A Fast Row-Stochastic Decentralized Optimization Method Over Directed Graphs

Diyako Ghaderyan, Necdet Serhat Aybat, A. Pedro Aguiar, and Fernando Lobo Pereira

Abstract

In this paper, we introduce a fast row-stochastic decentralized algorithm, referred to as FRSD, to solve consensus optimization problems over directed communication graphs. The proposed algorithm only utilizes row-stochastic weights, leading to certain practical advantages over those requiring column-stochastic weights. Thus, in contrast to the majority of existing methods, FRSD does not employ a gradient tracking technique, rather it uses a novel momentum term. Under the assumption that each node-specific function is smooth and strongly convex, we show that FRSD admits constant step-size and momentum parameters such that the iterate sequence converges linearly to the optimal consensus solution. In the numerical tests, we compare FRSD with other state-of-the-art methods, which use row-stochastic and/or column-stochastic weights.

Index Terms

Distributed optimization, consensus, directed graphs, linear convergence, row-stochastic weights.

I. INTRODUCTION

In recent years, rapid advances in artificial intelligence and communication technologies have led to large-scale network systems over which one has to solve optimization problems

This work was supported by PDMA-NORTE-08-5369-FSE-000061, UIDB/00147/2020 SYSTEC through the FCT/MCTES (PIDDAC);

Diyako Ghaderyan, A. Pedro Aguiar, and Fernando Lobo Pereira are with the Research Center for Systems and Technologies(SYSTEC) and the Faculty of Engineering of the University of Porto(FEUP), Portugal, (emails: dghaderyan@fe.up.pt, pedro.aguiar@fe.up.pt, flp@fe.up.pt).

Necdet Serhat Aybat is with the Industrial and Manufacturing Engineering Department, The Pennsylvania State University, University Park, PA 16802 USA (email: nsa10@psu.edu).

The authors are listed according to their contribution to the work, from the most to the least.
with enormous, physically distributed and/or private data sets in order to achieve system level objectives such that every agent (represented by a node in the network) has to agree on these decisions. To reach an optimal consensus decision, one resorts to decentralized optimization techniques to solve the consensus optimization problems in a distributed manner employing only local computations and communication among neighboring computing nodes that can directly communicate with each other. The classic consensus optimization problem has the following form:

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

where the objective function $\bar{f}$ is the average of all individual cost functions $\{f_i\}_{i=1}^{n}$, where $f_i: \mathbb{R}^p \rightarrow \mathbb{R}$ is the private function of agent $i$. This problem appears in a variety of applications, e.g., sensor networks [1], [2], distributed control [3], large-scale machine learning [4], [5], [6], [7], distributed estimation [8].

Below we first discuss the previous work focusing on undirected networks, and then we briefly go over the methods proposed for distributed consensus optimization over directed networks.

Inspired by the seminal work [9], the authors in [10] proposed a distributed (sub)gradient descent method for solving (1). When each $f_i$ is closed convex, the method in [10] is shown to have a sublinear convergence rate, i.e., to compute an $\epsilon$-optimal solution, one needs to evaluate $O(1/\epsilon^2)$ subgradients – the slower rate of subgradient methods is due to their employing of a diminishing step-size sequence or of a small fixed step size $\alpha = O(\epsilon)$. Moreover, when agents have simple constraint sets, e.g., $X$ is closed convex set, distributed projected subgradient methods are proposed for solving

$$\min \{ \bar{f}(x) : x \in \mathcal{X} \};$$

for instance, the method in [11] solves (2) employing exact subgradient evaluations, while the method in [12] can handle subgradients corrupted by stochastic noise. In [13], algorithms based on dual averaging of subgradients are studied for solving (2) assuming $0 \in \mathcal{X}$. In [12], it is shown that an $\epsilon$-optimal solution can be computed with $O(n^3/\epsilon^2)$ iteration complexity that is independent of the network topology, whereas the algorithm proposed in [13] requires iteration complexity of $O(n^2/\epsilon^2)$ for paths or simple cycle graphs, $O(n/\epsilon^2)$ for 2-d grids, and $O(1/\epsilon^2)$ for bounded degree expander graphs.

Authors in [14] propose a primal-dual subgradient algorithm to solve problems with a global constraint set defined as the intersection of local constraint sets, i.e., $\mathcal{X} = \mathcal{X}_0 \cap (\bigcap_{i=1}^{n}) \mathcal{X}_i$ in (2)
such that each agent-\textit{i} only knows \(f_i, x_i\) and \(x_0\). In [15], under bounded and Lipschitz gradients assumption, an improved convergence rate of \(O(\log(k)/k^2)\) is obtained by employing Nesterov acceleration. For smooth convex objective functions, the method EXTRA [16], utilizing the difference of two consecutive gradients in its updates and a fixed step size, generates a sequence that converges with a \(O(1/k)\) rate; the rate can be improved to a linear rate under the additional assumption of strong convexity. There are also distributed methods based on alternating direction method of multipliers (ADMM) achieving similar rates, e.g., [17], [18], [19], [20], [21].

The methods we discussed above are designed for undirected networks; hence, they correspond to balanced graphs if we treat undirected networks as a special case of directed networks. However, the directed networks in general may well be unbalanced; this situation arises especially for directed time-varying networks. For general directed networks, the subgradient-push method proposed in [22] combines the push-sum protocol [23] (for computing an average over directed networks) with the classic subgradient method [10] (for minimization of convex functions). More precisely, the method applies to (1) when each \(f_i\) is a closed convex function and the directed communication network is time-varying; a sublinear rate of \(O(\log(k)/\sqrt{k})\) can be achieved using a column-stochastic weight matrix and a diminishing step-size sequence. DEXTRA proposed in [24] is a distributed method for directed graphs with R-linear convergence rate; it combines EXTRA [16] with push-sum approach [23]. The step-size in DEXTRA is constant and should be carefully chosen belonging to a specific interval that may be unknown to the agents. Compared to DEXTRA, Push-DIGing [25] and ADD-OPT [26] have a simpler step-size rule and can achieve R-linear rate on directed graphs with time-varying and static topology, respectively, using sufficiently small constant step-size. These approaches employ column-stochastic weight to achieve R-linear rate over strongly connected networks. Unlike the previous methods that are based on push-sum, there are also others achieving a linear convergence rate through employing both column-stochastic and row-stochastic weights, e.g., AB, ABM and Push-Pull [27], [28], [29].

It is important to note that designing column-stochastic weights requires the knowledge of neighbors’ out-degree for each node; this requirement is impractical within broadcast-based communication systems. To address this issue, in [30], the authors proposed a method that only uses the row-stochastic weights. In follow-up works, [31] extended the method in [30] to handle uncoordinated step-sizes, and [32] improved the rate in [30] employing gradient tracking and nonuniform step-sizes.
Notation: In this paper, we consider the bold letter to denote vectors, $\mathbf{x} \in \mathbb{R}^p$, and $[\mathbf{x}]_j$ denotes the $j$-th element of $\mathbf{x}$. The vector $0_n$ and $1_n$ represent the $n$-dimensional vectors of all zeros and ones. The uppercase of letters are reserved for matrices; given $X \in \mathbb{R}^{n \times n}$, $\text{diag}(X) \in \mathbb{R}^{n \times n}$ denotes the diagonal matrix of which diagonal is equal to that of $X \in \mathbb{R}^{n \times n}$. Moreover, given $\mathbf{v} \in \mathbb{R}^n$, $\text{diag} (\mathbf{v})$ is a diagonal matrix with its diagonal equal to $\mathbf{v}$. $I_n = [e_i]_i^n$ denotes the $n \times n$ identity matrix, where $e_i$ denotes the $i$-th unit vector. Throughout $\| \cdot \|$ denotes the Euclidean and the spectral norms depending on whether the argument is a vector or a matrix.

Definition 1. Define $\mathbf{x} \triangleq [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times p}$ and $\mathbf{y} \triangleq [y_1, \ldots, y_n]^\top \in \mathbb{R}^{n \times p}$, where $x_i, y_i \in \mathbb{R}^p$ are the local variables of agent-$i$ for $i \in V \triangleq \{1, \ldots, n\}$, and in an algorithmic framework, their values at iteration $k$ are denoted by $x_i(k)$ and $y_i(k)$ for $i \in V$ and $k \geq 0$. Let $f : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}$ a function of local variables $\{x_i\}_{i \in V}$ such that $f(\mathbf{x}) \triangleq \sum_{i \in V} f_i(x_i)$ for $\mathbf{x} \in \mathbb{R}^{n \times p}$ and $\nabla f(\mathbf{x}) \triangleq [\nabla f_1(x_1), \ldots, \nabla f_n(x_n)]^\top \in \mathbb{R}^{n \times p}$, where $\nabla f_i(x_i) \in \mathbb{R}^p$ denotes the gradient of $f_i$ at $x_i \in \mathbb{R}^p$.

Contributions: In this paper, we design a fast row-stochastic decentralized method, referred to as FRSD, for distributed consensus optimization over directed communication networks. FRSD employs only row-stochastic weights, and we show that when $\{f_i\}_{i=1}^n$ are strongly convex and smooth, FRSD iterate sequence corresponding to a constant stepsize converges to the optimal consensus decision with a linear rate. While previous methods [30], [31], [32] crucially depend on the gradient tracking technique to establish linear rate, in this paper we achieve the same result through introducing a novel momentum term. In the numerical tests, we also empirically show that FRSD achieves a better convergence rate compared to other state-of-the-art methods: Xi-row, AB, Push-DIGing and Push-Pull.

II. Problem Formulation and Algorithm

The goal is to solve the consensus optimization problem in (1) over a communication network which is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is a set of nodes (agents), and $\mathcal{E}$ is a set of directed communication links between the nodes. Each node $i \in \mathcal{V}$ has a private cost function $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$, only known to node $i$. Furthermore, for each node $i \in \mathcal{V}$, we define its out-neighbors as the set of nodes receiving information from node $i$, i.e., $\mathcal{N}_i^{\text{out}} = \{j | (i, j) \in \mathcal{E}\} \cup \{i\}$, and in-neighbors as the set of nodes that can send information to node $i$, i.e., $\mathcal{N}_i^{\text{in}} = \{j | (j, i) \in \mathcal{E}\} \cup \{i\}$.

Throughout the paper we make the following assumptions.
Assumption 1. $G$ is directed and strongly connected.

Assumption 2. For all $i \in V$, the local function $f_i$ is $L$-smooth, i.e., it is differentiable with a Lipschitz gradient:
\[ \| \nabla f_i(x) - \nabla f_i(x') \| \leq L \| x - x' \|, \quad \forall x, x' \in \mathbb{R}^p. \] (3)

Assumption 3. For all $i \in V$, $f_i$ is $\mu$-strongly convex, i.e.,
\[ f_i(x') \geq f_i(x) + \nabla f_i(x)^\top (x' - x) + \frac{\mu}{2} \| x' - x \|^2 \] (4)
for all $x, x' \in \mathbb{R}^p$.

Remark 1. Under Assumption 3, $x^*$ is the unique optimal solution to (1).

We next propose our decentralized optimization algorithm FRSD to solve the consensus optimization problem in (1).

Algorithm 1 FRSD

Input: $x_i(0) \in \mathbb{R}^p$ for $i \in V$, $\alpha, \beta > 0$ such that $0 < \alpha \beta < 1$, row-stochastic $R = [r_{ij}] \in \mathbb{R}^{n \times n}$ as in (8).

1: $y_i(0) \leftarrow 0$, $v_i(0) \leftarrow e_i \in \mathbb{R}^n$ for $i \in V$

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

3: for all $i \in V$ do

4: \[ x_i(k+1) \leftarrow \sum_{j \in V} r_{ij} x_j(k) - \alpha \left( \frac{\nabla f_i(x_i(k))}{[v_i(k)]_i} + y_i(k) \right) \] (5)
\[ y_i(k+1) \leftarrow y_i(k) + \beta \left( x_i(k+1) - \sum_{j \in V} r_{ij} x_j(k+1) \right) \] (6)
\[ v_i(k+1) \leftarrow \sum_{j \in V} r_{ij} v_j(k) \] (7)

5: end for

6: end for

A. FRSD Algorithm

We now describe in detail the distributed algorithm FRSD to solve (1). At each iteration $k \geq 0$, each agent $i \in V$ updates three variables $x_i(k), y_i(k) \in \mathbb{R}^p$ and $v_i(k) \in \mathbb{R}^n$ as described
in the Algorithm 1 where $\alpha, \beta > 0$ and $R = \{r_{ij}\}$ are the parameters of the algorithm: $\alpha$ is the constant step-size and $\beta$ is a momentum parameter such that $\alpha \beta < 1$, and $R = [r_{ij}] \in \mathbb{R}^{n \times n}$ is a row-stochastic matrix such that

$$ r_{ij} = \begin{cases} > 0, & j \in \mathcal{N}_i^{in}, \\ 0, & \text{otherwise}; \sum_{j \in V} r_{ij} = 1, \quad \forall i \in V. \end{cases} $$

(8)

**Remark 2.** Since $\mathcal{G}$ is strongly connected and has finitely many nodes, the Markov chain corresponding to the transition probability matrix $R$ is irreducible and positive recurrent; moreover, since $R$ has a positive diagonal, it is also aperiodic; therefore, there exists a stationary distribution $\pi \in \mathbb{R}^n$, i.e., $\pi \geq 0$ and $1^\top \pi = 1$ such that $\pi^\top R = \pi^\top$.

**Definition 2.** Each node $i \in \mathcal{N}$, initialized from $v_i(0) = 0$ generates $\{v_i(k)\}_{k \geq 0}$ as in (7) of FRSD Algorithm. Let $V(k) \triangleq [v_1(k), ..., v_n(k)] \in \mathbb{R}^{n \times n}$, and $\tilde{V}(k) \triangleq \text{diag}(V(k))$.

Given arbitrary $x(0) \in \mathbb{R}^{n \times p}$, we initialize $y(0) \in \mathbb{R}^{n \times p}$ such that $y_i(0) = 0_n$ for $i \in V$ and $V(0) = I_n$. We present FRSD stated in (5)-(7) in a compact form as follows:

$$ x(k+1) = Rx(k) - \alpha \left( y(k) + \tilde{V}^{-1}(k) \nabla f(x(k)) \right), $$

(9a)

$$ y(k+1) = y(k) + \beta \left( I_n - R \right) x(k+1), $$

(9b)

$$ V(k+1) = RV(k). $$

(9c)

**B. Related Methods**

Next, we discuss some existing related distributed optimization methods for a directed graph $\mathcal{G}$ satisfying Assumption 1.

1) **Push-DIGing:** Push-DIGing algorithm, proposed in [25], achieves a linear convergence rate for solving (1) over directed graphs (possibly time-varying) with a constant step-size under Assumptions 1-3. Given $\mathcal{G}$, Push-DIGing updates four variables $\mathbf{x}_i(k), \mathbf{y}_i(k), \mathbf{z}_i(k) \in \mathbb{R}^p$ and
for each agent $i \in V$ as follows:

$$v_i(k + 1) = \sum_{j \in V} b_{ij} v_j(k),$$

$$x_i(k + 1) = \sum_{j \in V} b_{ij} (x_j(k) - \alpha y_j(k)), $$

$$z_i(k + 1) = x_i(k + 1)/v_i(k + 1),$$

$$y_i(k + 1) = \sum_{j \in V} b_{ij} y_j(k) + \nabla f_i(z_i(k + 1)) - \nabla f_i(z_i(k)),$$

where $B = [b_{ij}] \in \mathbb{R}^{n \times n}$ is a column-stochastic weights compatible with $G$ and $\alpha > 0$. The Push-DIGing algorithm is initialized with $v_i(0) = 1$, $y_i(0) = \nabla f_i(z_i(0))$ and from an arbitrary $x_i(0)$ for each $i \in V$. Since directed graphs are not balanced in general, Push-DIGing adopts a push-sum strategy along with utilizing a column-stochastic weights, which requires each agent to know its out-degree –this may not be practical within broadcast-based communication systems. Applying row-stochastic weights are easier than column-stochastic weights in such a distributed environment as each agent only manages the weights on information pertaining its in-neighbors.

2) AB/Push-Pull: In contrast to Push-DIGing, AB approach [27] could get away with the nonlinear update due to eigenvector estimation. The AB method uses both row-stochastic and column-stochastic weights simultaneously to stay feasible in directed graphs. At each iteration $k \geq 0$, AB updates two variables $x_i(k), y_i(k) \in \mathbb{R}^p$ for each agent $i \in V$:

$$x_i(k + 1) = \sum_{j \in V} r_{ij} x_j(k) - \alpha y_i(k),$$

$$y_i(k + 1) = \sum_{j \in V} b_{ij} (y_j(k) + \nabla f_j(x_j(k + 1)) - \nabla f_j(x_j(k))),$$

where $\alpha > 0$ is the step-size, $R = [r_{ij}] \in \mathbb{R}^{n \times n}$ and $B = [b_{ij}] \in \mathbb{R}^{n \times n}$ denote the row-stochastic and column-stochastic weights, respectively, compatible with $G$. The AB iterate sequence, initialized with an arbitrary $x_i(0)$ and $y_i(0) = \nabla f_i(x_i(0))$ for each $i \in V$, converges linearly to the optimal solution under Assumptions [1][5]. There is a variant of the AB algorithm, ABm [28] that combines the gradient tracking with a momentum term and can deal with nonuniform step-sizes.

Push-Pull, proposed in [29], is related to AB, it is only different in its $x_i(k + 1)$ update:

$$x_i(k + 1) = \sum_{j \in V} r_{ij} (x_j(k) - \alpha y_i(k)), $$
while \( y_i(k+1) \) update is the same with AB. AB approach is based on the Combine-And-Adapt based scheme; on the other hand, Push-Pull method can be considered as an Adapt-Then-Combine based approach—for more details see [33].

3) Xi-row: The method proposed in [30], which we call it as Xi-row in this paper, can solve (1) over directed networks with a linear convergence rate using an uniform fixed step-size. Similar to our FRSD method, it only employs row-stochastic weights. Each agent \( i \in V \) updates three variables \( x_i(k), y_i(k), v_i(k) \in \mathbb{R}^p \) as follows:

\[
x_i(k+1) = \sum_{j \in V} r_{ij} x_j(k) - \alpha y_i(k),
\]
\[
v_i(k+1) = \sum_{j \in V} r_{ij} v_j(k),
\]
\[
y_i(k+1) = \sum_{j \in V} r_{ij} y_i(k) + \frac{\nabla f_i(x_i(k+1))}{[v_i(k+1)]^i} - \frac{\nabla f_i(x_i(k))}{[v_i(k)]^i},
\]

where \( R = [r_{ij}] \in \mathbb{R}^{n \times n} \) is the row-stochastic weights compatible with \( G \) and \( \alpha > 0 \) is the step-size. The Xi-row iterates are initialized with \( v_i(0) = e_i, y_i(0) = \nabla f_i(x_i(0)) \) from an arbitrary \( x_i(0) \) for each \( i \in V \). There is a variant of the Xi-row method, Frost [31] that extends to nonuniform step-sizes.

All the methods we have reviewed that use a uniform step-size also employ column-stochastic weights, except for FRSD and Xi-row which only use row-stochastic weights. Therefore, FRSD and Xi-row are the method of choice for the broadcast-based distributed computational setting. On the other hand, comparing FRSD and Xi-row, FRSD has additional momentum parameter \( \beta > 0 \); thus, it is natural to expect that it can be tuned to converge faster than Xi-row—indeed, we observed this expected behavior empirically in our numerical experiments—see Section III.

Next, we write \( x(k+2) \) in a recursive manner for AB, Xi-row and FRSD to understand the similarity among them.

\textbf{AB:} For \( k \geq 0 \),
\[
x(k+2) = (R + B)x(k+1) - BRx(k) - \alpha B(\nabla f(x(k+1)) - \nabla f(x(k))).
\]

\textbf{Xi-row:} For \( k \geq 0 \),
\[
x(k+2) = 2Rx(k+1) - R^2x(k) - \alpha (\hat{V}^{-1}(k+1)\nabla f(x(k+1)) - \hat{V}^{-1}(k)\nabla f(x(k))),
\]

where as in Definition 2, \( \hat{V}(k) \triangleq \text{diag}(V(k)) \) and \( V(k) \triangleq [v_1(k), ..., v_n(k)]^T \in \mathbb{R}^{n \times n} \) for \( k \geq 0 \).

\textbf{FRSD:} For \( k \geq 0 \),
\[
x(k+2) = ((1 + \alpha \beta)R + (1 - \alpha \beta)I_n)x(k+1) - Rx(k) - \alpha (\hat{V}^{-1}(k+1)\nabla f(x(k+1)) - \hat{V}^{-1}(k)\nabla f(x(k))).
\]
Note that if we set $\beta = 0$, FRSD updates reduces to
\[
x(k + 2) = (R + I_n)x(k + 1) - Rx(k) - \alpha(\hat{V}^{-1}(k + 1)\nabla f(x(k + 1)) - \hat{V}^{-1}(k)\nabla f(x(k))).
\]

Note that for $R = B = I_n$, all of them generate the same sequence. That said, for arbitrary $R$ and $B$ compatible with a non-trivial directed graph $G$, AB, Xi-row and FRSD are all different. Compared to AB and Xi-row, FRSD is more flexible as it has an additional momentum parameter $\beta > 0$ in addition to the constant step size $\alpha > 0$ like the others.

C. Main Results

In this section, we will show that the iterate sequence generated by the algorithm FRSD as stated in (9) converges to the optimal solution $x^*$ linearly. Without loss of generality, we consider $p = 1$; hence, the local iterates $x_i(k), y_i(k) \in \mathbb{R}$.

Remark 3. Since we assume $p = 1$, $x = [x_i]_{i=1}^n \in \mathbb{R}^n$ and $f$ and $\nabla f$ defined in Definition 7 become $f : \mathbb{R}^n \to \mathbb{R}$ and $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ such that $f(x) \triangleq \sum_{i=1}^n f_i(x_i)$ and $\nabla f(x) \triangleq [\nabla f_i(x_i)]_{i=1}^n \in \mathbb{R}^n$.

Remark 4. Assumptions 2 and 3 imply that $f$ is $L$-smooth, i.e., $\|\nabla f(x) - \nabla f(x')\| \leq L \|x - x'\|$, and $\mu$-strongly convex.

Remark 5. Since $R$ is row-stochastic, spectral radius of $R$ is 1, $\rho(R) = 1$; thus, $\lim_{k \to \infty} R^k$ exists. In particular, since $R$ corresponds to an ergodic Markov chain, we get $\lim_{k \to \infty} R^k = 1_n \pi^\top$ – see Remark 2.

Definition 3. Define $V_\infty \triangleq \lim_{k \to \infty} V(k)$ and $\hat{V}_\infty = \lim_{k \to \infty} \hat{V}(k)$. Since $V(0) = I_n$, $V_\infty = \lim_{k \to \infty} R^k = 1_n \pi^\top$ and $\hat{V}_\infty = \text{diag}(\pi)$. Thus, $v \triangleq \sup_{k \geq 0} \|V(k)\| \in \mathbb{R}$ and $\tilde{v} \triangleq \sup_{k \geq 0} \|\hat{V}^{-1}(k)\| \in \mathbb{R}$ are well-defined.

Next, we define some auxiliary sequences that will be used within the analysis. For $k \geq 0$, let $\hat{x}(k) = V_\infty x(k) = 1_n \pi^\top x(k) = \bar{x}(k)1_n$, where $\bar{x}(k) = \pi^\top x(k) \in \mathbb{R}$. Let $x^* = x^*1_n$, where $x^* \in \mathbb{R}$ is the unique optimal solution to (1). Thus, Remark 3 implies that for any $x \in \mathbb{R}^n$ such that $x = x1_n$ for some $x \in \mathbb{R}$, we have $\nabla f(x) = [\nabla f_1(x), \ldots, \nabla f_n(x)]^\top \in \mathbb{R}^n$; hence, $\nabla f(\hat{x}(k)) = [\nabla f_1(\bar{x}(k)), \ldots, \nabla f_n(\bar{x}(k))]^\top \in \mathbb{R}^n$ and $\nabla f(x^*) = [\nabla f_1(x^*), \ldots, \nabla f_n(x^*)]^\top \in \mathbb{R}^n$.

Remark 6. From the optimality condition for (1), $1_n^\top \nabla f(x^*) = 0$. 

DRAFT
The structure of our proof was inspired by [30] and [34]. In particular, we construct a linear system of inequalities and use the deterministic version of the celebrated supermartingale convergence theorem [35] to prove the convergence results. We were able to show that FRSD iterates converge to the optimal consensus solution with a linear rate as in [29], [27], [32].

In the rest of this section, we establish the linear convergence; but, first, we state some preliminary results which will be used later.

**Definition 4.** Given $\alpha, \beta > 0$ such that $\alpha \beta \in (0, 1)$, let $C \triangleq (1 - \alpha \beta)I_n + \alpha \beta R$, where $R = [r_{ij}] \in \mathbb{R}^{n \times n}$ is the row-stochastic matrix as given in (8).

Note $C$ corresponds to the lazy version of the Markov chain corresponding to $R$; thus, it has the same stationary distribution, i.e., $\lim_{k \to \infty} C^k = \lim_{k \to \infty} R^k = 1_n \pi^\top$. Next, we state two technical results that will help us derive our main result.

**Lemma 1.** Given $R$ and $C$ as defined above, there exist vector norms $\|\cdot\|_R, \|\cdot\|_C$ such that $\|\cdot\| \leq \|\cdot\|_R$ and $\|\cdot\| \leq \|\cdot\|_C$, and there exist constants $\sigma_R, \sigma_C \in (0, 1)$ such that

$$\|Rx - \hat{x}\|_R \leq \sigma_R \|x - \hat{x}\|_R, \tag{10}$$

$$\|Cx - \hat{x}\|_C \leq \sigma_C \|x - \hat{x}\|_C, \tag{11}$$

for any $x \in \mathbb{R}^n$ and $\hat{x} = V_\infty x$.

**Remark 7.** Let $\|\cdot\|$ represent the matrix norm induced by $\|\cdot\|_R$. According to [36, Lemma 5.6.10], the constant $\sigma_R \in (0, 1)$ in Lemma 1 has an explicit form $\sigma_R = \|R - V_\infty\|$.

Lemma 1 directly follows from (8) and Assumption 1 – for the proof of (10), see [30, Lemma 2], and (11) can be shown similarly since $\lim_{k \to \infty} C^k = \lim_{k \to \infty} R^k = 1_n \pi^\top$. Indeed, one can argue that $\rho(R - V_\infty) < 1$; thus, [36, Lemma 5.6.10] implies that there exists invertible $S \in \mathbb{R}^n$ such that $\|x\|_R = \|Sx\|_1$; moreover, the matrix norm $\|\cdot\|$ induced by $\|\cdot\|_R$ satisfies $\|R - V_\infty\| \in (0, 1)$. Finally, through properly scaling $\|\cdot\|_R$, we immediately get $\|\cdot\| \leq \|\cdot\|_R$, which does not affect $\|\cdot\|$ since $\|B\| = \max\{\|Bx\|_R / \|x\|_R : x \neq 0\}$ for any $B \in \mathbb{R}^{n \times n}$. Same arguments can be used for showing (11) as we also have $\rho(C - V_\infty) < 1$.

First, we remark that all vector norms on a finite dimensional vector spaces are equivalent,
i.e., there exist $\kappa_1, \kappa_2, \kappa_3, \kappa_4 > 0$ such that
\[
\| \cdot \| \leq \kappa_1 \| C \|, \quad \| \cdot \| \leq \kappa_2 \| R \|
\]
(12)

**Remark 8.** Since $R$ corresponds to an Ergodic Markov chain, Remarks 2 and 5 imply that $V_\infty R = RV_\infty = V_\infty V_\infty = V_\infty$.

It is shown in [22] that $\| V(k) - V_\infty \| \leq \lambda \lambda^k$ for some $0 < \lambda \in \mathbb{R}$ and $\lambda \in (0, 1)$. Below we analyze the dependence of $\lambda$ and $\Lambda$ on $R$.

**Lemma 2.** Let $V(k) = R^k$ for $k \geq 0$ and $V_\infty = \lim_{k \to \infty} R^k$. Then, for $\kappa_3 > 0$ defined in (12) and $\sigma_R \in (0, 1)$ given in Remark 7 the following bound holds:
\[
\| V(k) - V_\infty \| \leq \kappa_3 \sigma_R, \quad \forall \ k \geq 0.
\]
(13)

**Proof.** It immediately follows from Remark 8 that for $k \geq 1$:
\[
\| V(k) - V_\infty \| \leq \|(R - V_\infty)^k\| \leq \kappa_3 \|(R - V_\infty)^k\| \leq \kappa_3 \sigma_R^k,
\]
where the second inequality follows from
\[
\| A \| = \max_{\| v \| \leq 1} \| Av \| \leq \max_{\| v \| \leq \kappa_3} \| Av \|_R = \| A \|, \quad \forall \ A \in \mathbb{R}^{n \times n};
\]
and the third inequality is due to $\| \cdot \|$ being submultiplicative as it is an induced norm. \[ \blacksquare \]

**Lemma 3.** The following inequalities hold for all $k \geq 0$:
\[
\| \tilde{V}^{-1}(k) - \tilde{V}_\infty^{-1} \| \leq \tilde{v}^2 \sqrt{n} \kappa_3 \sigma_R^k \]
(14a)
\[
\| \tilde{V}^{-1}(k) - \tilde{V}^{-1}(k - 1) \| \leq 2 \tilde{v}^2 \sqrt{n} \kappa_3 \sigma_R^k.
\]
(14b)

The proof Lemma 3 follows from [30, Lemma 3] and Lemma 2 and using $\| A \|_F \leq \sqrt{n} \| A \|_2$ for any $A \in \mathbb{R}^{n \times n}$. 

DRAFT
Lemma 4. The following inequality holds for all $k$:

\[(a) \quad \|V_\infty \tilde{V}^{-1}(k) \nabla f(x(k))\| \leq \nu \tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k} \|\nabla f(x(k))\| + nL\|x(k) - \dot{x}(k)\|_C + nL\|\ddot{x}(k) - x^*\| \]

\[(b) \quad \|V_\infty \tilde{V}^{-1}(k - 1) \nabla f(x(k))\| \leq 3\nu \tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k} \|\nabla f(x(k))\| + nL\|x(k) - \dot{x}(k)\|_C + nL\|\ddot{x}(k) - x^*\| \]

\[(c) \quad \|\dot{x}(k) - \ddot{x}(k - 1)\| \leq \alpha \nu \tilde{v} L \|x(k) - x(k - 1)\|_R + \alpha 3\nu \tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k} \|\nabla f(x(k))\| + \alpha nL\|x(k) - \dot{x}(k)\|_C + \alpha nL\|\ddot{x}(k) - x^*\| \]

\[(d) \quad \|\tilde{V}^{-1}(k) \nabla f(x(k)) - \tilde{V}^{-1}(k - 1) \nabla f(x(k - 1))\| \leq \tilde{v} L\|x(k) - x(k - 1)\|_R + 2\nu \tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k} \|\nabla f(x(k))\|.\]

Proof. First, we prove the part \(a\).

\[\|V_\infty \tilde{V}^{-1}(k) \nabla f(x(k))\| \leq \|V_\infty \tilde{V}^{-1}(k) \nabla f(x(k)) - V_\infty \tilde{V}^{-1}_\infty \nabla f(x(k))\| + \|V_\infty \tilde{V}^{-1}_\infty \nabla f(x(k))\| \]

\[\leq \|V_\infty\| \|\tilde{V}^{-1}(k) - \tilde{V}^{-1}_\infty\| \|\nabla f(x(k))\| + \|V_\infty \tilde{V}^{-1}_\infty \nabla f(x(k)) - 1_n 1_n^T \nabla f(x^*)\| \]

\[\leq \nu \tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k} \|\nabla f(x(k))\| + nL\|x(k) - x^*\|,\]

which implies \(a\) using triangular inequality, where $\tilde{V}_\infty$ is defined in Definition 3. In the second inequality, we use Remark 6 and the third inequality follows from \(14a\) in Lemma 3 and we also use $V_\infty \tilde{V}^{-1} = 1_n 1_n^T$, $\|1_n 1_n^T\| = n$ and Remark 4. Next, we prove part \(b\):

\[\|V_\infty \tilde{V}^{-1}(k - 1) \nabla f(x(k))\| \leq \|V_\infty \tilde{V}^{-1}(k - 1) \nabla f(x(k)) - V_\infty \tilde{V}^{-1}_\infty \nabla f(x(k))\| + \|V_\infty \tilde{V}^{-1}_\infty \nabla f(x(k))\| \]

\[\leq \|V_\infty\| \|\tilde{V}^{-1}(k - 1) - \tilde{V}^{-1}_\infty\| \|\nabla f(x(k))\| + \|V_\infty \tilde{V}^{-1}_\infty \nabla f(x(k))\| \]

\[\leq \nu \tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k} \|\nabla f(x(k))\| + nL\|x(k) - x^*\|,\]

hence, the part \(b\) follows from \(14b\) in Lemma 5 and from the part \(a\) of Lemma 4.

Now we consider part \(c\). Since $y(0) = 0_n$, it follows from \(9b\) that $y(k) = \beta(I_n - R) \sum_{\ell=1}^k x(\ell)$. Since $V_\infty R = V_\infty$ – see Remark 8 we have $V_\infty y(k) = 0_n$, for all $k \geq 0$ as $V_\infty (I_n - R) = 0_{n \times n}$. Hence, using $\dot{x}(k) = V_\infty x(k)$ for $k \geq 0$, when we multiply $V_\infty$ on both side of \(9a\), we get

$$\dot{x}(k) = \dot{x}(k - 1) - \alpha V_\infty \tilde{V}^{-1}(k - 1) \nabla f(x(k - 1)).$$
Therefore, the part \((c)\) immediately follows from using Remark 4 and the part \((b)\) of Lemma 4 on
\[
\|\hat{x}(k) - \hat{x}(k-1)\| \leq \alpha \|V_{\infty}^{-1}(k-1)[\nabla f(x(k-1)) - \nabla f(x(k))]| + \alpha \|V_{\infty}^{-1}(k-1)\nabla f(x(k))|.
\]
Finally, we consider the part \((d)\).
\[
\|\hat{V}^{-1}(k)\nabla f(x(k)) - \hat{V}^{-1}(k-1)\nabla f(x(k-1))\| \\
\leq \|\hat{V}^{-1}(k-1)\nabla f(x(k)) - \hat{V}^{-1}(k-1)\nabla f(x(k-1))\| \\
+ \|\hat{V}^{-1}(k)\nabla f(x(k)) - \hat{V}^{-1}(k-1)\nabla f(x(k))\| \\
\leq \|\hat{V}^{-1}(k-1)\|\|\nabla f(x(k)) - \nabla f(x(k-1))\| \\
+ \|\hat{V}^{-1}(k) - \hat{V}^{-1}(k-1)\|\|\nabla f(x(k))\|.
\]
Hence, the part \((d)\) follows from \((14b)\) of Lemma 3 and Remark 4.

For the sake of completeness we provide another technical result –for its proof, see [34, Lemma 10].

**Lemma 5.** Under Assumptions 2 and 3 holds, for all \(x \in \mathbb{R}^p\) and \(\alpha \in (0, \frac{2}{nL})\), one has
\[
\|x - \alpha \sum_{i=1}^{n} \nabla f_i(x) - x^*\| \leq \eta \|x - x^*\|
\]
where \(\eta \triangleq \max \{|1 - nL\alpha|, |1 - n\mu\alpha|\}\).

Next, we will obtain bounds on \(\|x(k+1) - \hat{x}(k+1)\|_C, \|\hat{x}(k+1) - x^*\|\) and \(\|x(k+1) - x(k)\|_R\). Combining these results will help us establish the linear rate for FRSD.

**Lemma 6.** The following inequality holds for all \(k \geq 0:\)
\[
\|x(k+1) - \hat{x}(k+1)\|_C \\
\leq (\sigma_C + \alpha \kappa_4 nL)\|x(k) - \hat{x}(k)\|_C(\kappa_2\|R\| + \alpha \kappa_4 \tilde{v}L)\|x(k) - x(k-1)\|_R \\
+ \alpha \kappa_4 nL\|\hat{x}(k) - x^*\| + \alpha \kappa_4(2 + \tilde{v})\tilde{v}^2 \sqrt{n\kappa_3 \sigma_R^k}\|\nabla f(x(k))\|
\]
where \(\|\cdot\|\) denotes the induced matrix norm corresponding to the vector norm \(\|\cdot\|_R\).
Proof. Using (9a) twice, one for $x(k + 1)$ and one for $\hat{x}(k + 1) = V_{\infty}x(k + 1)$, and using $V_{\infty}R = V_{\infty}$ together with $V_{\infty}y(k) = 0$, we get the first equality below:

$$\|x(k + 1) - \hat{x}(k + 1)\|_C \tag{15}$$

$$= \|Rx(k) - \alpha y(k) - \alpha \hat{V}^{-1}(k)\nabla f(x(k)) - \hat{x}(k) + \alpha V_{\infty}V^{-1}(k)\nabla f(x(k))\|_C$$

$$= \|Rx(k) - Rx(k - 1) + x(k) - \hat{x}(k) - \alpha \beta(I_n - R)x(k) + \alpha \hat{V}^{-1}(k)\nabla f(x(k)) + \alpha V_{\infty}V^{-1}(k)\nabla f(x(k))\|_C$$

$$\le \|(1 - \alpha \beta)I_n + \alpha \beta R)x(k) - \hat{x}(k)\|_C + \kappa_2\|Rx(k) - Rx(k - 1)\|_R + \alpha \kappa_4(\hat{V}^{-1}(k)\nabla f(x(k)) - \hat{V}^{-1}(k - 1)\nabla f(x(k - 1))\|_R + \alpha \kappa_4\|V_{\infty}V^{-1}(k)\nabla f(x(k))\|,$$

where in the second equality we first use (9b) to represent $y(k)$ in terms of $x(k)$ and $y(k - 1)$, and next we use (9a) to get rid of the term $-\alpha y(k - 1)$.

Next, using (11) of Lemma 1, we can bound the first term on the right-hand-side of (16) as follows:

$$\|(1 - \alpha \beta)I_n + \alpha \beta R)x(k) - \hat{x}(k)\|_C = \|Cx(k) - \hat{x}(k)\|_C \le \sigma_C\|x(k) - \hat{x}(k)\|_C,$$

where $C$ is given in Definition 4. Clearly, we can also bound the second term in (16) with $\|R\|\|x(k) - x(k - 1)\|_R$. Finally, using the parts (d) and (a) of Lemma 4 for the third and the fourth terms, respectively, we get the desired result.

Remark 9. The FRSD stepsize bound, $\alpha = O(\frac{1}{n})$, compares similarly to the stepsizes used in other related works, e.g., the AB, Push-DiGing, Xi-row methods.

Lemma 7. When $0 < \alpha < \frac{2}{nL}$, it holds that for $k \ge 0$:

$$\|\hat{x}(k + 1) - \hat{x}^*\| \le \eta\|\hat{x}(k) - \hat{x}^*\| + \alpha nL\|x(k) - \hat{x}(k)\|_C + \alpha n\bar{v}^2\sqrt{n\kappa_3}\sigma_R\|\nabla f(x(k))\|.$$

Proof. Using (9a) for $\hat{x}(k + 1) = V_{\infty}x(k + 1)$ together with $V_{\infty}R = V_{\infty}$ and $V_{\infty}y(k) = 0$, we get

$$\|\hat{x}(k + 1) - \hat{x}^*\| = \|\hat{x}(k) - \alpha V_{\infty}\hat{V}^{-1}(k)\nabla f(x(k)) - \hat{x}^*\| \le \|\hat{x}(k) - \alpha 1_n1_n^\top\nabla f(\hat{x}(k)) - \hat{x}^*\| + \alpha\|1_n1_n^\top\nabla f(\hat{x}(k)) - V_{\infty}\hat{V}^{-1}(k)\nabla f(x(k))\|.$$ 

(17)
The first term on the right-hand-side of (17) can be bounded using Lemma 5:

\[ \| \hat{x}(k) - \alpha 1_n 1_n^T \nabla f(\hat{x}(k)) - x^* \| \leq \eta \| \hat{x}(k) - x^* \| \] (18)

where \( \eta = \max \{ |1 - nL\alpha|, |1 - n\mu\alpha| \} \). Next, we bound the second term in (17) as follows:

\[
\| 1_n 1_n^T \nabla f(\hat{x}(k)) - V_\infty \tilde{V}^{-1}(k) \nabla f(x(k)) \|
\leq \| 1_n 1_n^T \nabla f(\hat{x}(k)) - V_\infty \tilde{V}^{-1}(k) \nabla f(x(k)) \| + \| V_\infty \tilde{V}^{-1}(k) \nabla f(x(k)) - V_\infty \tilde{V}^{-1}(k) \nabla f(x(k)) \|
\leq nL \| x(k) - \hat{x}(k) \|_C + vv^2 \sqrt{n} \kappa_3 \sigma_R^k \| \nabla f(x(k)) \|, \] (19)

where we used \( V_\infty \tilde{V}^{-1} = 1_n 1_n^T \), Assumption 2 and Lemma 3. Finally, Lemma 7 follows from (17)-(19).

**Lemma 8.** The following inequality holds for all \( k \geq 0 \):

\[
\| x(k+1) - x(k) \|_R \leq (\sigma_R + \alpha (1 + v) \kappa_3 \tilde{v}L) \| x(k+1) - x(k) \|_R + \alpha (\beta \kappa_1 \| I_n - R \| \\
+ \kappa_3 nL) \| x(k) - \hat{x}(k) \|_C + \alpha \kappa_3 nL \| \hat{x}(k) - x^* \| \\
+ \kappa_3^2 \alpha (3v + 2) \tilde{v} \sqrt{n} \| \nabla f(x(k)) \|. \]

**Proof.** We use (9a) and (9b) for rewriting \( x(k+1) \) and \( y(k) \) respectively, to derive the first two equations:

\[
\| x(k+1) - x(k) \|_R = \| R x(k) - \alpha y(k) - \alpha \tilde{V}^{-1}(k) \nabla f(x(k)) - x(k) \|_R \\
= \| R x(k) - \alpha y(k-1) - \alpha \beta (I_n - R)x(k) - \alpha \tilde{V}^{-1}(k) \nabla f(x(k)) - x(k) \|_R \\
= \| R (x(k) - x(k-1)) + \alpha \tilde{V}^{-1}(k-1) \nabla f(x(k-1)) - \alpha \beta (I_n - R)x(k) \\
- \alpha \tilde{V}^{-1}(k) \nabla f(x(k)) \|_R \\
\leq \| R x(k) - R x(k-1) - \hat{x}(k) + \hat{x}(k-1) \|_R + \kappa_3 \| \hat{x}(k) - \hat{x}(k-1) \| \\
+ \alpha \beta \| (I_n - R)x(k) \|_R \\
+ \alpha \kappa_3 \| \tilde{V}^{-1}(k) \nabla f(x(k)) - \tilde{V}^{-1}(k-1) \nabla f(x(k-1)) \| \]

where in the third equation, we use (9a) to get rid of the term \( -\alpha y(k-1) \) as we did previously to derive (16). We bound the first term above using Remark 7 i.e.,

\[
\| R x(k) - R x(k-1) - \hat{x}(k) + \hat{x}(k-1) \|_R = \| (R - V_\infty)(x(k) - x(k-1)) \|_R \\
\leq \sigma_R \| x(k) - x(k-1) \|_R. \] (21)

DRAFT
We can use Lemma 4 (c) to bound $\|\hat{x}(k) - \hat{x}(k - 1)\|$, and Lemma 4 (d) to bound the fourth term. Then, the remaining third term in (20) can be bounded as

$$\|(I_n - R)x(k)\|_R = \|(I_n - R)(x(k) - \hat{x}(k))\|_R \leq \|I_n - R\|\|x(k) - \hat{x}(k)\|_R \leq \kappa_1\|I_n - R\|\|x(k) - \hat{x}(k)\|_C,$$

(22)

where in the first equality follows from $(I_n - R)V_\infty = 0$ due to $RV_\infty = V_\infty$; hence, we can add $(I_n - R)\hat{x}(k)$ to $(I_n - R)x(k)$. Combining all bounds gives the desired result. ■

Combining the results of Lemmas 6, 7 and 8, we will construct a linear dynamical system prove the linear convergence of the proposed algorithm. For the sake of notational simplicity, we define some constants below:

\[
\begin{align*}
    s_1 & = \kappa_4 nL, & s_2 & = \kappa_4 nL, & s_3 & = \kappa_4 \tilde{v}L, \\
s_4 & = \tilde{v}L, & s_5 & = \kappa_1 \beta \|I_n - R\| + \kappa_3 nL, & s_6 & = \kappa_3 nL, \\
s_7 & = \kappa_3 (1 + v)\tilde{v}L, & s_8 & = \kappa_3 \kappa_4 (2 + v)\tilde{v}^2 \sqrt{n}, & s_9 & = \kappa_3 v\tilde{v}^2 \sqrt{n}, \\
s_{10} & = \kappa_3^2 (3v + 2)\tilde{v}^2 \sqrt{n}.
\end{align*}
\]

For $\alpha \in (0, \frac{2}{nL})$ and $\beta > 0$ such that $\alpha \beta < 1$, FRSD sequence $\{x(k)\}_{k \geq 0}$ satisfies the following system:

$$\theta_{k+1} \leq \Upsilon \theta_k + \Phi_k \Psi_k, \quad \forall \ k \geq 0,$$

(23)

where $\theta_k$, $\Phi_k$, $\Psi_k$ and $\Upsilon$ are defined as

\[
\begin{align*}
    \theta_k &= \begin{bmatrix}
    \|x(k) - \hat{x}(k)\|_C \\
    \|\hat{x}(k) - x^*\| \\
    \|x(k) - x(k-1)\|_R
    \end{bmatrix}, & \Phi_k &= \sigma_R^k \begin{bmatrix}
    s_8 \alpha & 0 & 0 \\
    s_9 \alpha & 0 & 0 \\
    s_{10} \alpha & 0 & 0
    \end{bmatrix}, \\
    \Upsilon &= \begin{bmatrix}
    \sigma_C + s_1 \alpha & s_2 \alpha & \kappa_2 \|R\| + s_3 \alpha \\
    s_4 \alpha & \eta & 0 \\
    s_5 \alpha & s_6 \alpha & \sigma_R + s_7 \alpha
    \end{bmatrix}, & \Psi_k &= \begin{bmatrix}
    \|\nabla f(x(k))\| \\
    0 \\
    0
    \end{bmatrix}.
\end{align*}
\]
Theorem 1. Suppose Assumptions \[1\]-\[3\] holds. Let \(\alpha, \beta > 0\) such that \(\alpha \in (0, \bar{\alpha})\) and \(\alpha \beta < 1\), where

\[
\bar{\alpha} \triangleq \sup_{\delta_1, \delta_2} \min \left\{ \frac{(1 - \sigma_c) - \kappa_2 \| R \| \delta_2}{s_1 + s_2 \delta_1 + s_3 \delta_2}, \frac{(1 - \sigma_R)\delta_2}{s_5 + s_6 \delta_1 + s_7 \delta_2}, \frac{1}{nL} \right\}
\]

\[
\text{s.t.} \quad \frac{L}{\mu} < \delta_1, \quad 0 < \delta_2 < \frac{1 - \sigma_c}{\kappa_2 \| R \|}.
\]

(24)

Then, the spectral radius \(\rho(\Upsilon) < 1\) holds.

Proof. Given \(\alpha \in (0, \frac{2}{nL})\) and \(\beta > 0\) such that \(\alpha \beta < 1\), it follows from Lemmas [6,8] that (23) holds for \(k \geq 0\). Next, we show \(\rho(\Upsilon) < 1\). Since \(\Upsilon\) has all non-negative entries, it is sufficient to show that \(\Upsilon \gamma < \gamma\) for some positive \(\gamma = [\gamma_1, \gamma_2, \gamma_3]^\top \in \mathbb{R}_+^3\) –see [36, Corollary 8.1.29]. Since \(L \geq \mu\), according to the definition of \(\eta\) given in Lemma [5], \(\eta = 1 - \alpha n\mu\) for \(\alpha \in (0, \frac{1}{nL})\). Hence, \(\Upsilon \gamma < \gamma\) is equivalent to

\[
(s_1 \gamma_1 + s_2 \gamma_2 + s_3 \gamma_3) \alpha < \gamma_1 (1 - \sigma_c) - \kappa_2 \| R \| \gamma_3,
\]

\[
(s_4 \gamma_1 \alpha - \gamma_2 n \mu \alpha < 0,
\]

\[
(s_5 \gamma_1 + s_6 \gamma_2 + s_7 \gamma_3) \alpha < \gamma_3 (1 - \sigma_R).
\]

(25)

Clearly, (25) holds for all \(\alpha \in (0, \bar{\alpha})\) and \(\gamma \in \mathbb{R}_+^3\) such that \(\gamma_2 = \delta_1 \gamma_1\) and \(\gamma_3 = \delta_2 \gamma_1\) for any \(\gamma_1 > 0\) and \(\delta_1, \delta_2 > 0\) satisfying (24); thus, we get \(\rho(\Upsilon) < 1\).

\[\blacksquare\]

Remark 10. Note \(\delta_1\) and \(\delta_2\) are free parameters that need to satisfy only (24). To provide a lower bound on an admissible \(\alpha\), we compute a lower bound on \(\bar{\alpha}\) by setting \(\delta_2 = \frac{1 - \sigma_c}{2 \kappa_2 \| R \|}\) satisfying (24). Note the supremum over \(\delta_1\) subject to (24) is achieved at \(\delta_1 = \frac{1}{\mu}\). For this particular choice we get \(\bar{\alpha} < \frac{1}{nL}\) and \(\bar{\alpha} \geq \min\{\alpha_1, \alpha_2\}\), where

\[
\alpha_1 \triangleq \left[ \frac{2 \kappa_4}{1 - \sigma_c} \left( \frac{L}{\mu} + 1 \right) n L + \frac{\kappa_4}{\kappa_2} \bar{v} L \right]^{-1},
\]

\[
\alpha_2 \triangleq (1 - \sigma_R) \left[ \frac{\kappa_2 \kappa_3 \| R \|}{1 - \sigma_c} \left( \frac{L}{\mu} + 1 \right) n L + \frac{\kappa_1 \kappa_2 \| R \| \| I - R \| \beta}{1 - \sigma_c} + \kappa_3 (1 + v) \bar{v} L \right]^{-1},
\]

where we used \(1 = \rho(R) \leq \| R \|\).

Finally, in the next theorem, we prove that FRSD iterate sequence converges linearly through showing a linear decay for \(\{\Phi_k\}\). First, we state a classic result that will be useful in our analysis; for its proof, see [35, 37].

DRAFT
Lemma 9. Let \( \{a_k\} \), \( \{b_k\} \), \( \{c_k\} \) and \( \{d_k\} \) be non-negative sequences such that \( \sum_{k=0}^{\infty} c_k < \infty \), \( \sum_{k=0}^{\infty} d_k < \infty \), and
\[
a_{k+1} \leq (1 + c_k)a_k - b_k + d_k, \quad \forall \ k \geq 0.
\]
Then \( \{a_k\} \) converges and \( \sum_{k=0}^{\infty} b_k < \infty \).

Theorem 2. Let Assumptions 1-3 hold. Then, the sequence \( \{x(k)\} \) converges to \( x^* \) for any sufficiently small step-size \( \alpha \in (0, \bar{\alpha}) \), where \( \bar{\alpha} \) is defined in the Theorem 1.

Proof. Theorem 1 shows that the spectral radius \( \Upsilon \) is less than 1; hence, using the same arguments in the proof of [30, Lemma 5], we conclude that there exists some \( \Gamma > 0 \) and \( \bar{\lambda} \in (\sigma_R, 1) \) such that for all \( 0 \leq j \leq k - 1 \), we have
\[
\|\Upsilon^k\| \leq \Gamma \bar{\lambda}^k, \quad \|\Upsilon^{k-j-1}\Phi_j\| \leq \Gamma \bar{\lambda}^k.
\] (26)

By writing (23) recursively, we get, for all \( k \geq 0 \),
\[
\theta_k \leq \Upsilon^k\theta_0 + \sum_{j=0}^{k-1} \Upsilon^{k-j-1}\Phi_j\Psi_j.
\] (27)

Since all the terms in (27) have non-negative entries, using (26), we get for all \( k \geq 0 \),
\[
\|\theta_k\| \leq \|\Upsilon^k\||\theta_0|| + \sum_{j=0}^{k-1} \|\Upsilon^{k-j-1}\Phi_j\||\Psi_j|
\leq \Gamma \bar{\lambda}^k (\|\theta_0\| + \sum_{j=0}^{k-1} ||\Psi_j||).
\] (28)

For any \( k \geq 0 \), we can bound \( ||\Psi_k|| \) as follows:
\[
||\Psi_k|| \leq ||\nabla f(x(k)) - \nabla f(x^*)|| + ||\nabla f(x^*)||
\leq L||x(k) - x(k)|| + L||x(k) - x^*|| + ||\nabla f(x^*)||
\leq 2L||\theta_k|| + ||\nabla f(x^*)||.
\] (29)

Thus, for all \( k \geq 0 \), combining (28) and (30) we get
\[
||\theta_k|| \leq \left( ||\theta_0|| + 2L \sum_{j=0}^{k-1} ||\theta_j|| + k||\nabla f(x^*)|| \right) \Gamma \bar{\lambda}^k.
\]
For $k \geq 0$, let $a_k \triangleq \sum_{j=0}^{k-1} \|\theta_j\|$, $b_k \triangleq 0$, $\bar{c} \triangleq 2L\Gamma$, and $\bar{d}_k \triangleq \Gamma \|\theta_0\| + k\Gamma \|\nabla f(x^*)\|$; hence, we get

$$\|\theta_k\| = a_{k+1} - a_k \leq (\bar{c}a_k + \bar{d}_k)\bar{\lambda}^k, \quad \forall \ k \geq 0. \quad (31)$$

Define $c_k \triangleq \bar{c}\bar{\lambda}^k \geq 0$ and $d_k \triangleq \bar{d}_k\bar{\lambda}^k \geq 0$ for $k \geq 0$. Since $\bar{\lambda} \in (0, 1)$, we have $\sum_{k=0}^{\infty} c_k + d_k < \infty$; therefore, Lemma 9 implies that $\{a_k\}$ converges. Furthermore, since $\{a_k\}$ is bounded, (31) implies that for all $\xi \in (0, 1 - \bar{\lambda})$, we get

$$\lim_{k \to \infty} \frac{\|\theta_k\|}{(\lambda + \xi)^k} \leq \frac{(\bar{c}a_k + \bar{d}_k)\bar{\lambda}^k}{(\lambda + \xi)^k} = 0. \quad (32)$$

Thus, there exist $p > 0$ such that

$$\|\theta_k\| \leq p(\bar{\lambda} + \xi)^k, \quad \forall \ k \geq 0, \quad (33)$$

Thus, we get the desired result by showing for all $k \geq 0$,

$$\|x(k) - x^*\| \leq \|x(k) - \hat{x}(k)\| + \|\hat{x}(k) - x^*\| \leq 2\|\theta_k\| \leq 2p(\bar{\lambda} + \xi)^k.$$

\[\square\]

III. Numerical Results

In this section, we provide some numerical results to demonstrate the performance of the proposed method against the state-of-the-art competitive algorithms designed for directed graphs. We compare FRSD with Xi-row [30], which uses only row-stochastic weights as our method, Push-DIGing [25], which utilizes column-stochastic weights, and also with AB [27], and Push-Pull [29], which use both row-stochastic and column-stochastic weights. We consider two different time-invariant directed graphs with $n = 10$ nodes (agents), see Figure 1. In our experiments, we considered two types of distributed regression problems, of the form given in (1); one with Huber loss and the other is the logistic regression as described in Sections III-A and III-B respectively. Throughout the experiments, we use the uniform weighting strategy to set up the row-stochastic weights in (8), i.e., $r_{ij} = 1/|N_i^m|$ for all $i \in \mathcal{V}$. For each $i \in \mathcal{V}$, let $M_i \in \mathbb{R}^{m_i \times p}$ represent $m_i$ data points with $p - 1$ features and the last column of $M_i$ is the vector of all ones to model intercept.
A. Distributed Regression with Huber Loss

Suppose $\tilde{x} \in \mathbb{R}^p$ is the unknown linear model, and for each $i \in \mathcal{V}$, let $y_i \in \mathbb{R}^{m_i}$ be the corresponding noisy measurement vector, i.e., $y_i = M_i\tilde{x} + e_i$ where $e_i \in \mathbb{R}^{m_i}$ is some noise. Given parameter $\xi > 0$, the Huber loss function $H_\xi : \mathbb{R} \to \mathbb{R}_+$ is defined as

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

For any $m \in \mathbb{Z}_+$, we also define $H_\xi : \mathbb{R}^m \to \mathbb{R}^m$ such that $H_\xi(z) = [H_\xi(z_j)]_{j=1}^m$ where $z = [z_j]_{j=1}^m$.

In this experiment, the goal is to estimate $\tilde{x}$ with an optimal solution $x^*$ to the Huber loss problem:

$$x^* = \arg\min_{x \in \mathbb{R}^p} \bar{f}(x) \triangleq \frac{1}{n} \sum_{i=1}^n H_\xi(M_i\tilde{x} - y_i),$$

(34)

In the experiments, following a similar setup as in [25], $\xi = 2$ and $m_i = 1$ for $i \in \mathcal{V}$ and we set $p = 6$ and we solve (34) over the directed graphs in Fig. 1 with $n = 10$. For each $i \in \mathcal{V}$, we generated $f_i(x) = H_\xi(M_i\tilde{x} - y_i)$ as described in [25, Sec. 6] such that $L_i = 1$. Moreover, we also initialized all the methods from $x_i(0) = 0$ for all $i \in \mathcal{V}$. As noted in [25], as $n = 10$ and $p = 6$, $\bar{f}$ is restricted strongly convex while $f_i$ is merely convex for $i \in \mathcal{V}$.

In Fig. 2 we plot the residual sequence $\{r(k)\}_{k \geq 0}$ for all the methods where $r(k) \triangleq \frac{\|x(k) - x^*\|}{\|x(0) - x^*\|}$. To optimize the convergence rate, we tuned the step size, $\alpha$, for all algorithms. It is worth emphasizing that as FRSD has an additional parameter, $\beta$, while the other only has $\alpha$ to tune; therefore, we were able to tune $(\alpha, \beta)$ so that for both graphs in Fig. 1, FRSD exhibits a faster convergence compared to the others methods.

Fig. 1: The strongly-connected directed graphs.
B. Distributed Logistic Regression

We now consider the distributed binary classification problem using the logistic regression to train a linear classifier. Suppose each node (agent) \( i \in \mathcal{V} \) has access to \((M_i, y_i) \in \mathbb{R}^{m_i \times p} \times \{-1, +1\}^{m_i}\). Let \( L : \mathbb{R} \times \{-1, 1\} \rightarrow \mathbb{R}_+ \) such that \( L(u, v) = \ln(1 + \exp(-uv)) \); and for any \( m \in \mathbb{Z}_+ \), we also define \( \mathbf{L} : \mathbb{R}^m \times \{-1, 1\}^m \rightarrow \mathbb{R}_+^m \) such that \( \mathbf{L}(\mathbf{u}, \mathbf{v}) = [L(u_j, v_j)]_{j=1}^m \) where \( \mathbf{u} = [u_j]_{j=1}^m \) and \( \mathbf{v} = [v_j]_{j=1}^m \). The linear classifier \( x^* \) is computed by solving the regularized logistic regression problem:

\[
    x^* = \arg\min_{x \in \mathbb{R}^p} \bar{f}(x) \triangleq \frac{1}{n} \sum_{i=1}^n \left( L(M_i x, y_i) + \frac{\lambda}{2} \|x\|_2^2 \right),
\]

where using regularization parameter \( \lambda > 0 \) improves the statistical properties of \( x^* \) – see \[38\].

In the experiments, we use the australian-scale dataset \[39\] with 790 data points where each data point consists of a 14-dimensional feature vector, i.e., \( p = 15 \) to model the intercept, and the corresponding binary label. Suppose each agent \( i \) samples \( m_i = 10 \) data points uniformly at random from the training set with replacement. We test the proposed method FRSD against those methods that we compared with in Section III-A. The residual sequence \( \{r(k)\}_{k \geq 1} \) for all the methods are shown in Fig. 3 where \( r(k) \) is defined in Section III-A. Our algorithm FRSD exhibits a faster convergence rate for both graphs displayed in Fig. 1. We have observed that the improvement in the rate becomes more significant especially for when the graphs are sparse, which is indeed the case for most of the real-life networks in practice.

IV. CONCLUSION

In this paper, we proposed a distributed optimization algorithm, FRSD, for decentralized co-
sensus optimization over directed graphs. FRSD only employs a row-stochastic matrix for local messaging with neighbors, making it desirable for broadcast-based communication systems. The proposed algorithm achieves a geometric convergence to the global optimal when agents’ cost functions are strongly convex with Lipschitz continuous gradients. Empirical results demonstrated the efficacy of the novel momentum term employed by FRSD, which performed better in practice than the other-state-of-the-art methods we compared.

**REFERENCES**

[1] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 20–27.

[2] U. A. Khan, S. Kar, and J. M. Moura, “DILAND: An algorithm for distributed sensor localization with noisy distance measurements,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1940–1947, 2009.

[3] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009, vol. 27.

[4] V. Cevher, S. Becker, and M. Schmidt, “Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics,” *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 32–43, 2014.

[5] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[6] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

[7] H. Raja and W. U. Bajwa, “Cloud k-svd: A collaborative dictionary learning algorithm for big, distributed data,” *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 173–188, 2015.

[8] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning—part i: Algorithm development,” *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708–723, 2018.

[9] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
[10] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” IEEE Transactions on Automatic Control, vol. 54, no. 1, pp. 48–61, 2009.

[11] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” IEEE Transactions on Automatic Control, vol. 55, no. 4, pp. 922–938, 2010.

[12] S. S. Ram, A. Nedić, and V. V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” Journal of optimization theory and applications, vol. 147, no. 3, pp. 516–545, 2010.

[13] J. C. Duchi, A. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: Convergence analysis and network scaling,” IEEE Transactions on Automatic control, vol. 57, no. 3, pp. 592–606, 2011.

[14] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints,” IEEE Transactions on Automatic Control, vol. 57, no. 1, pp. 151–164, 2011.

[15] D. Jakovetić, J. Xavier, and J. M. Moura, “Fast distributed gradient methods,” IEEE Transactions on Automatic Control, vol. 59, no. 5, pp. 1131–1146, 2014.

[16] W. Shi, Q. Ling, G. Wu, and W. Yin, “Extra: An exact first-order algorithm for decentralized consensus optimization,” SIAM Journal on Optimization, vol. 25, no. 2, pp. 944–966, 2015.

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

[18] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, “On the linear convergence of the ADMM in decentralized consensus optimization,” IEEE Transactions on Signal Processing, vol. 62, no. 7, pp. 1750–1761, 2014.

[19] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, “Distributed linearized alternating direction method of multipliers for composite convex consensus optimization,” IEEE Transactions on Automatic Control, vol. 63, no. 1, pp. 5–20, 2017.

[20] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, “DQM: Decentralized quadratically approximated alternating direction method of multipliers,” IEEE Transactions on Signal Processing, vol. 64, no. 19, pp. 5158–5173, 2016.

[21] ——, “A decentralized second-order method with exact linear convergence rate for consensus optimization,” IEEE Transactions on Signal and Information Processing over Networks, vol. 2, no. 4, pp. 507–522, 2016.

[22] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” IEEE Transactions on Automatic Control, vol. 60, no. 3, pp. 601–615, 2014.

[23] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information,” in 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings. IEEE, 2003, pp. 482–491.

[24] C. Xi and U. A. Khan, “Dextra: A fast algorithm for optimization over directed graphs,” IEEE Transactions on Automatic Control, vol. 62, no. 10, pp. 4980–4993, 2017.

[25] A. Nedic, 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.

[26] C. Xi, R. Xin, and U. A. Khan, “ADD-OPT: Accelerated distributed directed optimization,” IEEE Transactions on Automatic Control, vol. 63, no. 5, pp. 1329–1339, 2017.

[27] 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.

[28] ——, “Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking,” IEEE Transactions on Automatic Control, 2019.

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

[30] 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.
[31] R. Xin, C. Xi, and U. A. Khan, “FROST—Fast row-stochastic optimization with uncoordinated step-sizes,” EURASIP Journal on Advances in Signal Processing, vol. 2019, no. 1, pp. 1–14, 2019.

[32] Q. Lü, X. Liao, H. Li, and T. Huang, “A nesterov-like gradient tracking algorithm for distributed optimization over directed networks,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020.

[33] A. H. Sayed, “Adaptation, learning, and optimization over networks,” Foundations and Trends in Machine Learning, vol. 7, no. ARTICLE, pp. 311–801, 2014.

[34] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1245–1260, 2017.

[35] H. Robbins and D. Siegmund, Optimizing methods in statistics (Proc. Symp., Ohio State Univ., Columbus, Ohio, 1971). New York: Academic Press, 1971, ch. A convergence theorem for non negative almost supermartingales and some applications, pp. 233 – 257.

[36] R. A. Horn and C. R. Johnson, Matrix analysis. Cambridge university press, 2012.

[37] B. T. Polyak, “Introduction to optimization. optimization software,” Inc., Publications Division, New York, vol. 1, 1987.

[38] K. Sridharan, S. Shalev-Shwartz, and N. Srebro, “Fast rates for regularized objectives,” Advances in neural information processing systems, vol. 21, pp. 1545–1552, 2008.

[39] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM transactions on intelligent systems and technology (TIST), vol. 2, no. 3, pp. 1–27, 2011.