Accelerated AB/Push-Pull Methods for Distributed Optimization over Time-Varying Directed Networks

Duong Thuy Anh Nguyen, *Student Member, IEEE*, Duong Tung Nguyen, *Member, IEEE*, and Angelia Nedić, *Member, IEEE*

***Abstract*—This paper investigates a novel approach for solving the distributed optimization problem in which multiple agents collaborate to find the global decision that minimizes the sum of their individual cost functions. First, the AB/Push-Pull gradient-based algorithm is considered, which employs row- and column-stochastic weights simultaneously to track the optimal decision and the gradient of the global cost function, ensuring consensus on the optimal decision. Building on this algorithm, we then develop a general algorithm that incorporates acceleration techniques, such as heavy-ball momentum and Nesterov momentum, as well as their combination with non-identical momentum parameters. Previous literature has established the effectiveness of acceleration methods for various gradient-based distributed algorithms and demonstrated linear convergence for static directed communication networks. In contrast, we focus on time-varying directed communication networks and establish linear convergence of the methods to the optimal solution, when the agents' cost functions are smooth and strongly convex. Additionally, we provide explicit bounds for the step-size value and momentum parameters, based on the properties of the cost functions, the mixing matrices, and the graph connectivity structures. Our numerical results illustrate the benefits of the proposed acceleration techniques on the AB/Push-Pull algorithm.**

***Index Terms*—Distributed optimization, accelerated algorithm, time-varying graph, directed graph.**

I. Introduction

Distributed optimization has attracted significant interest in recent years due to its wide range of applications in large-scale multi-agent systems, such as sensor networks [1], formation control [2], and machine learning [3]. In these systems, data samples are distributed across multiple agents with computation tasks divided among them. Communication between agents only occurs through established communication links. This paper considers a system of n agents, where each agent's local cost f_i is determined by its data sample. The goal is for the agents to collaborate and reach a consensus on an optimal solution for the global cost f, by solving the following optimization problem:

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

The use of decentralized and collaborative approaches for solving the optimization of the sum of convex functions has garnered significant attention in the literature, with many algorithms proposed, including gradient-based methods [4], dual averaging methods [5], ADMM [6], and Newton methods [7], [8]. Early works often assume doubly-stochastic weights, which require underlying networks to be undirected or balanced [6], [9], [10]. To address directed graphs, [11] introduced subgradient-push algorithm, a decentralized subgradient method based on the push-sum technique, which is later extended to time-varying graphs in [12] with a convergence rate of $O(\ln t/\sqrt{t})$ for diminishing step-sizes. Algorithms such as ADD-OPT [13] and Push-DIGing [14] further improve the convergence rate by incorporating the push-sum protocol with a gradient estimation approach. These methods require knowledge of agents' out-degree to construct a column-stochastic weight matrix, while algorithms such as [15] and FROST [16] only use row-stochastic weights. These algorithms introduce a nonlinear term through division by the Perron eigenvector estimation of the weight matrix, which may result in stability issues. The AB/Push-Pull method, introduced by [4], [17], eliminates the need for Perron eigenvector estimation by using both row- and column-stochastic weights simultaneously, and demonstrates linear convergence for static directed communication networks. References [18] and [19] further establish linear convergence of this method for time-varying directed graphs, with the latter work also providing an improved analysis and explicit range for the step-size.

The heavy-ball method [20] and Nesterov's momentum [21] are popular acceleration techniques for gradient-based methods to achieve faster convergence. Several distributed algorithms have been proposed in the literature that incorporate these momentum methods. In [22], two variants of an accelerated distributed Nesterov gradient method are proposed for convex (and strongly convex) smooth objective function when the communication network is static undirected. For a static directed network, papers [23] and [24] propose methods that combine the AB/Push-Pull gradient tracking method with a heavy-ball momentum and Nesterov momentum, respectively. In particular, the linear convergence for the AB/Push-Pull method with a heavy-ball momentum term is proved in [23], while [24] only shows convergence through numerical examples for the AB/Push-Pull method with a Nesterov momentum term. Reference [25] further proposes a double-accelerated method based on AB/Push-Pull by incorporating both momentum terms, while [26] proposes to combine Nesterov momentum term with the FROST method. All the aforementioned acceleration methods are studied for a static directed graph. For time-varying communication networks,

The authors are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, United States. Email: {dtnguy52, duongnt, Angelia.Nedich}@asu.edu. This material is based in part upon work supported by the NSF award CCF-2106336. *Corresponding Author*: Duong Thuy Anh Nguyen.

[27] proposes to utilize heavy-ball and Nesterov techniques to accelerate the Push-DIGing algorithm [14] that uses column-stochastic matrices only. Furthermore, all prior studies on the double-accelerated method [25], [27] mandate that the parameters for heavy-ball and Nesterov acceleration are identical.

In this paper, we focus on a general network setting where agent communication is given by a sequence of time-varying directed graphs. Building on the AB/Push-Pull algorithm, we propose a general formulation that incorporates acceleration techniques such as heavy-ball momentum and Nesterov momentum, as well as their combination with non-identical momentum parameters. This is an innovative departure from existing works, which mandate identical momentum parameters. The AB/Push-Pull algorithm does not rely on Perron eigenvector estimation, making it well-suited for time-varying weights and serving as the foundation for the development of these acceleration methods. A key challenge in the analysis is the time-varying nature of the mixing matrices. Our analysis uses time-varying weighted averages and norms to establish consensus contractions for each update step for both row- and column-stochastic mixing matrices.

We prove that the accelerated algorithm converges linearly to the optimal solution when the agents' cost functions are smooth and strongly convex. The convergence result is derived based on appropriate values for constant step-size and momentum parameters, with explicit upper bounds provided in terms of the cost function properties, mixing matrices, and graph connectivity structures. Numerical results demonstrate that the acceleration of the AB/Push-Pull method leads to faster convergence. The results also show that allowing for different values of acceleration parameters provides increased flexibility and the potential for faster convergence. Our main contributions can be summarized as follows:

- We propose a novel algorithm that combines the AB/Push-Pull method with acceleration techniques such as heavy-ball momentum, Nesterov momentum, and a combination of both, using *non-identical* coefficients.
- We consider a general, *directed*, and *time-varying* communication network and rigorously prove the linear convergence of the proposed algorithm. The proof extends and improves the previous results [18], [19], [23]–[25] by providing a comprehensive analysis and explicit ranges for the step-size and momentum parameters in terms of cost function properties and communication structures.

The structure of this paper is as follows. We outline the distributed optimization problem in Section II. In Section III, we introduce the accelerated algorithm, and its convergence analysis is presented in Section IV. The performance of the proposed algorithm is demonstrated in Section V. Finally, we conclude with some key points in Section VI.

Notations. Unless otherwise stated, all vectors are considered to be column vectors. The standard Euclidean norm is denoted by $\|\cdot\|$. The notation $\mathbf{1}$ represents a vector with all entries equal to 1, and $\mathbb{I}$ denotes the identity matrix. The i-th entry of a time-varying vector u_k is denoted by $[u_k]_i$. For a vector v, we use notation $\min(v) = \min_i v_i$ and $\max(v) = \max_i v_i$. A vector is considered to be stochastic if its entries are nonnegative and sum to 1.

To denote the ij-th entry of a matrix A_k, we write $[A_k]_{ij}$. The minimum positive entry of a nonnegative matrix is denoted by $\min^+(A)$. A nonnegative matrix $A \in \mathbb{R}^{n\times n}$ is considered to be row-stochastic if $A\mathbf{1} = \mathbf{1}$, and a nonnegative matrix $B \in \mathbb{R}^{n\times n}$ is considered to be column-stochastic if $\mathbf{1}^\mathsf{T}B = \mathbf{1}^\mathsf{T}$. Given a positive vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$, the $\mathbf{a}$-weighted norm is as follows:

$$\|\mathbf{x}\|_{\mathbf{a}} = \sqrt{\sum_{i=1}^{n} a_i \|x_i\|^2} \quad \text{for } \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^p \times \cdots \times \mathbb{R}^p.$$

When $\mathbf{a} = \mathbf{1}$, the norm $\|\mathbf{x}\|_{\mathbf{a}}$ reduces to the Euclidean norm of $\mathbf{x}$, and we simply write $\|\mathbf{x}\|$. We have the following inequality,

$$\|\mathbf{x}\| \leq \sqrt{\tfrac{1}{\min(\mathbf{a})}}\, \|\mathbf{x}\|_{\mathbf{a}} \quad \text{for } \mathbf{x} \in \mathbb{R}^p \times \cdots \times \mathbb{R}^p,\ \mathbf{a} > \mathbf{0}. \tag{2}$$

We let $[n] = \{1, \ldots, n\}$ for an integer $n \geq 1$. A directed graph $\mathbb{G} = ([n], \mathcal{E})$ is specified by the edge set $\mathcal{E} \subseteq [n] \times [n]$ of ordered pairs of nodes. Given a directed graph $\mathbb{G} = ([n], \mathcal{E})$, the sets $\mathcal{N}_i^{\text{out}}$ and $\mathcal{N}_i^{\text{in}}$ denote the out-neighbors and the in-neighbors of a node i, i.e.,

$$\mathcal{N}_i^{\text{out}} = \{j \mid (i,j) \in \mathcal{E}\} \quad \text{and} \quad \mathcal{N}_i^{\text{in}} = \{j \mid (j,i) \in \mathcal{E}\}.$$

When the graph varies over time, we use a subscript k to indicate the time instance. For example, $\mathcal{E}_k$ denotes the edge-set of a graph $\mathbb{G}_k$, $\mathcal{N}_{ik}^{\text{in}}$ and $\mathcal{N}_{ik}^{\text{out}}$ denote the in-neighbors and the out-neighbors of a node i, respectively, at time k.

A directed graph is *strongly connected* if there is a directed path from any node to all other nodes in the graph. Given a directed path, the length of the path is the number of edges in the path. For a strongly connected directed graph $\mathbb{G} = ([n], \mathcal{E})$, we use the following definitions:

Definition 1 (Graph Diameter): The diameter $\mathrm{D}(\mathbb{G})$ is the length of the longest path in a collection of all shortest directed paths connecting all ordered pairs of distinct nodes in $\mathbb{G}$.

Let $\mathbf{p}_{jl}$ denote a *shortest directed path from node j to node l*, where $j \neq l$. A collection $\mathcal{P}$ of directed paths in $\mathbb{G}$ is a shortest-path graph covering if $\mathbf{p}_{jl} \in \mathcal{P}$ and $\mathbf{p}_{lj} \in \mathcal{P}$ for any two nodes $j,\ l \in [n],\ j \neq l$. The *utility of the edge* (j,l) with respect to the covering $\mathcal{P}$ is the number of shortest paths in $\mathcal{P}$ that pass through the edge (j,l). Define $\mathrm{K}(\mathcal{P})$ as the maximum edge-utility in $\mathcal{P}$ taken over all edges in the graph, i.e., $\mathrm{K}(\mathcal{P}) = \max\limits_{(j,l)\in\mathcal{E}} \sum\limits_{\mathbf{p}\in\mathcal{P}} \chi_{\{(j,l)\in\mathbf{p}\}}$, where $\chi_{\{(j,l)\in\mathbf{p}\}}$ is the indicator function taking value 1 when $(j,l) \in \mathbf{p}$ and, otherwise, taking value 0. Denote by $\mathcal{S}(\mathbb{G})$ the collection of all possible shortest-path coverings of the graph $\mathbb{G}$.

Definition 2 (Maximal Edge-Utility): The maximal edge-utility $\mathrm{K}(\mathbb{G})$ is the maximum value of $\mathrm{K}(\mathcal{P})$ taken over all possible shortest-path coverings $\mathcal{P} \in \mathcal{S}(\mathbb{G})$, i.e., $\mathrm{K}(\mathbb{G}) = \max_{\mathcal{P}\in\mathcal{S}(\mathbb{G})} \mathrm{K}(\mathcal{P})$.

II. Problem Formulation

We consider a system of n agents that are connected by a communication network, with the aim of collaboratively solving the optimization problem in (1), where each function $f_i : \mathbb{R}^p \to \mathbb{R}$ represents the cost function known only to

agent i. The agents aim to find a globally optimal solution by performing local computations and exchanging information with their neighboring agents through a sequence of directed communication networks, represented by a time-varying graph sequence $\{\mathbb{G}_k\}$.

At each time step k, agents communicate over a directed graph $\mathbb{G}_k$, and their updates are governed by two non-negative matrices A_k and B_k, which *align* with the connectivity structure of the graph $\mathbb{G}_k$, in the following sense:

$$[A_k]_{ij} > 0 \ \forall j \in \mathcal{N}_{ik}^{\text{in}} \cup \{i\}, \ [A_k]_{ij} = 0 \ \forall j \notin \mathcal{N}_{ik}^{\text{in}} \cup \{i\}, \quad (3)$$

$$[B_k]_{ji} > 0 \ \forall j \in \mathcal{N}_{ik}^{\text{out}} \cup \{i\}, \ [B_k]_{ji} = 0 \ \forall j \notin \mathcal{N}_{ik}^{\text{out}} \cup \{i\}. \quad (4)$$

Moreover, each matrix A_k is row-stochastic and each matrix B_k is column-stochastic for all $k \geq 0$. Additionally, we assume that there exist scalars $a > 0$ and $b > 0$ such that $\min^+(A_k) \geq a$ and $\min^+(B_k) \geq b$ for all $k \geq 0$.

We consider the problem under the following assumptions:

Assumption 1: For each k, the directed graph $\mathbb{G}_k = ([n], \mathcal{E}_k)$ is strongly connected.

Remark 1: We can relax Assumption 1 by considering a sequence of C-strongly connected graphs, i.e., for every $k \geq 0$, there exists an integer $C \geq 1$ such that the graph formed by the edge set $\mathcal{E}_k^C = \bigcup_{i=kC}^{(k+1)C-1} \mathcal{E}_i$ is strongly connected.

Assumption 2: Each f_i is continuously differentiable and has L-Lipschitz continuous gradients, i.e., for some $L > 0$,

$$\|\nabla f_i(x) - \nabla f_i(u)\| \leq L\|x - u\|, \quad \text{for all } x, u \in \mathbb{R}^p.$$

Assumption 3: The average-sum function $f = \frac{1}{n}\sum_{i=1}^n f_i$ is μ-strongly convex, i.e., for some $\mu > 0$,

$$\langle \nabla f(x) - \nabla f(u), x - u \rangle \geq \mu\|x - u\|^2 \quad \text{for all } x, u \in \mathbb{R}^p.$$

Remark 2: The strong convexity condition implies that problem (1) has a unique optimal solution denoted by x^*, i.e.,

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^p} f(x).$$

III. Accelerated AB/Push-Pull Methods

The AB/Push-Pull method, initially proposed in [4], [17], is a general framework that unifies many of the existing decentralized first-order methods with gradient tracking. Thus, it is worthwhile to consider adding momentum terms to accelerate its convergence. In this section, we introduce a comprehensive approach that encompasses three methods for accelerating the distributed AB/Push-Pull algorithm for time-varying directed graphs. These methods include the heavy-ball method [20] (also, in Section 3.2.1 of [28]), the Nesterov gradient method [21], and a combination of these two methods.

Let each agent $i \in \{1, 2, \ldots, n\}$ possess two local copies $x_k^i \in \mathbb{R}^p$ and $s_k^i \in \mathbb{R}^p$ of the decision variable and a gradient-tracking variable $y_k^i \in \mathbb{R}^p$ which is an estimate of a "global update direction", at iteration k. These variables are maintained and updated over time, as follows:

To begin, agents are directed to use the step-size $\alpha > 0$, the heavy-ball momentum parameter $\beta \geq 0$, and the Nesterov momentum parameter $\gamma \geq 0$. Additionally, each agent i initializes their updates with arbitrary vectors x_{-1}^i, x_0^i, s_0^i and with $y_0^i = \nabla f_i(s_0^i)$, without the need for coordination among agents. Each agent i also independently decide on the entries of A_k in the i-th row for their in-neighbors $j \in \mathcal{N}_{ik}^{\text{in}}$, while the value $[B_k]_{ij}$ is determined by agent $j \in \mathcal{N}_{ik}^{\text{in}}$. At every time k, every agent i sends its vector s_k^i and a scaled direction $[B_k]_{\ell i} y_k^i$ to its out-neighbors $\ell \in \mathcal{N}_{ik}^{\text{out}}$. Every agent i also receives these vectors sent by its in-neighbors $j \in \mathcal{N}_{ik}^{\text{in}}$. Upon the information exchange step, every agent i updates its vectors using equations (5)–(7) for all $k \geq 0$. The proposed procedure is outlined in Algorithm 1.

Algorithm 1: Accelerated AB/Push-Pull

Agents are directed to use $\alpha > 0$, $\beta \geq 0$ and $\gamma \geq 0$.
Every agent $i \in [n]$ initializes with arbitrary initial vectors $x_{-1}^i, x_0^i, s_0^i \in \mathbb{R}^p$ and $y_0^i = \nabla f_i(s_0^i)$.
for $k = 0, 1, \ldots,$ every agent $i \in [n]$ does the following:
In-bounds mixing weights $[A_k]_{ij}$, for all $j \in \mathcal{N}_{ik}^{\text{in}}$;
Out-bounds pushing weights $[B_k]_{\ell i}$, for all $\ell \in \mathcal{N}_{ik}^{\text{out}}$;
Receives s_k^j and $[B_k]_{ij} y_k^j$ from in-neighbors $j \in \mathcal{N}_{ik}^{\text{in}}$;
Sends s_k^i and $[B_k]_{\ell i} y_k^i$ to out-neighbors $\ell \in \mathcal{N}_{ik}^{\text{out}}$;
Updates x_{k+1}^i, s_{k+1}^i and y_{k+1}^i by

$$x_{k+1}^i = \sum_{j=1}^n [A_k]_{ij} s_k^j - \alpha y_k^i + \beta(x_k^i - x_{k-1}^i), \quad (5)$$

$$s_{k+1}^i = x_{k+1}^i + \gamma(x_{k+1}^i - x_k^i), \quad (6)$$

$$y_{k+1}^i = \sum_{j=1}^n [B_k]_{ij} y_k^j + \nabla f_i(s_{k+1}^i) - \nabla f_i(s_k^i), \quad (7)$$

end for

The method in Algorithm 1 is a generalization of three methods to accelerate the AB/Push-Pull algorithm, namely,

- $\beta > 0, \gamma = 0$: Heavy-ball method
- $\beta = 0, \gamma > 0$: Nesterov gradient method
- $\beta > 0, \gamma > 0$: Combination of Nesterov gradient and heavy-ball methods.

Relations to the AB/Push-Pull algorithm: From the viewpoint of an agent, the information about the gradients is pushed to the neighbors, while the information about the decision variable is pulled from the neighbors, as noted in [17]. Hence, the update step for aggregating the decision variables using the row-stochastic matrix A_k is referred to as a *pull-step*, while the step for tracking the average gradients using the column-stochastic matrix B_k is referred to as a *push-step* as it mimics the push-sum consensus method, originally proposed in [29], later used in decentralized gradient-based methods [3], [11], [12], and recently studied in [30]. Moreover, the role of gradient tracking is to account for the heterogeneity of the local data distributions among agents.

Acceleration techniques: Intuitively, the heavy-ball momentum term, represented by $\beta(x_k^i - x_{k-1}^i)$, accelerates the gradient method by adding inertia to the updates. The Nesterov momentum term, represented by $\gamma(x_{k+1}^i - x_k^i)$, is mathematically shown to improve convergence rate, but its underlying intuition is not immediately clear. A geometric interpretation of the Nesterov accelerated algorithm can be found in [31]. We discuss the impact of incorporating the momentum terms on the convergence rate of the standard gradient method, i.e.,

$$x_{k+1} = x_k - \alpha \nabla f(x_k),$$

where $\alpha > 0$ is a stepsize. We recall the following updates of the heavy-ball method (see [20], or Section 3.2.1 of [28]):

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}),$$

where $\beta > 0$, and the Nesterov gradient method [21]:

$$\begin{aligned} x_{k+1} &= y_k - \alpha \nabla f(y_k), \\ y_{k+1} &= x_{k+1} + \gamma(x_{k+1} - x_k), \end{aligned}$$

where $\gamma > 0$. The convergence rate of the gradient method is well-known to be $\mathcal{O}\big(\big(\frac{Q-1}{Q+1}\big)^k\big)$, where $Q = \frac{L}{\mu}$ is the condition number of the objective function $f(x)$ (see [28]). By properly choosing the parameters α and β, the heavy-ball method can achieve a faster, locally accelerated rate of $\mathcal{O}\big(\big(\frac{\sqrt{Q}-1}{\sqrt{Q}+1}\big)^k\big)$. The Nesterov gradient method also has the rate of $\mathcal{O}\big(\big(\frac{\sqrt{Q}-1}{\sqrt{Q}}\big)^k\big)$ when $0 < \alpha \leq 1/L$ and $\gamma = (\sqrt{Q}-1)/(\sqrt{Q}+1)$, which is faster than the gradient method but slower than the heavy-ball method in terms of the dependence on the condition number Q (since $Q \geq 1$). These acceleration methods, as well as their combination, have been studied in the context of distributed optimization algorithms. It has been shown numerically and/or theoretically in [23]–[25] that the accelerated AB/Push-Pull algorithms converge linearly for static directed graphs with appropriate choices of step-size and momentum parameters. However, despite this progress, a comprehensive theoretical analysis for the convergence on time-varying directed graphs is still an open problem, which will be addressed in the next section.

IV. Convergence Analysis

In this section, we present a detailed convergence analysis of the generic accelerated AB/Push-Pull algorithm over a time-varying directed communication network. We begin by introducing some preliminary results. We then proceed to establish the estimates for the four quantities of interest, namely the optimality gap, the consensus error, the state difference, and the gradient estimation error. By combining these estimates, we demonstrate the linear convergence of the algorithm. Finally, we provide a step-size selection rule and the bounds for the acceleration parameters that ensure the convergence of the proposed algorithm.

A. Preliminary Results

We first present the contraction property of the gradient mapping, assuming that the objective function is strongly convex and has Lipschitz continuous gradients, in the following:

Lemma 1 ([28]): Let f be a μ-strongly convex and L-smooth function. For $0 < \alpha < 2L^{-1}$, we have

$$\|x - x^* - \alpha \nabla f(x)\| \leq q(\alpha)\|x - x^*\| \qquad \text{for all } x,$$

where $q(\alpha) = \max\{|1 - \alpha\mu|, |1 - \alpha L|\} < 1$.

We then proceed by presenting some foundational results that will support our later analysis.

Lemma 2 ([32], Corollary 5.2): Consider a vector collection $\{u_i,\, i \in [n]\} \subset \mathbb{R}^p$, and a scalar collection $\{\gamma_i,\, i \in [n]\} \subset \mathbb{R}$ of scalars such that $\sum_{i=1}^n \gamma_i = 1$. For all $u \in \mathbb{R}^p$, we have the following relation:

$$\left\|\sum_{i=1}^n \gamma_i u_i - u\right\|^2 = \sum_{i=1}^n \gamma_i \|u_i - u\|^2 - \sum_{i=1}^n \gamma_i \left\|u_i - \sum_{\ell=1}^n \gamma_\ell u_\ell\right\|^2.$$

Lemma 3 ([32], Lemma 5.4): Consider a sequence $\{A_k\}$ of row-stochastic matrices. Then, there exists a sequence $\{\phi_k\}$ of stochastic vectors such that

$$\phi_{k+1}^{\mathsf{T}} A_k = \phi_k^{\mathsf{T}} \qquad \text{for all } k \geq 0. \tag{8}$$

Moreover, if Assumption 1 holds and A_k is aligned with the graph $\mathbb{G}_k$ (see (3)) with $\min^+(A_k) \geq a > 0$ for all $k \geq 0$, then the entries of each ϕ_k have a uniform lower bound, i.e., $[\phi_k]_i \geq \frac{a^n}{n}$ for all $i \in [n]$ and for all $k \geq 0$.

Lemma 4 ([19], Lemma 3.4): Consider a sequence $\{B_k\}$ of column-stochastic matrices and the vector sequence $\{\pi_k\}$, defined as follows:

$$\pi_{k+1} = B_k \pi_k, \qquad \text{initialized with } \ \pi_0 = \tfrac{1}{n}\mathbf{1}. \tag{9}$$

Then, the vectors π_k are stochastic vectors. Moreover, if Assumption 1 holds, where B_k is aligned with the graph $\mathbb{G}_k$ and $\min^+(B_k) \geq b > 0$ for all $k \geq 0$, then $[\pi_k]_i \geq \frac{b^n}{n}$ for all $i \in [n]$ and $k \geq 0$.

Remark 3: When the graph sequence is C-strongly-connected (Remark 1), the product of weight matrices $A_{k+C-1} \dots A_{k+1} A_k$ and $B_{k+C-1} \dots B_{k+1} B_k$ are row- and column-stochastic, respectively. These matrices represent the directed paths among the nodes in the composition of the graphs $\mathbb{G}_k, \dots, \mathbb{G}_{k+C-1}$, capturing the underlying connectivity patterns and facilitating the understanding of information flow dynamics. Moreover, the more general results of Lemma 3 and Lemma 4 indicate the existence of stochastic vector sequences ϕ_k and π_k, such that for all $k \geq 0$,

$$\begin{aligned} \phi_{k+C}' \left(A_{k+C-1} \dots A_{k+1} A_k\right) &= \phi_k' \\ \text{and } \pi_{k+C} = \left(B_{k+C-1} \dots B_{k+1} B_k\right) \pi_k. & \end{aligned}$$

Furthermore,

$$[\phi_k]_i \geq \frac{a^{nC}}{n} \text{ and } [\pi_k]_i \geq \frac{b^{nC}}{n} \quad \text{for all } i \in [n].$$

Consider a strongly connected directed graph $\mathbb{G} = ([n], \mathcal{E})$, and weight matrices A and B that are aligned with the graph $\mathbb{G}$ (in sense of equations (3) and (4)). Let $\mathsf{D}(\mathbb{G})$ and $\mathsf{K}(\mathbb{G})$ be the diameter and the maximal edge-utility of the graph $\mathbb{G}$, respectively. We have the following two results:

Lemma 5 ([32], Lemma 6.1): Let A be a row-stochastic matrix, ϕ be a stochastic vector and let π be a nonnegative vector such that $\pi^{\mathsf{T}} A = \phi^{\mathsf{T}}$. Consider a collection of vectors $x_1, \dots, x_n \in \mathbb{R}^p$. For $\hat{x}_\phi = \sum_{i=1}^n \phi_i x_i$, we have

$$\sqrt{\sum_{i=1}^n \pi_i \left\|\sum_{j=1}^n A_{ij} x_j - \hat{x}_\phi\right\|^2} \leq c \sqrt{\sum_{j=1}^n \phi_j \|x_j - \hat{x}_\phi\|^2},$$

where $c = \sqrt{1 - \frac{\min(\pi)(\min^+(A))^2}{\max^2(\phi)\mathsf{D}(\mathbb{G})\mathsf{K}(\mathbb{G})}} \in (0,1)$ is a scalar.

Lemma 6 ([19], Lemma 4.5): Let B be a column-stochastic matrix, ν be a stochastic vector with positive

entries, i.e., $\nu_i > 0,\ \forall i \in [n]$, and let the vector π be given by $\pi = B\nu$. Consider the vectors $y_1, \ldots, y_n \in \mathbb{R}^p$ and vectors $w_i = \sum_{j=1}^n B_{ij} y_j$ for all $i \in [n]$, we have

$$\sqrt{\sum_{i=1}^n \pi_i \left\| \frac{w_i}{\pi_i} - \sum_{\ell=1}^m y_\ell \right\|^2} \leq \tau \sqrt{\sum_{i=1}^n \nu_i \left\| \frac{y_i}{\nu_i} - \sum_{\ell=1}^n y_\ell \right\|^2},$$

where $\tau = \sqrt{1 - \frac{\min^2(\nu)\,(\min^+(B))^2}{\max^2(\nu)\max(\pi)\,\mathsf{D}(\mathbb{G})\mathsf{K}(\mathbb{G})}} \in (0,1)$ is a scalar.

The column stochastic property of the matrices B_k ensures that the sum of the y-iterates, at any time k, is equal to the sum of the gradients $\nabla f_i(s_k^i)$, as seen in the following lemma (the proof follows from mathematical induction on k, and uses the column-stochasticity of B_k and the initialization of the y-variables; see Algorithm 1).

Lemma 7: Consider Algorithm 1, and assume that each B_k is column-stochastic. Then, we have

$$\sum_{i=1}^n y_k^i = \sum_{i=1}^n \nabla f_i(s_k^i) \quad \text{for all } k \geq 0.$$

B. Main Results

The convergence analysis of the accelerated AB/Push-Pull method is based on establishing a contraction relationship between the following four quantities: (i) the optimality gap, (ii) the consensus error, (iii) the state difference, and (iv) the gradient estimation error, given respectively as follows:

$$\|\hat{x}_k - x^*\|, \quad D(\mathbf{x}_k, \phi_k) = \sqrt{\sum_{i=1}^n [\phi_k]_i \|x_k^i - \hat{x}_k\|^2}, \tag{10a}$$

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\|,\ S(\mathbf{y}_k, \pi_k) = \sqrt{\sum_{j=1}^n [\pi_k]_j \left\| \frac{y_k^j}{[\pi_k]_j} - \sum_{\ell=1}^n y_k^\ell \right\|^2}, \tag{10b}$$

where $\hat{x}_k = \sum_{i=1}^n [\phi_k]_i x_k^i$, $\mathbf{x}_k = (x_k^1, \ldots, x_k^n)$ and x^* is the optimal solution of problem (1). We define the constants $\varphi_k > 0$, $r_k > 0$, $c_k \in (0,1)$ and $\tau_k \in (0,1)$ as follows where $\hat{x}_k = \sum_{i=1}^n [\phi_k]_i x_k^i$, $\mathbf{x}_k = (x_k^1, \ldots, x_k^n)$ and x^* is the optimal solution of problem (1). We define the constants $\varphi_k > 0$, $r_k > 0$, $c_k \in (0,1)$ and $\tau_k \in (0,1)$ as follows

$$r_k = \sqrt{n} + \frac{1}{\sqrt{\min(\pi_{k+1})}},\ c_k = \sqrt{1 - \frac{\min(\phi_{k+1})\,a^2}{\max^2(\phi_k)\,\mathsf{D}(\mathbb{G}_k)\mathsf{K}(\mathbb{G}_k)}},$$
$$\varphi_k = \sqrt{\frac{1}{\min(\phi_k)}},\ \tau_k = \sqrt{1 - \frac{\min^2(\pi_k)\,b^2}{\max^2(\pi_k)\max(\pi_{k+1})\mathsf{D}(\mathbb{G}_k)\mathsf{K}(\mathbb{G}_k)}}. \tag{11}$$

We first establish the recursive relation for the weighted average $\{\hat{x}_k\}$ that will be utilized in the subsequent analysis.

Lemma 8: The weighted average sequence $\{\hat{x}_k\}$ satisfies,

$$\hat{x}_{k+1} = \hat{x}_k + \sum_{i=1}^n (\beta[\phi_{k+1}]_i + \gamma[\phi_k]_i)(x_k^i - x_{k-1}^i) - \alpha \sum_{i=1}^n [\phi_{k+1}]_i y_k^i, \qquad \text{for all } k \geq 0.$$

Proof: Plugging in the update for s_k^j from (6) into the update for x_k^j in (5) and rearranging the terms results in:

$$x_{k+1}^i = \sum_{j=1}^n [A_k]_{ij} x_k^j - \alpha y_k^i + \gamma \sum_{j=1}^n [A_k]_{ij}(x_k^j - x_{k-1}^j) + \beta(x_k^i - x_{k-1}^i). \tag{12}$$

By taking a weighted average of x_{k+1}^i using the ϕ_{k+1} weights, from relation (12) we obtain

$$\sum_{i=1}^n [\phi_{k+1}]_i x_{k+1}^i = \sum_{i=1}^n [\phi_{k+1}]_i \sum_{j=1}^n [A_k]_{ij} x_k^j - \alpha \sum_{i=1}^n [\phi_{k+1}]_i y_k^i$$
$$+\beta \sum_{i=1}^n [\phi_{k+1}]_i (x_k^i - x_{k-1}^i) + \gamma \sum_{i=1}^n [\phi_{k+1}]_i \sum_{j=1}^n [A_k]_{ij}(x_k^j - x_{k-1}^j).$$

Since $\phi_{k+1}^\mathsf{T} A_k = \phi_k^\mathsf{T}$, for the double-sum term we have

$$\sum_{i=1}^n [\phi_{k+1}]_i \sum_{j=1}^n [A_k]_{ij} x_k^j = \sum_{j=1}^n \Big(\sum_{i=1}^n [\phi_{k+1}]_i [A_k]_{ij} \Big) x_k^j = \sum_{j=1}^n [\phi_k]_j x_k^j.$$

By using the definition of $\hat{x}_k = \sum_{j=1}^n [\phi_k]_j x_k^j$, and combining the preceding two equations, we arrive at the desired relation. ■

We next examine the behavior of the directions y_k^i generated by the update in (7). The analysis will make use of some weighted norms of scaled directions y_k^i, where the scalings are time-varying and defined by a stochastic vector sequence $\{\pi_k\}$ associated with the matrix sequence $\{B_k\}$.

Lemma 9: Under Assumption 1, we have,

$$\sqrt{\sum_{i=1}^n \frac{\|y_k^i\|^2}{[\pi_k]_i}} \leq S(\mathbf{y}_k, \pi_k) + \left\| \sum_{\ell=1}^n y_k^\ell \right\| \qquad \text{for all } k \geq 0.$$

Proof: By Lemma 4, we have $\pi_k > 0$ for all $k \geq 0$, thus

$$\sum_{i=1}^n \frac{\|y_k^i\|^2}{[\pi_k]_i} = \sum_{i=1}^n [\pi_k]_i \left\| \frac{y_k^i}{[\pi_k]_i} \right\|^2.$$

Using the relation in Lemma 2 with $\gamma_i = [\pi_k]_i$ and $u_i = y_k^i / [\pi_k]_i$ for all i, and $u = 0$, we obtain

$$\sum_{i=1}^n [\pi_k]_i \left\| \frac{y_k^i}{[\pi_k]_i} \right\|^2 = \sum_{i=1}^n [\pi_k]_i \left\| \frac{y_k^i}{[\pi_k]_i} - \sum_{\ell=1}^n y_k^\ell \right\|^2 + \left\| \sum_{\ell=1}^n y_k^\ell \right\|^2$$
$$= S^2(\mathbf{y}_k, \pi_k) + \left\| \sum_{\ell=1}^n y_k^\ell \right\|^2,$$

where the last equality is implied from the definition of the S-quantity in (10b). Therefore,

$$\sqrt{\sum_{i=1}^n \frac{\|y_k^i\|^2}{[\pi_k]_i}} = \sqrt{S^2(\mathbf{y}_k, \pi_k) + \left\| \sum_{\ell=1}^n y_k^\ell \right\|^2} \leq S(\mathbf{y}_k, \pi_k) + \left\| \sum_{\ell=1}^n y_k^\ell \right\|,$$

where the inequality is obtained by using $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$, which is valid for any two scalars $a, b \geq 0$. ∎

Lemma 10: Under Assumption 1, Assumption 2, and Assumption 3, we have the following relation for the sum of the y-iterates, which holds for all $k \geq 0$:

$$\left\|\sum_{i=1}^n y_k^i\right\| \leq L\sqrt{n}\,(\varphi_k\|\hat{x}_k - x^*\| + \varphi_k D(\mathbf{x}_k,\phi_k) + \gamma\|\mathbf{x}_k - \mathbf{x}_{k-1}\|),$$

where $\varphi_k > 0$ is as given in (11).

Proof: By Lemma 7, we have

$$\left\|\sum_{i=1}^n y_k^i\right\| = \left\|\sum_{i=1}^n \nabla f_i(s_k^i)\right\| = \left\|\sum_{i=1}^n \Big(\nabla f_i(s_k^i) - \nabla f_i(x^*)\Big)\right\|,$$

where we use $\sum_{i=1}^n \nabla f_i(x^*) = 0$, valid for the optimal solution x^* to problem (1) (which exists and is unique due to Assumption 3). By Assumption 2 that each f_i has Lipschitz continuous gradients with a constant $L > 0$, we obtain

$$\left\|\sum_{i=1}^n y_k^i\right\| \leq \sum_{i=1}^n \left\|\nabla f_i(s_k^i) - \nabla f_i(x^*)\right\| \leq L\sum_{i=1}^n \|s_k^i - x^*\|.$$

We define $\mathbf{s}_k = (s_k^1, \ldots, s_k^n)$ and $\mathbf{x}^* = (x^*, \ldots, x^*)$. By Hölder's inequality we have $\sum_{i=1}^n a_i \leq \sqrt{n\sum_{i=1}^n a_i^2}$, for all a_i, $i \in [n]$, implying that

$$\begin{aligned}\left\|\sum_{i=1}^n y_k^i\right\| &\leq L\sqrt{n}\|\mathbf{s}_k - \mathbf{x}^*\| \leq L\sqrt{n}(\|\mathbf{x}_k - \mathbf{x}^*\| + \gamma\|\mathbf{x}_k - \mathbf{x}_{k-1}\|)\\ &\leq L\sqrt{n}(\varphi_k|\mathbf{x}_k - \mathbf{x}^*\|_{\phi_k} + \gamma\|\mathbf{x}_k - \mathbf{x}_{k-1}\|), \qquad (13)\end{aligned}$$

where the second inequality follows from the updates of the s-iterates in (6), while the last inequality follows from the relation for the norms in (2) and the fact that $[\phi_k]_i > 0$ for all i and k (by Assumption 1 and Lemma 3). For the quantity $\|\mathbf{x}_k - \mathbf{x}^*\|_{\phi_k}$, applying the relation in Lemma 2 with $u_i = x_k^i$, $\gamma_i = [\phi_k]_i$ for all i, and $u = x^*$ yields

$$\sum_{i=1}^n [\phi_k]_i\|x_k^i - x^*\|^2 = \|\hat{x}_k - x^*\|^2 + \sum_{i=1}^n [\phi_k]_i\|x_k^i - \hat{x}_k\|^2,$$

where $\hat{x}_k = \sum_{\ell=1}^n [\phi_k]_\ell x_k^\ell$. Using the definition of $D(\mathbf{x}_k, \phi_k)$ in (10), we further derive the following inequality:

$$\|\mathbf{x}_k - \mathbf{x}^*\|_{\phi_k} \leq \|\hat{x}_k - x^*\| + D(\mathbf{x}_k, \phi_k), \qquad (14)$$

which uses $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. The desired relation follows by combining the relations in (13) and (14) with the inequality $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\phi_k} \leq \|\mathbf{x}_k - \mathbf{x}_{k-1}\|$. ∎

In the following, we derive upper bounds for each of the four quantities defined in (10). We begin by assessing the optimality gap in the subsequent proposition.

Proposition 1: Let Assumption 1, Assumption 2, and Assumption 3 hold. Let the step-size α in Algorithm 1 be such that $0 < \alpha < \frac{2}{n\min(\pi_k)L}$, where L is the gradient Lipschitz constant. Then, we have for all $k \geq 0$:

$$\begin{aligned}\|\hat{x}_{k+1} - x^*\| &\leq q_k(\alpha)\|\hat{x}_k - x^*\| + \alpha L\sqrt{n}\varphi_k D(\mathbf{x}_k, \phi_k)\\ &\quad + \alpha S(\mathbf{y}_k, \pi_k) + \big(\beta + (1+\alpha L\sqrt{n})\gamma\big)\|\mathbf{x}_k - \mathbf{x}_{k-1}\|,\end{aligned}$$

where $q_k(\alpha) = \max\{|1 - \alpha n\min(\pi_k)\mu|, |1 - \alpha n\min(\pi_k)L|\}$.

Proof: See Appendix A. ∎

In the next proposition, we investigate the behavior of the deviation of the iterates x_k^i, $i \in [n]$, from their weighted average $\hat{x}_k$, as measured by the ϕ_k-weighted dispersion $D(\mathbf{x}_k, \phi_k)$.

Proposition 2: Under Assumption 1, Assumption 2, and Assumption 3, the following inequality holds for all $k \geq 0$,

$$\begin{aligned}D(\mathbf{x}_{k+1}, \phi_{k+1}) &\leq (c_k + \alpha L\sqrt{n}\varphi_k)D(\mathbf{x}_k, \phi_k) + \alpha S(\mathbf{y}_k, \pi_k)\\ &\quad + \alpha L\sqrt{n}\varphi_k\|\hat{x}_k - x^*\| + (\beta + \gamma(c_k + \alpha L\sqrt{n}))\|\mathbf{x}_k - \mathbf{x}_{k-1}\|.\end{aligned}$$

Proof: See Appendix B. ∎

The next result establishes an upper bound for the state difference of the x-sequence produced by the update in (5).

Proposition 3: Let Assumption 1, Assumption 2, and Assumption 3 hold. Then, for all $k \geq 0$, we have:

$$\begin{aligned}\|\mathbf{x}_{k+1} - \mathbf{x}_k\| &\leq (\beta + \gamma\sqrt{n}(1+\alpha L))\|\mathbf{x}_k - \mathbf{x}_{k-1}\| + \alpha S(\mathbf{y}_k, \pi_k)\\ &\quad + \alpha L\sqrt{n}\varphi_k\|\hat{x}_k - x^*\| + (c_k\varphi_{k+1} + \varphi_k + \alpha L\sqrt{n}\varphi_k)D(\mathbf{x}_k, \phi_k).\end{aligned}$$

Proof: Let us denote $z_k^i = \sum_{j=1}^n [A_k]_{ij}x_k^j$ and $v_k^i = \sum_{j=1}^n [A_k]_{ij}(x_k^j - x_{k-1}^j)$. Define the vectors $\mathbf{z}_k = (z_k^1, \ldots, z_k^n)$, $\mathbf{v}_k = (v_k^1, \ldots, v_k^n)$, $\mathbf{y}_k = (y_k^1, \ldots, y_k^n)$ and $\hat{\mathbf{x}}_k = (\hat{x}_k, \ldots, \hat{x}_k)$. Then, we can write the x-update in (12) compactly as follows:

$$\mathbf{x}_{k+1} = \mathbf{z}_k + \gamma\mathbf{v}_k - \alpha\mathbf{y}_k + \beta(\mathbf{x}_k - \mathbf{x}_{k-1}). \qquad (15)$$

By adding and subtracting $\hat{\mathbf{x}}_k$ and using the triangle inequality, we obtain:

$$\begin{aligned}\|\mathbf{x}_{k+1} - \mathbf{x}_k\| &\leq \|\mathbf{x}_{k+1} - \hat{\mathbf{x}}_k\| + \|\hat{\mathbf{x}}_k - \mathbf{x}_k\|\\ &\leq \|\mathbf{z}_k - \hat{\mathbf{x}}_k\| + \alpha\|\mathbf{y}_k\| + \beta\|\mathbf{x}_k - \mathbf{x}_{k-1}\| + \gamma\|\mathbf{v}_k\| + \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|, \qquad (16)\end{aligned}$$

where the last inequality follows from the compact representation of the x-update in (15). For the first term in (16), we use the relation for norms in (2) and Lemma 5 with $A = A_k$, $x_i = x_k^i$, and $\phi_{k+1}^{\mathsf{T}}A_k = \phi_k^{\mathsf{T}}$ to obtain the following bound:

$$\|\mathbf{z}_k - \hat{\mathbf{x}}_k\| \leq \varphi_{k+1}\|\mathbf{z}_k - \hat{\mathbf{x}}_k\|_{\phi_{k+1}} \leq c_k\varphi_{k+1}\|\mathbf{x}_k - \hat{\mathbf{x}}_k\|_{\phi_k}.$$

For the second term in (16), using the fact that the vector π_k is stochastic, Lemma 9 and Lemma 10, we derive

$$\begin{aligned}\|\mathbf{y}_k\| &= \sqrt{\sum_{i=1}^n [\pi_k]_i\frac{\|y_k^i\|^2}{[\pi_k]_i}} \leq \sqrt{\sum_{i=1}^n \frac{\|y_k^i\|^2}{[\pi_k]_i}} \leq S(\mathbf{y}_k, \pi_k)\\ &\quad + L\sqrt{n}\,(\varphi_k\|\hat{x}_k - x^*\| + \varphi_k D(\mathbf{x}_k, \phi_k) + \gamma\|\mathbf{x}_k - \mathbf{x}_{k-1}\|)\,.\end{aligned}$$

For the fourth term in (16), we have

$$\begin{aligned}\|\mathbf{v}_k\|^2 &= \sum_{i=1}^n \left\|\sum_{j=1}^n [A_k]_{ij}(x_k^j - x_{k-1}^j)\right\|^2\\ &\leq \sum_{i=1}^n \sum_{j=1}^n [A_k]_{ij}\|x_k^j - x_{k-1}^j\|^2 \leq n\|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2,\end{aligned}$$

where we use the fact that A_k is row-stochastic.

The last term in (16) follows from the relation for norms in (2) and the definition of $D(\mathbf{x}_k, \phi_k)$, as follows

$$\|\mathbf{x}_k - \hat{\mathbf{x}}_k\| \leq \varphi_k\|\mathbf{x}_k - \hat{\mathbf{x}}_k\|_{\phi_k} = \varphi_k D(\mathbf{x}_k, \phi_k).$$

Combining the estimates for each of the terms in (16), we obtain the desired relation. ∎

Next, we provide a recursive relation for the gradient estimation error $S(\mathbf{y}_k, \pi_k)$, as given in the following proposition.

Proposition 4: Under Assumption 1 and Assumption 2, the following inequality holds for all $k \geq 0$,

$$S(\mathbf{y}_{k+1}, \pi_{k+1}) \leq \tau_k\, S(\mathbf{y}_k, \pi_k) + Lr_k(1+\gamma)\|\mathbf{x}_{k+1} - \mathbf{x}_k\| + Lr_k\gamma\|\mathbf{x}_k - \mathbf{x}_{k-1}\|.$$

Proof: See Appendix C. ∎

We now present a composite relation for the four quantities defined in (10), by defining the vector V_k as follows:

$$V_k = \Big(\|\hat{x}_k - x^*\|, D(\mathbf{x}_k,\phi_k), S(\mathbf{y}_k,\pi_k), \|\mathbf{x}_k - \mathbf{x}_{k-1}\|\Big)^{\mathsf{T}}. \quad (17)$$

For the vector V_k we have the following result.

Proposition 5: Let Assumption 1, Assumption 2, and Assumption 3 hold. Then, for the iterates produced by the accelerated AB/Push-Pull method in Algorithm 1 with the step-size $\alpha \in (0, 2(nL)^{-1})$, we have

$$V_{k+1} \leq M_k(\alpha,\beta,\gamma)V_k \qquad \text{for all } k \geq 0,$$

where the ij-th element $m_k^{i,j}$ of the matrix $M_k(\alpha,\beta,\gamma)$ are given as follows:

$$\begin{aligned}
&m_k^{1,1} = q_k(\alpha),\ m_k^{1,2} = \alpha L\sqrt{n}\varphi_k,\ m_k^{1,4} = \beta+\gamma(1+\alpha L\sqrt{n}),\\
&m_k^{1,3} = \alpha,\ m_k^{2,1} = \alpha L\sqrt{n}\varphi_k,\ m_k^{2,2} = c_k + \alpha L\sqrt{n}\varphi_k,\\
&m_k^{2,3} = \alpha,\ \ m_k^{2,4} = \beta + \gamma(c_k + \alpha L\sqrt{n}),\\
&m_k^{3,1} = Lr_k(1+\gamma)m_k^{4,1},\ \ m_k^{3,2} = Lr_k(1+\gamma)m_k^{4,2},\\
&m_k^{3,3} = \tau_k + Lr_k(1+\gamma)m_k^{4,3},\ m_k^{3,4} = Lr_k\gamma + Lr_k(1+\gamma)m_k^{4,4},\\
&m_k^{4,1} = \alpha L\sqrt{n}\varphi_k,\ \ m_k^{4,2} = c_k\varphi_{k+1} + \varphi_k + \alpha L\sqrt{n}\varphi_k,\\
&m_k^{4,3} = \alpha,\ \ m_k^{4,4} = \beta + \gamma\sqrt{n}(1+\alpha L).
\end{aligned}$$

Proof: The stated relation follows directly from Proposition 1, Proposition 2, Proposition 3, and Proposition 4. ∎

Remark 4: From Proposition 5, to prove that V_k tends to 0 at a geometric rate, it is sufficient to show that

$$M_k(\alpha,\beta,\gamma) \leq M(\alpha,\beta,\gamma),$$

for some matrix $M(\alpha,\beta,\gamma)$, where the preceding inequality is to be understood entry-wise. Then, we select an appropriate step-size α in the range $(0, 2(nL)^{-1})$ and the acceleration parameters β, γ, such that the eigenvalues of $M(\alpha,\beta,\gamma)$ lie inside the unit circle, i.e., the spectral radius of $M(\alpha,\beta,\gamma)$ is less than 1.

We now determine an upper bound matrix $M(\alpha,\beta,\gamma)$ for $M_k(\alpha,\beta,\gamma)$. To do this, we define upper bounds for the constants c_k, τ_k, r_k, and φ_k defined in (11) as $c \in (0,1)$, $\tau \in (0,1)$, r, and φ, respectively, i.e.,

$$\max_{k\geq 0} c_k \leq c,\ \ \max_{k\geq 0}\tau_k \leq \tau,\ \ \max_{k\geq 0} r_k \leq r,\ \ \max_{k\geq 0}\varphi_k \leq \varphi. \quad (18)$$

For the quantity $q_k(\alpha)$ in Lemma 1, when $\alpha \in (0, 2(nL + n\mu)^{-1})$, we have $q_k(\alpha) = 1 - \alpha n \min(\pi_k)\mu < 1$. Let σ be a lower bound for $\min(\pi_k)$, $k \geq 0$, i.e., $\sigma \leq \min_{k\geq 0}\{\min(\pi_k)\}$. Note that Lemma 4 provides such a lower bound applicable to any sequence of strongly connected graphs $\{\mathbb{G}_k\}$. Better lower bounds can be obtained when the graphs have more specific structures. We have the following upper bound for $q_k(\alpha)$:

$$\max_{k\geq 0} q_k(\alpha) \leq 1 - \alpha n\sigma\mu \in (0,1). \quad (19)$$

Using the upper-bounds given in (18) and (19), for $\alpha \in (0, 2(nL+n\mu)^{-1})$, we have $M_k(\alpha,\beta,\gamma) \leq M(\alpha,\beta,\gamma)$, for all $k \geq 0$, with the matrix $M(\alpha,\beta,\gamma)$ given by

$$\begin{bmatrix}
1-\alpha n\sigma\mu & \alpha L\sqrt{n}\varphi & \alpha & \beta+\gamma(1+\alpha L\sqrt{n})\\
\alpha L\sqrt{n}\varphi & c+\alpha L\sqrt{n}\varphi & \alpha & \beta+\gamma(c+\alpha L\sqrt{n})\\
u_1 & u_2 & \tau+u_3 & Lr\gamma+u_4\\
\alpha L\sqrt{n}\varphi & (c+1)\varphi+\alpha L\sqrt{n}\varphi & \alpha & \beta+\gamma\sqrt{n}(1+\alpha L)
\end{bmatrix} \quad (20)$$

where the third row of the matrix $M(\alpha,\beta,\gamma)$ is co-linear with the fourth row, i.e.,

$$(u_1,u_2,u_3,u_4) = Lr(1+\gamma)[M(\alpha,\beta,\gamma)]_{4,:}.$$

We next define constants η_i, $i \in [6]$, as follows:

$$\eta_1 = (1-\tau)(1-c)n\sigma\mu, \quad (21a)$$
$$\eta_2 = (1-\tau)(n\sigma\mu L\sqrt{n}\varphi + L^2 n\varphi^2), \quad (21b)$$
$$\eta_3 = Lr[(1+c)\varphi + 1 - c](n\sigma\mu + L\sqrt{n}\varphi), \quad (21c)$$
$$\eta_4 = (1-\tau)[(1+c)\varphi + 1 - c], \quad (21d)$$
$$\eta_5 = \eta_1(\sqrt{n}-c) + [n\sigma\mu(1+c+L\sqrt{n}) + 2L\sqrt{n}\varphi]\eta_4, \quad (21e)$$
$$\eta_6 = (1+\gamma c - \gamma\sqrt{n})\eta_2 + (1+\gamma)\eta_3 + L^2 n\varphi\eta_4. \quad (21f)$$

We now present the main result of this paper, which states that the accelerated AB/Push-Pull algorithm (Algorithm 1) converges to the global minimizer at a linear rate.

Theorem 1: Let Assumption 1, Assumption 2, and Assumption 3 hold. Consider the iterates produced by Algorithm 1, the notations in (18)-(19) and the constants $\eta_i,\ i \in [6]$, defined in (21). If the step-size $\alpha > 0$ and the acceleration parameters $\beta \geq 0$ and $\gamma \geq 0$ are chosen such that

$$\begin{cases}
\alpha \leq \min\left\{\dfrac{1-c}{L\sqrt{n}\varphi}, \dfrac{1-\tau}{Lr}, \dfrac{\eta_1 - \kappa\eta_5}{\eta_6}, \dfrac{2}{n(L+\mu)}\right\},\\
\max\{\beta,\gamma\} < \dfrac{\eta_1}{\eta_5},\\
\beta + \gamma\sqrt{n} < 1,
\end{cases} \quad (22)$$

then $\rho_M < 1$ where ρ_M is the spectral radius of $M(\alpha,\beta,\gamma)$ and, thus, $\lim_{k\to\infty}\|x_k^i - x^*\| = 0$ with a linear convergence rate of the order of $\mathcal{O}\Big(\rho_M^k\Big)$ for all $i \in [n]$.

Proof: Recall that by Proposition 5, we have

$$V_{k+1} \leq M_k(\alpha,\beta,\gamma)V_k, \qquad \text{for all } k \geq 0.$$

With the matrix $M(\alpha,\beta,\gamma)$ defined as in (20), we obtain

$$V_{k+1} \leq M(\alpha,\beta,\gamma)V_k, \qquad \text{for all } k \geq 0. \quad (23)$$

Thus, $\|\hat{x}_k - x^*\|,\ D(\mathbf{x}_k,\phi_k),\ \|\mathbf{x}_k - \mathbf{x}_{k-1}\|$, and $S(\mathbf{y}_k,\pi_k)$ all converge to 0 linearly at rate $\mathcal{O}\Big(\rho_M^k\Big)$ if the spectral radius ρ_M of $M(\alpha,\beta,\gamma)$ satisfies $\rho_M < 1$. By Lemma 8 of [17], we

will have $\rho_M < 1$ if all diagonal entries of $M(\alpha,\beta,\gamma)$ are less than 1 and $\det(\mathbb{I} - M(\alpha,\beta,\gamma)) > 0$ where

$$\begin{aligned}&\det(\mathbb{I} - M(\alpha,\beta,\gamma))\\ =&\alpha\eta_1(1+\gamma c-\gamma\sqrt{n}) - \alpha^2(1+\gamma c-\gamma\sqrt{n})\eta_2 - \alpha^2(1+\gamma)\eta_3\\ -&\alpha[n\sigma\mu(\beta+\gamma c+\gamma L\sqrt{n}) + L\sqrt{n}\varphi(\beta+\gamma)]\eta_4 - \alpha^2L^2n\varphi\eta_4,\end{aligned}$$

with positive constants $\eta_1 > 0$, $\eta_2 > 0$, $\eta_3 > 0$ and $\eta_4 > 0$ defined as in (21a)–(21d). Let $\kappa = \max\{\beta,\gamma\}$, we can further simplify the determinant as follows

$$\det(\mathbb{I} - M(\alpha,\beta,\gamma)) = \alpha\left[(\eta_1 - \kappa\eta_5) - \alpha\eta_6\right],$$

where η_5 and η_6 are positive constants defined in equations (21e) and (21f), given that $\gamma\sqrt{n} < 1$. Hence, we need to choose $\alpha \in (0, 2(nL+n\mu)^{-1})$ and $\beta \geq 0$, $\gamma \geq 0$, so that the following conditions are satisfied

$$\begin{cases} c + \alpha L\sqrt{n}\varphi < 1, \quad \tau + \alpha L r < 1, \\ \beta + \gamma\sqrt{n} < 1, \quad (\eta_1 - \kappa\eta_5) - \alpha\eta_6 > 0. \end{cases}$$

which yields the range in (22). ■

We note that Theorem 1 includes the case when $\beta = \gamma = 0$, thus recovering the convergence rate of the AB/Push-Pull for time-varying graphs without accelaration, which has been shown in [18] and, recently, in [19] (with explicit bounds on the stepsize selection). Theorem 1 also encompasses the results of acceleration for static directed graphs where the graph remains unchanged over time, $\mathbb{G}_k = \mathbb{G}$, for all time steps, $k > 0$. This includes the case when $\gamma = 0$ from [23], the missing convergence analysis for $\beta = 0$ in [24], and the case $\beta = \gamma > 0$ presented in [25].

Remark 5: The convergence analysis for C-strongly-connected graph sequences is performed similarly to our analysis above by utilizing the results in Remark 3, and by recognizing that contractions resulting from row- and column-stochastic matrices occur after $k = C$.

V. Numerical Simulations

In this section, we evaluate the performance of the proposed algorithm by testing it on several real-world datasets, and assessing its accuracy and efficiency. We compare the results between the AB/Push-Pull algorithm (ABPP) and its variants using different acceleration techniques, including the heavy-ball momentum (ABPP-m), Nesterov momentum (ABPP-N), and the combination of the two momentum techniques (ABPP-mN). This comparison is sufficient as the performance of the AB/Push-Pull algorithm and other existing algorithms, such as Push-DIGing [14] and subgradient-push [12], as discussed in our introduction, have already been evaluated (see, for example, [18], [19]).

In our simulations, all the communication graphs are directed, time-varying, and have self-loops. To ensure the graphs are strongly connected, a directed cycle linking all agents is established at each iteration. Utilizing time-varying directed communication graphs proves to be highly practical in numerous scenarios characterized by dynamic communication networks among agents, where the flow of information or commands between agents can be directed. These scenarios often arise due to various factors such as communication delays, user mobility, and the influence of straggler effects.

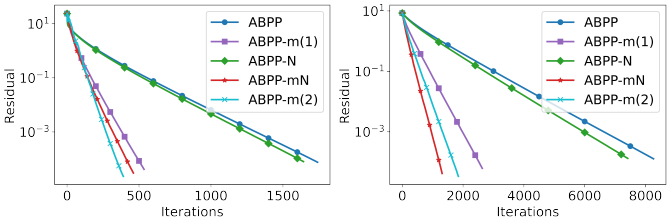
A. Distributed Ridge Regression

Consider a sensor fusion problem, as described in [17], [33]. The goal of the sensor fusion problem is to estimate an unknown parameter x by utilizing data from n sensors. Each sensor i has a measurement matrix $H_i \in \mathbb{R}^{s\times p}$, and a noisy observation $z_i = H_i x + \omega_i \in \mathbb{R}^s$ of x, where ω_i represents the noise. The resulting sensor fusion problem is given by the following minimization problem:

$$\min_{x\in\mathbb{R}^p} \sum_{i=1}^{n} \Big(\|z_i - H_i x\|^2 + \lambda_i\|x\|^2\Big),$$

where $\lambda_i > 0$ is the regularization parameter for the local cost function of sensor i.

We follow the setup in [17] where $n = 20$, $p = 20$, and $s = 1$ are chosen to make the local cost function ill-conditioned, requiring coordination among agents for fast convergence. The measurement matrix H_i is generated from a uniform distribution in the unit $\mathbb{R}^{s\times p}$ space and, then, normalized so that its Lipschitz constant is equal to 1. The noise ω_i follows an i.i.d. Gaussian process with zero mean and unit variance $\mathcal{N}(0,1)$. The regularization parameter is chosen as $\lambda_i = 0.01$ for all $i \in [n]$ to ensure the strong convexity of the loss function. Figure 1(a) illustrates the accelerated linear convergence of the proposed algorithm using different acceleration techniques (ABPP, ABPP-m(1), ABPP-N, ABPP-mN), using $\alpha = 0.25$, $\beta = 0.7$ and $\gamma = 0.05$. When Nesterov momentum is not used, a larger value of $\beta = 0.8$ may be selected (ABPP-m(2)). The plot measures performance based on the residual between the iterates x_k^i, $i \in [n]$, at time step k and the optimal value x^*, given by $\frac{1}{n}\sum_{i=1}^{n}\|x_k^i - x^*\|$.



(a) Sensor fusion problem (b) Diabetes dataset

Fig. 1: Residual plots

We also test how the changes in the momentum parameters β and γ affect the convergence rate, as shown in Figure 2. The results imply that the algorithm converges faster with higher momentum parameter values. Also, the range of parameters satisfies the conditions in (22). Note that the value of γ is much smaller due to the condition $\beta + \gamma\sqrt{n} < 1$ in (22).

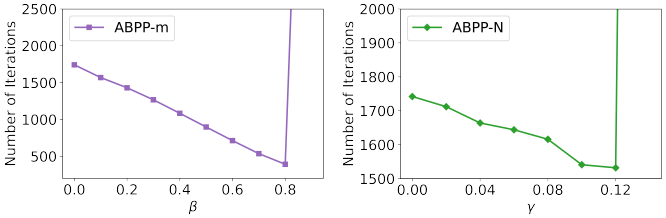


Fig. 2: Effects of varying momentum parameters ($\alpha = 0.25$)

B. Distributed Logistic Regression (L2-Regularization)

In this experiment, we examine binary classification problems using real-world datasets to evaluate the performance among the different acceleration technique over a time-varying directed network. We consider a total of N labeled data point for training, with each node i possessing a local batch of m_i training samples. The j-th sample at node i is a tuple $\{b_{ij}, y_{ij}\} \subseteq \mathbb{R}^p \times \{+1, -1\}$. To construct an estimate $x = [x_0, x_{1:}^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{p+1}$ of the coefficients, where $x_{1:} = [x_1, \ldots, x_p]^{\mathsf{T}}$, we will use the principle of maximum likelihood and define the local logistic regression cost function f_i at node i as:

$$f_i(x) = \frac{1}{m_i} \sum_{j=1}^{m_i} \ln \left[1 + \exp \left\{ -(x_{1:}^{\mathsf{T}} b_{ij} + x_0) y_{ij} \right\} \right] + \frac{\lambda}{2} \|x\|^2,$$

which is smooth and strongly convex due to the inclusion of the L2-regularization. The nodes cooperate to solve the following decentralized optimization problem:

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

We examine the performance of the proposed algorithm using two datasets.

Pima Indians Diabetes Dataset: We evaluate the performance of our algorithm using the Diabetes dataset, which consists of $N = 700$ training samples and 68 test samples. The dataset is divided among $n = 7$ agents, with each agent having $m_i = 100$ samples. A regularization parameter of $\lambda = 0.001$ is used and the algorithm stops when the consensus error among agents is less than 10^{-7}. The accuracy on the test set is 79.41% for all 4 algorithms. Figure 1(b) illustrates the accelerated linear convergence of the proposed algorithm using different acceleration techniques for $\alpha = 0.5$, $\beta = 0.7$ (ABPP-m(1), ABPP-mN) and $\gamma = 0.1$ while $\beta = 0.8$ for ABPP-m(2).

MNIST Dataset: The task at hand is to perform digit classification on the MNIST dataset. Figure 3 shows a part of randomly selected samples, where each image is featured as a 784-dimensional vector. We use 2000 training samples and 1000 test samples. The problem is divided among 10 agents, with each agent handling 200 samples. The regularization parameter is set to $\lambda = 0.001$, and the algorithm terminates when the consensus error among agents is below 10^{-3}.

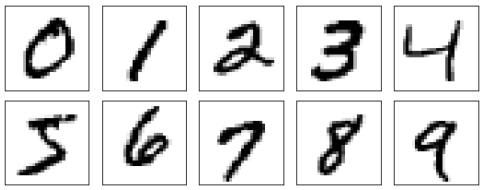


Fig. 3: Samples from MNIST Dataset

As a sanity check of accuracy, we visually examine the coefficients generated by the proposed algorithm for the binary classification task of hand-written digit 0 versus non-zero digits. The visualization, shown in Figure 4, highlights the most important features identified by the algorithm for classifying a digit as 0. The blue pixels indicate the highest probability of a digit being classified as 0, while the red pixels indicate the lowest probability. As seen in Figure 4, the blue pixels form the shape of a 0, with more red pixels in the center, indicating that these pixels are less likely to be shaded in images of a 0.

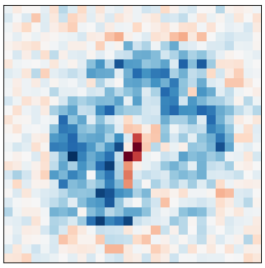


Fig. 4: Heatmap of Coefficients for Logistic Regression in Classifying Zero and Non-Zero Digits using ABPP-mN

We now assess the performance of the four algorithms in classifying hand-written digits. The tasks are to classify hand-written digits $\{1, 2\}$ and classify hand-written digits $\{3, 5\}$. The results of the numerical experiments are shown in Table I which include the number of iterations, the computational time and the accuracy on the test set.

	Classify $\{1,2\}$			Classify $\{3,5\}$		
	# Iterations	Time (s)	Accuracy	# Iterations	Time (s)	Accuracy
ABPP	1326	75.5	98.3%	3287	246.0	95.49%
ABPP–m	1008	51.9	98.6%	1704	137.4	95.30%
ABPP–N	1380	95.8	98.3%	3261	266.5	95.49%
ABPP–mN	944	47.6	98.6%	1606	117.8	95.70%

TABLE I: Performance for $\alpha = 0.01$, $\beta = 0.3$ and $\gamma = 0.01$.

Overall, the results demonstrate that the proposed algorithm with acceleration techniques substantially enhances the convergence rate while having comparable performance to the AB/Push-Pull algorithm. The heavy-ball momentum, in particular, significantly improves the convergence, and incorporating both heavy-ball and Nesterov momentum may be beneficial in some cases (Figure 1). The Nesterov parameter γ is influenced by the number of agents in the network, as it is multiplied by $\sqrt{n}$ (see (22)). The heavy-ball parameter β, on the other hand, can be set to a larger value for faster convergence. By considering different values for the Nesterov and heavy-ball acceleration parameters β and γ, our proposed algorithm offers greater flexibility and the potential for faster convergence.

VI. Conclusion

In this paper, we propose a novel approach for solving distributed optimization problems over time-varying directed networks. The proposed algorithm incorporates acceleration techniques to improve the performance of the AB/Push-Pull algorithm. Theoretical analysis is provided to prove linear convergence to the optimal solution under certain conditions. Additionally, explicit bounds for the step-size and momentum parameters are derived based on the properties of the cost functions and network structure. The numerical results demonstrate the benefits of the proposed acceleration techniques on the AB/Push-Pull algorithm. An interesting open direction is to theoretically analyze the acceleration over the AB/Push-Pull algorithm.

REFERENCES

[1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd IPSN*, 2004, pp. 20–27.

[2] D. M. Stipanović, G. İnalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," in *41st IEEE Conf. Decis. Control*, vol. 3, 2002, pp. 2829–2835 vol.3.

[3] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *50th Annu. Allerton Conf. Commun. Control Comput.*, 2012, pp. 1543–1550.

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

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, 2012.

[6] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, 2014.

[7] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 994–1009, 2016.

[8] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network Newton distributed optimization methods," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 146–161, 2017.

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

[10] A. Olshevsky, "Linear time average consensus and distributed optimization on fixed graphs," *SIAM J. Control Optim.*, vol. 55, no. 6, pp. 3990–4014, 2017.

[11] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proc. of the 51st IEEE Conf. Decis. Control*, 2012, pp. 5453–5458.

[12] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, 2015.

[13] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1329–1339, 2018.

[14] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[15] C. Xi, V. S. Mai, R. Xin, E. H. Abed, and U. A. Khan, "Linear convergence in optimization over directed graphs with row-stochastic matrices," *IEEE Trans. Autom. Control*, 2018.

[16] R. Xin, C. Xi, and U. A. Khan, "FROST—fast row-stochastic optimization with uncoordinated step-sizes," *EURASIP J. Adv. Signal Process.*, pp. 1–14, 2019.

[17] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push-Pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, 2021.

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

[19] A. Nedić, D. T. A. Nguyen, and D. T. Nguyen, "AB/Push-Pull method for distributed optimization in time-varying directed networks," *arXiv preprint arXiv:2209.06974*, 2022.

[20] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964.

[21] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2003, vol. 87.

[22] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, 2020.

[23] R. Xin and U. A. Khan, "Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2627–2633, 2020.

[24] R. Xin, D. Jakovetić, and U. A. Khan, "Distributed Nesterov gradient methods over arbitrary graphs," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1247–1251, 2019.

[25] H. Li, H. Cheng, Z. Wang, and G.-C. Wu, "Distributed Nesterov gradient and heavy-ball double accelerated asynchronous optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5723–5737, 2021.

[26] Q. Lu, X. Liao, H. Li, and T. Huang, "A Nesterov-like gradient tracking algorithm for distributed optimization over directed networks," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 10, pp. 6258–6270, 2021.

[27] X. Shi, H. Liu, J. Chen, and X. Wang, "An accelerated distributed optimization algorithm over time-varying digraphs with column-stochastic matrices," in *2021 33rd Chin. Control Decis. Conf. (CCDC)*, 2021, pp. 2124–2129.

[28] B. Polyak, *Introduction to Optimization*. New York : Optimization Software, Inc., 1987.

[29] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th IEEE FOCS*, 2003, pp. 482–491.

[30] P. Rezaienia, B. Gharesifard, T. Linder, and B. Touri, "Push-sum on random graphs: Almost sure convergence and convergence rate," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1295–1302, 2020.

[31] S. Bubeck, Y. T. Lee, and M. Singh, "A geometric alternative to Nesterov's accelerated gradient descent," *arXiv preprint arXiv:1506.08187*, 2015.

[32] D. T. A. Nguyen, D. T. Nguyen, and A. Nedić, "Distributed Nash equilibrium seeking over time-varying directed communication networks," *arXiv preprint arXiv:2201.02323*, 2022.

[33] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, 2018.

Duong Thuy Anh Nguyen (Graduate Student Member, IEEE) received the M.Sc. degree in Applied Mathematics from the University of Louisiana at Lafayette, LA, USA in 2019. She is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Energy Engineering at Arizona State University, AZ, USA. Her current research interests include distributed optimization, operations research and game theory. The research focuses on developing mathematical models for decision-making under uncertainty, fair and privacy-preserving mechanism designs and distributed algorithms for large-scale network in multi-agent systems. Research applications include cloud/edge computing, electric vehicles, non-cooperative games over communication networks.

Duong Tung Nguyen received the Ph.D. degree in electrical and computer engineering from the University of British Columbia, BC, Canada. He is currently an assistant professor in the School of Electrical, Computer and Energy Engineering at Arizona State University, AZ, USA. His research lies at the intersection of operations research, AI, economics, and engineering, with a focus on developing new mathematical models and techniques for decision-making and economic analysis of large-scale networked systems such as cloud/edge computing, smart grids, intelligent transportation, and crowdsourcing.

Angelia Nedić has a Ph.D. from Moscow State University, Moscow, Russia, in Computational Mathematics and Mathematical Physics (1994), and a Ph.D. from Massachusetts Institute of Technology, Cambridge, USA, in Electrical and Computer Science Engineering (2002). She has worked as a senior engineer in BAE Systems North America, Advanced Information Technology Division at Burlington, MA. Currently, she is a faculty member of the School of Electrical, Computer and Energy Engineering at Arizona State University at Tempe. Prior to joining Arizona State University, she has been a Willard Scholar faculty member at the University of Illinois, Urbana-Champaign. She is a recipient (jointly with her co-authors) of the Best Paper Award at the Winter Simulation Conference 2013 and the Best Paper Award at the International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) 2015. Her general research interest is in optimization, large scale complex systems dynamics, variational inequalities, and games.

APPENDIX

A. Proof of Proposition 1

Proof: Under Assumption 3, the unique minimizer x^* of $f(x)$ over $x \in \mathbb{R}^p$ exists. By subtracting x^* from both sides of the relation for $\hat{x}_k$ in Lemma 8, and by adding and subtracting $\sum_{i=1}^n [\phi_{k+1}]_i \alpha n [\pi_k]_i \nabla f(\hat{x}_k)$, we obtain:

$$\begin{aligned}\hat{x}_{k+1} - x^* = \hat{x}_k - x^* - \sum_{i=1}^n [\phi_{k+1}]_i \alpha n [\pi_k]_i \nabla f(\hat{x}_k) \\ + \alpha \sum_{i=1}^n [\phi_{k+1}]_i \Big(n[\pi_k]_i \nabla f(\hat{x}_k) - y_k^i \Big) \\ + \sum_{i=1}^n (\beta [\phi_{k+1}]_i + \gamma [\phi_k]_i)(x_k^i - x_{k-1}^i).\end{aligned}$$

As a result of the convexity of the norm and the stochastic nature of ϕ_{k+1}, we can deduce that:

$$\begin{aligned}\|\hat{x}_{k+1} - x^*\| \leq \sum_{i=1}^n [\phi_{k+1}]_i \|\hat{x}_k - x^* - \alpha n [\pi_k]_i \nabla f(\hat{x}_k)\| \\ + \alpha \sum_{i=1}^n [\phi_{k+1}]_i \|y_k^i - n[\pi_k]_i \nabla f(\hat{x}_k)\| \\ + \sum_{i=1}^n (\beta [\phi_{k+1}]_i + \gamma [\phi_k]_i) \|x_k^i - x_{k-1}^i\|. \quad (24)\end{aligned}$$

For a step-size α satisfying $\alpha \in (0, \frac{2}{n[\pi_k]_i L})$, for all $i \in [n]$, by Lemma 1, it follows that

$$\|\hat{x}_k - x^* - \alpha n [\pi_k]_i \nabla f(\hat{x}_k)\| \leq q_{i,k}(\alpha) \|\hat{x}_k - x^*\|,$$

with $q_{i,k}(\alpha) = \max\{|1 - \alpha n [\pi_k]_i \mu|, |1 - \alpha n [\pi_k]_i L|\}$.

We have the following estimate for the first term on the right-hand-side of (24):

$$\begin{aligned}&\sum_{i=1}^n [\phi_{k+1}]_i \|\hat{x}_k - x^* - \alpha n [\pi_k]_i \nabla f(\hat{x}_k)\| \\ &\leq \sum_{i=1}^n [\phi_{k+1}]_i q_{i,k}(\alpha) \|\hat{x}_k - x^*\| \leq q_k(\alpha) \|\hat{x}_k - x^*\|,\end{aligned}$$

using the stochasticity of ϕ_{k+1} and $q_k(\alpha) = \max\{|1 - \alpha n \min(\pi_k)\mu|, |1 - \alpha n \min(\pi_k) L|\}$. Since $\max(\phi_{k+1}) \leq 1$, to estimate the second term on the right-hand-side in (24), we factor out $[\pi_k]_i$ (which is positive by Assumption 1 and Lemma 4), as follows

$$\begin{aligned}&\sum_{i=1}^n [\phi_{k+1}]_i \|y_k^i - n[\pi_k]_i \nabla f(\hat{x}_k)\| \\ &\leq \sum_{i=1}^n \|y_k^i - n[\pi_k]_i \nabla f(\hat{x}_k)\| = \sum_{i=1}^n [\pi_k]_i \left\| \frac{y_k^i}{[\pi_k]_i} - n \nabla f(\hat{x}_k) \right\| \\ &\leq \sum_{i=1}^n [\pi_k]_i \left\| \frac{y_k^i}{[\pi_k]_i} - \sum_{\ell=1}^n y_k^\ell \right\| + \sum_{i=1}^n [\pi_k]_i \left\| \sum_{\ell=1}^n y_k^\ell - n \nabla f(\hat{x}_k) \right\| \\ &\leq \sqrt{\sum_{i=1}^n [\pi_k]_i \left\| \frac{y_k^i}{[\pi_k]_i} - \sum_{\ell=1}^n y_k^\ell \right\|^2} + \left\| \sum_{\ell=1}^n y_k^\ell - n \nabla f(\hat{x}_k) \right\| \\ &\leq S(\mathbf{y}_k, \pi_k) + \left\| \sum_{\ell=1}^n y_k^\ell - n \nabla f(\hat{x}_k) \right\|,\end{aligned}$$

where we add and subtract $\sum_{\ell=1}^n y_k^\ell$, and use the triangle inequality to obtain the second inequality. We use the fact that the vector sequence $\{\pi_k\}$ is stochastic to derive the third inequality (see Lemma 3), and the last inequality follows from the definition of the S-quantity in (10b). We now estimate the last term in the preceding relation. By Lemma 7 we have $\sum_{\ell=1}^n y_k^\ell = \sum_{\ell=1}^n \nabla f_\ell(s_k^\ell)$; hence, in view of $\nabla f = \frac{1}{n} \sum_{\ell=1}^n \nabla f_\ell$, it follows that

$$\begin{aligned}&\left\| \sum_{\ell=1}^n y_k^\ell - n \nabla f(\hat{x}_k) \right\| = \left\| \sum_{\ell=1}^n \Big(\nabla f_\ell(s_k^\ell) - \nabla f_\ell(\hat{x}_k) \Big) \right\| \\ &\leq \sum_{\ell=1}^n \|\nabla f_\ell(s_k^\ell) - \nabla f_\ell(\hat{x}_k)\| \leq L \sum_{\ell=1}^n \|s_k^\ell - \hat{x}_k\| \\ &= L\sqrt{n} \varphi_k D(\mathbf{x}_k, \phi_k) + L\sqrt{n}\gamma \|\mathbf{x}_k - \mathbf{x}_{k-1}\|,\end{aligned}$$

where we use the gradient Lipschitz continuity property for each f_i, the s-update in (6) the definitions of the D-quantity in (10a) and the constant φ_k in (11). Hence, we obtain the following relation for the second term in (24):

$$\begin{aligned}&\sum_{i=1}^n [\phi_{k+1}]_i \|y_k^i - n[\pi_k]_i \nabla f(\hat{x}_k)\| \\ &\leq S(\mathbf{y}_k, \pi_k) + L\sqrt{n} \varphi_k D(\mathbf{x}_k, \phi_k) + L\sqrt{n}\gamma \|\mathbf{x}_k - \mathbf{x}_{k-1}\|.\end{aligned}$$

For the final term in (24), since ϕ_k is stochastic, we have

$$\sum_{i=1}^n [\phi_k]_i \|x_k^i - x_{k-1}^i\| \leq \sqrt{\sum_{i=1}^n [\phi_k]_i \|x_k^i - x_{k-1}^i\|^2} \leq \|\mathbf{x}_k - \mathbf{x}_{k-1}\|.$$

We can obtain similar relation when the weight is ϕ_{k+1}, thus,

$$\sum_{i=1}^n (\beta [\phi_{k+1}]_i + \gamma [\phi_k]_i) \|x_k^i - x_{k-1}^i\| \leq (\beta + \gamma) \|\mathbf{x}_k - \mathbf{x}_{k-1}\|.$$

Substituting the estimates obtained above for each term on the right-hand-side of (24) gives the desired relation. ∎

B. Proof of Proposition 2

Proof: Let $u_k^i = x_k^i - x_{k-1}^i$ and $\hat{u}_k = \sum_{i=1}^n [\phi_k]_i u_k^i$. Subtracting the relation for $\hat{x}_k$ given in Lemma 8 from the x-update in equation (12), and using the triangle inequality, we have

$$\begin{aligned}D(\mathbf{x}_{k+1}, \phi_{k+1}) \leq &\sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| \sum_{j=1}^n [A_k]_{ij} x_k^j - \hat{x}_k \right\|^2} \\ &+ \alpha \sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| y_k^i - \sum_{j=1}^n [\phi_{k+1}]_j y_k^j \right\|^2} \\ &+ \gamma \sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| \sum_{j=1}^n [A_k]_{ij} u_k^j - \hat{u}_k \right\|^2} \\ &+ \beta \sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| u_k^i - \sum_{j=1}^n [\phi_{k+1}]_j u_k^j \right\|^2}. \quad (25)\end{aligned}$$

Under Assumption 1, we use Lemma 5 to estimate the first term in (25), with $A = A_k$, $x_i = x_k^i$ and $\phi_{k+1}^\mathsf{T} A_k = \phi_k^\mathsf{T}$:

$$\sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| \sum_{j=1}^n [A_k]_{ij} x_k^j - \hat{x}_k \right\|^2} \leq c_k \sqrt{\sum_{j=1}^n [\phi_k]_j \|x_k^j - \hat{x}_k\|^2} \leq c_k D(\mathbf{x}_k, \phi_k). \quad (26)$$

Similarly for the third term in (25), using Lemma 5 with $A = A_k$, $x_i = u_k^i$ and $\phi_{k+1}^\mathsf{T} A_k = \phi_k^\mathsf{T}$, we obtain

$$\sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| \sum_{j=1}^n [A_k]_{ij} u_k^j - \hat{u}_k \right\|^2} \leq c_k \sqrt{\sum_{j=1}^n [\phi_k]_j \|u_k^j - \hat{u}_k\|^2}.$$

Next, using the relation in Lemma 2 with $\gamma_i = [\phi_k]_i$, $u_i = u_k^i$ and $u = 0$, it follows that

$$\sum_{i=1}^n [\phi_k]_i \left\| u_k^i - \sum_{j=1}^n [\phi_k]_j u_k^j \right\|^2 \leq \sum_{i=1}^n [\phi_k]_i \|u_k^i\|^2 \leq \|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2,$$

where the last inequality follows from the stochasticity of ϕ_k and the definition of u_k^i, for all $i \in [n]$. Thus,

$$\sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| \sum_{j=1}^n [A_k]_{ij} u_k^j - \hat{u}_k \right\|^2} \leq c_k \|\mathbf{x}_k - \mathbf{x}_{k-1}\|. \quad (27)$$

Similar argument can be used to estimate the forth term in (25), which yields

$$\sum_{i=1}^n [\phi_{k+1}]_i \left\| u_k^i - \sum_{j=1}^n [\phi_{k+1}]_j u_k^j \right\|^2 \leq \sum_{i=1}^n [\phi_{k+1}]_i \|u_k^i\|^2 \leq \|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2, \quad (28)$$

and, for the second term in (25), as follows

$$\begin{aligned}
&\sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \left\| y_k^i - \sum_{j=1}^n [\phi_{k+1}]_j y_k^j \right\|^2} \leq \sqrt{\sum_{i=1}^n [\phi_{k+1}]_i \|y_k^i\|^2} \\
&\leq \sqrt{\max_{j\in[n]} ([\phi_{k+1}]_j [\pi_k]_j)} \sqrt{\sum_{i=1}^n \frac{\|y_k^i\|^2}{[\pi_k]_i}} \leq \sqrt{\sum_{i=1}^n \frac{\|y_k^i\|^2}{[\pi_k]_i}} \\
&\leq L\sqrt{n}\,(\varphi_k \|\hat{x}_k - x^*\| + \varphi_k D(\mathbf{x}_k, \phi_k) + \gamma \|\mathbf{x}_k - \mathbf{x}_{k-1}\|) \\
&\quad + S(\mathbf{y}_k, \pi_k), \quad (29)
\end{aligned}$$

where the last inequality follows from Lemmas 9 and 10. By combining the estimates for each term in (25), as given by relations (26)–(29), we arrive at the desired relation. ∎

C. Proof of Proposition 4

Proof: By defining $w_k^i = \sum_{j=1}^n [B_k]_{ij} y_k^j$, $\mathbf{w}_k = (w_k^1, \ldots, w_k^n)$ and $\mathbf{g}_k = (\nabla f_1(s_k^1), \ldots, \nabla f_n(s_k^n))$, the update for the y-iterate in compact form is given as

$$\mathbf{y}_{k+1} = \mathbf{w}_k + \mathbf{g}_{k+1} - \mathbf{g}_k \qquad \text{for all } k \geq 0. \quad (30)$$

Let $\Lambda = \operatorname{diag}^{-1}(\pi_{k+1})$, we can write

$$\mathbf{y}_{k+1}\Lambda = \mathbf{w}_k\Lambda + (\mathbf{g}_{k+1} - \mathbf{g}_k)\Lambda \qquad \text{for all } k \geq 0.$$

By subtracting the vector $\bar{\mathbf{y}}_{k+1} = (\bar{y}_{k+1}, \ldots, \bar{y}_{k+1})$, where $\bar{y}_{k+1} = \sum_{j=1}^n y_{k+1}^j$, from both sides of the preceding relation, we have for all $k \geq 0$,

$$\mathbf{y}_{k+1}\Lambda - \bar{\mathbf{y}}_{k+1} = \mathbf{w}_k\Lambda - \bar{\mathbf{y}}_k + (\bar{\mathbf{y}}_k - \bar{\mathbf{y}}_{k+1}) + (\mathbf{g}_{k+1} - \mathbf{g}_k)\Lambda.$$

By taking π_{k+1}-induced norm on both sides and noting that $S(\mathbf{y}_{k+1}, \pi_{k+1}) = |\mathbf{y}_{k+1}\Lambda - \bar{\mathbf{y}}_{k+1}|\pi k + 1$, we obtain

$$S(\mathbf{y}_{k+1}, \pi_{k+1}) \leq \|\mathbf{w}_k\Lambda - \bar{\mathbf{y}}_k\|_{\pi_{k+1}} + \|\bar{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_k\|_{\pi_{k+1}} + \|(\mathbf{g}_{k+1} - \mathbf{g}_k)\Lambda\|_{\pi_{k+1}}. \quad (31)$$

For the first term in (31), by using the definitions of $\mathbf{w}_k$ and $\bar{\mathbf{y}}_k$, we can deduce that

$$\begin{aligned}
\|\mathbf{w}_k\Lambda - \bar{\mathbf{y}}_k\|_{\pi_{k+1}} &= \sqrt{\sum_{i=1}^n [\pi_{k+1}]_i \left\| \frac{w_k^i}{[\pi_{k+1}]_i} - \sum_{\ell=1}^n y_k^\ell \right\|^2} \\
&\leq \tau_k \sqrt{\sum_{i=1}^n [\pi_k]_i \left\| \frac{y_i}{[\pi_k]_i} - \sum_{\ell=1}^n y_\ell \right\|^2} = \tau_k\, S(\mathbf{y}_k, \pi_k),
\end{aligned}$$

where the first inequality follows from Lemma 6 with $\mathbb{G} = \mathbb{G}_k$, a strongly connected directed graph (see Assumption 1), $B = B_k$, $\pi = \pi_{k+1}$, and $\nu = \pi_k$. The last equality follows from the definition of $S(\mathbf{y}_k, \pi_k)$.

For the second term in (31), since $\bar{\mathbf{y}}_k = \sum_{i=1}^n y_k^i = \sum_{i=1}^n \nabla f_i(s_k^i)$ (as stated in Lemma 7), we have

$$\begin{aligned}
\|\bar{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_k\|_{\pi_{k+1}} &= \sqrt{\sum_{i=1}^n [\pi_{k+1}]_i \|\bar{y}_{k+1} - \bar{y}_k\|^2} = \|\bar{y}_{k+1} - \bar{y}_k\| \\
&= \left\| \sum_{i=1}^n \big(\nabla f_i(s_{k+1}^i) - \nabla f_i(s_k^i)\big) \right\| \leq \sum_{i=1}^n \|\nabla f_i(s_{k+1}^i) - \nabla f_i(s_k^i)\| \\
&\leq L \sum_{i=1}^n \|s_{k+1}^i - s_k^i\| \leq L\sqrt{n} \|\mathbf{s}_{k+1} - \mathbf{s}_k\|,
\end{aligned}$$

where the last inequality follows from the Lipschitz continuity of the gradients ∇f_i (Assumption 2).

For the last term in relation (31), we have

$$\begin{aligned}
\|(\mathbf{g}_{k+1} - \mathbf{g}_k)\Lambda\|_{\pi_{k+1}} &= \sqrt{\sum_{i=1}^n \frac{\|\nabla f_i(s_{k+1}^i) - \nabla f_i(s_k^i)\|^2}{[\pi_{k+1}]_i}} \\
&\leq L \sqrt{\sum_{i=1}^n \frac{\|s_{k+1}^i - s_k^i\|^2}{[\pi_{k+1}]_i}} \leq \frac{L}{\sqrt{\min(\pi_{k+1})}} \|\mathbf{s}_{k+1} - \mathbf{s}_k\|,
\end{aligned}$$

where the first inequality follows by the Lipschitz continuity of the gradients ∇f_i. Thus,

$$S(\mathbf{y}_{k+1}, \pi_{k+1}) \leq \tau_k\, S(\mathbf{y}_k, \pi_k) + L r_k \|\mathbf{s}_{k+1} - \mathbf{s}_k\|,$$

where $r_k = \sqrt{n} + \frac{1}{\sqrt{\min(\pi_{k+1})}}$. Using the compact form of the s-update of the method in (6), we further obtain

$$\|\mathbf{s}_{k+1} - \mathbf{s}_k\| \leq (1+\gamma)\|\mathbf{x}_{k+1} - \mathbf{x}_k\| + \gamma\|\mathbf{x}_k - \mathbf{x}_{k-1}\|.$$

The desired relation follows from the preceding two relations. ∎