and a gradient tracking technique. The DIGing algorithm uses doubly stochastic mixing matrices and employs fixed step-sizes and, yet, drives all the agents' iterates to a global and consensual minimizer. When the graphs are directed, in which case the implementation of doubly stochastic mixing matrices is unrealistic, we construct an algorithm that incorporates the push-sum protocol into the DIGing structure, thus obtaining Push-DIGing algorithm. The Push-DIGing uses column stochastic matrices and fixed step-sizes, but it still converges to a global and consensual minimizer. Under the strong convexity assumption, we prove that the algorithms converge at R-linear (geometric) rates as long as the step-sizes do not exceed some upper bounds. We establish explicit estimates for the convergence rates. When the graph is undirected it shows that DIGing scales polynomially in the number of agents. We also provide some numerical experiments to demonstrate the efficacy of the proposed algorithms and to validate our theoretical findings.

Key words. distributed optimization, time-varying graphs, linear convergence, small gain theorem, inexact gradient

1. Introduction. This paper focuses on the following distributed convex optimization problem:

$$(1) \qquad \min_{x \in \mathbb{R}^p} \ f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x),$$

where each function $f_i : \mathbb{R}^p \to \mathbb{R}$ is held privately by agent i to encode the agent's objective function. We assume that the agents are connected through a communication network which can be time-varying. The agents want to collaboratively solve the problem, while each agent can only receive/send the information from/to its immediate neighbors (to be specified precisely soon). Problems of the form (1) that require distributed computing have appeared in various domains including information processing and decision making in sensor networks, networked vehicle/UAV coordination/control, as well as distributed estimation and learning. Some examples include distributed averaging [6, 37, 57], distributed spectrum sensing [1], formation control [36, 44], power system control [13, 41], statistical inference and learning [12, 32, 39]. In general, distributed optimization framework fits the scenarios where the data is collected and/or stored in a network of agents and having a fusion center is either inapplicable or unaffordable. In such scenarios, data processing and computing is to be performed in a distributed but collaborative manner by the agents within the network.

We assume that the functions f_i in problem (1) are convex and continuously differentiable. For such a problem, we propose a class of distributed algorithms that solve the problem over time-varying connectivity graphs for two different cases, namely, the case when the graphs are undirected and the case when they are directed. The algorithms employ consensus ideas for estimating the gradient of the global objective function in (1). When at least one of the objective functions is strongly convex, we show that the algorithms achieve R-linear convergence rates [1].

[1]Suppose that a sequence $\{x(k)\}$ converges to x^* in some norm $\|\cdot\|$. We say that the convergence is: (i) Q-linear if there exists $\lambda \in (0,1)$ such that $\frac{\|x(k+1)-x^*\|}{\|x(k)-x^*\|} \leq \lambda$ for all k; (ii) R-linear if there exists $\lambda \in (0,1)$ and some positive constant C such that $\|x(k) - x^*\| \leq C\lambda^k$ for all k. Both of these rates are geometric, and they are often referred to as global rates to be distinguished from the case when the given relations are valid for some sufficiently large indices k. The

1.1. Literature Review. The research on distributed optimization dates back to 1980s [2, 52]. Since then, due to the emergence of large-scale networks, the development of distributed algorithms for solving problem in (1) has received significant attention recently. Some earlier methods include distributed incremental (sub)gradient methods [25–27, 40] and incremental proximal methods [3, 53], while a more recent work includes incremental aggregated gradient methods [15] and its proximal gradient variants [4]. All of the incremental methods require a special ring networks due to the nature of these methods. To handle a more general (possibly time-varying) networks, distributed subgradient algorithm was proposed in [34], while its stochastic variant was studied in [42] and its asynchronous variant in [24] with provable convergence rates. These algorithms are intuitive and simple but usually slow due to the fact that even if the objective functions are differentiable and strongly convex, these methods still need to use diminishing step-size to converge to a consensual solution. Other works on distributed algorithms that also require the use of diminishing step-sizes include [11, 28, 63]. With a fixed step-size, these distributed methods can be fast, but they only converge to a neighborhood of the solution set. This phenomenon creates an exactness-speed dilemma. A different class of distributed approaches that bypasses this dilemma is based on introducing Lagrangian dual variables and working with the Lagrangian function. The resulting algorithms include distributed dual decomposition [50] and decentralized alternating direction method of multipliers (ADMM) [5, 21]. Specifically, the decentralized ADMM can employ a fixed step-size and it has nice provable rates [54]. Under the strong convexity assumption, the decentralized ADMM has been shown to have linear convergence for time-invariant undirected graphs [49]. Building on (augmented) Lagrangian, a few improvements have been made via proximal-gradient [7], stochastic gradient [16], and second-order approximation [22, 23]. In particular, ADMM over a random undirected network has been shown to have $O(1/k)$ rate for convex functions [16]. However, de-synchronization and extensions of these methods to time-varying undirected graphs are more involved [16, 54], while their extensions to directed graphs are non-existent in the current literature.

Some distributed methods exist that do not (explicitly) use dual variables but can still converge to an exact consensual solution while using fixed step-sizes. In particular, work in [8] employs multi-consensus inner loop and Nesterov's acceleration method, which gives a proximal-gradient algorithm with a rate at least $O(1/k)$. By utilizing multi-consensus inner loop, *adapt-then-combine* (ATC) strategy, and Nesterov's acceleration, the algorithm proposed in [17] is shown to have $O\left(\ln(k)/k^2\right)$ rate under the assumption of bounded and Lipschitz gradients. For least squares, the general diffusion strategy (a generalization of ATC) can converge to the global minimizer' [45]. Although it is unknown to the literature, the above algorithms that do not use dual variable but use fixed step-size are not likely to reach linear convergence even under the strong convexity assumption. References [46–48] use a difference structure to cancel the steady state error in decentralized gradient descent [34, 60], thereby developing the algorithm EXTRA and its proximal-gradient variant. EXTRA converges at an $o(1/k)$ rate when the objective function in (1) is convex, and it has a Q-linear rate when the objective function is strongly convex.

Aside from the diminishing step-size issue, another topic of interest is distributed optimization over *time-varying directed graphs*. The distributed algorithms over time-varying graphs require the use of doubly stochastic weight matrices, which are not easily constructed in a distributed fashion when the graphs are directed. To overcome this issue, reference [28] is the first to propose a different distributed approach, namely a subgradient-push algorithm that combines the distributed subgradient method [34] with the push-sum protocol [18]. While the subgradient-push eliminates the requirement of graph balancing [14], it suffers from a slow sublinear[2] convergence rate even for strongly convex smooth functions due to its employment of diminishing step-size [29]. On the other hand, noticing that EXTRA has satisfactory convergence rates for undirected graphs, references [55, 61] combine

---

difference between these two types of geometric rate is in that Q-linear rate implies monotonic decrease of $\|x(k) - x^*\|$, while R-linear rate does not.

[2]When an algorithm has convergence rate of $O(\theta(k))$, we say that the rate is sublinear if $\lim_{k \to +\infty} \frac{\lambda^k}{\theta(k)} = 0$ for any constant $\lambda \in (0, 1)$. A typical sublinear rates include $O(1/k^p)$ with $p > 0$.

EXTRA with the push-sum protocol [18] to produce DEXTRA (ExtraPush) algorithm in hope of making it work over directed graph. It turns out that for a time-invariant strongly connected directed graph, DEXTRA converges at an R-linear rate under strong convexity assumption but the step-size has to be carefully chosen in some interval. However, the feasible set of step-sizes for DEXTRA can even be empty in some situations [55]. References [58, 59] utilize an ATC strategy [45] of dynamic average consensus approach [62] and, thereby, it develops Aug-DGM algorithm, which is capable of working over time-invariant directed or undirected graphs (though doubly stochastic matrix is needed). Specifically, Aug-DGM is proved to be convergent for general convex smooth objectives when the step-size is sufficiently small. The most promising part of Aug-DGM is that it allows different agents to use different (constant) step-size value and, yet, all agents iterates converge to a consensual minimizer. Another algorithm has been recently developed in [20], referred to as NEXT, which also utilizes dynamic average consensus strategy [62] but the algorithm deals with a more general non-convex problem and its convergence rate is yet to be studied.

1.2. Summary of Contributions. Prior to this work, an open question was how to construct a linearly convergent method for distributed optimization over time-varying (undirected or directed) graphs. This question motivates the work in the present paper that resolves the issue. Specifically, we construct linearly convergent methods along with establishment of explicit bounds on their convergence rates. It should be noted that, unlike the DEXTRA (ExtraPush) algorithm proposed in [55, 61], our algorithm has a linear convergence as long as the nonnegative step-size is smaller than an explicit upper bound and no positive lower bound is needed. Also, our convergence results hold for time-varying graphs which is in contrast with the results in [55, 61] that apply only to static graphs.

In addition, we show that when the graphs are time-varying and undirected, if the mixing matrices in the algorithm satisfy some specific assumption (this assumption can be fulfilled by simple rules such as Metropolis mixing strategy), the iteration complexity (number of iterations needed to reach a specific accuracy) of the introduced algorithm scales polynomially in the number of agents. By "polynomial scaling" we mean that the number of iterations needed by the algorithm to reach a specific accuracy is $O(n^m)$ for some nonnegative integer m, where n is the number of agents.

1.3. Paper Organization. The rest of this paper is organized as follows. To facilitate the description of the technical ideas, the algorithms, and the analysis, we first introduce the notation in Subsection 1.4. In Section 2 we consider the case of undirected time-varying graphs, and we introduce a distributed consensus-based algorithm in Section 2.1. The algorithm uses "distributed inexact gradients" and, also, employs a "gradient tracking" technique, thus we term the algorithm as DIGing to account for its main design features. In Section 3 we establish that the DIGing algorithm converges at an R-linear rate under standard assumptions including uniform joint strong connectivity of the graphs, the strong convexity of the objective function, and the Lipschitz continuity of the gradients. Moreover, we show that the convergence rate of DIGing scales polynomially in the total number of agents in the network. After this, we consider the case of time-varying directed graphs, and we propose a push-sum consensus-based variant of DIGing in Section 4. We establish its R-linear rate in Section 5. Finally, some numerical simulations are given in Section 6, and the paper concludes with some final remarks in Section 7.

1.4. Notation. Throughout the paper, the variable $x \in \mathbb{R}^p$ in the original problem (1) is viewed as a column vector. We let agent i hold a *local copy* of the variable x of the problem in (1), which is denoted by $x_i \in \mathbb{R}^p$; its value at iteration/time k is denoted by $x_i(k)$. We introduce an *aggregate objective function* of the local variables: $\mathbf{f}(\mathbf{x}) \triangleq \sum_{i=1}^n f_i(x_i)$, where its argument and gradient are defined as

$$\mathbf{x} \triangleq \begin{pmatrix} - & x_1^\top & - \\ - & x_2^\top & - \\ & \vdots & \\ - & x_n^\top & - \end{pmatrix} \in \mathbb{R}^{n \times p} \quad \text{and} \quad \nabla \mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} - & (\nabla f_1(x_1))^\top & - \\ - & (\nabla f_2(x_2))^\top & - \\ & \vdots & \\ - & (\nabla f_n(x_n))^\top & - \end{pmatrix} \in \mathbb{R}^{n \times p},$$

respectively. Each row i of $\mathbf{x}$ and $\nabla \mathbf{f}(\mathbf{x})$ is associated with agent i. We say that $\mathbf{x}$ is *consensual* if all of its rows are identical, i.e., $x_1 = x_2 = \cdots = x_n$. The analysis and results of this paper hold for all $p \geq 1$. The reader can assume $p = 1$ for convenience (so $\mathbf{x}$ and $\nabla \mathbf{f}$ become vectors) without loss of generality. The notation is not standard but it enables us to present our algorithm and analysis in a compact form.

We let $\mathbf{1}$ denote a column vector with all entries equal to one (its size is to be understood from the context). For any matrix $\mathbf{v} \in \mathbb{R}^{n \times p}$, we denote its average across the rows (a row corresponds to an agent) as $\overline{v} = \frac{1}{n}\mathbf{v}^\top \mathbf{1} \in \mathbb{R}^p$, and its consensus violation as $\check{\mathbf{v}} = \mathbf{v} - \mathbf{1}\overline{v}^\top = \mathbf{v} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \mathbf{v} = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\mathbf{v} \triangleq \mathbf{L}\mathbf{v}$, where $\mathbf{L} = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is a symmetric matrix. For any $n \times p$ matrices $\mathbf{a}$ and $\mathbf{b}$, their inner product is denoted as $\langle \mathbf{a}, \mathbf{b} \rangle = \text{Trace}(\mathbf{a}^\top \mathbf{b})$. For a given matrix $\mathbf{a}$, the Frobenius norm is given by $\|\mathbf{a}\|_\text{F}$, the (entry-wise) max norm is given by $\|\mathbf{a}\|_\text{max}$, while the spectral norm is given by $\|\mathbf{a}\|_2$. $\|\mathbf{a}\|_2$ equals to the largest singular value $\sigma_\text{max}\{\mathbf{a}\}$. We also use $\|\mathbf{a}\|_\mathbf{L}$ to denote its $\mathbf{L}$ weighted (semi)-norm, that is, $\|\mathbf{a}\|_\mathbf{L} = \sqrt{\langle \mathbf{a}, \mathbf{L}\mathbf{a} \rangle}$. Note that since $\mathbf{L} = \mathbf{L}^\top \mathbf{L}$, we always have $\|\mathbf{a}\|_\mathbf{L} = \|\mathbf{L}\mathbf{a}\|_\text{F}$. For any $n \times p$ matrices $\mathbf{a}, \mathbf{b}, \mathbf{c}$ with $\mathbf{c} = \mathbf{a} + \mathbf{b}$, in view of the definition of the consensus violation, it holds that

$$\|\check{\mathbf{c}}\|_\text{F} = \|\mathbf{c}\|_\mathbf{L} = \|\mathbf{a} + \mathbf{b}\|_\mathbf{L} \leq \|\mathbf{a}\|_\mathbf{L} + \|\mathbf{b}\|_\mathbf{L} = \|\check{\mathbf{a}}\|_\text{F} + \|\check{\mathbf{b}}\|_\text{F}$$

and we will use this property in the analysis without specific explanation. Finally, for any matrix $A \in \mathbb{R}^{m \times n}$, $\text{null}\{A\} \triangleq \{x \in \mathbb{R}^n | Ax = 0\}$ is the null space of A and $\text{span}\{A\} \triangleq \{y \in \mathbb{R}^m | y = Ax, x \in \mathbb{R}^n\}$ is the linear span of all the columns of A.

2. Distributed Optimization over Undirected Graphs. In this section, we consider the case when the agents want to jointly solve the problem (1) while interacting over time-varying undirected graphs. We describe our algorithm, provide an interpretation of its steps and discuss its connection to some of the existing approaches.

2.1. The DIGing Algorithm. We introduce the algorithm and provide some insights into its iterations. In what follows we will make use of the following proposition, which provides the optimality conditions for problem (1).

PROPOSITION 1 ([48]).   *Assume* $\text{null}\{I - \mathbf{W}\} = \text{span}\{\mathbf{1}\}$ *where* $\mathbf{W} \in \mathbb{R}^{n \times n}$. *If* $\mathbf{x}^*$ *satisfies conditions:* (i) $\mathbf{x}^* = \mathbf{W}\mathbf{x}^*$ *(consensus),* (ii) $\mathbf{1}^\top \nabla \mathbf{f}(\mathbf{x}^*) = 0$ *(optimality), then the rows of* $\mathbf{x}^*$ *are the same as each other and the transpose of each row is an optimal solution of the problem* (1).

To illustrate the idea of the DIGing algorithm, let us focus on the case of a static graph for the moment. Consider the distributed gradient descent (DGD), given as follows:

$$\mathbf{x}(k + 1) = \mathbf{W}\mathbf{x}(k) - \alpha \nabla \mathbf{f}(\mathbf{x}(k)),$$

where $\mathbf{W}$ is a doubly stochastic mixing matrix and $\alpha > 0$ is a fixed step-size. The mixing part "$\mathbf{W}\mathbf{x}(k)$" is necessary for reaching consensus while DGD exhibits undesirable behavior due to its use of the gradient direction, "$-\alpha \nabla \mathbf{f}(\mathbf{x}(k))$". To see this, let us break the update into steps per agent: for every agent i, we have $x_i(k+1) = W_{ii}x_i(k) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(k) - \alpha \nabla f_i(x_i(k))$, where $\mathcal{N}_i$ is the set of the neighbors of agent i in the given graph. Thus, each agent is updating using only the gradient of its local objective function f_i. Suppose now that the values $x_i(k)$ have reached consensus and that $x_i(k) = x^*$ for all i and some solution x^* of the problem (1). Then, the mixing part gives $W_{ii}x_i(k) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(k) = x^*$ for all i. However, the gradient-based term gives $-\alpha \nabla f_i(x_i(k))$ for all i, which need not be zero in general, thus resulting in $x_i(k + 1)$ that will move away from the solution x^* (recall that a solution to the problem (1) is at a point x where $\sum_{j=1}^n \nabla f_j(x) = 0 \ \forall i$ and not necessarily a point where $\nabla f_i(x) = 0 \ \forall i$).

Conceptually, one (non-distributed) scheme that bypasses this limitation is the update

$$(2) \qquad\qquad \mathbf{x}(k + 1) = \mathbf{W}\mathbf{x}(k) - \alpha \frac{1}{n}\mathbf{1}\mathbf{1}^\top \nabla \mathbf{f}(\mathbf{x}(k))$$

which can be implemented if every agent has access to the average of all the gradients $\nabla f_j(x_j(k))$, $j = 1, \ldots, n$ (evaluated at each agent's local copy). One can verify that if (2) converges, its limit point

$\mathbf{x}(\infty)$ satisfies the optimality conditions as given in Proposition 1. However, the update in (2) is not distributed among the agents as it requires a central entity to provide the average of the gradients.

Nevertheless, one may approximate the update in (2) through a surrogate direction that tracks the gradient average. To track the average of the gradients, namely, $\frac{1}{n}\mathbf{1}\mathbf{1}^\top\nabla\mathbf{f}(\mathbf{x}(k))$, we introduce a variable $\mathbf{y}(k)$ that is updated as follows:

$$\text{(3)} \qquad \mathbf{y}(k+1) = \mathbf{W}\mathbf{y}(k) + \nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k)),$$

with initialization $\mathbf{y}(0) = \nabla\mathbf{f}(\mathbf{x}(0))$ and where each row i of $\mathbf{y}(k) \in \mathbb{R}^{n \times p}$ is associated with agent i. A similar technique has been introduced in [62] for dynamically tracking the average state of a multi-agent system, and for tracking some network-wide aggregate quantities in [19, 43]. If $\mathbf{x}(k+1)$ converges to some point $\mathbf{x}(\infty)$ and the underlying graph is connected, then it can be seen that the sequence $\mathbf{y}(k)$ generated by the gradient tracking procedure (3) will converge to the point $\mathbf{y}(\infty)$ given by

$$\mathbf{y}(\infty) = \frac{1}{n}\mathbf{1}\mathbf{1}^\top\nabla\mathbf{f}(\mathbf{x}(\infty)),$$

which is exactly what we need in view of (2). Replacing $\frac{1}{n}\mathbf{1}\mathbf{1}^\top\nabla\mathbf{f}(\mathbf{x}(k))$ in (2) by its dynamic approximation $\mathbf{y}(k)$ is exactly what we use to construct the DIGing algorithm. Furthermore, to accommodate time-varying graphs, the static weight matrix $\mathbf{W}$ is replaced by a time varying matrix $\mathbf{W}(k)$, thus resulting in the DIGing algorithm, as given below.

Algorithm 1: DIGing

Choose step-size $\alpha > 0$ and pick any $\mathbf{x}(0) \in \mathbb{R}^{n \times p}$;
Initialize $\mathbf{y}(0) = \nabla\mathbf{f}(\mathbf{x}(0))$;
for $k = 0, 1, \ldots$ **do**
  $\mathbf{x}(k+1) = \mathbf{W}(k)\mathbf{x}(k) - \alpha\mathbf{y}(k)$;
  $\mathbf{y}(k+1) = \mathbf{W}(k)\mathbf{y}(k) + \nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k))$;
end

Looking at an individual agent i, the initialization of DIGing uses an arbitrary $x_i(0) \in \mathbb{R}^p$ and sets $y_i(0) = \nabla f_i(x_i(0))$ for all $i = 1, \ldots, n$. At each iteration k, every agent i maintains two vectors, namely, $x_i(k), y_i(k) \in \mathbb{R}^p$, which are updated as follows:

$$x_i(k+1) = W_{ii}(k)x_i(k) + \sum_{j \in \mathcal{N}_i(k)} W_{ij}(k)x_j(k) - \alpha y_i(k),$$
$$y_i(k+1) = W_{ii}(k)y_i(k) + \sum_{j \in \mathcal{N}_i(k)} W_{ij}(k)y_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)),$$

where $\mathcal{N}_i(k)$ is the set of all neighbors of agent i at time k. At every iteration k, each agent i sends its current solution estimate $x_i(k)$ and average gradient estimate $y_i(k)$ to all of its neighbors $\mathcal{N}_i(k)$ while receiving all of its neighbors' solution estimates $x_j(k)$ and average gradient estimates $y_j(k), \forall j \in \mathcal{N}_i(k)$. Then, each agent i updates its vector $x_i(k+1)$ by mixing its own $x_i(k)$ and the neighbors' copies $x_j(k)$, for $j \in \mathcal{N}_i(k)$, with specific weights, and adjusting along the direction of $-y_i(k)$. Also, each agent i updates its direction $y_i(k+1)$ by mixing its own $y_i(k)$ and the neighbors' directions $y_j(k)$, for $j \in \mathcal{N}_i(k)$ with specific weights, and by taking into account only the new information contained in the most recent gradient evaluation, as captured in the gradient-difference term $\nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k))$.

2.2. Relation of DIGing to some of the existing approaches. This section explains how the introduced DIGing algorithm is related to some of the other distributed algorithms.

2.2.1. Connection with EXTRA [48]. When $\mathbf{W}(k) = \mathbf{W}$ (time-invariant case) and $\mathbf{W}$ is symmetric, the DIGing algorithm shares some similarity with EXTRA. If we eliminate the variables $\mathbf{y}(k)$ in the recursion of DIGing, we will obtain

$$\mathbf{x}(k+2) = (I + (2\mathbf{W} - I))\mathbf{x}(k+1) - \mathbf{W}^2\mathbf{x}(k) - \alpha[\nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k))].$$

In this form, $(2\mathbf{W} - I)$ and $\mathbf{W}^2$ will be the two mixing matrices in EXTRA. As long as we have

$$(2\mathbf{W} - I) \preccurlyeq \mathbf{W}^2 \preccurlyeq (I + (2\mathbf{W} - I))/2 \quad \text{and} \quad \mathbf{W}^2 \succ 0,$$

the convergence properties of DIGing will follow from the results in [48]. It can be seen that, when $\mathbf{W} \succ 0$, the convergence of DIGing follows from the convergence of EXTRA immediately. In this paper, we conduct the convergence analysis with more general choices of time-varying $\mathbf{W}(k)$.

2.2.2. Connection with the general primal-dual. DIGing has a primal-dual interpretation when the mixing matrices are static and symmetric. Indeed, suppose that $\mathbf{W}(k) = \mathbf{W}$ where $\mathbf{W}$ is a symmetric doubly stochastic matrix, and consider the augmented Lagrangian function

$$(4) \qquad \mathcal{L}_\alpha(\mathbf{x}, \mathbf{r}) = \mathbf{f}(\mathbf{x}) + \frac{1}{\alpha}\langle \mathbf{r}, (I - \mathbf{W})\mathbf{x}\rangle + \frac{1}{2\alpha}\|\mathbf{x}\|^2_{I-\mathbf{W}^2}.$$

If we apply the basic gradient method with step-size α in Gauss-Seidel-like order for seeking the saddle point of the augmented Lagrangian function (4), we will have

$$\begin{aligned} \mathbf{x}\text{-update: } \mathbf{x}(k+1) &= \mathbf{x}(k) - \alpha\nabla\mathbf{f}(\mathbf{x}(k)) - (I - \mathbf{W})\mathbf{r}(k) - (I - \mathbf{W}^2)\mathbf{x}(k), \\ \mathbf{r}\text{-update: } \mathbf{r}(k+1) &= \mathbf{r}(k) + (I - \mathbf{W})\mathbf{x}(k+1). \end{aligned}$$

By eliminating the dual variable $\mathbf{r}$, we will obtain the same updates as in the DIGing algorithm (when the variable $\mathbf{y}$ is eliminated). The same will happen, if alternatively we consider the augmented Lagrangian function

$$(5) \qquad \mathcal{L}_\alpha(\mathbf{x}, \mathbf{r}) = \mathbf{f}(\mathbf{x}) + \frac{1}{\alpha}\langle \mathbf{r}, (I - \mathbf{W})\mathbf{x}\rangle + \frac{1}{2\alpha}\|\mathbf{x}\|^2_{2I-2\mathbf{W}},$$

and apply the basic gradient method with step-size α in Jacobi-like order for seeking the saddle point of the augmented Lagrangian function (5).

REMARK 1. *The above discussion assumes a time-invariant matrix $\mathbf{W}$ which is symmetric. Even in the time-invariant graph case but for asymmetric $\mathbf{W}$, it appears to be difficult to adapt the classical primal-dual analysis for the recursion of DIGing. Our arguments in this paper are from a pure primal perspective and do not assume any symmetry property of $\mathbf{W}(k)$. This suggests the possibility of its extension to the case of directed graphs, which will be addressed in Section 4.*

2.2.3. Connection with Aug-DGM [59]. A very recent reference that comes to our attention is [59] that proposes a distributed consensus optimization algorithm, Aug-DGM, which is applicable to general time-invariant graphs and is quite similar to DIGing. The proposed algorithm is based on the combination of *Adapt-then-Combine* (ATC) strategy of [45] and the dynamic average consensus for gradient tracking of [62]. It differs from DIGing only in the dynamic average gradient-consensus update, which uses an ATC variant. The updates of Aug-DGM are given by

$$\begin{aligned} \mathbf{x}\text{-update: } \quad & \mathbf{x}(k+1) = \mathbf{W}\left(\mathbf{x}(k) - \mathbf{D}\mathbf{y}(k)\right), \\ \mathbf{y}\text{-update: } \quad & \mathbf{y}(k+1) = \mathbf{W}\left(\mathbf{y}(k) + \nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k))\right), \end{aligned}$$

where $\mathbf{W}$ is a doubly stochastic matrix and $\mathbf{D}$ is a diagonal step-size matrix. When $\mathbf{D}$ is chosen as αI, it turns into an ATC variant of DIGing. With a general (positive) diagonal matrix $\mathbf{D}$, Aug-DGM allows different agents to use different step-sizes and it still drives all the agent to reach a consensus on a global minimizer. The convergence of Aug-DGM is provided under general convexity and Lipschitz gradient assumptions.

3. Convergence Analysis for DIGing over Undirected Graphs. In this section we establish the linear convergence of DIGing over time-varying undirected graphs. Let us formally describe the assumptions we make on the graphs and on the mixing matrices $\mathbf{W}(k)$, which are closely related. Consider a time-varying undirected graph sequence $\{\mathcal{G}^{\mathrm{un}}(0), \mathcal{G}^{\mathrm{un}}(1), \ldots\}$. Every graph instance $\mathcal{G}^{\mathrm{un}}(k)$ consists of a time-invariant set of agents $\mathcal{V} = \{1, 2, \ldots, n\}$ and a set of time-varying edges $\mathcal{E}(k)$. The unordered pair of vertices $(j, i) \in \mathcal{E}(k)$ if and only if agents j and i can exchange information at time (iteration) k. The set of neighbors of agent i at time k is defined as $\mathcal{N}_i(k) = \{j | (j, i) \in \mathcal{E}(k)\}$.

In the sequel, we use the following notation:

$$\mathbf{W}_b(k) \triangleq \mathbf{W}(k)\mathbf{W}(k-1)\cdots\mathbf{W}(k-b+1) \quad \text{for any } k = 0, 1, \ldots \text{ and any } b = 0, 1, \ldots,$$

with the convention that $\mathbf{W}_b(k) = I$ for any needed $k < 0$ and $\mathbf{W}_0(k) = I$ for any k.

Next, we give the basic assumption that we impose on the weight matrices.

ASSUMPTION 1 (**Mixing matrix sequence** $\{\mathbf{W}(k)\}$). *For any $k = 0, 1, \ldots$, the mixing matrix* $\mathbf{W}(k) = [W_{ij}(k)] \in \mathbb{R}^{n \times n}$ *satisfies*
   (i) *(Decentralized property) If $i \neq j$ and the edge $(j, i) \notin \mathcal{E}(k)$, then $W_{ij}(k) = 0$;*
   (ii) *(Double stochasticity) $\mathbf{W}(k)\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top\mathbf{W}(k) = \mathbf{1}^\top$;*
   (iii) *(Joint spectrum property) There exists a positive integer B such that*

$$\sup_{k \geq B-1} \delta(k) < 1 \quad \text{where} \quad \delta(k) = \sigma_{\max}\left\{\mathbf{W}_B(k) - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right\} \quad \text{for all} \quad k = 0, 1, \ldots.$$

In Assumption 1, item (i) is due to the physical restriction of the network. Properties (ii) and (iii) are commonly used in the analysis of the rate of consensus algorithms. Several different mixing rules exist that yield the matrix sequences which have property (iii) (see subsection 2.4 of reference [48]).

In particular, the following two assumptions taken together imply Assumption 1 [31].

ASSUMPTION 2 (B-**connected graph sequence**). *The time-varying undirected graph sequence* $\{\mathcal{G}^{\mathrm{un}}(k)\}$ *is B-connected. Specifically, there exists some positive integer B such that the undirected graph $\mathcal{G}_B^{\mathrm{un}}(k) = \left\{\mathcal{V}, \bigcup_{b=k}^{k+B-1} \mathcal{E}(b)\right\}$ is connected for all $k = 0, 1, \ldots$.*

Assumption 2 is typical for many results in multi-agent coordination and distributed optimization [30]. It is much weaker than the assumption of every $\mathcal{G}^{\mathrm{un}}(k)$ being connected.

ASSUMPTION 3 (**Mixing matrix sequence** $\{\mathbf{W}(k)\}$). *For any $k = 0, 1, \ldots$, the mixing matrix* $\mathbf{W}(k) = [W_{ij}(k)] \in \mathbb{R}^{n \times n}$ *satisfies*
   (i) *(Double stochasticity) $\mathbf{W}(k)\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top\mathbf{W}(k) = \mathbf{1}^\top$;*
   (ii) *(Positive diagonal) For all i, $\mathbf{W}_{ii}(k) > 0$;*
   (iii) *(Edge utilization) If $(j, i) \in \mathcal{E}(k)$, then $W_{ij}(k) > 0$; otherwise $W_{ij}(k) = 0$;*
   (iv) *(Non-vanishing weights) There exists some $\tau > 0$ such that if $W_{ij}(k) > 0$, then $W_{ij}(k) > \tau$;*

Assumption 3 is strong but typical for multi-agent coordination and optimization. For undirected graph it can be fulfilled, for example, by using Metropolis weights:

$$W_{ij}(k) = \begin{cases} 1/\left(1 + \max\{d_i(k), d_j(k)\}\right), & \text{if } (j, i) \in \mathcal{E}(k), \\ 0, & \text{if } (j, i) \notin \mathcal{E}(k)) \text{ and } j \neq i, \\ 1 - \sum_{l \in \mathcal{N}_i(k)} W_{il}(k), & \text{if } j = i, \end{cases}$$

where $d_i(k) = |\mathcal{N}_i(k)|$ is the degree of agent i at time k. In this case, Assumption 3 will be satisfied with the choice of $\tau = 1/n$.

The following lemma provides an important relation for later use.

LEMMA 2 (B-**step consensus contraction**). *Under Assumption 1, for any $k = B-1, B, \ldots$, and any matrix $\mathbf{b}$ with appropriate dimensions, if $\mathbf{a} = \mathbf{W}_B(k)\mathbf{b}$, then we have $\|\mathbf{a}\|_{\mathbf{L}} \leq \delta(k)\|\mathbf{b}\|_{\mathbf{L}}$, where $\delta(k)$ is as given in Assumption 1(iii).*

We do not claim any originality of this lemma. Clearly, this lemma is fairly standard in consensus theory and is a direct consequence of Assumption 1 due to the fact that $\mathbf{W}_B(k)$ is doubly stochastic:

$$\begin{aligned} \|\mathbf{a}\|_{\mathbf{L}} &= \|(I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{W}_B(k)\mathbf{b}\|_{\mathrm{F}} \\ &= \|(\mathbf{W}_B(k) - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)(I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{b}\|_{\mathrm{F}} \\ &\leq \sigma_{\max}\left\{\mathbf{W}_B(k) - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top\right\}\|\mathbf{b}\|_{\mathbf{L}}. \end{aligned}$$

To make our arguments more concise, we will use $\delta = \sup_{k \geq B-1}\{\delta(k)\}$ in our analysis of the algorithm. An explicit expression of δ in terms of n can be found in [31] if the more specific Assumption 3 is made.

We also need the following two assumptions on the objective functions, which are standard for deriving linear (geometric) rate of strongly convex smooth functions.

ASSUMPTION 4 (**Smoothness**). *For every agent i, its objective $f_i : \mathbb{R}^p \to \mathbb{R}$ is differentiable and has Lipschitz gradient, i.e., there exists a Lipschitz constant $L_i \in (0, +\infty)$ such that*

$$\|\nabla f_i(x_a) - \nabla f_i(x_b)\|_{\mathrm{F}} \leq L_i \|x_a - x_b\|_{\mathrm{F}} \quad \text{for any} \quad x_a, x_b \in \mathbb{R}^p.$$

When Assumption 4 holds, we will also say that each ∇f_i is L_i-Lipschitz (continuous). In the forthcoming analysis we will use $L \triangleq \max_i \{L_i\}$, which is the Lipschitz constant of $\nabla \mathbf{f}(\mathbf{x})$, and $\bar{L} \triangleq (1/n) \sum_{i=1}^{n} L_i$ which is the Lipschitz constant of $\nabla f(x)$.

ASSUMPTION 5 (**Strong convexity**). *For every agent i, its objective $f_i : \mathbb{R}^p \to \mathbb{R}$ satisfies*

$$f_i(x_a) \geq f_i(x_b) + \langle \nabla f_i(x_b), x_a - x_b \rangle + \frac{\mu_i}{2} \|x_a - x_b\|_{\mathrm{F}}^2 \quad \text{for any} \quad x_a, x_b \in \mathbb{R}^p,$$

where $\mu_i \in [0, +\infty)$ and at least one μ_i is nonzero.

When Assumption 5 holds, we also say that f_i is μ_i-strongly convex. In the analysis we will use $\hat{\mu} \triangleq \max_i \{\mu_i\}$ and $\bar{\mu} \triangleq (1/n) \sum_{i=1}^{n} \mu_i$. Assumption 5 implies the $\bar{\mu}$-strong convexity of $f(x)$. Under this assumption, the optimal solution to problem (1) is guaranteed to exist and to be unique since $\bar{\mu} > 0$. We note that all the convergence results in our analysis are achieved under Assumption 5. We will also use $\bar{\kappa} \triangleq L/\bar{\mu}$.

To establish the R-linear rate of the algorithm, one of our technical innovations will be to resort to a somewhat unusual version of small gain theorem under a well-chosen metric, whose original version has received an extensive research and been widely applied in control theory [9]. We will give an intuition of the whole analytical approach shortly, after stating the small gain theorem at first.

3.1. The Small Gain Theorem. Let us adopt the notation $\mathbf{s}^i$ for the infinite sequence $\mathbf{s}^i = (\mathbf{s}^i(0), \mathbf{s}^i(1), \mathbf{s}^i(2), \ldots)$ where $\mathbf{s}^i(k) \in \mathbb{R}^{n \times p}$, $\forall i$. Furthermore, let us define

$$(6) \qquad \|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda, K} \triangleq \max_{k=0,\ldots,K} \frac{1}{\lambda^k} \|\mathbf{s}^i(k)\|_{\mathrm{F}} \quad \text{and} \quad \|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda} \triangleq \sup_{k \geq 0} \frac{1}{\lambda^k} \|\mathbf{s}^i(k)\|_{\mathrm{F}},$$

where the parameter $\lambda \in (0,1)$ will serve as the linear rate parameter later in our analysis. While $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda, K}$ is always finite, $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda}$ may be infinite. If $n = p = 1$, i.e., each $\mathbf{s}^i(k)$ is a scalar, we will just write $|\mathbf{s}^i|^{\lambda, K}$ and $|\mathbf{s}^i|^{\lambda}$ for these quantities. Intuitively, $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda, K}$ is a weighted "ergodic norm" of $\mathbf{s}^i$. Noticing that the weight $\frac{1}{\lambda^k}$ is exponentially growing with respect to k, if we can show that $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda}$ is bounded, then it would imply that $\|\mathbf{s}^i(k)\|_{\mathrm{F}} \to 0$ geometrically fast. This ergodic definition enables us to give analysis to those algorithms which do not converge Q-linearly. Next we will state the small gain theorem which gives a sufficient condition to for the boundedness of $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda}$. The theorem is a basic result in control systems and a detailed discussion about its result can be found in [9]. For the sake of completeness, we include the proof.

THEOREM 3 (**The small gain theorem**). *Suppose that $\mathbf{s}^1, \ldots, \mathbf{s}^m$ are sequences such that for all positive integers K and for each $i = 1, \ldots, m$, we have an arrow $\mathbf{s}^i \to \mathbf{s}^{(i \bmod m)+1}$, that is*

$$(7) \qquad \|\mathbf{s}^{(i \bmod m)+1}\|_{\mathrm{F}}^{\lambda, K} \leq \gamma_i \|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda, K} + \omega_i,$$

where the constants (gains) $\gamma_1, \ldots, \gamma_m$ are nonnegative and satisfy $\gamma_1 \gamma_2 \cdots \gamma_m < 1$. Then

$$\|\mathbf{s}^1\|_{\mathrm{F}}^{\lambda} \leq \frac{1}{1 - \gamma_1 \gamma_2 \cdots \gamma_m} \left(\omega_1 \gamma_2 \gamma_3 \cdots \gamma_m + \omega_2 \gamma_3 \gamma_4 \cdots \gamma_m + \cdots + \omega_{m-1} \gamma_m + \omega_m \right).$$

Proof. By iterating inequality (7) for i from m down to 1, we obtain

$$\begin{aligned} \|\mathbf{s}^1\|_{\mathrm{F}}^{\lambda, K} &\leq \gamma_m \gamma_{m-1} \cdots \gamma_1 \|\mathbf{s}^1\|_{\mathrm{F}}^{\lambda, K} + \gamma_m \gamma_{m-1} \cdots \gamma_2 \omega_1 \\ &\quad + \gamma_m \gamma_{m-1} \cdots \gamma_3 \omega_2 + \cdots + \gamma_m \omega_{m-1} + \omega_m. \end{aligned}$$

Thus,

$$(8) \qquad \|\mathbf{s}^1\|_{\mathrm{F}}^{\lambda,K} \quad \leq \quad \frac{1}{1-\gamma_1\gamma_2\cdots\gamma_m} \left(\omega_1\gamma_2\gamma_3\cdots\gamma_m + \omega_2\gamma_3\gamma_4\cdots\gamma_m + \cdots + \omega_{m-1}\gamma_m + \omega_m \right).$$

Since (8) holds for all K and its right-hand side does not depend on K, taking $K \to \infty$ implies the desired relation. $\quad\square$

Clearly, the small gain theorem involves a cycle $\mathbf{s}^1 \to \mathbf{s}^2 \to \cdots \to \mathbf{s}^m \to \mathbf{s}^1$. Due to this cyclic structure [cf. (7)], similar bounds hold for $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda}$, $\forall i$.

LEMMA 4 (**Bounded norm $\Rightarrow$ R-linear rate**). *For any matrix sequence $\mathbf{s}^i$, if $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda}$ is bounded, then $\|\mathbf{s}^i(k)\|_{\mathrm{F}}$ converges at a global R-linear (geometric) rate $O(\lambda^k)$.*

Proof. If $\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda} \leq U$ where U is some nonnegative constant, then by the definition we have $\sup_{k\geq 0} \frac{1}{\lambda^k}\|\mathbf{s}^i(k)\|_{\mathrm{F}} \leq U$, thus $\frac{1}{\lambda^k}\|\mathbf{s}^i(k)\|_{\mathrm{F}} \leq U$, $\forall k$. The conclusion follows immediately from $\|\mathbf{s}^i(k)\|_{\mathrm{F}} \leq U\lambda^k$, $\forall k$. $\quad\square$

3.2. Sketch of the Main Idea. Before summarizing our main proof idea, let us define some quantities which we will use frequently in our analysis. We define $\mathbf{x}^* \triangleq \mathbf{1}(x^*)^\top$ where x^* is the optimal solution of problem (1). Also, define

$$\mathbf{q}(k) \triangleq \mathbf{x}(k) - \mathbf{x}^* \text{ for any } k = 0, 1, \ldots,$$

which is the optimality residual of the iterates $\mathbf{x}(k)$ (at the k-th iteration). Moreover, let us adopt the notation

$$\mathbf{z}(k) \triangleq \nabla\mathbf{f}(\mathbf{x}(k)) - \nabla\mathbf{f}(\mathbf{x}(k-1)) \text{ for any } k = 1, 2, \ldots,$$

and with the convention that $\mathbf{z}(0) \triangleq \mathbf{0}$.

We will apply the small gain theorem with the $\|\cdot\|_{\mathrm{F}}^{\lambda,K}$ metric and a right choice of $\lambda < 1$ around the following circle of arrows:

$$(9) \qquad\qquad\qquad \text{Algorithm 1: } \mathbf{q} \to \mathbf{z} \to \check{\mathbf{y}} \to \check{\mathbf{x}} \to \mathbf{q},$$

where, recall, $\mathbf{q}$ is the difference between local copies and the global optimizer, $\mathbf{z}$ is the successive difference of gradients, $\check{\mathbf{y}}$ is the consensus violation of the estimation of gradient average across agents, and $\check{\mathbf{x}}$ is the consensus violation of local copies (see Subsection 1.4 for the definition of operator "$\check{}$").

Intuitively: as long $\mathbf{q}$ is small, the successive difference of the gradients $\mathbf{z}$ is small since the gradients are close to zero in the neighborhood of the optimal point; as long as the successive difference of the gradients $\mathbf{z}$ is small, the structure of DIGing implies that $\mathbf{y}$ is close to consensual; as long as $\mathbf{y}$ is close to consensual, then by the structure of DIGing so is $\mathbf{x}$; and, finally, as long as $\mathbf{x}$ is close to consensual, DIGing is very similar to gradient descent and drives the distance to the optimal point $\mathbf{q}$ to zero and thus completes the cycle.

After the establishment of each arrow, we will apply the small gain theorem to conclude that every corresponding quantities under the metric $\|\cdot\|_{\mathrm{F}}^{\lambda,K}$ is bounded and hence conclude that all quantities in the "circle of arrows" decay at an R-linear rate $O(\lambda^k)$.

Note that to apply the small gain theorem, we would need to have gains (γ_i) that multiply to less than one. This is achieved by choosing an appropriate step-size α. Indeed, by looking at the algorithm, we can see that the step-size appears only in one place – the third arrow (i.e., the arrow $\check{\mathbf{y}} \to \check{\mathbf{x}}$), and the dependence of the corresponding gain in that arrow is *linear* in α. Thus we should be able to apply the small gain theorem after choosing small enough α.

3.3. The Establishment of Each Arrow. We now discuss the establishment of each arrow/relation in the sketch above [cf. (9)].

The first arrow demonstrated in Lemma 5 is a simple consequence of Assumption 4 (namely, it is a consequence of the fact that the gradient of $\mathbf{f}$ is L-Lipschitz).

LEMMA 5 (**Algorithm 1: The first arrow $\mathbf{q} \to \mathbf{z}$**). *Under Assumption 4, we have that for all $K = 0, 1, \ldots$ and any $\lambda \in (0, 1)$,*

$$\|\mathbf{z}\|_{\mathrm{F}}^{\lambda, K} \leq L \left(1 + \frac{1}{\lambda} \right) \|\mathbf{q}\|_{\mathrm{F}}^{\lambda, K}.$$

Proof. By Assumption 4, $\nabla \mathbf{f}(\mathbf{x})$ is L-Lipschitz and we have

(10)
$$
\begin{aligned}
\|\nabla \mathbf{f}(\mathbf{x}(k+1)) - \nabla \mathbf{f}(\mathbf{x}(k))\|_{\mathrm{F}} &\leq L \|\mathbf{x}(k+1) - \mathbf{x}(k)\|_{\mathrm{F}} \\
&= L \|(\mathbf{x}(k+1) - \mathbf{x}^*) - (\mathbf{x}(k) - \mathbf{x}^*)\|_{\mathrm{F}} \\
&\leq L \|\mathbf{x}(k+1) - \mathbf{x}^*\|_{\mathrm{F}} + L \|\mathbf{x}(k) - \mathbf{x}^*\|_{\mathrm{F}}.
\end{aligned}
$$

By the definition of $\mathbf{z}$ and $\mathbf{q}$, it follows from (10) that

(11)
$$\lambda^{-(k+1)} \|\mathbf{z}(k+1)\|_{\mathrm{F}} \leq L \lambda^{-(k+1)} \|\mathbf{q}(k+1)\|_{\mathrm{F}} + \tfrac{L}{\lambda} \lambda^{-k} \|\mathbf{q}(k)\|_{\mathrm{F}}.$$

Taking $\max_{k=0,1,\ldots,K-1}\{\cdot\}$ on both sides of (11) gives

$$\|\mathbf{z}\|_{\mathrm{F}}^{\lambda, K} \leq L \|\mathbf{q}\|_{\mathrm{F}}^{\lambda, K} + \tfrac{L}{\lambda} \|\mathbf{q}\|_{\mathrm{F}}^{\lambda, K-1} \leq L \left(1 + \tfrac{1}{\lambda} \right) \|\mathbf{q}\|_{\mathrm{F}}^{\lambda, K}. \qquad \square$$

Next we provide the lemmata for the second and third arrows in the cycle (9). They are proved by an almost identical analysis based on Lemma 2: indeed, a glance at the structure of DIGing implies that some (semi)-norm of $\mathbf{x}$ can be bounded in terms of some (semi)-norm of $\mathbf{y}$, while some (semi)-norm of $\mathbf{y}$ can be bounded in terms of some (semi)-norm of $\mathbf{z}$. This is a fairly straightforward application of Lemma 2, which shows how multiplication by $\mathbf{W}(k)$ shrinks the distance toward the consensus subspace.

LEMMA 6 (**Algorithm 1: The second arrow $\mathbf{z} \to \check{\mathbf{y}}$**). *Let Assumption 1 hold, and let $\delta = \sup_{k \geq B-1}\{\delta(k)\}$, where $\delta(k)$ is as given in Assumption 1(iii). Also, let λ be such that $\delta < \lambda^B < 1$. Then, we have for all $K = 0, 1, \ldots$,*

(12)
$$\|\check{\mathbf{y}}\|_{\mathrm{F}}^{\lambda, K} \leq \frac{\lambda(1 - \lambda^B)}{(\lambda^B - \delta)(1 - \lambda)} \|\mathbf{z}\|_{\mathrm{F}}^{\lambda, K} + \frac{\lambda^B}{\lambda^B - \delta} \sum_{t=1}^{B} \lambda^{1-t} \|\check{\mathbf{y}}(t-1)\|_{\mathrm{F}}.$$

Proof. The equivalent relation in DIGing involving $\mathbf{y}$ and $\mathbf{z}$ is

(13)
$$\mathbf{y}(k+1) = \mathbf{W}(k)\mathbf{y}(k) + \mathbf{z}(k+1).$$

From (13), using Lemma 2, for all $k \geq B - 1$, it follows that

(14)
$$
\begin{aligned}
\|\check{\mathbf{y}}(k+1)\|_{\mathrm{F}} &= \|\mathbf{y}(k+1)\|_{\mathbf{L}} \\
&\leq \|\mathbf{W}_B(k)\mathbf{y}(k+1-B)\|_{\mathbf{L}} + \|\mathbf{W}_{B-1}(k)\mathbf{z}(k+2-B)\|_{\mathbf{L}} \\
&\quad + \cdots + \|\mathbf{W}_1(k)\mathbf{z}(k)\|_{\mathbf{L}} + \|\mathbf{W}_0(k)\mathbf{z}(k+1)\|_{\mathbf{L}} \\
&\leq \delta \|\check{\mathbf{y}}(k+1-B)\|_{\mathrm{F}} + \sum_{t=1}^{B} \|\mathbf{z}(k+2-t)\|_{\mathrm{F}},
\end{aligned}
$$

and therefore, for all $k = B - 1, B, \ldots$,

(15)
$$\lambda^{-(k+1)} \|\check{\mathbf{y}}(k+1)\|_{\mathrm{F}} \leq \tfrac{\delta}{\lambda^B} \lambda^{-(k+1-B)} \|\check{\mathbf{y}}(k+1-B)\|_{\mathrm{F}} + \sum_{t=1}^{B} \tfrac{1}{\lambda^{t-1}} \lambda^{-(k+2-t)} \|\mathbf{z}(k+2-t)\|_{\mathrm{F}}.$$

To utilize the norm $\|\cdot\|_{\mathrm{F}}^{\lambda, K}$, we need to take $\max_{k=0,\ldots,K}$, which in turn requires a relation for $\lambda^{-(k+1)} \|\check{\mathbf{y}}(k+1)\|_{\mathrm{F}}$ with $k < B - 1$. To obtain such a relation, we complement the initial relation for (15), i.e.,

(16)
$$\lambda^{-(k+1)} \|\check{\mathbf{y}}(k+1)\|_{\mathrm{F}} \leq \lambda^{-(k+1)} \|\check{\mathbf{y}}(k+1)\|_{\mathrm{F}}$$

for $k = -1, \ldots, B - 2$. Taking the maximum over $k = -1, 0, \ldots, B - 2$ on both sides of (16) and the maximum over $k = B - 1, \ldots, K$ in (15), and then by combining the obtained relations, we obtain

$$
\begin{aligned}
\|\check{\mathbf{y}}\|_{\mathrm{F}}^{\lambda,K} & \leq \frac{\delta}{\lambda^B}\|\check{\mathbf{y}}\|_{\mathrm{F}}^{\lambda,K-B} + \sum_{t=1}^{B}\frac{1}{\lambda^{t-1}}\|\mathbf{z}\|_{\mathrm{F}}^{\lambda,K+1-t} + \sum_{t=1}^{B}\lambda^{1-t}\|\check{\mathbf{y}}(t-1)\|_{\mathrm{F}} \\
& \leq \frac{\delta}{\lambda^B}\|\check{\mathbf{y}}\|_{\mathrm{F}}^{\lambda,K} + \sum_{t=1}^{B}\frac{1}{\lambda^{t-1}}\|\mathbf{z}\|_{\mathrm{F}}^{\lambda,K} + \sum_{t=1}^{B}\lambda^{1-t}\|\check{\mathbf{y}}(t-1)\|_{\mathrm{F}}.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\|\check{\mathbf{y}}\|_{\mathrm{F}}^{\lambda,K} & \leq \frac{\lambda^B \sum_{t=1}^{B}\frac{1}{\lambda^{t-1}}}{\lambda^B - \delta}\|\mathbf{z}\|_{\mathrm{F}}^{\lambda,K} + \frac{\lambda^B}{\lambda^B - \delta}\sum_{t=1}^{B}\lambda^{1-t}\|\check{\mathbf{y}}(t-1)\|_{\mathrm{F}} \\
& = \frac{\lambda(1 - \lambda^B)}{(\lambda^B - \delta)(1 - \lambda)}\|\mathbf{z}\|_{\mathrm{F}}^{\lambda,K} + \frac{\lambda^B}{\lambda^B - \delta}\sum_{t=1}^{B}\lambda^{1-t}\|\check{\mathbf{y}}(t-1)\|_{\mathrm{F}},
\end{aligned}
$$

which is exactly (12). $\qquad\square$

LEMMA 7 (**Algorithm 1: The third arrow $\check{\mathbf{y}} \to \check{\mathbf{x}}$**). *Let Assumption 1 hold, and let $\delta = \sup_{k \geq B-1}\{\delta(k)\}$, where $\delta(k)$ is as given in Assumption 1(iii). Furthermore, let λ be such that $\delta < \lambda^B < 1$. Then, we have for all $K = 0, 1, \ldots,$*

$$
\|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} \leq \frac{\alpha(1 - \lambda^B)}{(\lambda^B - \delta)(1 - \lambda)}\|\check{\mathbf{y}}\|_{\mathrm{F}}^{\lambda,K} + \frac{\lambda^B}{\lambda^B - \delta}\sum_{t=1}^{B}\lambda^{1-t}\|\check{\mathbf{x}}(t-1)\|_{\mathrm{F}}.
$$

The relation in DIGing involving $\mathbf{x}$ and $\mathbf{y}$ is given by

(17) $$ \mathbf{x}(k+1) = \mathbf{W}(k)\mathbf{x}(k) - \alpha\mathbf{y}(k). $$

Noticing the similarity between (17) and (13), we omit the proof of Lemma 7 since it is almost identical to that of Lemma 6.

With all the above lemmata in place, the last arrow of our proof sketch remains to be addressed. For this, we need an interlude on gradient descent with errors in the gradient. Since this part is relatively independent from the preceding development, we provide it in the next subsection.

3.4. The Inexact Gradient Descent on a Sum of Strongly Convex Functions. In this subsection, we consider the basic (centralized) first-order method for problem (1) under inexact first-order oracle. To distinguish from the notation used for our distributed optimization problem/algorithm/analysis, let us make some definitions that are only used in this subsection. Problem (1) is restated as follows with different notation,

$$
\min_{x \in \mathbb{R}^d} g(x) = \frac{1}{n}\sum_{i=1}^{n}g_i(x),
$$

where all g_i's satisfy Assumptions 4 and 5 with f_i being replaced by g_i. Let us consider the inexact gradient descent (IGD) on the function g:

(18) $$ p^{k+1} = p^k - \theta\frac{1}{n}\sum_{i=1}^{n}\nabla g_i(s_i^k), $$

where θ is the step-size. Note that since this subsection has nothing to do with time-varying setup, to avoid heavy notation, we use the upper right corner p^k instead of $p(k)$ to denote the value of p at iteration k. In particular, we use $(p)^a$ instead of p^a to denote the a-th power of p when it may cause confusion. Let p^* be the global minimum of g, and define

$$
r^k \triangleq \|p^k - p^*\|_{\mathrm{F}} \text{ for any } k = 0, 1, \ldots.
$$

The main lemma of this subsection is stated next; it is basically obtained by following the ideas in [10].

LEMMA 8 (**The error bound on the IGD**). *Suppose that*

$$(19) \qquad \sqrt{1 - \frac{\theta \bar{\mu} \beta}{\beta + 1}} \le \lambda < 1 \quad and \quad \theta \le \frac{1}{(1 + \eta) \bar{L}},$$

where $\beta > 0$ and $\eta > 0$. Then under Assumptions 4 and 5 with f_i's replaced by g_i's, the tuple sequence $\{r^k, p^k; s_1^k, s_2^k, \ldots, s_n^k\}$ generated by the inexact gradient method (18) obeys

$$|r|^{\lambda, K} \le 2r^0 + (\lambda \sqrt{n})^{-1} \left(\sqrt{\frac{L(1 + \eta)}{\bar{\mu} \eta} + \frac{\hat{\mu}}{\bar{\mu}} \beta} \right) \sum_{i=1}^{n} \|p - s_i\|_{\mathrm{F}}^{\lambda, K} \text{ for any } K = 0, 1, \ldots.$$

Proof. By assumptions, for each $i \in \{1, 2, \ldots, n\}$ and $k = 0, 1, \ldots$, we have

$$(20) \qquad g_i(p^*) \ge g_i(s_i^k) + \langle \nabla g_i(s_i^k), p^* - s_i^k \rangle + \frac{\mu_i}{2} \|p^* - s_i^k\|_{\mathrm{F}}^2.$$

Through using the basic inequality $\|s_i^k - p^*\|_{\mathrm{F}}^2 \ge \frac{\beta}{\beta + 1} \|p^k - p^*\|_{\mathrm{F}}^2 - \beta \|p^k - s_i^k\|_{\mathrm{F}}^2$ where $\beta > 0$ is a tunable parameter, it follows from (20) that

$$\begin{aligned} g_i(p^*) &\ge g_i(s_i^k) + \langle \nabla g_i(s_i^k), p^k - s_i^k \rangle + \langle \nabla g_i(s_i^k), p^* - p^k \rangle \\ &\quad + \frac{\mu_i}{2} \left(\frac{\beta}{\beta + 1} \|p^k - p^*\|_{\mathrm{F}}^2 - \beta \|p^k - s_i^k\|_{\mathrm{F}}^2 \right) \end{aligned}$$

and therefore

$$(21) \qquad \begin{aligned} \langle \nabla g_i(s_i^k), p^* - p^k \rangle &\le g_i(p^*) - g_i(s_i^k) - \langle \nabla g_i(s_i^k), p^k - s_i^k \rangle \\ &\quad - \frac{\mu_i \beta}{2(\beta + 1)} \|p^k - p^*\|_{\mathrm{F}}^2 + \frac{\mu_i \beta}{2} \|s_i^k - p^k\|_{\mathrm{F}}^2. \end{aligned}$$

Averaging (21) over i through 1 to n gives

$$(22) \qquad \begin{aligned} &\frac{1}{n} \sum_{i=1}^{n} \langle \nabla g_i(s_i^k), p^* - p^k \rangle \\ &\le g(p^*) - \frac{1}{n} \sum_{i=1}^{n} \left(g_i(s_i^k) + \langle \nabla g_i(s_i^k), p^k - s_i^k \rangle - \frac{\mu_i \beta}{2} \|s_i^k - p^k\|_{\mathrm{F}}^2 \right) \\ &\quad - \frac{\bar{\mu} \beta}{2(\beta + 1)} \|p^k - p^*\|_{\mathrm{F}}^2. \end{aligned}$$

On the other hand, we also have that for any vector Δ,

$$\begin{aligned} g_i(p^k + \Delta) &= g_i \left(s_i^k + \Delta + p^k - s_i^k \right) \\ &\le g_i(s_i^k) + \langle \nabla g_i(s_i^k), \Delta + p^k - s_i^k \rangle + \frac{L_i}{2} \|\Delta + p^k - s_i^k\|_{\mathrm{F}}^2 \\ &\le g_i(s_i^k) + \langle \nabla g_i(s_i^k), \Delta \rangle + \langle \nabla g_i(s_i^k), p^k - s_i^k \rangle \\ &\quad + \frac{L_i(1 + \eta)}{2} \|\Delta\|_{\mathrm{F}}^2 + \frac{L_i(1 + \eta)}{2\eta} \|p^k - s_i^k\|_{\mathrm{F}}^2 \end{aligned}$$

where $\eta > 0$ is some tunable parameter, and therefore

$$(23) \qquad \begin{aligned} -\langle \nabla g_i(s_i^k), \Delta \rangle &\le -g_i(p^k + \Delta) + g_i(s_i^k) + \langle \nabla g_i(s_i^k), p^k - s_i^k \rangle \\ &\quad + \frac{L_i(1 + \eta)}{2\eta} \|p^k - s_i^k\|_{\mathrm{F}}^2 + \frac{L_i(1 + \eta)}{2} \|\Delta\|_{\mathrm{F}}^2. \end{aligned}$$

Averaging (23) over i through 1 to n gives

$$(24) \qquad \begin{aligned} -\langle \frac{1}{n} \sum_{i=1}^{n} \nabla g_i(s_i^k), \Delta \rangle &\le -g(p^k + \Delta) + \frac{\bar{L}(1 + \eta)}{2} \|\Delta\|_{\mathrm{F}}^2 \\ &\quad + \frac{1}{n} \sum_{i=1}^{n} \left(g_i(s_i^k) + \langle \nabla f_i(s_i^k), p^k - s_i^k \rangle + \frac{L_i(1 + \eta)}{2\eta} \|p^k - s_i^k\|_{\mathrm{F}}^2 \right). \end{aligned}$$

Having (22) and (24) at hand, we are ready to show how r^{k+1} is related to r^k. First, plugging $a = p^{k+1} - p^*$ and $b = p^k - p^{k+1}$ into the basic equality $\|a\|_{\mathrm{F}}^2 = \|a + b\|_{\mathrm{F}}^2 - 2\langle a, b\rangle - \|b\|_{\mathrm{F}}^2$ yields

$$
\begin{aligned}
(r^{k+1})^2 &= (r^k)^2 - 2\langle p^{k+1} - p^*, p^k - p^{k+1}\rangle - \|p^{k+1} - p^k\|_{\mathrm{F}}^2 \\
&\quad \text{(substituting (18))} \\
&= (r^k)^2 - 2\langle p^{k+1} - p^*, \theta\tfrac{1}{n}\sum_{i=1}^n \nabla g_i(s_i^k)\rangle - \|p^{k+1} - p^k\|_{\mathrm{F}}^2 \\
&= (r^k)^2 + 2\theta\langle \tfrac{1}{n}\sum_{i=1}^n \nabla g_i(s_i^k), p^* - p^k\rangle \\
&\quad - 2\theta\langle \tfrac{1}{n}\sum_{i=1}^n \nabla g_i(s_i^k), p^{k+1} - p^k\rangle - \|p^{k+1} - p^k\|_{\mathrm{F}}^2.
\end{aligned}
$$
(25)

Next, in (25), we substitute (22) for the second term, and we substitute (24) with $\Delta = p^{k+1} - p^k$ for the third term. Thus, we obtain that

$$
\begin{aligned}
(r^{k+1})^2 &\leq (r^k)^2 + 2\theta\left(g(p^*) - \tfrac{1}{n}\sum_{i=1}^n \left(g_i(s_i^k) + \langle \nabla g_i(s_i^k), p^k - s_i^k\rangle\right.\right. \\
&\quad \left.\left. - \tfrac{\mu_i\beta}{2}\|s_i^k - p^k\|_{\mathrm{F}}^2\right) - \tfrac{\bar{\mu}\beta}{2(\beta+1)}\|p^k - p^*\|_{\mathrm{F}}^2\right) \\
&\quad + 2\theta\left(-g(p^{k+1}) + \tfrac{1}{n}\sum_{i=1}^n \left(g_i(s_i^k) + \langle \nabla g_i(s_i^k), p^k - s_i^k\rangle\right.\right. \\
&\quad \left.\left. + \tfrac{L_i(1+\eta)}{2\eta}\|p^k - s_i^k\|_{\mathrm{F}}^2\right) + \tfrac{\bar{L}(1+\eta)}{2}\|p^{k+1} - p^k\|_{\mathrm{F}}^2\right) - \|p^{k+1} - p^k\|_{\mathrm{F}}^2 \\
&= (r^k)^2 + 2\theta(g(p^*) - g(p^{k+1})) - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\|p^k - p^*\|_{\mathrm{F}}^2 \\
&\quad + \tfrac{1}{n}\sum_{i=1}^n \left(\tfrac{\theta L_i(1+\eta)}{\eta} + \theta\mu_i\beta\right)\|p^k - s_i^k\|_{\mathrm{F}}^2 - (1 - \theta\bar{L}(1+\eta))\|p^{k+1} - p^k\|_{\mathrm{F}}^2 \\
&\leq \left(1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\right)(r^k)^2 - 2\theta(g(p^{k+1}) - g(p^*)) \\
&\quad + \left(\tfrac{\theta L(1+\eta)}{\eta} + \theta\hat{\mu}\beta\right)\tfrac{1}{n}\sum_{i=1}^n \|p^k - s_i^k\|_{\mathrm{F}}^2 - (1 - \theta\bar{L}(1+\eta))\|p^{k+1} - p^k\|_{\mathrm{F}}^2.
\end{aligned}
$$
(26)

Define $\epsilon^k = \tfrac{1}{n}\sum_{i=1}^n \|p^k - s_i^k\|_{\mathrm{F}}^2$. By choosing $\theta \leq \tfrac{1}{(1+\eta)\bar{L}}$ such that $1 - \theta\bar{L}(1+\eta)$ in (26) is nonnegative, we have

$$
(r^{k+1})^2 \leq \left(1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\right)(r^k)^2 - 2\theta(g(p^{k+1}) - g(p^*)) + \left(\tfrac{\theta L(1+\eta)}{\eta} + \theta\hat{\mu}\beta\right)\epsilon^k.
$$
(27)

Let us look into the last two terms of (27). Noticing that $\bar{\mu} = (1/n)\sum_{i=1}^n \mu_i$ is a strong convexity constant of $g(p)$, there are two possibilities that could happen at time k. Possibility A is that

$$
(r^{k+1})^2 \geq \left(\frac{L(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\beta\right)\epsilon^k,
$$

while possibility B is the opposite, namely that

$$
(r^{k+1})^2 < \left(\frac{L(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\beta\right)\epsilon^k.
$$

If possibility A occurs, we have

$$
2\theta(g(p^{k+1}) - g(p^*)) \geq \theta\bar{\mu}\|p^{k+1} - p^*\|_{\mathrm{F}}^2 = \theta\bar{\mu}(r^{k+1})^2 \geq \left(\tfrac{\theta L(1+\eta)}{\eta} + \theta\hat{\mu}\beta\right)\epsilon^k
$$

which together with (27) implies

$$
(r^{k+1})^2 \leq \left(1 - \frac{\theta\bar{\mu}\beta}{\beta+1}\right)(r^k)^2.
$$

Considering both possibilities A and B, it follows that

$$
(r^{k+1})^2 \leq \max\left\{\left(1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\right)(r^k)^2, \left(\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right)\epsilon^k\right\}.
$$
(28)

Recursively using the inequality (28) we can see that

$$(29) \qquad (r^{k+1})^2 \leq \max\left\{ \left(1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\right)^{k+1} (r^0)^2, \left(\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right) \max_{t=0,\dots,k}\left\{ \left(1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\right)^t \epsilon^{k-t} \right\} \right\}.$$

Taking square root on both sides of (29) gives us

$$(30) \qquad r^{k+1} \leq \left(\sqrt{1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}}\right)^{k+1} r^0 + \left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \max_{t=0,\dots,k}\left\{ \left(\sqrt{1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}}\right)^t \sqrt{\epsilon^{k-t}} \right\}.$$

Choose λ that satisfies (19) so to have $c \triangleq (\lambda)^{-2}\left(1 - \tfrac{\theta\bar{\mu}\beta}{\beta+1}\right) \leq 1$, then from (30) we get

$$
\begin{aligned}
(31) \quad (\lambda)^{-(k+1)} r^{k+1} &\leq (\sqrt{c})^{k+1} r^0 + (\lambda)^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \max_{t=0,\dots,k}\left\{ (\lambda)^{-(k-t)} (\sqrt{c})^t \sqrt{\epsilon^{k-t}} \right\} \\
&\leq r^0 + (\lambda)^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \max_{t=0,\dots,k}\left\{ (\lambda)^{-t} \sqrt{\epsilon^t} \right\}.
\end{aligned}
$$

Further observing that

$$\sqrt{\epsilon^k} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|p^k - s_i^k\|_{\mathrm{F}}^2} \leq \frac{1}{\sqrt{n}}\sum_{i=1}^{n} \|p^k - s_i^k\|_{\mathrm{F}},$$

and combining it with (31), it follows that

$$(32) \qquad (\lambda)^{-(k+1)} r^{k+1} \leq r^0 + (\lambda\sqrt{n})^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \sum_{i=1}^{n} \max_{t=0,\dots,k}\left\{ (\lambda)^{-t} \|p^t - s_i^t\|_{\mathrm{F}} \right\}.$$

Taking $\max_{k=0,1,\dots,K-1}\{\cdot\}$ on both sides of (32) gives

$$
\begin{aligned}
|r|^{\lambda,K} &\leq 2r^0 + (\lambda\sqrt{n})^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \sum_{i=1}^{n} \|p - s_i\|_{\mathrm{F}}^{\lambda,K-1} \\
&\leq 2r^0 + (\lambda\sqrt{n})^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \sum_{i=1}^{n} \|p - s_i\|_{\mathrm{F}}^{\lambda,K}. \qquad \square
\end{aligned}
$$

3.5. The Last Arrow. Now we prove the last arrow of our proof sketch [cf. (9)] in the following lemma. Its establishment will use the error bound on the IGD of Lemma 8, as a key ingredient.

LEMMA 9 (**Algorithm 1: The last arrow $\check{\mathbf{x}} \to \mathbf{q}$**). *Let Assumptions 1, 4, and 5 hold. In addition, assume that the stepsize $\alpha > 0$ and the parameter λ are such that*

$$\sqrt{1 - \frac{\alpha\bar{\mu}\beta}{\beta+1}} \leq \lambda < 1 \quad and \quad \alpha \leq \frac{1}{(1+\eta)\bar{L}},$$

where $\beta > 0$ and $\eta > 0$ are some tunable parameters. Then, we have

$$(33) \qquad \|\mathbf{q}\|_{\mathrm{F}}^{\lambda,K} \leq \left(1 + \tfrac{\sqrt{n}}{\lambda}\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta}\right) \|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} + 2\sqrt{n}\|\overline{x}(0) - x^*\|_{\mathrm{F}} \text{ for any } K = 0, 1, \dots.$$

Proof. First, let us consider the evolution of $\overline{x}(k)$. Noticing that

$$
\begin{aligned}
\overline{y}(k+1) - \tfrac{1}{n}(\nabla\mathbf{f}(\mathbf{x}(k+1)))^\top \mathbf{1} &= \overline{y}(k) - \tfrac{1}{n}(\nabla\mathbf{f}(\mathbf{x}(k)))^\top \mathbf{1} \\
&\vdots \\
&= \overline{y}(0) - \tfrac{1}{n}(\nabla\mathbf{f}(\mathbf{x}(0)))^\top \mathbf{1} \\
&= 0
\end{aligned}
$$

holds for all k, we then have that

$$(34) \qquad \begin{aligned} \overline{x}(k+1) &= \overline{x}(k) - \alpha \overline{y}(k) \\ &= \overline{x}(k) - \alpha \tfrac{1}{n} (\nabla \mathbf{f}(\mathbf{x}(k)))^\top \mathbf{1} \\ &= \overline{x}(k) - \alpha \tfrac{1}{n} \sum_{i=1}^{n} \nabla f_i(x_i(k)). \end{aligned}$$

Applying Lemma 8 to the recursion relation of $\overline{x}$, namely (34), we obtain

$$(35) \qquad \begin{aligned} \|\overline{x} - x^*\|_\mathrm{F}^{\lambda,K} &\leq 2\|\overline{x}(0) - x^*\|_\mathrm{F} + (\lambda\sqrt{n})^{-1} \left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta \right) \sum_{i=1}^{n} \|\overline{x} - x_i\|_\mathrm{F}^{\lambda,K} \\ &\leq 2\|\overline{x}(0) - x^*\|_\mathrm{F} + (\lambda\sqrt{n})^{-1} \left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta \right) \sqrt{n} \sqrt{\sum_{i=1}^{n} \left(\|\overline{x} - x_i\|_\mathrm{F}^{\lambda,K} \right)^2} \\ &\leq 2\|\overline{x}(0) - x^*\|_\mathrm{F} + (\lambda)^{-1} \left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta \right) \|\check{\mathbf{x}}\|_\mathrm{F}^{\lambda,K}. \end{aligned}$$

Since

$$\begin{aligned} \mathbf{q}(k) &= \mathbf{x}(k) - \mathbf{1}(\overline{x}(k))^\top + \mathbf{1}(\overline{x}(k))^\top - \mathbf{x}^* \\ &= \check{\mathbf{x}}(k) + \mathbf{1}(\overline{x}(k) - x^*)^\top, \end{aligned}$$

it follows that

$$(36) \qquad \|\mathbf{q}\|_\mathrm{F}^{\lambda,K} \leq \|\check{\mathbf{x}}\|_\mathrm{F}^{\lambda,K} + \sqrt{n}\|\overline{x} - x^*\|_\mathrm{F}^{\lambda,K}.$$

Substituting (35) into (36) yields (33). $\qquad\qquad \square$

3.6. Linear Convergence of DIGing. We now state our main results on the convergence rates of DIGing. Our first theorem gives an explicit convergence rate for DIGing in terms of the network parameters (B, n, and δ), objective parameters ($\bar{\mu}$ and $\bar{\kappa} = \tfrac{L}{\bar{\mu}}$), and the algorithmic step-size (α).

THEOREM 10 (**Algorithm 1: Explicit geometric rate over time-varying graphs**). *Suppose that Assumptions 1, 4, and 5 hold. Let*

$$\delta = \sup_{k \geq B-1} \left\{ \sigma_{\max} \left\{ \mathbf{W}_B(k) - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top \right\} \right\} \quad and \quad J_1 = 3\bar{\kappa}B^2 \left(1 + 4\sqrt{n}\sqrt{\bar{\kappa}} \right).$$

Then, for any step-size $\alpha \in \left(0, \frac{1.5(1-\delta)^2}{\bar{\mu}J_1} \right]$, the sequence $\{\mathbf{x}(k)\}$ generated by DIGing algorithm converges to the matrix $\mathbf{x}^* = \mathbf{1}(x^*)^\top$, where x^* is the unique optimal solution of problem (1) at a global R-linear (geometric) rate $O(\lambda^k)$, where the parameter λ is given by

$$\lambda = \begin{cases} \sqrt[2B]{1 - \frac{\alpha\bar{\mu}}{1.5}}, & if\ \alpha \in \left(0, \frac{1.5\left(\sqrt{J_1^2 + (1-\delta^2)J_1} - \delta J_1\right)^2}{\bar{\mu}J_1(J_1+1)^2} \right], \\[3mm] \sqrt[B]{\sqrt{\frac{\alpha\bar{\mu}J_1}{1.5}} + \delta}, & if\ \alpha \in \left(\frac{1.5\left(\sqrt{J_1^2 + (1-\delta^2)J_1} - \delta J_1\right)^2}{\bar{\mu}J_1(J_1+1)^2}, \frac{1.5(1-\delta)^2}{\bar{\mu}J_1} \right]. \end{cases}$$

Proof. Let us collect all the relations/arrows at hand [cf. Lemmata 5, 6, 7, and 9]:

i) $\quad \|\mathbf{z}\|_\mathrm{F}^{\lambda,K} \leq \gamma_1 \|\mathbf{q}\|_\mathrm{F}^{\lambda,K} + \omega_1$ where $\gamma_1 = L\left(1 + \tfrac{1}{\lambda}\right)$ and $\omega_1 = 0$;

ii) $\quad \|\check{\mathbf{y}}\|_\mathrm{F}^{\lambda,K} \leq \gamma_2 \|\mathbf{z}\|_\mathrm{F}^{\lambda,K} + \omega_2$ where $\gamma_2 = \frac{\lambda(1-\lambda^B)}{(\lambda^B - \delta)(1-\lambda)}$ and $\omega_2 = \frac{\lambda^B}{\lambda^B - \delta} \sum_{t=1}^{B} \lambda^{1-t}\|\check{\mathbf{y}}(t-1)\|_\mathrm{F}$;

iii) $\quad \|\check{\mathbf{x}}\|_\mathrm{F}^{\lambda,K} \leq \gamma_3 \|\check{\mathbf{y}}\|_\mathrm{F}^{\lambda,K} + \omega_3$ where $\gamma_3 = \frac{\alpha(1-\lambda^B)}{(\lambda^B - \delta)(1-\lambda)}$ and $\omega_3 = \frac{\lambda^B}{\lambda^B - \delta} \sum_{t=1}^{B} \lambda^{1-t}\|\check{\mathbf{x}}(t-1)\|_\mathrm{F}$;

iv) $\quad \|\mathbf{q}\|_\mathrm{F}^{\lambda,K} \leq \gamma_4 \|\check{\mathbf{x}}\|_\mathrm{F}^{\lambda,K} + \omega_4$ where $\gamma_4 = 1 + \frac{\sqrt{n}}{\lambda}\sqrt{\frac{L(1+\eta)}{\bar{\mu}\eta}} + \frac{\hat{\mu}}{\bar{\mu}}\beta$ and $\omega_4 = 2\sqrt{n}\|\overline{x}(0) - x^*\|_\mathrm{F}$.

To apply the small gain theorem (Theorem 3) to get that $\|\mathbf{q}\|_F^\lambda$ is bounded, we need $\gamma_1\gamma_2\gamma_3\gamma_4 < 1$, that is,

$$(37) \qquad \frac{\alpha L(\lambda+1)(1-\lambda^B)^2}{(\lambda^B-\delta)^2(1-\lambda)^2}\left(1 + \frac{\sqrt{n}}{\lambda}\sqrt{\frac{L(1+\eta)}{\bar\mu\eta} + \frac{\hat\mu}{\bar\mu}\beta}\right) < 1,$$

$$(38) \qquad \text{where} \quad \beta > 0 \quad \text{and} \quad \eta > 0,$$

along with other restrictions on parameters that appear in Lemmata 6, 7, and 9:

$$(39) \qquad \delta < \lambda^B < 1;$$

$$(40) \qquad \sqrt{1 - \frac{\alpha\bar\mu\beta}{\beta+1}} \leq \lambda < 1;$$

$$(41) \qquad \text{and} \quad \alpha \leq \frac{1}{(1+\eta)\bar L}.$$

We next use relations (37)–(41) with a specific values for the parameters β, η and the stepsize α, which yields the desired result. Specifically, let $\beta = 2L/\hat\mu$ and $\eta = 1$ in relation (38). By further using $0.5 \leq \lambda < 1$ and $(1-\lambda^B)/(1-\lambda) \leq B$, (37) and (41) together yield

$$(42) \qquad \alpha \leq \frac{(\lambda^B - \delta)^2}{2LB^2\left(1 + 4\sqrt{n}\sqrt{\bar\kappa}\right)}.$$

On the other hand, since $1.5 \geq 1 + 1/\beta$, relation (40) implies that

$$(43) \qquad \alpha \geq \frac{1.5(1-\lambda^2)}{\bar\mu}.$$

Using (42) and (43), it remains to show that there exists $\lambda \in (\sqrt[B]{\delta}, 1)$ [cf. (39)] such that

$$\left[\frac{1.5(1-\lambda^2)}{\bar\mu}, \frac{(\lambda^B-\delta)^2}{2LB^2\left(1 + 4\sqrt{n}\sqrt{\bar\kappa}\right)}\right] \neq \emptyset,$$

which is equivalent to

$$(44) \qquad \left[\frac{1.5(1-\lambda^2)}{\bar\mu}, \frac{1.5(\lambda^B-\delta)^2}{\bar\mu J_1}\right] \neq \emptyset,$$

where $J_1 = 3\bar\kappa B^2\left(1 + 4\sqrt{n}\sqrt{\bar\kappa}\right)$. We consider a smaller interval by enlarging the left-bound of the interval in (44), i.e., we will prove that

$$(45) \qquad \left[\frac{1.5(1-\lambda^{2B})}{\bar\mu}, \frac{1.5(\lambda^B-\delta)^2}{\bar\mu J_1}\right] \neq \emptyset.$$

When λ varies from $\sqrt[B]{\delta}$ to 1, the left-bound of the interval in (45) is monotonically decreasing from $\frac{1.5(1-\delta^2)}{\bar\mu}$ to 0, while its right-bound is monotonically increasing from 0 to $\frac{1.5(1-\delta)^2}{\bar\mu J_1}$. In particular, the relation (45) is valid when λ (as small as the current choice of all parameters can give) is given by

$$\lambda = \sqrt[B]{\frac{\sqrt{J_1^2 + (1-\delta^2)J_1} + \delta}{J_1 + 1}}.$$

Thus, for

$$\alpha \in \left(0, \frac{1.5\left(\sqrt{J_1^2 + (1-\delta^2)J_1} - \delta J_1\right)^2}{\bar{\mu}J_1(J_1+1)^2}\right],$$

we can set $\lambda = \sqrt[2B]{1 - \frac{\alpha\bar{\mu}}{1.5}}$, while for

$$\alpha \in \left(\frac{1.5\left(\sqrt{J_1^2 + (1-\delta^2)J_1} - \delta J_1\right)^2}{\bar{\mu}J_1(J_1+1)^2}, \frac{1.5(1-\delta)^2}{\bar{\mu}J_1}\right],$$

we can use $\lambda = \sqrt[B]{\sqrt{\frac{\alpha\bar{\mu}J_1}{1.5}} + \delta}$. The rest of the statements follow from Theorem 3 and Lemma 4. □

Other possible choices of β, η, α, and λ exist and may give tighter bounds but here we only aim to give an explicit estimation on the rate.

To see how the geometric rate scales with the number of agents, we further have the following corollary.

COROLLARY 11 (**Algorithm 1: Polynomial network scalability**). *Under the same assumptions in Theorem 10, if we further let the mixing matrices* $\mathbf{W}(k)$ *be such that Assumption 3 holds, and the agents choose the step-size to be*

$$(46) \qquad \alpha(\tau) = \frac{3\tau^2}{128B^2n^{4.5}L\sqrt{\bar{\kappa}}} - \frac{1.5}{\bar{\mu}}\left(\frac{\tau^2}{128B^2n^{4.5}\bar{\kappa}^{1.5}}\right)^2,$$

where τ is the smallest nonzero positive element of the nonnegative matrices $\mathbf{W}(k)$ *for all k [cf. Assumption 3]. Then, the sequence* $\{\mathbf{x}(k)\}$ *generated by DIGing converges to the unique optimal solution* $\mathbf{x}^*$ *at a global R-linear (geometric) rate of* $O\left((\lambda(\tau))^k\right)$ *where*

$$\lambda(\tau) = \sqrt[B]{1 - \frac{\tau^2}{128B^2n^{4.5}\bar{\kappa}^{1.5}}}.$$

Proof. Define $\varphi = 1 - \lambda^B$, then requiring that the interval in (45) be nonempty is equivalent to showing that the following inequality has a solution:

$$(1 - (1-\varphi)^2)J_1 \leq (1 - \delta - \varphi)^2.$$

Therefore, we can show that an achievable φ is

$$(47) \qquad \begin{aligned} \varphi &= \frac{2J_1 + 2(1-\delta) - \sqrt{(2J_1 + 2(1-\delta))^2 - 4(1-\delta)^2(J_1+1)}}{2(J_1+1)} \\ &= \frac{2(1-\delta)^2}{2J_1 + 2(1-\delta) + \sqrt{(2J_1 + 2(1-\delta))^2 - 4(1-\delta)^2(J_1+1)}} \\ &\geq \frac{(1-\delta)^2}{2(J_1+1)}. \end{aligned}$$

By Assumption 3, from Lemma 9 of reference [31], we have that $\delta \leq 1 - \frac{\tau}{2n^2}$. Substituting $J_1 = 3B^2\bar{\kappa}\left(1 + 4\sqrt{n}\sqrt{\bar{\kappa}}\right)$ into (47) gives us

$$\varphi \geq \frac{\tau^2}{8n^4(3B^2\bar{\kappa}\left(1 + 4\sqrt{n}\sqrt{\bar{\kappa}}\right) + 1)} \geq \frac{\tau^2}{128B^2n^{4.5}\bar{\kappa}^{1.5}}.$$

Thus the final rate is $\lambda = \sqrt[B]{1 - \frac{\tau^2}{128B^2n^{4.5}\bar{\kappa}^{1.5}}}$, and a step-size to reach this rate is $\alpha = \frac{1.5(1-\lambda^{2B})}{\bar{\mu}} = \frac{3\tau^2}{128B^2n^{4.5}L\sqrt{\bar{\kappa}}} - \frac{1.5}{\bar{\mu}}\left(\frac{\tau^2}{128B^2n^{4.5}\bar{\kappa}^{1.5}}\right)^2.$ □

Corollary 11 explicitly shows how the linear convergence rate of the DIGing algorithm depends on the condition number $\bar{\kappa}$, time-varying graph connectivity constant B, and the network size n. To reach ε-accuracy, the iteration complexity under conditions of Corollary 11 is $O\left(\tau^{-2}B^3n^{4.5}\bar{\kappa}^{1.5}\ln\frac{1}{\varepsilon}\right)$ which is polynomial in the number of agents n. Beyond the more general form of it, the advantage of Theorem 10 is that it explicitly depends on the parameter δ which measures the convergence speed of consensus. Indeed, Corollary 11 uses the bound $\delta \leq 1 - \tau/(2n^2)$ from [31], which may be very conservative since it applies to a rather general class of graphs. Moreover, any further advances in "consensus theory" deriving improved convergence bounds on consensus would immediately translate into improvements via Corollary 11 [cf. (47)], where better bounds would immediately arise with smaller values of δ.

COROLLARY 12 (**Algorithm 1: Iteration complexity under lazy Metropolis mixing**). *Suppose Assumptions 2, 4, and 5 hold. Also assume that the graphs are time-varying, undirected and connected (i.e., $B = 1$). Let each $\mathbf{W}(k)$ be a lazy Metropolis matrix, that is,*

$$
W_{ij}(k) = \left\{
\begin{array}{ll}
1/\left(2\max\{d_i(k), d_j(k)\}\right), & \text{if } (j,i) \in \mathcal{E}, \\
0, & \text{if } (j,i) \notin \mathcal{E} \text{ and } j \neq i, \\
1 - \sum_{l \in \mathcal{N}_i(k)} W_{il}(k), & \text{if } j = i.
\end{array}
\right.
$$

Then, with the agents choosing the step-size $\alpha(\frac{2}{71})$ [cf. (46) with $\tau = \frac{2}{71}$ and $B = 1$], the sequence $\{\mathbf{x}(k)\}$ generated by DIGing converges to the unique optimal solution $\mathbf{x}^*$ at a global R-linear (geometric) rate of $O(\lambda^k)$, where $\lambda = 1 - \frac{1}{161312n^{4.5}\bar{\kappa}^{1.5}}$. In particular, the number of iterations needed to reach ε-accuracy is $O\left(n^{4.5}\bar{\kappa}^{1.5}\ln\frac{1}{\varepsilon}\right)$.

We omit the proof for Corollary 12 since it is essentially the same as the proof for Corollary 11 in addition to which we further used $\delta = 1 - \frac{1}{71n^2}$ from Lemma 2.2 of reference [37].

REMARK 2. *Our analytical framework also applies to Aug-DGM of [59] when its step-size matrix $\mathbf{D}$ is set to αI (see Subsection 2.2.3). We also find that when the graph is well connected, the ATC strategy employed in Aug-DGM can improve the linear rate. But due to space limit we omit any detailed discussion on this aspect. In the following design of a variant of DIGing for directed graphs, we also partially employed the ATC strategy to accelerate the convergence.*

4. Distributed Optimization over Directed Graphs. We now focus our attention to directed graphs. For such graphs we want to design a distributed algorithm that can work with mixing matrices that need not be doubly stochastic. To do so, we employ the idea of push-sum protocol which relaxes the requirement of doubly stochastic mixing matrices to column stochastic matrices to achieve average consensus. We then introduce our algorithm that uses push-sum protocol for tracking the gradient average in time. The resulting algorithm is termed Push-DIGing (Algorithm 2), which we analyze later on in Section 5.

4.1. Motivation. Suppose there are n agents that can communicate over a static strongly connected directed graph. Each agent i initially holds row i of $\mathbf{x}(0) \in \mathbb{R}^{n \times p}$ and would like to calculate the average $\frac{1}{n}\mathbf{1}^\top\mathbf{x}(0)$. To do so, one possible decentralized approach is to construct a doubly stochastic matrix $\mathbf{W}$ and perform updates $\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k)$ starting from $\mathbf{x}(0)$. However, in a general directed graph, construction of a doubly stochastic matrix needs a weight balancing procedure and it is costly [14]. This becomes even less realistic when the graph is time-varying, as the maintenance of a doubly stochastic matrix sequence needs a real time weight balancing.

Instead, if every agent knows its out-degree, it is possible for the agents to construct a column stochastic matrix $\mathbf{C}$ and perform the following recursions with initialization $\mathbf{u}(0) = \mathbf{x}(0)$ and $\mathbf{v}(0) = \mathbf{1}$ to achieve the average (push-sum protocol [18]):

$$
\begin{array}{ll}
\mathbf{u}\text{-update:} & \mathbf{u}(k+1) = \mathbf{C}\mathbf{u}(k); \\
\mathbf{v}\text{-update:} & \mathbf{v}(k+1) = \mathbf{C}\mathbf{v}(k); \ \mathbf{V}(k+1) = \text{diag}\{\mathbf{v}(k+1)\}; \\
\mathbf{x}\text{-update:} & \mathbf{x}(k+1) = (\mathbf{V}(k+1))^{-1}\mathbf{u}(k+1).
\end{array}
$$

Intuitively, noticing that $\overline{u}(k+1) = \overline{u}(k)$ ($\mathbf{u}(k)$ is sum preserving), rows of $\mathbf{u}(k)$ are heading towards scaled averages with uneven scaling ratios across the vertices caused by the non-double stochasticity of $\mathbf{C}$ (the ratios are actually the elements of a right eigenvector of $\mathbf{C}$ corresponding to the eigenvalue 1), while $\mathbf{V}(k)$ is recording the ratios. By applying the recorded ratio inverse $(\mathbf{V}(k))^{-1}$ on $\mathbf{u}(k)$, the algorithm recovers the unscaled average of the rows in $\mathbf{x}(k)$.

4.2. The Push-DIGing Algorithm.
Next we formally state Push-DIGing in *Algorithm 2*.

Algorithm 2: Push-DIGing

Choose step-size $\alpha > 0$ and pick any $\mathbf{x}(0) = \mathbf{u}(0) \in \mathbb{R}^{n \times p}$;
Initialize $\mathbf{y}(0) = \nabla \mathbf{f}(\mathbf{x}(0))$, $\mathbf{v}(0) = \mathbf{1} \in \mathbb{R}^n$, and $\mathbf{V}(0) = \text{diag}\{\mathbf{v}(0)\}$;
for $k = 0, 1, \ldots$ **do**
    $\mathbf{u}(k+1) = \mathbf{C}(k)(\mathbf{u}(k) - \alpha\mathbf{y}(k))$;
    $\mathbf{v}(k+1) = \mathbf{C}(k)\mathbf{v}(k)$; $\mathbf{V}(k+1) = \text{diag}\{\mathbf{v}(k+1)\}$;
    $\mathbf{x}(k+1) = (\mathbf{V}(k+1))^{-1}\mathbf{u}(k+1)$;
    $\mathbf{y}(k+1) = \mathbf{C}(k)\mathbf{y}(k) + \nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k))$;
end for

Looking at the individual agents, the initialization of Push-DIGing uses an arbitrary $x_i(0) = u_i(0) \in \mathbb{R}^p$, and sets $y_i(0) = \nabla f_i(x_i(0))$ and $v_i(0) = 1$ for $i = 1, \ldots, n$. Then, at each iteration k, every agent i performs updates, as follows:

$$u_i(k+1) = C_{ii}(k)(u_i(k) - \alpha y_i(k)) + \sum_{j \in \mathcal{N}_i^{\text{in}}(k)} C_{ij}(k)(u_j(k) - \alpha y_j(k)),$$
$$v_i(k+1) = C_{ii}(k)v_i(k) + \sum_{j \in \mathcal{N}_i^{\text{in}}(k)} C_{ij}(k)v_j(k),$$
$$x_i(k+1) = u_i(k+1)/v_i(k+1),$$
$$y_i(k+1) = C_{ii}(k)y_i(k) + \sum_{j \in \mathcal{N}_i^{\text{in}}(k)} C_{ij}(k)y_j(k) + \nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)),$$

where $\mathcal{N}_i^{\text{in}}(k)$ is the set of agents that can send information to agent i (in-neighbors of agent i) at time k, while $\mathcal{N}_i^{\text{out}}(k)$ is the set of agents that can receive the information from agent i (in-neighbors of agent i) at time k. (Formal definition of the sets $\mathcal{N}_i^{\text{in}}(k)$ and $\mathcal{N}_i^{\text{out}}(k)$ is deferred to Section 5). At every iteration k, each agent i sends its $u_i(k) - \alpha y_i(k)$, $y_i(k)$, and $v_i(k)$ all scaled by $C_{ij}(k)$ to each of its out-neighbors $\mathcal{N}_i^{\text{out}}(k)$, and receives the corresponding messages from its in-neighbors $\mathcal{N}_i^{\text{in}}(k)$. Then, each agent i updates its own $u_i(k+1)$ by summing its own $C_{ii}(k)(u_i(k) - \alpha y_i(k))$ and the received $C_{ij}(k)(u_j(k) - \alpha y_j(k))$ from its in-neighbors $\mathcal{N}_i^{\text{in}}(k)$; a similar strategy applies to the update of $v_i(k+1)$; then $x_i(k+1)$ is given by scaling $u_i(k+1)$ with $(v_i(k+1))^{-1}$; finally each agent i updates its own $y_i(k+1)$ by summing its own $C_{ii}y_i(k)$ and the received $C_{ij}(k)y_j(k)$ from its in-neighbors $\mathcal{N}_i^{\text{in}}(k)$, and accumulating its current local gradient $\nabla f_i(x_i(k+1))$ and subtracting its previous local gradient $\nabla f_i(x_i(k))$ (in order to filter in only the new information contained in the most recent gradient). Unlike DIGing for undirected graphs in which each agent scales the received variables ($x_i(k)$ and $y_i(k)$) and then sums them up, in Push-DIGing, the variables ($u_i(k) - \alpha y_i(k)$, $y_i(k)$, and $v_i(k)$) are scaled before being sent out. This is due to the fact that, over directed graphs, usually a scaling weight $\mathbf{C}_{ij}$ can only be conveniently determined by the out-degree information of agent j which is not available to agent i.

5. Convergence Analysis for Push-DIGing.
In this section we conduct the convergence analysis for Push-DIGing over time-varying directed graphs. Consider a time-varying graph sequence $\{\mathcal{G}^{\text{dir}}(0), \mathcal{G}^{\text{dir}}(1), \ldots\}$. Every graph instance $\mathcal{G}^{\text{dir}}(k)$ consists of a static set of agents $\mathcal{V} = \{1, 2, \ldots, n\}$ and a set $\mathcal{A}(k)$ of time-varying arcs. An arc $(\overrightarrow{j, i}) \in \mathcal{A}(k)$ indicates that agent j can send information to agent i at time (iteration) k. The set of in- and out-neighbors of agent i at time k are defined as $\mathcal{N}_i^{\text{in}}(k) = \left\{j \big| (\overrightarrow{j, i}) \in \mathcal{A}(k)\right\}$ and $\mathcal{N}_i^{\text{out}}(k) = \left\{j \big| (\overrightarrow{i, j}) \in \mathcal{A}(k)\right\}$, respectively.

We make the following two assumptions for the setup of time-varying directed graphs.

ASSUMPTION 6 ($B_\ominus$-**strongly connected graph sequence**). *The time-varying directed graph sequence $\mathcal{G}^{\text{dir}}(k)$ is $B_\ominus$-strongly connected. Specifically, there exists an integer $B_\ominus > 0$ such that for any $k = 0, 1, \ldots$, the directed graph $\mathcal{G}_{B_\ominus}^{\text{dir}}(k) = \left\{\mathcal{V}, \bigcup_{b=k}^{k+B_\ominus-1} \mathcal{A}(b)\right\}$ is strongly connected.*

ASSUMPTION 7 (**Mixing matrix sequence $\{\mathbf{C}(k)\}$**). *For any $k = 0, 1, \ldots$, the mixing matrix $\mathbf{C}(k) = [C_{ij}(k)] \in \mathbb{R}^{n \times n}$ is given by*

$$C_{ij}(k) = \frac{1}{d_j^{\text{out}}(k) + 1} \ \ \text{if} \ (\overrightarrow{j, i}) \in \mathcal{A}(k), \ \text{and otherwise} \ C_{ij}(k) = 0,$$

where $d_j^{\text{out}}(k) = |\mathcal{N}_j^{\text{out}}(k)|$ is the out-degree of agent j at time k.

Assumption 6 has been used in distributed optimization over time-varying directed graphs [30]. Similar to the case of undirected graphs, we may have multiple options for assigning the weights $C_{ij}(k)$. In the existing literature on push-sum consensus protocol, the best understood are the matrices relying on the out-degree information (as stated in Assumption 7), which we use in establishing the bound on convergence rate of push-sum (see Lemma 13 and its proof). Generalizations of Assumption 6 may be of their own interest for the push-sum consensus algorithm, but that is beyond the scope of this paper.

A little algebra shows that the recursion relation of Push-DIGing is equivalent to

$$
(48) \qquad
\begin{aligned}
&\mathbf{v}(k + 1) = \mathbf{C}(k)\mathbf{v}(k), \mathbf{V}(k + 1) = \text{diag}\left\{\mathbf{v}(k + 1)\right\}, \\
&\mathbf{x}(k + 1) = \widetilde{\mathbf{R}}(k)\left(\mathbf{x}(k) - \alpha\mathbf{h}(k)\right), \\
&\mathbf{h}(k + 1) = \widetilde{\mathbf{R}}(k)\mathbf{h}(k) + (\mathbf{V}(k + 1))^{-1}\left(\nabla\mathbf{f}(\mathbf{x}(k + 1)) - \nabla\mathbf{f}(\mathbf{x}(k))\right),
\end{aligned}
$$

where $\widetilde{\mathbf{R}}(k) = (\mathbf{V}(k + 1))^{-1}\mathbf{C}(k)(\mathbf{V}(k))$ and $\mathbf{h}(k) = (\mathbf{V}(k))^{-1}\mathbf{y}(k)$. We note here that, under Assumptions 6 and 7, it can be seen that each matrix $\mathbf{V}(k)$ is invertible, and that

$$(49) \qquad \|\mathbf{V}^{-1}\|_{\max}^1 \triangleq \sup_{k \geq 0} \|(\mathbf{V}(k))^{-1}\|_{\max} \leq n^{nB_{\ominus}},$$

where $B_{\ominus}$ is the graph connectivity constant defined in Assumption 6. The preceding relation follows from Corollary 2(b) of [30] (here we borrow the notation for the special norm defined in (6)). Also, we note that $\widetilde{\mathbf{R}}(k)$ is actually a row stochastic matrix, i.e., every row of $\widetilde{\mathbf{R}}(k)$ sums to 1 (see Lemma 4 of [29]).

In what follows, we will use following notation

$$\mathbf{C}_b(k) \triangleq \mathbf{C}(k)\mathbf{C}(k - 1) \cdots \mathbf{C}(k + 1 - b)$$

for any $k = 0, 1, \ldots$ and $b = 0, 1, \ldots$ with the convention that $\mathbf{C}_b(k) = I$ for any needed $k < 0$ and $\mathbf{C}_0(k) = I$ for any k. The same notation rule applies to $\widetilde{\mathbf{R}}(k)$. In the sequel, we will give an upper bound on a norm of $(I - \mathbf{1}\mathbf{1}^\top/n)\widetilde{\mathbf{R}}_B(k)$, as provided in the next lemma. This lemma comes from the properties of push-sum protocol and can be obtained from references [33, 35].

LEMMA 13 (*B-step consensus contraction*). *Under Assumptions 6 and 7, let B be an integer satisfying $B \geq B_{\ominus}$ and such that*

$$\delta \triangleq Q_1(1 - \tilde{\tau}^{nB_{\ominus}})^{\frac{B-1}{nB_{\ominus}}} < 1,$$

where Q_1 and $\tilde{\tau}$ are given by

$$(50) \qquad Q_1 \triangleq 2n\frac{1 + \tilde{\tau}^{-nB_{\ominus}}}{1 - \tilde{\tau}^{nB_{\ominus}}} \quad \text{and} \quad \tilde{\tau} \triangleq \frac{1}{n^{2+nB_{\ominus}}}.$$

Then, for any $k = B - 1, B, \ldots$ and any matrix $\mathbf{b}$ with appropriate dimensions, if $\mathbf{a} = \widetilde{\mathbf{R}}_B(k)\mathbf{b}$, i.e., $\mathbf{a} = (\mathbf{V}(k + 1))^{-1}\mathbf{C}_B(k)\mathbf{V}(k + 1 - B)\mathbf{b}$, we have $\|\mathbf{a}\|_{\mathbf{L}} \leq \delta\|\mathbf{b}\|_{\mathbf{L}}$.

Proof. It has been shown in Lemma 4 of [30] that $\widetilde{\mathbf{R}}(k) = (\mathbf{V}(k + 1))^{-1}\mathbf{C}(k)\mathbf{V}(k)$ is a row stochastic matrix with entries being $\tilde{R}_{ij}(k) = C_{ij}(k)v_j(k)/v_i(k + 1)$, where $v_i(k)$ denotes the i-th

entry of the vector $\mathbf{v}(k)$. Denote $\tilde{\mathbf{d}}(k, B) \triangleq (\mathbf{1}^\top \widetilde{\mathbf{R}}_B(k))^\top / n$ which is a stochastic vector. Then it follows that

$$
\begin{aligned}
\|\mathbf{a}\|_{\mathbf{L}} &= \|(I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\widetilde{\mathbf{R}}_B(k)\mathbf{b}\|_{\mathrm{F}} \\
&= \|(\widehat{\mathbf{R}}_B(k) - \mathbf{1}(\tilde{\mathbf{d}}(k, B))^\top)(I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{b}\|_{\mathrm{F}} \\
&\leq \|\widetilde{\mathbf{R}}_B(k) - \mathbf{1}(\tilde{\mathbf{d}}(k, B))^\top\|_2 \|\mathbf{b}\|_{\mathbf{L}}.
\end{aligned}
$$

Now we can focus on establishment of an upper bound for $\|\widetilde{\mathbf{R}}_B(k) - \mathbf{1}(\tilde{\mathbf{d}}(k, B))^\top\|_2$. The first step is to analyse the constituent of $\tilde{R}_{ij}(k) = C_{ij}(k)v_j(k)/v_i(k+1)$. From Corollary 2(b) of reference [29], we know that $v_j(k) \geq \frac{1}{n^{nB_\ominus}}$. Also, obviously we have $1/v_j(k) \geq 1/n$, and for any $(\overrightarrow{j, i}) \in \mathcal{A}(k)$, $C_{ij}(k) \geq 1/n$. Based on those bounds on $C_{ij}(k)$, $v_j(k)$, and $1/v_i(k+1)$, we can have the following equivalent arc-utilization lower bound

$$
\tilde{R}_{ij}(k) \geq \tilde{\tau} \triangleq \frac{1}{n^{2+nB_\ominus}} \text{ if } (\overrightarrow{j, i}) \in \mathcal{A}(k).
$$

Thus, we have

(51)
$$
\begin{aligned}
\|\widetilde{\mathbf{R}}_B(k) - \mathbf{1}(\tilde{\mathbf{d}}(k, B))^\top\|_2 &\leq n\|\widetilde{\mathbf{R}}_B(k) - \mathbf{1}(\tilde{\mathbf{d}}(k, B))^\top\|_{\max} \\
&\leq 2n\frac{1+\tilde{\tau}^{-nB_\ominus}}{1-\tilde{\tau}^{nB_\ominus}}\left(1 - \tilde{\tau}^{nB_\ominus}\right)^{\frac{B-1}{nB_\ominus}},
\end{aligned}
$$

where the second inequality comes from Lemma 5 of reference [35]. By (9) and (51), we get $\|\mathbf{a}\|_{\mathbf{L}} \leq Q_1 \left(1 - \tilde{\tau}^{nB_\ominus}\right)^{\frac{B-1}{nB_\ominus}} \|\mathbf{b}\|_{\mathbf{L}}$ with $Q_1 \triangleq 2n\frac{1+\tilde{\tau}^{-nB_\ominus}}{1-\tilde{\tau}^{nB_\ominus}}$. $\qquad\square$

The convergence analysis of Push-DIGing will be based on analogous recursions illustrated in (48). Similar to the proof in Section 3, we will follow the proof sketch of the small gain theorem around the cycle:

(52)
$$
\text{Algorithm 2: } \mathbf{q} \to \mathbf{z} \to \check{\mathbf{h}} \to \check{\mathbf{x}} \to \mathbf{q}.
$$

REMARK 3. *In consensus-based algorithms for optimization over directed graphs, it is difficult to construct monotonically decreasing Lyapunov functions for convergence analysis [38] due to the presence of asymmetric operators in the iterations (arising from the asymmetric weight matrices). Also, to deal with time-varying graphs, one has to resort to time-varying or ergodic metrics [51] which are not easy to construct when the consensus protocol is combined with an optimization algorithm. In this situation, conventional approaches that heavily rely on every step contraction for proving Q-linear convergence are usually inapplicable. However, by defining a special metric and utilizing the small gain theorem, we manage to conveniently analyze the introduced algorithms and establish their linear convergence rates, but without relying on the monotonic decay of any Lyapunov function associated with the recursion.*

5.1. The Establishment of Each Arrow. Noticing that that Lemma 5 is a simple consequence of Assumption 4, so it also holds for Push-DIGing. For the sake of reference convenience, we restate it as follows without proof.

LEMMA 14 (**Algorithm 2: The first arrow $\mathbf{q} \to \mathbf{z}$**). *Under Assumption 4, we have that for all $K = 0, 1, \ldots$ and any $\lambda \in (0, 1)$,*

$$
\|\mathbf{z}\|_{\mathrm{F}}^{\lambda, K} \leq L\left(1 + \frac{1}{\lambda}\right)\|\mathbf{q}\|_{\mathrm{F}}^{\lambda, K}.
$$

The next two lemmata are provided by doing almost identical arguments as those for Lemmata 6 and 7: indeed, by noticing the similarity between the equivalent recursion of Push-DIGing (48) and the recursion of DIGing (see Algorithm 1), similar bounds to those in Lemmata 6 and 7 should be obtained by an application of Lemma 13, which shows how multiplication by a row stochastic matrix $\widetilde{\mathbf{R}}(k)$ shrinks the distance to the consensus subspace.

LEMMA 15 (**Algorithm 2: The second arrow $\mathbf{z} \to \check{\mathbf{h}}$**). *Let Assumptions 6 and 7 hold, and let λ be such that $\delta < \lambda^B < 1$, where B is the constant provided in Lemma 13. Then, we have*

$$(53) \qquad \|\check{\mathbf{h}}\|_{\mathrm{F}}^{\lambda,K} \leq \frac{Q_1 \|\mathbf{V}^{-1}\|_{\max}^1 \lambda(1-\lambda^B)}{(\lambda^B - \delta)(1-\lambda)} \|\mathbf{z}\|_{\mathrm{F}}^{\lambda,K} + \frac{\lambda^B}{\lambda^B - \delta} \sum_{t=1}^{B} \lambda^{1-t} \|\check{\mathbf{h}}(t-1)\|_{\mathrm{F}} \quad \text{for all } K = 0, 1, \dots,$$

where Q_1 and $\|\mathbf{V}^{-1}\|_{\max}^1$ are the constants defined by (50) and (49), respectively.

Proof. The equivalent recursion of Push-DIGing involving $\mathbf{h}$ and $\mathbf{z}$ is

$$(54) \qquad \mathbf{h}(k+1) = \widetilde{\mathbf{R}}(k)\mathbf{h}(k) + (\mathbf{V}(k+1))^{-1}\mathbf{z}(k+1).$$

From (54), using Lemma 13, for all $k \geq B - 1$, it follows that

$$(55) \quad \begin{aligned} \|\check{\mathbf{h}}(k+1)\|_{\mathrm{F}} &= \|\mathbf{h}(k+1)\|_{\mathbf{L}} \\ &\leq \left\|\widetilde{\mathbf{R}}_B(k)\mathbf{h}(k+1-B)\right\|_{\mathbf{L}} + \left\|\widetilde{\mathbf{R}}_{B-1}(k)(\mathbf{V}(k+2-B))^{-1}\mathbf{z}(k+2-B)\right\|_{\mathbf{L}} \\ &\quad + \cdots + \|\widetilde{\mathbf{R}}_1(k)(\mathbf{V}(k))^{-1}\mathbf{z}(k)\|_{\mathbf{L}} + \|\widetilde{\mathbf{R}}_0(k)(\mathbf{V}(k+1))^{-1}\mathbf{z}(k+1)\|_{\mathbf{L}} \\ &\leq \delta\|\check{\mathbf{h}}(k+1-B)\|_{\mathrm{F}} + Q_1 \sum_{t=1}^{B} \|(\mathbf{V}(k+2-t))^{-1}\mathbf{z}(k+2-t)\|_{\mathrm{F}} \\ &\leq \delta\|\check{\mathbf{h}}(k+1-B)\|_{\mathrm{F}} + Q_1\|\mathbf{V}^{-1}\|_{\max}^1 \sum_{t=1}^{B} \|\mathbf{z}(k+2-t)\|_{\mathrm{F}}, \end{aligned}$$

where Q_1 and $\|\mathbf{V}^{-1}\|_{\max}^1$ are the constants defined in (50) and (49), respectively.

Noticing the similarity between (55) and (14), by the same argument as we have applied in the proof of Lemma 6 starting from (15), we can obtain (53). $\qquad \square$

LEMMA 16 (**Algorithm 2: The third arrow $\check{\mathbf{h}} \to \check{\mathbf{x}}$**). *Let Assumptions 6 and 7 hold, and let λ be such that $\delta < \lambda^B < 1$, where B is the constant provided in Lemma 13. Then, we have*

$$(56) \qquad \|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} \leq \frac{\alpha}{\lambda^B - \delta}\left(\delta + Q_1\frac{1-\lambda^{B-1}}{1-\lambda}\right)\|\check{\mathbf{h}}\|_{\mathrm{F}}^{\lambda,K} + \frac{\lambda^B}{\lambda^B - \delta}\sum_{t=1}^{B}\lambda^{1-t}\|\check{\mathbf{x}}(t-1)\|_{\mathrm{F}}$$

for all $K = 0, 1, \dots$, where Q_1 is the constant as introduced in Lemma 15 (see (50)).

The equivalent recursions of Push-DIGing involving $\mathbf{x}$ and $\mathbf{h}$ is

$$(57) \qquad \mathbf{x}(k+1) = \widetilde{\mathbf{R}}(k)\left(\mathbf{x}(k) - \alpha\mathbf{h}(k)\right).$$

Noticing the similarity between (57) and (54), with almost identical argument as that illustrated in the proof of Lemma 15, we can get (56).

Similar to the proof of Lemma 9, Lemma 8 also serves as a key ingredient in establishing the last arrow of the proof sketch for Push-DIGing. We state it in the form of a lemma as follows.

LEMMA 17 (**Algorithm 2: The last arrow $\check{\mathbf{x}} \to \mathbf{q}$**). *Let Assumptions 4, 5, and 7 hold. Also, assume that*

$$\sqrt{1 - \frac{\alpha\bar{\mu}\beta}{\beta+1}} \leq \lambda < 1 \quad \text{and} \quad \alpha \leq \frac{1}{(1+\eta)\bar{L}}$$

where $\beta > 0$ and $\eta > 0$ are some tunable parameters. Then, we have that for all $K = 0, 1, \dots$,

$$(58) \qquad \|\mathbf{q}\|_{\mathrm{F}}^{\lambda,K} \leq (1 + \sqrt{n})\left(1 + \frac{\sqrt{n}}{\lambda}\sqrt{\frac{L(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\beta}\right)\|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} + 2\sqrt{n}\|\overline{x}(0) - x^*\|_{\mathrm{F}}.$$

Proof. First, by the same argument as in the proof of Lemma 9, we have

$$\overline{u}(k+1) = \overline{u}(k) - \alpha \frac{1}{n} \sum_{i=1}^{n} f_i(x_i(k)). \tag{59}$$

Applying Lemma 8 to the recursion relation of $\overline{u}$, namely (59), we obtain

$$\|\overline{u} - x^*\|_{\mathrm{F}}^{\lambda,K} \le 2\|\overline{u}(0) - x^*\|_{\mathrm{F}} + (\lambda\sqrt{n})^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right) \sum_{i=1}^{n} \|\overline{u} - x_i\|_{\mathrm{F}}^{\lambda,K}. \tag{60}$$

Let us look into the summation in the last term of (60). Since $\mathbf{u}(k) = \mathbf{V}(k)\mathbf{x}(k)$, it follows that

$$
\begin{aligned}
\sum_{i=1}^{n} \|\overline{u} - x_i\|_{\mathrm{F}}^{\lambda,K} &= \sum_{i=1}^{n} \|(\overline{u} - \overline{x}) + (\overline{x} - x_i)\|_{\mathrm{F}}^{\lambda,K} \\
&\le n\|\overline{u} - \overline{x}\|_{\mathrm{F}}^{\lambda,K} + \sum_{i=1}^{n} \|\overline{x} - x_i\|_{\mathrm{F}}^{\lambda,K} \\
&\le n\|\tfrac{1}{n}\mathbf{x}^\top\mathbf{V}\mathbf{1} - \tfrac{1}{n}\mathbf{x}^\top\mathbf{1}\|_{\mathrm{F}}^{\lambda,K} + \sqrt{n}\|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} \\
&= \|(\mathbf{1} - \mathbf{v})^\top\mathbf{x}\|_{\mathrm{F}}^{\lambda,K} + \sqrt{n}\|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K}.
\end{aligned} \tag{61}
$$

Thus, by (60) and (61) we have

$$
\begin{aligned}
\|\overline{u} - x^*\|_{\mathrm{F}}^{\lambda,K} &\le 2\|\overline{x}(0) - x^*\|_{\mathrm{F}} + (\lambda)^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right)\|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} \\
&\quad + (\lambda\sqrt{n})^{-1}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right)\|(\mathbf{1} - \mathbf{v})^\top\mathbf{x}\|_{\mathrm{F}}^{\lambda,K}
\end{aligned}
$$

Since

$$
\begin{aligned}
\mathbf{q}(k) &= \mathbf{x}(k) - \mathbf{1}(\overline{x}(k))^\top + \mathbf{1}(\overline{x}(k))^\top - \mathbf{1}(\overline{u}(k))^\top + \mathbf{1}(\overline{u}(k))^\top - \mathbf{x}^* \\
&= \check{\mathbf{x}}(k) + \tfrac{1}{n}\mathbf{1}(\mathbf{1} - \mathbf{v}(k))^\top\mathbf{x}(k) + \mathbf{1}(\overline{u}(k) - x^*)^\top,
\end{aligned}
$$

it follows that

$$
\begin{aligned}
\|\mathbf{q}\|_{\mathrm{F}}^{\lambda,K} &\le \|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} + (\sqrt{n})^{-1}\|(\mathbf{1} - \mathbf{v})^\top\mathbf{x}\|_{\mathrm{F}}^{\lambda,K} + \sqrt{n}\|\overline{u} - x^*\|_{\mathrm{F}}^{\lambda,K} \\
&\le \left(1 + \tfrac{\sqrt{n}}{\lambda}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right)\right)\|\check{\mathbf{x}}\|_{\mathrm{F}}^{\lambda,K} + 2\sqrt{n}\|\overline{x}(0) - x^*\|_{\mathrm{F}} \\
&\quad + \left((\sqrt{n})^{-1} + \tfrac{1}{\lambda}\left(\sqrt{\tfrac{L(1+\eta)}{\bar{\mu}\eta}} + \tfrac{\hat{\mu}}{\bar{\mu}}\beta\right)\right)\|(\mathbf{1} - \mathbf{v})^\top\mathbf{x}\|_{\mathrm{F}}^{\lambda,K}.
\end{aligned} \tag{62}
$$

Finally, we can bound the last term in (62) as follows

$$
\begin{aligned}
\|(\mathbf{1} - \mathbf{v}(k))^\top\mathbf{x}(k)\|_{\mathrm{F}} &= \|(\mathbf{1} - \mathbf{v}(k))^\top(I - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{x}(k)\|_{\mathrm{F}} \\
&\le \sqrt{n^2 - n}\|\check{\mathbf{x}}(k)\|_{\mathrm{F}} \\
&\le n\|\check{\mathbf{x}}(k)\|_{\mathrm{F}}.
\end{aligned} \tag{63}
$$

Substituting (63) into (62) gives (58). $\qquad\square$

5.2. Linear Convergence of Push-DIGing.
Next, we provide a convergence rate estimate for the Push-DIGing algorithm.

THEOREM 18 (**Algorithm 2: Explicit geometric rate over time-varying directed graphs**). *Suppose Assumptions 4, 5, 6, and 7 hold. Let B be a large enough integer constant such that*

$$\delta \triangleq Q_1\left(1 - \frac{1}{n^{(2+nB_\ominus)nB_\ominus}}\right)^{\frac{B-1}{nB_\ominus}} < 1.$$

Also define the constant J_2 as follows:

$$J_2 = 3Q_1\|\mathbf{V}^{-1}\|_{\max}^1\bar{\kappa}B(\delta + Q_1(B-1))(1 + \sqrt{n})\left(1 + 4\sqrt{n}\sqrt{\bar{\kappa}}\right).$$

Then, for any step-size $\alpha \in \left(0, \frac{1.5(1-\delta)^2}{\bar{\mu}J_2}\right]$, *the sequence* $\{\mathbf{x}(k)\}$ *generated by Push-DIGing converges to the unique optimal solution* $\mathbf{x}^*$ *at a global R-linear (geometric) rate* $O(\lambda^k)$, *where* λ *is given by*

$$
\lambda = \begin{cases} \sqrt[2B]{1 - \frac{\alpha\bar{\mu}}{1.5}}, & \text{if } \alpha \in \left(0, \frac{1.5\left(\sqrt{J_2^2 + (1-\delta^2)J_2} - \delta J_2\right)^2}{\bar{\mu}J_2(J_2+1)^2}\right], \\ \sqrt[B]{\sqrt{\frac{\alpha\bar{\mu}J_2}{1.5}} + \delta}, & \text{if } \alpha \in \left(\frac{1.5\left(\sqrt{J_2^2 + (1-\delta^2)J_2} - \delta J_2\right)^2}{\bar{\mu}J_2(J_2+1)^2}, \frac{1.5(1-\delta)^2}{\bar{\mu}J_2}\right]. \end{cases}
$$

Proof. The proof is similar to that of Theorem 10. Specifically, we collect all the gains as follows:

i) $\gamma_1 = L\left(1 + \frac{1}{\lambda}\right)$;

ii) $\gamma_2 = \frac{Q_1\|\mathbf{V}^{-1}\|_{\max}^1\lambda(1-\lambda^B)}{(\lambda^B - \delta)(1-\lambda)}$;

iii) $\gamma_3 = \frac{\alpha}{\lambda^B - \delta}\left(\delta + Q_1\frac{1-\lambda^{B-1}}{1-\lambda}\right)$;

iv) $\gamma_4 = (1 + \sqrt{n})\left(1 + \frac{\sqrt{n}}{\lambda}\sqrt{\frac{L(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\beta}\right)$.

To apply the small gain theorem (Theorem 3), we need $\gamma_1\gamma_2\gamma_3\gamma_4 < 1$, that is,

$$
\frac{Q_1\|\mathbf{V}^{-1}\|_{\max}^1\alpha L(1+\lambda)}{(\lambda^B - \delta)^2}\left(\frac{1-\lambda^B}{1-\lambda}\right)\left(\delta + Q_1\frac{1-\lambda^{B-1}}{1-\lambda}\right)(1 + \sqrt{n})\left(1 + \frac{\sqrt{n}}{\lambda}\sqrt{\frac{L(1+\eta)}{\bar{\mu}\eta} + \frac{\hat{\mu}}{\bar{\mu}}\beta}\right) < 1.
$$

The other restrictions on α, β, η and λ are the same as those in (38), (39), (40), and (41). Choosing $\beta = 2L/\hat{\mu}$ and $\eta = 1$, and further using $0.5 \leq \lambda < 1$ and $(1 - \lambda^{B-1})/(1 - \lambda) \leq B - 1$, relations (12) and (41) together imply that

(64) $$\alpha \leq \frac{(\lambda^B - \delta)^2}{2Q_1\|\mathbf{V}^{-1}\|_{\max}^1 LB(\delta + Q_1(B-1))(1 + \sqrt{n})\left(1 + 4\sqrt{n}\sqrt{\kappa}\right)}.$$

Next, similar to the proof of Lemma 4 [cf. (45)], it remains to show that there exists some $\lambda \in (\sqrt[B]{\delta}, 1)$ such that

(65) $$\left[\frac{1.5(1 - \lambda^{2B})}{\bar{\mu}}, \frac{1.5(\lambda^B - \delta)^2}{\bar{\mu}J_2}\right] \neq \emptyset,$$

where $J_2 = 3Q_1\|\mathbf{V}^{-1}\|_{\max}^1\bar{\kappa}B(\delta + Q_1(B-1))(1 + \sqrt{n})\left(1 + 4\sqrt{n}\sqrt{\kappa}\right)$. Noticing the similarity between (65) and (45), by the same argument as in the proof of Theorem 10 (starting from (45)), we can obtain the statement of the theorem. $\square$

6. Numerical Experiments. Consider a decentralized estimation problem: each agent $i \in \{1, \ldots, n\}$ has its own observation y_i given by $y_i = M_i x + e_i$, where $y_i \in \mathbb{R}^{m_i}$ and $M_i \in \mathbb{R}^{m_i \times p}$ are known data, $x \in \mathbb{R}^p$ is unknown, and $e_i \in \mathbb{R}^{m_i}$ is some noise. The goal is to estimate x. In this experiment we use the Huber loss, which is known to be robust to outliers, and it allows us to observe both sublinear and linear convergence. The corresponding optimization problem is: $\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^n \left\{\sum_{j=1}^{m_i} H_\xi(M_{i,j}x - y_{i,j})\right\}$, where $M_{i,j}$ and $y_{i,j}$ are the j-th row of matrix M_i and vector y_i, respectively. The Huber loss function H_ξ is defined by

$$
H_\xi(a) = \begin{cases} \frac{1}{2}a^2, & \text{if } |a| \leq \xi \ (\ell_1 \text{ zone}), \\ \xi(|a| - \frac{1}{2}\xi), & \text{otherwise } (\ell_2^2 \text{ zone}). \end{cases}
$$

In all experiments, we set $n = 12$, $m_i = 1$ for all i, and $p = 3$. The data M_i, as well as the noise e_i, $\forall i$, are firstly generated following the standard normal distribution and then re-normalized so that $L_i = 1$ for all i. The parameter ξ is set to 2. The algorithm starts from $x_i(0) = 0$ for all i. We scale e_i so that $x_i(0)$ is located in the ℓ_1 zone for all agents i. The optimal solution x^* is randomly but artificially selected so that x^* is in the ℓ_2^2 zone and $\|x^* - x_i(0)\| = 300$, $\forall i$.
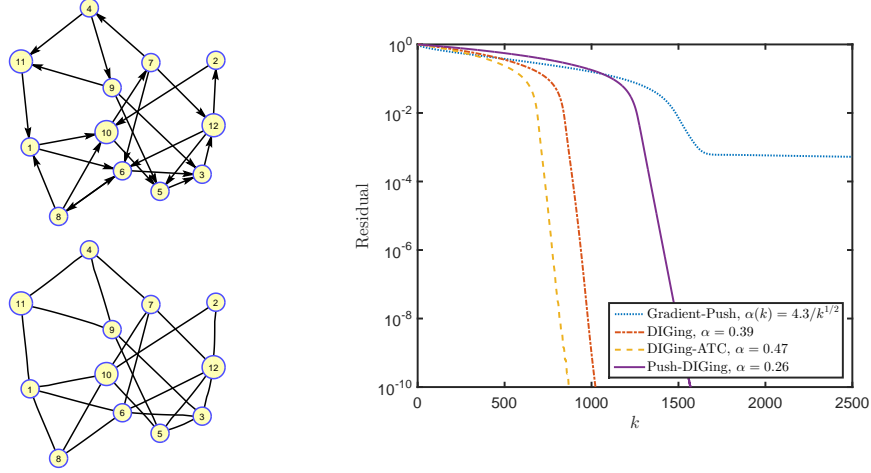
FIG. 1. *The plots are showing the underlying directed and undirected graphs for experiments. The plot to the right shows the residuals $\frac{\|\mathbf{x}(k)-\mathbf{x}^*\|_{\mathrm{F}}}{\|\mathbf{x}(0)-\mathbf{x}^*\|_{\mathrm{F}}}$ for the time-invariant directed graph illustrated on the top-left. Step-sizes have been* hand-optimized *to give faster convergence and more accurate solution for all algorithms.*
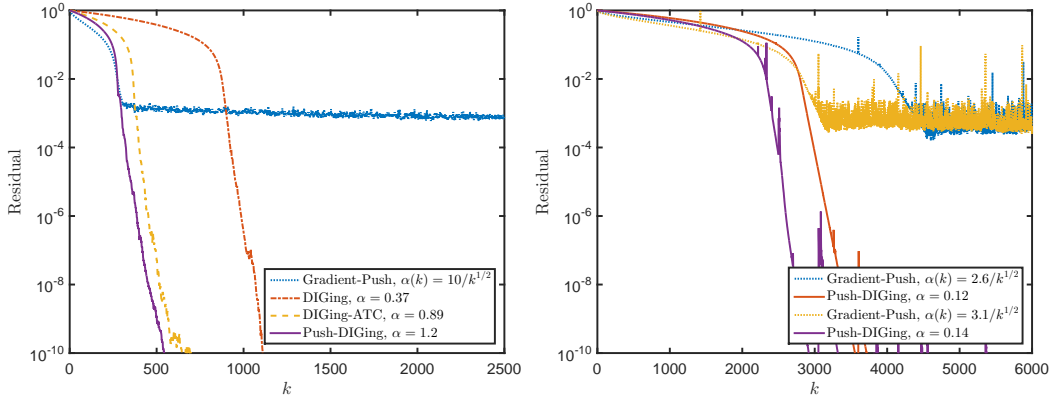

FIG. 2. *The plots to the left and to the right are showing the residuals $\frac{\|\mathbf{x}(k)-\mathbf{x}^*\|_{\mathrm{F}}}{\|\mathbf{x}(0)-\mathbf{x}^*\|_{\mathrm{F}}}$ for a time-varying undirected graph sequence and a time-varying directed graph sequence, respectively. Step-sizes have been* hand-optimized *to give faster convergence and more accurate solution for all algorithms.*

We conduct three experiments: (i) Time-invariant directed graphs; (ii) Time-varying undirected graphs; and (iii) Time-varying directed graphs. For (i), the underlying directed graph $\mathcal{G}_{\mathrm{TI}}^{\mathrm{dir}} = \{\mathcal{V}, \mathcal{A}\}$ is illustrated in the top left of Fig. 1. $\mathcal{G}_{\mathrm{TI}}^{\mathrm{dir}}$ is randomly generated with 24 arcs and 12 vertices, and guaranteed to be strongly connected. The mixing matrix $\mathbf{W}$ is chosen as the fastest (asymmetric) linear consensus matrix [56]. For (ii), the underlying undirected graph $\mathcal{G}_{\mathrm{TI}}^{\mathrm{un}} = \{\mathcal{V}, \mathcal{E}\}$ is generated by simply taking out the direction of every arc of $\mathcal{G}_{\mathrm{TI}}^{\mathrm{dir}}$ (see the bottom left of Fig. 1). $\mathcal{G}_{\mathrm{TI}}^{\mathrm{un}}$ has 23 edges in total. At iteration k, we generate the graph instance of the time-varying undirected graph $\mathcal{G}_{\mathrm{TV}}^{\mathrm{un}}(k) = \{\mathcal{V}, \mathcal{E}(k)\}$ through randomly uniformly sampling $\mathcal{E}(k)$ from $\mathcal{E}$ with 40 percent. The mixing matrix $\mathbf{W}(k)$ is generated through Metropolis weights. For (iii), we generate the graph instance of the time-varying directed graph $\mathcal{G}_{\mathrm{TV}}^{\mathrm{dir}}(k) = \{\mathcal{V}, \mathcal{A}(k)\}$ through randomly uniformly sampling $\mathcal{A}(k)$ from $\mathcal{A}$ of $\mathcal{G}_{\mathrm{TI}}^{\mathrm{dir}}$ with 80 percent. In all experiments, the mixing matrix $\mathbf{C}(k)$ for Push-DIGing is generated based on the out-degree information [30].

We plot push-gradient method [29], DIGing, DIGing-ATC (Aug-DGM with step-size matrix αI), and Push-DIGing in Fig. 1 and Fig. 2. In the experiments, we observe that DIGing and its variants

all have R-linear rates while push-gradient method only has sublinear rate even if the objective is smooth and strongly convex.

7. Conclusion. In this paper, we considered a class of protocols for distributed optimization based on the idea of "distributed inexact gradient" and "gradient tracking". Under strong convexity, we studied the convergence rates of the algorithms over time-varying directed/undirected graphs. Using the small gain theorem, we showed that our protocols converge at some global R-linear rates for strongly convex functions, and were able to obtain explicit bounds on the rates.

An open question is to obtain improved estimates on the convergence rates of our method, especially as far as scaling with the number of agents, n, goes. Furthermore, extensions to more complex optimization models containing local constraints, couplings among agents in the objectives would be of considerable interest.

REFERENCES

[1] J. BAZERQUE AND G. GIANNAKIS, *Distributed Spectrum Sensing for Cognitive Radio Networks by Exploiting Sparsity*, IEEE Transactions on Signal Processing, 58 (2010), pp. 1847–1862.
[2] D. BERTSEKAS, *Distributed Asynchronous Computation of Fixed Points*, Mathematical Programming, 27 (1983), pp. 107–120.
[3] D. P. BERTSEKAS, *Incremental proximal methods for large scale convex optimization*, Mathematical Programming, 129 (2011), pp. 163–195.
[4] D. P. BERTSEKAS, *Incremental aggregated proximal and augmented lagrangian algorithms*, tech. report, Laboratory for Information and Decision Systems Report LIDS-P-3176, MIT, 2015.
[5] D. P. BERTSEKAS AND J. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Nashua, 2nd ed., 1997.
[6] K. CAI AND H. ISHII, *Average Consensus on Arbitrary Strongly Connected Digraphs with Time-Varying Topologies*, IEEE Transactions on Automatic Control, 59 (2014), pp. 1066–1071.
[7] T. CHANG, M. HONG, AND X. WANG, *Multi-Agent Distributed Optimization via Inexact Consensus ADMM*, arXiv preprint arXiv:1402.6065, (2014).
[8] A. CHEN AND A. OZDAGLAR, *A Fast Distributed Proximal-Gradient Method*, in the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2012, pp. 601–608.
[9] C. DESOER AND M. VIDYASAGAR, *Feedback Systems: Input-Output Properties*, vol. 55, Siam, 2009.
[10] O. DEVOLDER, F. GLINEUR, AND Y. NESTEROV, *First-Order Methods with Inexact Oracle: The strongly convex case*, tech. report, UCL, 2013.
[11] J. DUCHI, A. AGARWAL, AND M. WAINWRIGHT, *Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling*, IEEE Transactions on Automatic Control, 57 (2012), pp. 592–606.
[12] P. FORERO, A. CANO, AND G. GIANNAKIS, *Consensus-Based Distributed Support Vector Machines*, Journal of Machine Learning Research, 59 (2010), pp. 1663–1707.
[13] L. GAN, U. TOPCU, AND S. LOW, *Optimal Decentralized Protocol for Electric Vehicle Charging*, IEEE Transactions on Power Systems, 28 (2013), pp. 940–951.
[14] B. GHARESIFARD AND J. CORTES, *Distributed Strategies for Generating Weight-Balanced and Doubly Stochastic Digraphs*, European Journal of Control, 18 (2012), pp. 539–557.
[15] M. GURBUZBALABAN, A. OZDAGLAR, AND P. PARRILO, *On the convergence rate of incremental aggregated gradient algorithms.* available on arxiv at http://arxiv.org/abs/1506.02081, 2015.
[16] M. HONG AND T. CHANG, *Stochastic Proximal Gradient Consensus over Random Networks*, arXiv preprint arXiv:1511.08905, (2015).
[17] D. JAKOVETIC, J. XAVIER, AND J. MOURA, *Fast Distributed Gradient Methods*, IEEE Transactions on Automatic Control, 59 (2014), pp. 1131–1146.
[18] D. KEMPE, A. DOBRA, AND J. GEHRKE, *Gossip-Based Computation of Aggregate Information*, in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003, pp. 482–491.
[19] J. KOSHAL, A. NEDIĆ, AND U. V. SHANBHAG, *Distributed algorithms for aggregative games on graphs*, Operations Research, 64 (2016), pp. 680–704.
[20] P. LORENZO AND G. SCUTARI, *NEXT: In-Network Nonconvex Optimization*, IEEE Transactions on Signal and Information Processing over Networks, (2016).
[21] G. MATEOS, J. BAZERQUE, AND G. GIANNAKIS, *Distributed Sparse Linear Regression*, IEEE Transactions on Signal Processing, 58 (2010), pp. 5262–5276.
[22] A. MOKHTARI, W. SHI, Q. LING, AND A. RIBEIRO, *DQM: Decentralized Quadratically Approximated Alternating Direction Method of Multipliers*, arXiv preprint arXiv:1508.02073, (2015).
[23] A. MOKHTARI, W. SHI, Q. LING, AND A. RIBEIRO, *A Decentralized Second-Order Method with Exact Linear Convergence Rate for Consensus Optimization*, arXiv preprint arXiv:1602.00596, (2016).

[24] A. NEDIĆ, *Asynchronous Broadcast-Based Convex Optimization over a Network*, IEEE Transactions on Automatic Control, 56 (2011), pp. 1337–1351.

[25] A. NEDIĆ AND D. BERTSEKAS, *Incremental Subgradient Methods for Nondifferentiable Optimization*, SIAM Journal on Optimization, 12 (2001), pp. 109–138.

[26] A. NEDIĆ AND D. P. BERTSEKAS, *Convergence Rate of Incremental Subgradient Algorithms*, Kluwer Academic Publishers, 2000, pp. 263–304.

[27] A. NEDIĆ, D. P. BERTSEKAS, AND V. BORKAR, *Distributed asynchronous incremental subgradient methods*, in Proceedings of the March 2000 Haifa Workshop "Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications", Elsevier, Amsterdam, 2001.

[28] A. NEDIĆ AND A. OLSHEVSKY, *Distributed Optimization over Time-Varying Directed Graphs*, in The 52nd IEEE Annual Conference on Decision and Control, 2013, pp. 6855–6860.

[29] A. NEDIĆ AND A. OLSHEVSKY, *Stochastic Gradient-Push for Strongly Convex Functions on Time-Varying Directed Graphs*, arXiv preprint arXiv:1406.2075, (2014).

[30] A. NEDIĆ AND A. OLSHEVSKY, *Distributed Optimization over Time-Varying Directed Graphs*, IEEE Transactions on Automatic Control, 60 (2015), pp. 601–615.

[31] A. NEDIĆ, A. OLSHEVSKY, A. OZDAGLAR, AND J. TSITSIKLIS, *On Distributed Averaging Algorithms and Quantization Effects*, IEEE Transactions on Automatic Control, 54 (2009), pp. 2506–2517.

[32] A. NEDIĆ, A. OLSHEVSKY, AND C. URIBE, *Fast Convergence Rates for Distributed Non-Bayesian Learning*, arXiv preprint arXiv:1508.05161, (2015).

[33] A. NEDIĆ AND A. OZDAGLAR, *Cooperative Distributed Multi-agent Optimization*, Cambridge University Press, 2008, ch. 1, Convex Optimization in Signal Processing and Communications, pp. 3–49.

[34] A. NEDIĆ AND A. OZDAGLAR, *Distributed Subgradient Methods for Multi-agent Optimization*, IEEE Transactions on Automatic Control, 54 (2009), pp. 48–61.

[35] A. NEDIĆ AND A. OZDAGLAR, *Convergence rate for consensus with delays*, Journal of Global Optimization, 47 (2010), pp. 437–456.

[36] A. OLSHEVSKY, *Efficient Information Aggregation Strategies for Distributed Control and Signal Processing*, PhD thesis, Massachusetts Institute of Technology, 2010.

[37] A. OLSHEVSKY, *Linear Time Average Consensus on Fixed Graphs and Implications for Decentralized Optimization and Multi-Agent Control*, arXiv preprint arXiv:1411.4186v6, (2016).

[38] A. OLSHEVSKY AND J. TSITSIKLIS, *On the Nonexistence of Quadratic Lyapunov Functions for Consensus Algorithms*, IEEE Transactions on Automatic Control, 53 (2008), pp. 2642–2645.

[39] M. RABBAT AND R. NOWAK, *Distributed Optimization in Sensor Networks*, in Proceedings of the 3rd international symposium on Information processing in sensor networks, ACM, 2004, pp. 20–27.

[40] S. RAM, A. NEDIĆ, AND V. VEERAVALLI, *Incremental Stochastic Subgradient Algorithms for Convex Optimization*, SIAM Journal on Optimization, 20 (2009), pp. 691–717.

[41] S. RAM, V. VEERAVALLI, AND A. NEDIĆ, *Distributed Non-Autonomous Power Control through Distributed Convex Optimization*, in INFOCOM, 2009, pp. 3001–3005.

[42] S. S. RAM, A. NEDIĆ, AND V. VEERAVALLI, *Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization*, Journal of Optimization Theory and Applications, 147 (2010), pp. 516–545.

[43] S. S. RAM, A. NEDIĆ, AND V. V. VEERAVALLI, *A new class of distributed optimization algorithms: Application to regression of distributed data*, Optimization Methods and Software, 27 (2012), pp. 71–88.

[44] W. REN, *Consensus Based Formation Control Strategies for Multi-Vehicle Systems*, in Proceedings of the American Control Conference, 2006, pp. 4237–4242.

[45] A. SAYED, *Diffusion Adaptation over Networks*, Academic Press Library in Signal Processing, 3 (2013), pp. 323–454.

[46] W. SHI, Q. LING, G. WU, AND W. YIN, *A Note: the Non-Ergodic $o\left(\frac{1}{k}\right)$ Rate of PG-EXTRA*. http://home.ustc.edu.cn/ ∼shiwei00/papers/PG_EXTRA_appendix_small_o_1_k_v0.1.pdf.

[47] W. SHI, Q. LING, G. WU, AND W. YIN, *A Proximal Gradient Algorithm for Decentralized Composite Optimization*, IEEE Transactions on Signal Processing, 63 (2015), pp. 6013–6023.

[48] W. SHI, Q. LING, G. WU, AND W. YIN, *EXTRA: An Exact First-Order Algorithm for Decentralized Consensus Optimization*, SIAM Journal on Optimization, 25 (2015), pp. 944–966.

[49] 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 (2014), pp. 1750–1761.

[50] H. TERELIUS, U. TOPCU, AND R. MURRAY, *Decentralized Multi-agent Optimization via Dual Decomposition*, in IFAC, 2011.

[51] B. TOURI, *Product of Random Stochastic Matrices and Distributed Averaging*, Springer Science & Business Media, 2012.

[52] J. TSITSIKLIS, D. BERTSEKAS, AND M. ATHANS, *Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms*, IEEE Transactions on Automatic Control, 31 (1986), pp. 803–812.

[53] M. WANG AND D. P. BERTSEKAS, *Incremental constraint projection-proximal methods for nonsmooth convex optimization*. Lab. for Information and Decision Systems Report LIDS-P-2907, MIT, July 2013; to appear in SIAM J. on Optimization, 2013.

[54] E. WEI AND A. OZDAGLAR, *On the $O(1/k)$ Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers*, arXiv preprint arXiv:1307.8254, (2013).

[55] C. XI AND U. KHAN, *On the Linear Convergence of Distributed Optimization over Directed Graphs*, arXiv preprint

arXiv:1510.02149, (2015).

[56] L. Xiao and S. Boyd, *Fast Linear Iterations for Distributed Averaging*, Systems and Control Letters, 53 (2004), pp. 65–78.

[57] L. Xiao, S. Boyd, and S. Kim, *Distributed Average Consensus with Least-mean-square Deviation*, Journal of Parallel and Distributed Computing, 67 (2007), pp. 33–46.

[58] J. Xu, *Augmented Distributed Optimization for Networked Systems*, PhD thesis, Nanyang Technological University, 2016.

[59] J. Xu, S. Zhu, Y. Soh, and L. Xie, *Augmented Distributed Gradient Methods for Multi-Agent Optimization Under Uncoordinated Constant Stepsizes*, in Proceedings of the 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 2055–2060.

[60] K. Yuan, Q. Ling, and W. Yin, *On the Convergence of Decentralized Gradient Descent*, arXiv preprint arXiv:1310.7063, (2013).

[61] J. Zeng and W. Yin, *ExtraPush for Convex Smooth Decentralized Optimization over Directed Networks*, arXiv preprint arXiv:1511.02942, (2015).

[62] M. Zhu and S. Martinez, *Discrete-Time Dynamic Average Consensus*, Automatica, 46 (2010), pp. 322–329.

[63] M. Zhu and S. Martinez, *On Distributed Convex Optimization under Inequality and Equality Constraints*, IEEE Transactions on Automatic Control, 57 (2012), pp. 151–164.