2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Projected Push-Pull For Distributed Constrained Optimization Over Time-Varying Directed Graphs

Orhan Eren Akgün*, Arif Kerem Dayı*, Stephanie Gil, and Angelia Nedić

***Abstract*— We introduce the Projected Push-Pull algorithm that enables multiple agents to solve a distributed constrained optimization problem with private cost functions and global constraints, in a collaborative manner. Our algorithm employs projected gradient method to deal with constraints and a lazy update rule to control the trade-off between the consensus and optimization steps in the protocol. We prove that our algorithm achieves geometric convergence over time-varying directed graphs while ensuring that decision variables always stay within the constraint set. We derive explicit bounds for step sizes that guarantee geometric convergence based on the strong-convexity and smoothness properties of cost functions, and graph properties. Moreover, we provide additional theoretical results on the usefulness of lazy updates, revealing the challenges in the analysis of any gradient tracking method that uses projection operators in a distributed constrained optimization setting. We validate our theoretical results with numerical studies over different graph types, showing that our algorithm achieves geometric convergence empirically.**

I. INTRODUCTION

In this paper, we are concerned with a class of distributed optimization problems where a set of n agents are trying to solve a problem with the structure:

$$\min_{x\in\mathcal{X}} f(x), \quad \text{where} \quad f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where x is the decision variable, each cost function $f_i : \mathbb{R}^d \to \mathbb{R}$ is known by agent i only and is strongly convex with Lipschitz continuous gradients, and the constraint set $\mathcal{X} \subseteq \mathbb{R}^d$ is closed and convex. We are interested in the case where agents communicate over a possibly time-varying directed graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ where $\mathcal{V}$ with $|\mathcal{V}| = n$ represents the set of agents and the set $\mathcal{E}_k$ represents the directed communication links at time k. This setup has various applications in control [1], robotics [2], and sensor networks [3].

Many distributed optimization applications demand fast algorithms due to time and computational constraints, which makes the convergence rate of the used algorithm critical. However, a simple extension of gradient descent to distributed optimization does not achieve geometric convergence even with strongly convex cost functions [4], [5]. Therefore, gradient tracking was introduced to achieve geometric convergence in undirected [4], [6], [7] and directed graphs [7]–[10]. In gradient tracking methods, agents maintain a decision variable and an estimate of the global gradient. At each step, agents first perform a consensus step and an optimization step on the decision variable using the estimated global gradient. Then, they update their estimate of their global gradient using their neighbors' estimates and their local gradient. In particular, the Push-Pull algorithm introduced in [9], [10] achieves geometric convergence in directed, time-varying graphs [11], [12]. Unlike other gradient tracking methods, Push-Pull uses row and column stochastic mixing matrices for averaging decision and gradient tracking variables, respectively. Therefore, it does not require estimating the non-one Perron vector of the mixing matrix, which would introduce additional communication and computation costs. However, it does not handle constrained optimization problems. Indeed, despite great progress in distributed unconstrained optimization algorithms, their counterparts in the constrained optimization space still remain underexplored. Our goal in this work is to develop a projected gradient descent based Push-Pull algorithm variant to achieve geometric convergence in *constrained optimization problems* over time-varying directed graphs.

Extending gradient tracking methods, including Push-Pull, to handle constrained optimization is a non-trivial task. Projection based algorithms for constrained optimization have some fundamental differences from their counterparts for unconstrained optimization. First, the non-linearity of the projection operator limits our ability to manipulate the mixing matrices in the analysis, which is an essential part of the analysis in the unconstrained case. Second, in the unconstrained case, the global gradient vanishes at the optimal point, which is heavily used in existing analyses. However, the gradient does not necessarily vanish at the optimum in the constrained setting. Since the gradient at the optimal point can be non-zero, the step size in the constrained case does not give fine-grained control over the tradeoff between different errors, such as the optimality error, consensus error, and the gradient tracking error, which are standard in the analysis of all gradient tracking methods.

The works in [13] and [14] propose gradient tracking based methods for the constrained optimization problems over static directed graphs. However, both algorithms require multiple consensus steps per optimization step which increases communication costs. Recent work in [15] eliminates the need for multiple consensus steps. Yet their results are limited to static undirected graphs, and the decision variables are not guaranteed to stay in the constraint set at every time

(*Co-primary authors). O. E. Akgün, A. K. Dayı, and S. Gil are with the School of Engineering and Applied Sciences, Harvard University, USA: erenakgun@g.harvard.edu, keremdayi@college.harvard.edu, sgil@seas.harvard.edu. A. Nedić is with the School of Electrical, Computer and Energy Engineering, Arizona State University, USA: Angelia.Nedich@asu.edu.

The authors gratefully acknowledge partial support through NSF CNS 2147641 and 2147694.

step. Conversely, the SONATA algorithm proposed in [16] is later on shown to have a geometric convergence rate for time-varying directed graphs in [17]. However, the SONATA and algorithms proposed in [13] and [14] all use only row stochastic mixing matrices and, therefore, require estimating the non-one Perron vector of the mixing matrix, increasing computation and communication costs.

Ideally, we want a distributed constrained optimization algorithm that 1) achieves a geometric convergence rate, 2) works for directed graphs and time-varying graphs, 3) has low communication cost (i.e., does not require multiple consensus steps or estimation of additional system parameters), 4) minimizes the number of costly operations such as projection, and 5) keeps the decision variable in the constraint set at all time steps. With this motivation, we introduce the Projected Push-Pull algorithm that satisfies all the aforementioned requirements. Similar to Push-Pull, we employ row and column stochastic mixing matrices. This allows our algorithm to work in directed time-varying graphs without needing to estimate the non-one Perron vector of the mixing matrices. To handle the constrained case, we use projection to keep the decision variables in the constraint set and an extra step size to control the tradeoff between consensus and optimization. We prove the geometric convergence rate of the algorithm for time-varying directed graphs. Our contributions can be summarized as follows

- We introduce a novel distributed projected gradient algorithm based on the Push-Pull to solve distributed constrained optimization problems with structure as in Equation (1) over time-varying directed graphs.
- We prove that with a small enough step size, our algorithm has a geometric convergence rate for time-varying directed graphs. We characterize the valid range for the step size based on various problem-based parameters, such as the smoothness and strong convexity of the cost functions and properties of the communication graph.
- We provide impossibility results that show some fundamental limitations of distributed gradient methods using projection in constrained optimization settings.
- We empirically show that our algorithm attains geometric convergence via numerical studies with different graph types.

II. Notation & Terminology

All vectors are column vectors by default unless stated otherwise. The i-th entry of a vector u is denoted by u_i; it is $[u_k]_i$ if u_k is time varying where $k \geq 0$ is the time step. For a vector u, $\min u$ and $\max u$ denote the smallest and largest entries of u, respectively. For any matrix A, we denote its ij-th entry by A_{ij}. If it is a time-varying matrix, we denote it by $[A_k]_{ij}$. We denote the smallest positive element of a non-negative matrix A by $\min\{A^+\}$. A non-negative matrix is row stochastic if all of its row sums are equal to 1, and it is column stochastic if all of its column sums are equal to 1.

We use $\langle a, b\rangle$ to denote the Euclidean inner product and $\|x\| = \sqrt{\langle x, x\rangle}$ to denote the Euclidean norm. For any vector $u \in \mathbb{R}^n$ with $u_i > 0\ \forall i$, we define the u-weighted norm of $\mathbf{x} \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$ (n copies of $\mathbb{R}^d$) as $\|\mathbf{x}\|_u = \sqrt{\sum_{i=1}^n u_i \|x_i\|^2}$ where $x_i \in \mathbb{R}^d$.

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be strongly connected if there is a directed path between any pair of the nodes in the graph. Finally, we define a projection operator as follows:

Definition 1 (Projection onto $\mathcal{X}$): Let $\mathcal{X} \subseteq \mathbb{R}^d$ be closed and convex. Then, we define the projection operator $\Pi_{\mathcal{X}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ as follows

$$\Pi_{\mathcal{X}}(x) = \arg\min_{z \in \mathcal{X}} \|x - z\|.$$

III. Problem Setup

We consider a distributed multi-agent system of n agents where agents need to solve a distributed optimization task in a collaborative manner. The agents' goal is to solve the following constrained minimization problem

$$\min_{x \in \mathcal{X}} f(x), \quad \text{where} \quad f(x) \triangleq \frac{1}{n}\sum_{i=1}^n f_i(x), \tag{2}$$

where each cost function $f_i : \mathbb{R}^d \to \mathbb{R}$ is known by agent i only and the constraint set $\mathcal{X} \subseteq \mathbb{R}^d$ is closed and convex. We make the following assumptions about the cost functions:

Assumption 1 (Strongly Convex Objective): For all agents i, $f_i(x)$ is μ-strongly convex, i.e, for some $\mu > 0$, we have $\langle \nabla f_i(x) - \nabla f_i(y), x - y\rangle \geq \mu \|x - y\|^2$, for all $x, y \in \mathbb{R}^d$.

Assumption 2 (Lipschitz Continuity of Gradients): For all agents i, $\nabla f_i(x)$ is L-Lipschitz continuous, i.e, for some $L > 0$, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$, for all $x, y \in \mathbb{R}^d$.

We assume that at each time step $k \in \mathbb{N}$, agents communicate over a directed graph, denoted by $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$. The set $\mathcal{V}$ with $|\mathcal{V}| = n$ represents the set of agents and the set $\mathcal{E}_k$ represents the directed communication links at time k. An edge $(i, j) \in \mathcal{E}_k$ indicates that agent i can send information to agent j at time k. Moreover, if $(i, j) \in \mathcal{E}_k$, we say that i is an in-neighbor of j and j is an out-neighbor of i. We make the following assumption on the communication graphs $\mathcal{G}_k$.

Assumption 3 (Strong Connectivity): $\mathcal{G}_k$ is strongly connected for all k.

IV. Algorithm

In this section, we introduce the Projected Push-Pull algorithm. In the algorithm, all agents maintain two decision variables $x_i[k]$ and $z_i[k]$, and a gradient tracking variable $y_i[k]$. Agents initialize $x_i[0] = z_i[0] \in \mathcal{X}$ arbitrarily, and $y_i[0] = \nabla f_i(x_i[0])$. At each communication round k, agents get $z_j[k]$ and the scaled gradient tracking variable $[C_k]_{ij} y_j[k]$ from their in-neighbors and perform the following updates:

$$x_i[k+1] = \sum_{j=1}^n [R_k]_{ij} z_j[k], \tag{3a}$$

$$y_i[k+1] = \sum_{j=1}^n [C_k]_{ij} y_j[k] + \nabla f_i(x_i[k+1]) - \nabla f_i(x_i[k]), \tag{3b}$$

$$z_i[k+1] = (1-\lambda)x_i[k+1] + \lambda \Pi_{\mathcal{X}}\left(x_i[k+1] - \eta y_i[k+1]\right), \tag{3c}$$

where $\eta > 0, \lambda \in (0, 1]$ are two different step sizes. We will refer to Equation (3c) as the lazy update rule. We formally describe how agent $i \in \mathcal{V}$ runs the protocol in Algorithm 1.

Algorithm 1 Projected Push-Pull

Input: Choose parameters η, λ according to Theorem 1.

1: Each agent i simultaneously does the following:
2: Initialize $x_i[0] = z_i[0] \in \mathcal{X}$ arbitrarily, and set $y_i[0] = \nabla f_i(x_i[0])$.
3: **while** $k = 0, 1, \ldots$ **do**
4: Determine coefficients $[R_k]_{ij}$, $[C_k]_{ji}$ for $j \in \mathcal{V}$ according to Assumption 4 and Assumption 5.
5: Send $z_i[k]$, $[C_k]_{ji} y_i[k]$ to out-neighbors.
6: Receive $z_j[k]$, $[C_k]_{ij} y_j[k]$ from in-neighbors.
7: Perform the consensus update using Equation (3a):

$$x_i[k+1] \leftarrow \sum_{j=1}^{n} [R_k]_{ij} z_j[k].$$

8: Perform the gradient tracking update using Equation (3b):

$$y_i[k+1] \leftarrow \sum_{j=1}^{n} [C_k]_{ij} y_j[k] + \nabla f_i(x_i[k+1]) - \nabla f_i(x_i[k]).$$

9: Perform the lazy optimization update using Equation (3c):

$$z_i[k+1] \leftarrow (1-\lambda) x_i[k+1] + \lambda \Pi_{\mathcal{X}} \left(x_i[k+1] - \eta y_i[k+1] \right).$$

10: **end while**

Coefficients $[R_k]_{ij}$ and $[C_k]_{ij}$ constitute the elements of mixing matrices R_k and C_k, respectively. We make the following assumptions on these matrices, which also show how agents can choose their coefficients $[R_k]_{ij}$ and $[C_k]_{ij}$:

Assumption 4 (Graph Compatibility of R_k): For all $k > 0$, the matrix R_k is row stochastic and it is compatible with the graph $\mathcal{G}_k$, i.e., $[R_k]_{ij} > 0$ if and only if $(j, i) \in \mathcal{E}_k$ or $i = j$ and $[R_k]_{ij} = 0$ otherwise. Moreover, for some $R_{\min} > 0$ we have $\min\{R_k^+\} \geq R_{\min}$ for all $k > 0$.

Assumption 5 (Graph Compatibility of C_k): For all $k > 0$, the matrix C_k is column stochastic and it is compatible with the graph $\mathcal{G}_k$, i.e., $[C_k]_{ij} > 0$ if and only if $(j, i) \in \mathcal{E}_k$ or $i = j$ and $[C_k]_{ij} = 0$. Moreover, for some $C_{\min} > 0$ we have $\min\{C_k^+\} \geq C_{\min}$ for all $k > 0$.

Under Assumption 5, the algorithm satisfies the gradient tracking property, that is $\sum_{i=1}^{n} y_i[k] = \sum_{i=1}^{n} \nabla f_i(x_i[k])$, at each time step k.

The key differences between this algorithm and the AB/Push-Pull algorithm [12] are as follows: 1) agents compute the gradients at $x_i[k]$, which is after the consensus step Equation (3a), 2) we introduce a projection operator in the calculation of $z_i[k]$, and 3) we use an additional step size λ to give agents more control over the trade-off between the consensus and optimization.

V. MAIN RESULTS

In this section, we state the main results concerning the convergence of our algorithm to the optimal point. First, will provide some core results about the behavior of graph compatible row stochastic and column stochastic matrices, and their contraction behavior. Then, we provide our main theorem showing the geometric convergence of our algorithm. The analysis accompanying these results is given in Section VI.

A. Preliminaries

We use the following lemmas to define stochastic vectors that will be used in our analysis.

Lemma 1 ([18], Lemma 5.4 and [12], Lemma 3.3): Let Assumption 3 hold and $\{R_k\}$ be a row stochastic matrix sequence satisfying Assumption 4. Then, there exists a sequence of positive stochastic vectors $\{\phi_k\}$ such that $\phi_{k+1}^{\intercal} R_k = \phi_k^{\intercal}$, where the entries of each ϕ_k are positive and have the uniform lower bound $[\phi_k]_i \geq \frac{(R_{\min})^n}{n}$ for all $i \in \mathcal{V}$ and $k \geq 0$.

Lemma 2 ([12], Lemma 3.4): Let Assumption 3 hold and $\{C_k\}$ be a matrix sequence satisfying Assumption 5. Set $\pi_0 = \frac{1}{n}\mathbf{1}$ and define the sequence $\pi_{k+1} = C_k \pi_k$. Then, each vector in $\{\pi_k\}$ is stochastic, and we have $[\pi_k]_i \geq \frac{(C_{\min})^n}{n}$. Now, we will define two lemmas about contractions of matrices R_k and C_k which allow the consensus of $x_i[k]$ and $y_i[k]$ values.

Lemma 3 ([18], Lemma 6.1): Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a strongly connected graph, and the row stochastic matrix R be compatible with the graph. Let ϕ be a stochastic vector and ϕ' be a non-negative vector such that $\phi'^{\intercal} R = \phi^{\intercal}$. Consider the vectors $z_1, z_2, \ldots, z_n \in \mathbb{R}^d$ and $x_i = \sum_{j=1}^{n} R_{ij} z_j$ for all $i \in \mathcal{V}$. Also define $\hat{z}_\phi \triangleq \sum_{i=1}^{n} \phi_i z_i$. Then, we have

$$\sum_{i=1}^{n} \phi_i' \|x_i - u\|^2 \leq \sum_{j=1}^{n} \phi_j \|z_j - u\|^2 - \frac{\min(\phi')(\min(R^+))^2}{\max^2(\phi)\mathsf{D}(\mathcal{G})\mathsf{K}(\mathcal{G})} \sum_{j=1}^{n} \phi_j \|z_j - \hat{z}_\phi\|^2,$$

where $\mathsf{D}(\mathcal{G})$ and $\mathsf{K}(\mathcal{G})$ are the diameter and the maximum edge utility of $\mathcal{G}$, respectively, as in [18, Lemma 6.1]. Define $\hat{x}_{\phi'} \triangleq \sum_{j=1}^{n} \phi_j' x_j$. Then, we get

$$\sqrt{\sum_{i=1}^{n} \phi_i' \|x_i - \hat{x}_{\phi'}\|^2} \leq \sigma \sqrt{\sum_{i=1}^{n} \phi_i \|z_i - \hat{z}_\phi\|^2},$$

where $\sigma = \sqrt{1 - \frac{\min(\phi')(\min R^+)^2}{\max^2(\phi)\mathsf{D}(\mathcal{G})\mathsf{K}(\mathcal{G})}} \in (0, 1)$.

Lemma 4 ([12], Lemma 4.5): Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a strongly-connected graph and C be a column stochastic matrix compatible with $\mathcal{G}$. Assume $y_1, y_2, \ldots, y_n \in \mathbb{R}^d$ and $v_i = \sum_{j=1}^{n} C_{ij} y_j$ for all $i \in \mathcal{V}$. Let $\pi \in \mathbb{R}^n$ be a positive stochastic vector and $\pi' = C\pi$. Then, we have

$$\sqrt{\sum_{i=1}^{n} \pi_i' \left\| \frac{v_i}{\pi_i'} - \sum_{l=1}^{n} y_l \right\|^2} \leq \tau \sqrt{\sum_{i=1}^{n} \pi_i \left\| \frac{y_i}{\pi_i} - \sum_{l=1}^{n} y_l \right\|^2},$$

where $\tau = \sqrt{1 - \frac{\min^2(\pi)(\min C^+)^2}{\max^2(\pi)\max(\pi')\mathsf{D}(\mathcal{G})\mathsf{K}(\mathcal{G})}} \in (0,1)$.

Lastly, we introduce a lemma showing the contraction properties of the projected gradient method. This lemma is an adaptation of a standard result in optimization for the projected gradient method (see [5, Lemma 10]).

Lemma 5 (Projected Gradient Contraction): Let $\mathcal{X} \subseteq \mathbb{R}^d$ be closed and convex set, and let $f : \mathbb{R}^d \to \mathbb{R}$ be μ-strongly convex and L-smooth. Define $\mathcal{T}_\eta(x) = \Pi_{\mathcal{X}}\left(x - \eta \nabla f(x)\right)$. For $0 < \eta < \frac{2}{\mu+L}$, we have

$$\|\mathcal{T}_\eta(x) - \mathcal{T}_\eta(y)\| \leq q(\eta)\,\|x - y\|$$

where $q(\eta) = 1 - \eta\mu < 1$.

B. Convergence Results

The convergence of Algorithm 1 will be determined entirely by 3 critical error terms, or distances: 1) agents' decision variables' distances to the optimal point, 2) the consensus error of the decision variables, and 3) the convergence of gradient tracking variables. We define these respective error terms mathematically as follows:

$$\|\mathbf{x}[k] - \mathbf{x}^*\|_{\phi_k} \triangleq \sqrt{\sum_{i=1}^{n} [\phi_k]_i \,\|x_i[k] - x^*\|^2}, \tag{4}$$

where $\mathbf{x}[k] = (x_1[k], \ldots, x_n[k])$, $\mathbf{x}^* = (x^*, \ldots, x^*)$, and the vectors ϕ_k satisfy Lemma 1.

$$D(\mathbf{x}[k], \phi_k) \triangleq \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{n} [\phi_k]_i [\phi_k]_j \,\|x_i[k] - x_j[k]\|^2}, \tag{5}$$

$$S(\mathbf{y}[k], \pi_k) \triangleq \sqrt{\sum_{i=1}^{n} [\pi_k]_i \left\| \frac{y_i[k]}{[\pi_k]_i} - \sum_{l=1}^{n} y_l[k] \right\|^2}, \tag{6}$$

where $\mathbf{y}[k] = (y_1[k], \ldots, y_n[k])$ and π_k satisfy Lemma 2. We call the term $\|\mathbf{x}[k] - \mathbf{x}^*\|_{\phi_k}$ the optimality gap, $D(\mathbf{x}[k], \phi_k)$ the consensus error, and $S(\mathbf{y}[k], \pi_k)$ the gradient tracking error. Now, we combine the errors in a single vector as $\mathbf{e}[k] = (\|\mathbf{x}[k] - \mathbf{x}^*\|_{\phi_k}, D(\mathbf{x}[k], \phi_k), S(\mathbf{y}[k], \pi_k))^\intercal$. We aim to show that $\lim_{k\to\infty} \mathbf{e}[k] = 0$ with a geometric rate. Hence, we want to find some matrix $M(\eta, \lambda)$ with spectral radius $\rho(M(\eta,\lambda)) < 1$ such that $\mathbf{e}[k+1] \leq M(\eta,\lambda)\mathbf{e}[k]$. This will give us the desired geometric rate. With this motivation, we now give the composite relation between the error terms at step $k+1$ and error terms at step k. First, define

$$\sigma_k \triangleq \sqrt{1 - \frac{\min(\phi_{k+1})(\min R_k^+)^2}{\max^2(\phi_k)\mathsf{D}(\mathcal{G}_k)\mathsf{K}(\mathcal{G}_k)}} \in (0,1),$$

$$\tau_k \triangleq \sqrt{1 - \frac{\min^2(\pi_k)(\min C_k^+)^2}{\max^2(\pi_k)\max(\pi_{k+1})\mathsf{D}(\mathcal{G}_k)\mathsf{K}(\mathcal{G}_k)}} \in (0,1),$$

which are the coefficents of contraction due to R_k and C_k respectively (as defined in Lemma 3 and Lemma 4), at time k. Notice that σ_k, τ_k are uniformly bounded above by constants less than 1 due to Assumption 4 and Assumption 5, and Lemma 1 and Lemma 2. Then, also define

$$r_k \triangleq \sqrt{\frac{1}{\min \pi_k}} + \sqrt{n}, \quad \varphi_k \triangleq \sqrt{\frac{1}{\min \phi_k}}.$$

Notice that since the entries of π_k and ϕ_k are bounded above and below uniformly across time, the min and max elements are also bounded uniformly over time. Therefore, we can define $r \triangleq \sup_{k\geq 0} r_k, \varphi \triangleq \sup_{k\geq 0}\varphi_k, \psi \triangleq \inf_{k\geq 0} \min \pi_k > 0, \sigma \triangleq \sup_{k\geq 0}\sigma_k < 1, \tau \triangleq \sup_{k\geq 0}\tau_k < 1$. Then, we have the following proposition describing the evolution of the errors.

Proposition 1 (Composite Relation): Let Assumptions 1–5 hold and let $\eta < \frac{1}{Ln}$. Then, we have

$$\mathbf{e}[k+1] \leq M(\eta,\lambda)\mathbf{e}[k], \tag{7}$$

where the inequality is elementwise and $M(\eta,\lambda)$ is equal to

$$\begin{bmatrix} 1 - \eta\lambda n\psi\mu & \lambda\varphi & \lambda L^{-1} \\ 2\lambda\sigma & \sigma + 2\lambda\sigma\varphi & 2\lambda\sigma L^{-1} \\ 2\lambda L r\varphi & Lr\varphi(1+\sigma) + \lambda L r\varphi^2 & \tau + \lambda r\varphi \end{bmatrix}.$$

Theorem 1 (Convergence): Let Assumptions 1–5 hold. Let $0 < \eta < \frac{1}{nL}$ and

$$0 < \lambda < \min\left\{ \frac{1-\sigma}{2\varphi\sqrt{n}}, \frac{1-\tau}{r\varphi}, \frac{\eta n\psi\mu(1-\sigma)(1-\tau)}{K} \right\},$$

where, $K = 2(1+\eta n\psi\mu)\varphi\sigma[(1-\tau) + r(1+\sigma)] + (2 + \eta n\psi\mu)r\varphi(1-\sigma)$. Then,

$$\lim_{k\to\infty} \|x_i[k] - x^*\| = 0 \;\text{ for all } i \in \mathcal{V},$$

where x^* is the solution to problem (2). Moreover, the convergence rate is geometric with rate $\rho(M(\eta,\lambda)) < 1$, where $\rho(\cdot)$ denotes the spectral radius of a matrix.

The proof of Theorem 1 is given in our extended technical report [19]. The proof shows that by choosing λ in the specified range, we can make the diagonals of M less than 1 and $\det(M(\eta,\lambda) - I) < 0$, which are sufficient to show $\rho(M(\eta,\lambda)) < 1$.

VI. ANALYSIS

In this section, at first, we provide all the necessary results for the proof of Theorem 1. Then, we provide two impossibility results providing insights into our algorithm design and the analysis. Due to space limitations, we provide some of the proofs in this section in our extended technical report [19].

A. Bounding Optimality Gap

We start the analysis of the optimality gap under our algorithm. First, notice that we have $\|\mathbf{x}[k+1] - \mathbf{x}^*\|_{\phi_{k+1}} \leq \|\mathbf{z}[k] - \mathbf{x}^*\|_{\phi_k}$ from Lemma 3 with $u = x^*$. Hence, we will focus on the analysis of $\|\mathbf{z}[k] - \mathbf{x}^*\|_{\phi_k}$. Our strategy is to split the error into two cases: the error we would have if agents had the perfect gradient knowledge and the error

coming from the gradient tracking. To represent the case where agents have the perfect gradient knowledge, we define

$$w_i[k] = (1-\lambda)x_i[k] - \lambda\Pi_{\mathcal{X}}\left(x_i - \eta n[\pi_k]_i \nabla f(x_i[k])\right),$$

for each agent i where $\nabla f(x_i[k]) \triangleq \frac{1}{n}\sum_{l=1}^{n}\nabla f_l(x_i[k])$ and stack these vectors in the matrix $\mathbf{w}[k]$. With this definition, we have $\|\mathbf{z}[k]-\mathbf{x}^*\|_{\phi_k} \leq \|\mathbf{z}[k]-\mathbf{w}[k]\|_{\phi_k} + \|\mathbf{w}[k]-\mathbf{x}^*\|_{\phi_k}$ by the triangular inequality. We establish a bound on the first term with the following lemma.

Proposition 2 (Bounding Error from Imperfect Gradients): Let Assumptions 2–5 hold. Then, we have for all $k \geq 0$,

$$\|\mathbf{z}[k]-\mathbf{w}[k]\|_{\phi_k} \leq \eta\lambda L\varphi_k\sqrt{n}D(\mathbf{x}[k],\phi_k) + \eta\lambda S(\mathbf{y}[k],\pi_k).$$

Next, we define the following terms to capture the contraction due to the lazy update rule Equation (3c):

$$q(\eta,\lambda) = 1-\lambda+\lambda q(\eta), \text{ and} \tag{8}$$

$$q_k(\eta,\lambda) = \max_i q(\eta n[\pi_k]_i, \lambda), \tag{9}$$

where $q(\eta)$ is the contraction we have in the projected gradient method as defined in Lemma 5. Now, we can derive our main result of the optimality gap:

Lemma 6 (Optimality Gap Bound): Let Assumptions 1–5 hold. Let $\eta < \frac{1}{nL}$ and $\lambda \in (0,1]$. Then, we have for all $k \geq 0$,

$$\|\mathbf{x}[k+1]-\mathbf{x}^*\|_{\phi_{k+1}} \leq q_k(\eta,\lambda)\|\mathbf{x}[k]-\mathbf{x}^*\|_{\phi_k} + \eta\lambda L\varphi_k\sqrt{n}D(\mathbf{x}[k],\phi_k) + \eta\lambda S(\mathbf{y}[k],\pi_k).$$

This lemma shows that we can control the error contributions coming from the consensus and gradient tracking errors by choosing smaller step sizes λ or η.

B. Bounding Consensus Error

Similar to the analysis of the optimality error in the previous section, we want to isolate the gradient tracking error. Let $\mathbf{u} \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$ (n copies of $\mathbb{R}^d$) and $a \in \mathbb{R}^n$ be a positive stochastic vector. Then, similar to the consensus error $D(\mathbf{x}[k],\phi_k)$, we can define

$$D(\mathbf{u},a) \triangleq \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \|u_i-u_j\|^2}. \tag{10}$$

Hence, in light of Lemma 3, notice that $D(\mathbf{x}[k+1],\phi_{k+1}) \leq \sigma_k D(\mathbf{z}[k],\phi_k)$. Then, we isolate the gradient tracking error contained in $D(\mathbf{z}[k],\phi_k)$ with the following proposition:

Proposition 3 (Isolating Gradient Tracking Error): Let Assumptions 3–4 hold. Then, we have for all $k \geq 0$,

$$D(\mathbf{z}[k],\phi_k) \leq 2\|\mathbf{z}[k]-\mathbf{w}[k]\|_{\phi_k} + D(\mathbf{w}[k],\phi_k).$$

We already have a bound on the term $\|\mathbf{z}[k]-\mathbf{w}[k]\|_{\phi_k}$ from Proposition 2. Therefore, we can complete the consensus error analysis by analyzing consensus under global gradient knowledge, which is captured by the term $D(\mathbf{w}[k],\phi_k)$.

Proposition 4: Let Assumptions 1–5 hold. Let $\eta < \frac{1}{nL}$ and $\lambda \in (0,1]$. Then, we have for all $k \geq 0$,

$$D(\mathbf{w}[k],\phi_k) \leq q_k(\eta,\lambda)D(\mathbf{x}[k],\phi_k) + 2\lambda q_k(\eta,1)\|\mathbf{x}[k]-\mathbf{x}^*\|_{\phi_k}.$$

Now, we can combine Propositions 2–4 to obtain the final bound for $D(\mathbf{x}[k+1],\phi_{k+1})$.

Lemma 7 (Consensus Error Bound): Let Assumptions 1–5 hold. Let $\eta < \frac{1}{nL}$ and $\lambda \in (0,1]$. Then, we have for $k \geq 0$,

$$D(\mathbf{x}[k+1],\phi_{k+1}) \leq 2\lambda\sigma_k q_k(\eta,1)\|\mathbf{x}[k]-\mathbf{x}^*\|_{\phi_k} + (\sigma_k q_k(\eta,\lambda) + 2\eta\lambda\sigma_k L\varphi_k\sqrt{n})D(\mathbf{x}[k],\phi_k) + 2\eta\lambda\sigma_k S(\mathbf{y}[k],\pi_k).$$

The error contribution from the optimality gap and gradient tracking error can be made small by choosing a small step size λ. Moreover, the contribution from the consensus error in previous step comes with a contraction coefficient σ_k and some additional error which can be made small with λ.

C. Bounding Gradient Tracking Error

In this section, we analyze the gradient tracking error $S(\mathbf{y}[k+1],\pi_{k+1})$. Recall that

$$y_i[k+1] = \sum_{j=1}^{n}[C_k]_{ij}y_j[k] + \nabla f_i(x_i[k+1]) - \nabla f_i(x_i[k]).$$

Here, the mixing term $\sum_{j=1}^{n}[C_k]_{ij}y_j[k]$ helps the agents agree on the direction of y-variables, while the $\nabla f_i(x_i[k+1]) - \nabla f_i(x_i[k])$ steer the y-variables towards the gradient direction. Therefore, we start by isolating the contraction in $S(\mathbf{y}[k],\pi_k)$ coming from the mixing and the error introduced by the gradient update $\nabla f_i(x_i[k+1]) - \nabla f_i(x_i[k])$:

Proposition 5: Let Assumptions 2–3 and Assumption 5 hold. Then, we have for all $k \geq 0$,

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

where $\mathbf{1}$ denotes the all ones vector.

Now, we have established that the agreement in the y-variables (i.e., $S(\mathbf{y}[k],\pi_k)$) can be distorted by $\|\mathbf{x}[k+1]-\mathbf{x}[k]\|_{\mathbf{1}}$. This is because as the x-variables change, the gradient evaluated at the previous location becomes less relevant. Hence, we now bound the error coming from this term:

Proposition 6: Let Assumptions 1–5 hold. Let $\eta < \frac{1}{nL}$ and $\lambda \in (0,1]$. Then, we have for all $k \geq 0$,

$$\|\mathbf{x}[k+1]-\mathbf{x}[k]\|_{\mathbf{1}} \leq \lambda\varphi_{k+1}(1+q_k(\eta,1))\|\mathbf{x}[k]-\mathbf{x}^*\|_{\phi_k} + \left[\frac{1}{\sqrt{2}}(\varphi_k+\sigma_k\varphi_{k+1}) + \eta\lambda L\varphi_k\varphi_{k+1}\sqrt{n}\right]D(\mathbf{x}[k],\phi_k) + \eta\lambda\varphi_{k+1}S(\mathbf{y}[k],\pi_k).$$

Finally, we combine the results in Proposition 5 and Proposition 6 to get the bound for $S(\mathbf{y}[k+1],\pi_{k+1})$.

Lemma 8 (Gradient Tracking Error Bound): Let Assumptions 1–5 hold. Let $\eta < \frac{1}{nL}$ and $\lambda \in (0,1]$. Then, we have for all $k \geq 0$,

$$\begin{aligned} &S(\mathbf{y}[k+1],\pi_{k+1}) \\ &\leq \lambda L r_k\varphi_{k+1}(1+q_k(\eta,1))\|\mathbf{x}[k]-\mathbf{x}^*\|_{\phi_k} \\ &+ Lr_k\left[(\varphi_k+\sigma_k\varphi_{k+1}) + \eta\lambda L\varphi_k\varphi_{k+1}\sqrt{n}\right]D(\mathbf{x}[k],\phi_k) \\ &+ (\tau_k + \eta\lambda L r_k\varphi_{k+1})S(\mathbf{y}[k],\pi_k). \end{aligned}$$

Similar to the consensus error, we get contraction in the gradient tracking error with coefficient τ_k. All the other errors can be made small by choosing a smaller step size λ. This completes all the necessary results needed to establish Proposition 1. Using Lemma 6, Lemma 7, and Lemma 8, and the upper bounds on r_k, φ_k, etc. (see the paragraph preceding Proposition 1), we obtain the composite relation matrix $M(\eta, \lambda)$.

D. Impossibility Results

In this section, we give some theoretical results highlighting the need for including an extra step size λ. Consider the case where we set $\lambda = 1$ in Equation (3c), thus removing the lazy update, so that

$$z_i[k+1] = \Pi_{\mathcal{X}}\left(x_i[k+1] - \eta y_i[k+1]\right).$$

We will establish that with this update rule, it is not possible to bound the term $\|\mathbf{x}[k+1] - \mathbf{x}[k]\|$ such that

$$\|\mathbf{x}[k+1] - \mathbf{x}[k]\| \leq c_1(\eta)\left\|\mathbf{x}[k] - \mathbf{x}^*\right\|_{\phi_k} + c_2(\eta)D(\mathbf{x}[k], \phi_k) + c_3(\eta)S(\mathbf{y}[k], \pi_k),$$

where $\lim_{\eta\to 0} c_1(\eta) = 0$ for every configuration of the problem. The term $\mathbf{x}[k+1] - \mathbf{x}[k]$ is essential for the analysis of the gradient tracking error since it is directly related to the term $\nabla f_i(x_i[k+1]) - \nabla f_i(x_i[k])$ by both the L-smoothness and strong convexity. The following result shows that we cannot control the error contribution coming from this term by the optimality error by simply decreasing the step size. This problem persist even when the system consists of a single agent with perfect gradient knowledge, as in centralized projected gradient method.

Lemma 9: Assume that the function $f(x) : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and its gradient $\nabla f(x)$ is L-Lipschitz continuous. Moreover, assume that the constraint set $\mathcal{X} \subseteq \mathbb{R}^d$ is convex and closed. Consider the sequence $\{x[k]\}_{k=1}^{\infty}$ generated by the centralized projected gradient method:

$$x[k+1] = \Pi_{\mathcal{X}}\left(x[k+1] - \eta\nabla f(x[k])\right), \tag{11}$$

for some $x[0] \in \mathbb{R}^d$. Suppose that there exist a coefficient $c(\eta)$ that depends on η such that for all $k \geq 0$ we have

$$\|x[k+1] - x[k]\| \leq c(\eta)\left\|x[k] - x^*\right\|, \tag{12}$$

where x^* is the unique minimizer of $f(x)$ over the set $\mathcal{X}$. Then, there exists a function f and a constraint set $\mathcal{X}$ where $c(\eta) \geq b$ where $b > 0$ for any $x[0] \in \mathcal{X} \setminus \{x^*\}$ and any $c(\eta)$ with $\eta > 0$.

Proof: We prove this result by constructing an example. Let $f(x) = \frac{L}{2}x^2$ and $\mathcal{X} = \{x \in \mathbb{R} \mid x \geq 1\}$. Then, the function f is strongly convex with L-Lipschitz continuous gradients and the set $\mathcal{X}$ is closed and convex. Notice that the update rule in this example is $x[k+1] = \Pi_{\mathcal{X}}\left((1-\eta L)x[k]\right)$ since $\nabla f(x) = Lx$. Let $c(\eta)$ satisfy Equation (12). Let the initial point $x[0] \in \mathcal{X} \setminus \{x^*\}$. We split the proof into two cases. First, assume that $\eta \geq \frac{1}{L}$. Hence, we have $x[1] = \Pi_{\mathcal{X}}\left((1-\eta L)x[0]\right) = 1 = x^*$. Then,

$$\|x[1] - x[0]\| \leq c(\eta)\left\|x[0] - x^*\right\|$$
$$\|x^* - x[0]\| \leq c(\eta)\left\|x[0] - x^*\right\|.$$

Hence, we have $1 \leq c(\eta)$. Since Equation (13) should hold for any $k \geq 0$, it must be that $c(\eta) \geq 1$ for any $c(\eta)$.

Next, we consider the case where $0 < \eta < \frac{1}{L}$. Define the set $\mathcal{C}_\eta = \{x \in \mathbb{R} \mid 1 < x \leq \frac{1}{1-\eta L}\}$. For any $x[k] \in \mathcal{C}_\eta$, we have $(1-\eta L)x[k] \leq 1$ since $x[k] \leq \frac{1}{1-\eta L}$. Therefore, $x[k+1] = x^*$. Then, using similar steps to the proof with $\eta > \frac{1}{L}$ we have $c(\eta) \geq 1$. The only remaining part is to show that for any $x[0] \in \mathcal{X}\setminus\{x^*\}$, there exists an $x[k] \in \mathcal{C}_\eta$. When $x[0] \in \mathcal{C}_\eta$, this is true trivially. Assume that $x[0] \notin \mathcal{C}_\eta$, i.e., $x[k] > \frac{1}{1-\eta L}$. First, notice that for any $x[k] \notin \mathcal{C}_\eta$, $x[k+1] > 1$. We know that $x[k]$ should converge to $x^* = 1$ since the iterates follow the projected gradient update rule with $\eta < \frac{1}{L}$ [20, Chapter 7.2]. Then, there must be an $x[k] \in \mathcal{C}_\eta$, which completes the proof. ∎

Remark 1: Let Assumptions 1–5 hold. Let there be a single agent in the system, i.e., $n = 1$. Then, if $\lambda = 1$, the Projected Push-Pull algorithm in Equation (3) is equivalent to the centralized projected gradient descent given in Lemma 9. Therefore, the impossibility results that we have shown for the projected gradient method also apply to the Projected Push-Pull algorithm.

Corollary 1: Let Assumptions 1–5 hold true. Assume that the agents follow the Projected Push-Pull algorithm given in Equation (3) with $\lambda = 1$ and $\eta > 0$. Suppose that there exist coefficients $c_1(\eta)$, $c_2(\eta)$, and $c_3(\eta)$ that depend on η such that for all $k \geq 0$ we have

$$\|x[k+1] - x[k]\| \leq c_1(\eta)\left\|\mathbf{x}[k] - \mathbf{x}^*\right\|_{\phi_k} + c_2(\eta)D(\mathbf{x}[k], \phi_k) + c_3(\eta)S(\mathbf{y}[k], \pi_k). \tag{13}$$

Then, there exist functions f_i, a constraint set $\mathcal{X}$, and initial points $x_i[0]$ where $c_1(\eta) \geq b$ where $b > 0$ for any $c_1(\eta)$.

Proof: Choose $f_i(x) = f_j(x)$ for all $i, j \in \mathcal{V}$. Consider a fully connected graph with $[R_k]_{ij} = [C_k]_{ij} = \frac{1}{n}$ for all $i, j \in \mathcal{V}$ and for all $k \geq 0$. Let each agent initialize the algorithm from the same point, i.e., $x_i[0] = x_j[0]$ for all $i, j \in \mathcal{V}$. Then, the Projected Push-Pull algorithm is equivalent to following a centralized projected gradient descent for all agents. Moreover, we have $D(\mathbf{x}[k], \phi_k) = 0$ and $S(\mathbf{y}[k], \pi_k) = 0$. Then, by Lemma 9, we know that there exist a function $f_i(x)$ and a constraint set $\mathcal{X}$ such that $c_1(\eta) \geq b$ where $b > 0$ for any $c_1(\eta)$ with $\eta > 0$. ∎

Hence, we have established that we cannot fully control the bound on the term $\|\mathbf{x}[k+1] - \mathbf{x}[k]\|$ by simply changing the step size η. This term has to arise in our analysis due to the definition of gradient tracking, which poses an important challenge to analyzing gradient tracking algorithms using projections. However, this problem does not happen when $\mathcal{X} = \mathbb{R}^d$, i.e., when the problem is unconstrained. The main challenge in the constrained case is that the gradient at x^* is typically non-zero, and therefore, we reach a pathological case where the agents do not slow down as they reach the

optimal point. In the unconstrained case, as the agents reach the optimum, gradient also slows down since it vanishes.

In a similar fashion to Corollary 1, we have a fundamental limitation in the analysis of the consensus error when $\lambda = 1$ in the algorithm. The following result shows this limitation.

Lemma 10: Let Assumptions 1–5 hold true. Assume that $n = 2$, i.e. there are two agents in the system. Let the agents follow the Projected Push-Pull algorithm given in Equation (3) with $\lambda = 1$ and $\eta > 0$. Suppose that there exist coefficients $c_1(\eta)$, $c_2(\eta)$, and $c_3(\eta)$ that depend on η such that for all $k \geq 0$ we have

$$D(\mathbf{x}[k+1], \phi_k) \leq c_1(\eta) \left\|\mathbf{x}[k] - \mathbf{x}^*\right\|_{\phi_k} + c_2(\eta) D(\mathbf{x}[k], \phi_k) + c_3(\eta) S(\mathbf{y}[k], \pi_k). \tag{14}$$

Then, there exist functions f_1, f_2, a constraint set $\mathcal{X}$, and initial points $x_1[0]$, $x_2[0]$ such that $\lim_{\eta \to 0} c_1(\eta) \neq 0$ for any $c_1(\eta)$.

Proof: Let $f(x) = \frac{L}{2}x^2 = \sum_{i=1}^{2} f_i(x)$ where $f_1(x) = \pi_1 \frac{L}{2}x^2$ and $f_2(x) = \pi_2 \frac{L}{2}x^2$. Let $\mathcal{X} = \{x \in \mathbb{R} \mid x \geq 1\}$ be the closed and convex constraint set. Without loss of generality, assume $R_k = R$ and $C_k = C$ for all k. Furthermore, assume that R is doubly stochastic and C is a column stochastic matrix with the right eigenvector π such that $C\pi = \pi$ with $\pi_1 > \pi_2$. Let $\eta \in \mathbb{R}$ with $0 < \eta < \frac{1}{L\pi_2}$ be arbitrary. Also, construct the initial conditions as $x_1[0] = x_2[0] = \frac{1}{1 - \eta L \pi_1}$. Then, clearly, $D(\mathbf{x}[0], \phi) = S(\mathbf{y}[0], \pi) = 0$. So, it must be that

$$D(\mathbf{x}[1], \phi) \leq c_1(\eta) \left\|\mathbf{x}[0] - \mathbf{x}^*\right\|_{\phi}.$$

By construction, we have that $\left\|\mathbf{x}[0] - \mathbf{x}^*\right\|_{\phi} = \frac{\eta L \pi_1}{1 - \eta L}$ and $D(\mathbf{x}[1], \phi) = \frac{1}{\sqrt{2}} \frac{\eta L}{1 - \eta L}(\pi_1 - \pi_2)$. Then, we must have

$$0 < \frac{1}{\sqrt{2}} \eta L (\pi_1 - \pi_2) \leq c_1(\eta) \eta L \pi_1$$
$$0 < \frac{1}{\sqrt{2}} (1 - \frac{\pi_2}{\pi_1}) \leq c_1(\eta),$$

which means that $\lim_{\eta \to 0} c_1(\eta) \neq 0$ as the relation above holds for any η in the range $0 < \eta < \frac{1}{L\pi_2}$. ∎

Essentially, Corollary 1 and Lemma 10 show that no matter how we bound the error terms, we cannot gain full control over the non-diagonal entries of the composite relation matrix $M(\eta, \lambda)$, with $\lambda = 1$. These entries are essential in controlling the spectral radius of $M(\eta, 1)$, and guaranteeing convergence. However, being able to choose a λ in range $(0, 1]$ gives us more flexibility.

VII. NUMERICAL STUDIES

In this section, we empirically demonstrate the convergence of our algorithm on a sample optimization problem and also investigate the effect of the graph properties and mixing times of matrices R_k and C_k on convergence.

A. Convergence of Protocol on Time Varying Communication Networks

Optimization Problem: We have $n = 50$ agents. Agent i has objective function $(x_i - x_i^c)^\intercal P_i (x_i - x_i^c)$ where $x_i^c \in \mathbb{R}^d$ is the center of the quadratic and the matrix $P_i \in \mathbb{R}^{d \times d}$ is positive definite with $\mu I \preceq P_i \preceq LI$ for some μ and L. Hence, the global objective is $\sum_{i=1}^{n} (x_i - x_i^c)^\intercal P_i (x_i - x_i^c)$.

For this experiment, we set $d = 2$ and sample the coordinates of x_i^c from the uniform distribution $\mathcal{U}[-2, 8]$ independently for each i. Similarly, we set P_i to be diagonal for each i, and sample each entry in the diagonal independently from $\mathcal{U}[0, 1]$. Notice that the objectives are strongly convex and smooth. We set $\mathcal{X} = \overline{B((6,6), 2)}$, the closed ball around $(6, 6)$ with radius 2. This setup is likely to result in a optimal point x^* in the boundary of $\mathcal{X}$, which helps us demonstrate the effectiveness of our algorithm when $\nabla f(x^*) \neq 0$, in contrast with the unconstrained setting.

Communication graphs: We construct the time-varying graphs by iterating over the sequence of graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_T\}$ where $T = 5$. That is, $\mathcal{G}_k = \mathcal{G}_{(k-1 \mod T)+1}$. We generate $\mathcal{G}_i$ independently for each $i \in [T]$ as follows. For all $j, l \in \mathcal{V}$ we add the edge (j, l) to $\mathcal{E}_i$ with probability $p = 0.1$. We regenerate the graph if it is not strongly connected.

Mixing matrices R_k and C_k: Let $\mathcal{N}_i^{\mathsf{in}}[k]$ and $\mathcal{N}_i^{\mathsf{out}}[k]$ denote the in/out neighborhoods of agent i at time k respectively. That is, $j \in \mathcal{N}_i^{\mathsf{in}}[k]$ if and only if $(j, i) \in \mathcal{E}_k$ and $j \in \mathcal{N}_i^{\mathsf{out}}[k]$ if and only if $(i, j) \in \mathcal{E}_k$. Agent i sets the i'th row of R_k and the i'th column of C_k as follows:

$$[R_k]_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_i^{\mathsf{in}}[k]|+1} & \text{if } j \in \mathcal{N}_i^{\mathsf{in}}[k] \text{ or } j = i \\ 0 & \text{otherwise} \end{cases},$$

$$[C_k]_{ji} \begin{cases} \frac{1}{|\mathcal{N}_i^{\mathsf{out}}[k]|+1} & \text{if } j \in \mathcal{N}_i^{\mathsf{out}}[k] \text{ or } j = i \\ 0 & \text{otherwise} \end{cases}.$$

Optimization parameters: We set $\eta = 0.5, \lambda = 0.7$. We initialize $x_i[0]$ by sampling each coordinate from $\mathcal{U}[0, 10]$ and projecting onto $\mathcal{X}$.

We plot the error terms[1] in Figure 1. As we can see in Figure 1, all of the error goes to 0 with geometric (linear) rate. The convergence rates (slopes) of all the terms are similar, which highlights the interdependence of these errors.

B. Effect of Graph Type on Convergence

Because the contraction of matrices R_k and C_k are related to the communication graph as given in Lemma 3 and Lemma 4, graph type will affect the convergence rate of our algorithm. Therefore, we investigate the effect of graph type on the convergence rate. In this section, we set $n = 15$ and $T = 1$ (static graphs) but otherwise use the same setup for the objective function and constraint set as the previous section. We generate three different graph types as follows: 1) *Random*: Same as described in the previous section. 2) *Cyclic*: We have $(i, i+1) \in \mathcal{E}$ for $i = 1, \ldots, n-1$ and $(n, 1) \in \mathcal{E}$. 3) *Unbalanced*: Graph where certain nodes have very high in-degrees and low out-degrees and vice versa.

We set R_k and C_k to be compatible for each graph as described in the previous section. We fix $\eta = 0.5$ for all

[1]One minor difference between these error terms and the ones used in the analysis is that we cannot compute the sequence $\{\phi_k\}$ as used in the analysis. Therefore, we choose $\phi_k = \frac{1}{n}\mathbf{1}$ for all k. We initialize $\pi_0 = \frac{1}{n}\mathbf{1}$ and let $\pi_{k+1} = C_k \pi_k$.

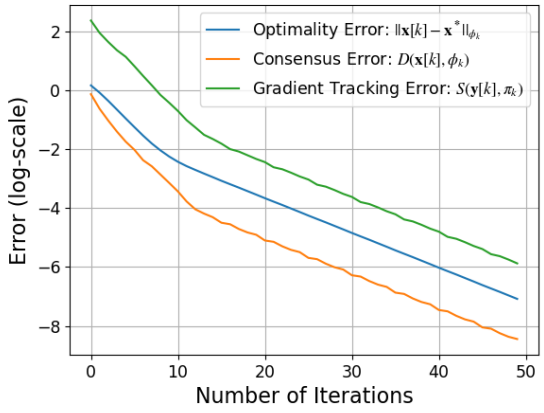

Fig. 1. Optimality, consensus, and gradient tracking errors vs. number of iterations on a log-scale. The errors converge to 0 as geometric (linear) rate with similar convergence rates.

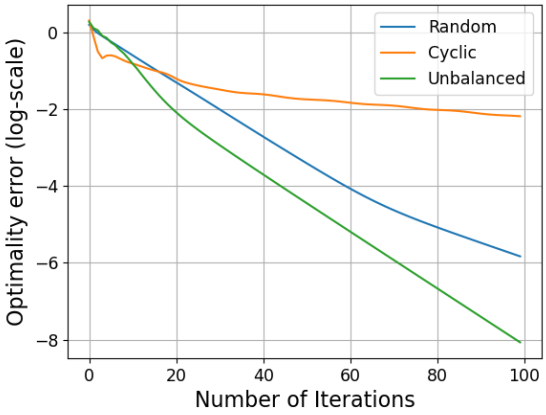

Fig. 2. Optimality error $\|\mathbf{x}[k]-\mathbf{x}^*\|_{\phi_k}$ vs. number of iterations k on a log-scale for different graph types. Even though the error converges to 0 geometrically for all graphs, unbalanced and random graphs have a much higher convergence rate compared to cyclic graphs, which are slowly mixing.

graphs, and for each graph, we choose λ to be the largest value that allows convergence. Specifically, the random graph requires $\lambda = 0.15$, the unbalanced graph requires $\lambda = 0.3$, and the cyclic graph requires $\lambda = 0.6$ to have convergence. The comparison of the convergence rate between these graphs is shown in Figure 2. We see that the random and unbalanced graphs have a faster convergence due to having higher-connectivity.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce the Projected Push-Pull algorithm, which combines gradient tracking and projected gradient method to address distributed constrained optimization problems on time-varying directed graphs. We prove that our algorithm achieves geometric convergence with sufficiently small step sizes. We derive explicit bounds for the step sizes based on the characteristics of the cost functions and the communication graph. Moreover, we provide additional theoretical results showing that having a non-zero gradient at the optimal point in constrained problems poses additional challenges in the analysis of gradient tracking methods employing projection. Finally, we demonstrate the geometric convergence of our algorithm via numerical studies over various graph types. An interesting direction for future work is the incorporation of random and adversarial noise.

REFERENCES

[1] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018.

[2] G. Carnevale, A. Camisa, and G. Notarstefano, "Distributed online aggregative optimization for dynamic multirobot coordination," *IEEE Transactions on Automatic Control*, vol. 68, no. 6, pp. 3736–3743, 2023.

[3] S. S. Ram, V. V. Veeravalli, and A. Nedić, "Distributed and recursive parameter estimation in parametrized linear state-space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 488–492, 2010.

[4] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 159–166.

[5] ——, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

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

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

[8] 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 Transactions on Automatic Control*, vol. 63, no. 10, pp. 3558–3565, 2018.

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

[10] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push–pull gradient methods for distributed optimization in networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1–16, 2021.

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

[12] 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.

[13] H. Liu, W. Yu, and G. Chen, "Discrete-time algorithms for distributed constrained convex optimization with linear convergence rates," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4874–4885, 2020.

[14] M. Luan, G. Wen, H. Liu, T. Huang, G. Chen, and W. Yu, "Distributed discrete-time convex optimization with closed convex set constraints: Linearly convergent algorithm design," *IEEE Transactions on Cybernetics*, pp. 1–13, 2023.

[15] X. Meng, Q. Liu, and J. Xiong, "An accelerated gradient tracking algorithm with projection error for distributed optimization," in *2023 15th International Conference on Advanced Computational Intelligence (ICACI)*, 2023, pp. 1–6.

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

[17] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM Journal on Optimization*, vol. 32, no. 2, pp. 354–385, 2022.

[18] 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.

[19] O. E. Akgün, A. K. Dayı, S. Gil, and A. Nedić, "Projected push-pull for distributed constrained optimization over time-varying directed graphs (extended version)," *arXiv preprint arXiv:2310.06223*, 2023.

[20] B. T. Polyak, *Introduction to optimization*. New York, Optimization Software,, 1987.