OPTIMIZATION OVER DIRECTED GRAPHS: LINEAR CONVERGENCE RATE

Chenguang Xi and Usman A. Khan

ECE Department, Tufts University, Medford MA
chenguang.xi@tufts.edu, khan@ece.tufts.edu

ABSTRACT

This paper considers distributed multi-agents optimization problems where agents collaborate to minimize the sum of locally known convex functions. We focus on the case when the communication between agents is described by a *directed* graph. The proposed algorithm achieves the best known rate of convergence for this class of problems, $O(\mu^k)$ for $0 < \mu < 1$, given that the objective functions are strongly-convex, where k is the number of iterations. Moreover, it provides a wider and more realistic range of step-size compared with existing methods.

Index Terms— optimization, distributed algorithms, directed graphs, sensor networks

1. INTRODUCTION

We consider the problem of minimizing a sum of objective, $\sum_{i=1}^{n} f_i(\mathbf{z})$, where $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$ is a private objective known only to the ith agent in the network. This model has various applications in the signal processing research in the context of wireless communication, [1, 2], sensor networks, [3, 4], large-scale machine learning, [5, 6], etc. Most of the existing algorithms, [7–12], assume information exchange over undirected networks where the communication between agents is bidirectional. On the contrary, we consider optimization over directed networks in this paper. Such cases arises, e.g., when agents broadcast at different power levels.

We report the literature considering directed graphs here. Broadly, the following three approaches refer to the techniques of reaching average consensus over directed graphs and extend the results to solve the distributed optimization. Subgradient-Push (SP), [13–16], combines Distributed Subgradient Descent (DSD), [17], and push-sum consensus, [18, 19]. Directed-Distributed Subgradient Descent (D-DSD), [20, 21], applies surplus consensus, [22], to DSD. The algorithm in [23] is a combination of weight-balancing technique, [24], and DSD. These gradient-based methods, [13–16, 20, 21, 23], converge at $O(\frac{\ln k}{\sqrt{k}})$. When the objective functions are strongly-convex, the convergence rate can be accelerated to $O(\frac{\ln k}{k})$, [25].

Recently, we proposed DEXTRA, [26], which achieves a linear convergence rate, $O(\mu^k)$ for $0 < \mu < 1$, given that the objective functions are strongly-convex. However, a restriction of DEXTRA is the range of allowable step-sizes. In particular, the greatest lower bound of DEXTRA's step-size is strictly greater than zero. In this paper, we propose an algorithm to solve distributed optimization over directed graphs. The proposed algorithm achieves a linear convergence rate when the objective functions are strongly-convex. Compared to DEXTRA, the proposed algorithm's step-size, α, lies in $\alpha \in (0, \overline{\alpha})$. The rest of the paper is organized as follows. Section 2 formulates the problem and describes the algorithm. We provide the main result in Section 3. Section 4 shows simulations and Section 5 concludes the paper.

Notation: We use lowercase bold letters for vectors and uppercase italic letters for matrices. The matrix, I_n, represents the $n \times n$ identity, and $\mathbf{1}_n$ is the n-dimensional vector of all 1's for any n. The spectral radius of a matrix, A, is represented by $\rho(A)$, and $\lambda(A)$ denotes any eigenvalue of A.

2. ALGORITHM DEVELOPMENT

Consider a strongly-connected network of n agents communicating over a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents, and $\mathcal{E}$ is the set of edges. We are interested in the following optimization problem that is distributed over the above *directed* multi-agent network:

$$\text{P1}: \quad \min f(\mathbf{z}) = \sum_{i=1}^{n} f_i(\mathbf{z}),$$

where each local objective function, $f_i : \mathbb{R}^p \rightarrow \mathbb{R}$, is convex and differentiable, and known only by agent i.

To solve Problem P1, we describe the implementation of the algorithm as follows. Each agent, $j \in \mathcal{V}$, maintains three vector variables: $\mathbf{x}_{k,j}$, $\mathbf{z}_{k,j}$, $\mathbf{w}_{k,j} \in \mathbb{R}^p$, as well as a scalar variable, $y_{k,j} \in \mathbb{R}$, where k is the discrete-time index. At kth iteration, agent j weights its states, $a_{ij}\mathbf{x}_{k,j}$, $a_{ij}y_{k,j}$, as well as $a_{ij}\mathbf{w}_{k,j}$, and sends these to each of its out-neighbors, $i \in \mathcal{N}_j^{\text{out}}$, where the weights, a_{ij}'s are such that:

$$a_{ij} = \begin{cases} > 0, & i \in \mathcal{N}_j^{\text{out}}, \\ 0, & \text{otw.}, \end{cases} \qquad \sum_{i=1}^{n} a_{ij} = 1, \forall j. \tag{1}$$

This work has been partially supported by NSF Award # CCF-1350264.

With agent i receiving the information from its in-neighbors, $j \in \mathcal{N}_i^{\text{in}}$, it updates $\mathbf{x}_{k+1,i}$, $y_{k+1,i}$, $\mathbf{z}_{k+1,i}$ and $\mathbf{w}_{k+1,i}$ as

$$\mathbf{x}_{k+1,i} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij}\mathbf{x}_{k,j} - \alpha \mathbf{w}_{k,i}, \tag{2a}$$

$$y_{k+1,i} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} y_{k,j}, \qquad \mathbf{z}_{k+1,i} = \frac{\mathbf{x}_{k+1,i}}{y_{k+1,i}}, \tag{2b}$$

$$\mathbf{w}_{k+1,i} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij}\mathbf{w}_{k,j} + \nabla f_i(\mathbf{z}_{k+1,i}) - \nabla f_i(\mathbf{z}_{k,i}). \tag{2c}$$

In the above, $\nabla f_i(\mathbf{z}_{k,i})$ in the gradient of the function $f_i(\mathbf{z})$ at $\mathbf{z} = \mathbf{z}_{k,i}$, for all $k \geq 0$. The step-size, α, is a positive number within a certain interval. We will explicitly show the range of α in Section 3. For any agent i, it is initiated with an arbitrary vector, $\mathbf{x}_{0,i}$, and with $\mathbf{w}_{0,i} = \nabla f_i(\mathbf{z}_{0,i})$ and $y_{0,i} = 1$. We note that the implementation of Eq. (2) needs each agent to at least have the knowledge of its out-neighbors degree. See [13–16, 20, 21, 23, 26] for the similar assumptions.

To simplify the analysis, we assume from now on that all sequences updated by Eq. (2) have only one dimension, i.e., $p = 1$; thus $x_{k,i}$, $y_{k,i}$, $w_{k,i}$, $z_{k,i} \in \mathbb{R}, \forall i, k$. For $\mathbf{x}_{k,i}$, $\mathbf{w}_{k,i}$, $\mathbf{z}_{k,i} \in \mathbb{R}^p$ being p-dimensional vectors, the proof is the same for every dimension by applying the results to each coordinate. Therefore, assuming $p = 1$ is without the loss of generality. We next write Eq. (2) in a matrix form. Define $\mathbf{x}_k$, $\mathbf{y}_k$, $\mathbf{w}_k$, $\mathbf{z}_k$, $\nabla \mathbf{f}_k \in \mathbb{R}^n$ as $\mathbf{x}_k = [x_{k,1}, \cdots, x_{k,n}]^\top$, $\mathbf{y}_k = [y_{k,1}, \cdots, y_{k,n}]^\top$, $\mathbf{w}_k = [w_{k,1}, \cdots, w_{k,n}]^\top$, $\mathbf{z}_k = [z_{k,1}, \cdots, z_{k,n}]^\top$, and $\nabla \mathbf{f}_k = [\nabla f_1(z_{k,1}), \cdots, \nabla f_n(z_{k,n})]^\top$. Let $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$ be the collection of weights a_{ij}. It is clear that A is a column-stochastic matrix. Define a diagonal matrix, $Y_k \in \mathbb{R}^{n \times n}$, for each k, as follows:

$$Y_k = \text{diag}\left(\mathbf{y}_k\right). \tag{3}$$

Given that the graph, $\mathcal{G}$, is strongly-connected and the corresponding weighting matrix, A, is non-negative, it follows that Y_k is invertible for any k. Then, we can write Eq. (2) in the matrix form equivalently as follows:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k - \alpha \mathbf{w}_k, \qquad \mathbf{y}_{k+1} = A\mathbf{y}_k, \tag{4a}$$

$$\mathbf{z}_{k+1} = Y_{k+1}^{-1}\mathbf{x}_{k+1}, \quad \mathbf{w}_{k+1} = A\mathbf{w}_k + \nabla \mathbf{f}_{k+1} - \nabla \mathbf{f}_k, \tag{4b}$$

where similarly we have the initial condition $\mathbf{w}_0 = \nabla \mathbf{f}_0$.

Based on Eq. (4), we now give an intuitive interpretation on the convergence of the algorithm to the optimal solution. By combining Eqs. (4a) and (4b), we obtain that

$$\mathbf{x}_{k+1} = 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left[\nabla \mathbf{f}_k - \nabla \mathbf{f}_{k-1}\right]. \tag{5}$$

Assume that the sequences generated by Eq. (4) converge to their limits (not necessarily true), denoted by $\mathbf{x}_\infty$, $\mathbf{y}_\infty$, $\mathbf{w}_\infty$, $\mathbf{z}_\infty$, $\nabla \mathbf{f}_\infty$, respectively. It follows from Eq. (5) that

$$\mathbf{x}_\infty = 2A\mathbf{x}_\infty - A^2\mathbf{x}_\infty - \alpha\left[\nabla \mathbf{f}_\infty - \nabla \mathbf{f}_\infty\right], \tag{6}$$

which implies that $(I_n - A)^2\mathbf{x}_\infty = \mathbf{0}_n$, or $\mathbf{x}_\infty \in \text{span}\{\mathbf{y}_\infty\}$, considering that $\mathbf{y}_\infty = A\mathbf{y}_\infty$. Therefore, we obtain that

$$\mathbf{z}_\infty = Y_\infty^{-1}\mathbf{x}_\infty \in \text{span}\{\mathbf{1}_n\}, \tag{7}$$

where the consensus is reached. By summing up Eq. (5) over k from 0 to ∞, we obtain that

$$\mathbf{x}_\infty = A\mathbf{x}_\infty + \sum_{r=1}^{\infty}(A - I_n)\mathbf{x}_r - \sum_{r=0}^{\infty}(A^2 - A)\mathbf{x}_r - \alpha \nabla \mathbf{f}_\infty.$$

Noting that $\mathbf{x}_\infty = A\mathbf{x}_\infty$ showed above, it follows

$$\alpha \nabla \mathbf{f}_\infty = \sum_{r=1}^{\infty}(A - I_n)\mathbf{x}_r - \sum_{r=0}^{\infty}(A^2 - A)\mathbf{x}_r.$$

Therefore, we obtain that

$$\alpha \mathbf{1}_n^\top \nabla \mathbf{f}_\infty = \mathbf{1}_n^\top (A - I_n)\sum_{r=1}^{\infty}\mathbf{x}_r - \mathbf{1}_n^\top (A^2 - A)\sum_{r=0}^{\infty}\mathbf{x}_r = 0,$$

which is the optimality condition of Problem P1. To conclude, if we assume the sequences updated in Eq. (4) have limits, $\mathbf{x}_\infty$, $\mathbf{y}_\infty$, $\mathbf{w}_\infty$, $\mathbf{z}_\infty$, $\nabla \mathbf{f}_\infty$, we have the fact that $\mathbf{z}_\infty$ achieves consensus and reaches the optimal solution of Problem P1. This reveals the convergence of the algorithm.

3. ASSUMPTIONS AND MAIN RESULT

With appropriate assumptions, our main result states that the proposed algorithm converges to the optimal solution of Problem P1 linearly. In this paper, we assume that the graph, $\mathcal{G}$, is strongly-connected; each local function, $f_i(z)$, is convex and differentiable, and the optimal solution, f^*, of Problem P1 and the corresponding optimal value, z^*, exists. Besides the above assumptions, we formally present the following assumptions.

Assumption A1. *Each private function, f_i, is differentiable and strongly-convex, and the gradient is Lipschitz continuous, i.e., for any i and $z_1, z_2 \in \mathbb{R}$,*

(a) there exists a positive constant l such that,

$$\|\nabla f_i(z_1) - \nabla f_i(z_2)\| \leq l\|z_1 - z_2\|;$$

(b) there exists a positive constant s such that,

$$s\|z_1 - z_2\|^2 \leq \langle \nabla f_i(z_1) - \nabla f_i(z_2), z_1 - z_2\rangle.$$

With these assumptions, we are able to present the convergence result, the representation of which are based on the following notations. Based on earlier notations, $\mathbf{x}_k$, $\mathbf{w}_k$, and $\nabla \mathbf{f}_k$, we further define $\overline{\mathbf{x}}_k = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\mathbf{x}_k$, $\overline{\mathbf{w}}_k = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\mathbf{w}_k$, $\mathbf{z}^* = z^*\mathbf{1}_n$, $\mathbf{g}_k = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\nabla \mathbf{f}_k$, $\mathbf{h}_k \in \mathbb{R}^n = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\nabla \mathbf{f}(\overline{\mathbf{x}}_k)$, where $\nabla \mathbf{f}(\overline{\mathbf{x}}_k) = [\nabla f_1(\frac{1}{n}\mathbf{1}_n^\top\mathbf{x}_k), ..., \nabla f_n(\frac{1}{n}\mathbf{1}_n^\top\mathbf{x}_k)]^\top$. We denote some constants:$\tau = \|A - I_n\|, \epsilon = \|I_n - A_\infty\|, \eta = \max(|1 - \alpha l|, |1 - \alpha s|)$, where $A_\infty = \lim_{k\to\infty} A^k$ is the limit of A. Let Y_∞ be the limit of Y_k in Eq. (3),

$$Y_\infty = \lim_{k\to\infty} Y_k, \tag{8}$$

and y and y_- be the maximum of $\|Y_k\|$ and $\|Y_k^{-1}\|$ over k, respectively: $y = \max_k \|Y_k\|, \quad y_- = \max_k \left\|Y_k^{-1}\right\|$. Moreover, we define two constants σ and γ_1 in the following two lemmas.

Lemma 1. *(Nedic et al. [13]) Consider Y_k, generated from the column-stochastic matrix, A, and its limit Y_∞. There exist $0 < \gamma_1 < 1$ and $0 < T < \infty$ such that for all k*

$$\|Y_k - Y_\infty\| \leq T\gamma_1^k. \tag{9}$$

Lemma 2. *(Olshevsky et al. [27]) Consider Y_∞ in Eq. (8), and A the weighting matrix used in Eq. (4). For any $\mathbf{a} \in \mathbb{R}^n$, define $\overline{\mathbf{a}} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\mathbf{a}$. There exists $0 < \sigma < 1$ such that*

$$\|A\mathbf{a} - Y_\infty\overline{\mathbf{a}}\| \leq \sigma \|\mathbf{a} - Y_\infty\overline{\mathbf{a}}\|. \tag{10}$$

We finally denote $\mathbf{t}_k, \mathbf{s}_k \in \mathbb{R}^3$, and G, $H_k \in \mathbb{R}^{3\times3}, \forall k$:

$$\mathbf{t}_k = \begin{bmatrix} \|\mathbf{x}_k - Y_\infty\overline{\mathbf{x}}_k\| \\ \|\overline{\mathbf{x}}_k - \mathbf{z}^*\| \\ \|\mathbf{w}_k - Y_\infty\mathbf{g}_k\| \end{bmatrix}, \mathbf{s}_k = \begin{bmatrix} \|\mathbf{x}_k\| \\ 0 \\ 0 \end{bmatrix},$$

$$G = \begin{bmatrix} \sigma & 0 & \alpha \\ \alpha(ly_-) & \eta & 0 \\ \epsilon l\tau y_- + \alpha(\epsilon l^2 yy_-^2) & \alpha(\epsilon l^2 yy_-) & \sigma + \alpha(\epsilon ly_-) \end{bmatrix},$$

$$H_k = \begin{bmatrix} 0 & 0 & 0 \\ \alpha ly_- T\gamma_1^k & 0 & 0 \\ (\alpha ly + 2)\epsilon ly_-^2 T\gamma_1^k & 0 & 0 \end{bmatrix}, \tag{11}$$

We now state the key relation in this paper, the proof of which appears in [28].

Theorem 1. *The following inequality holds for all $k \geq 1$,*

$$\mathbf{t}_k \leq G\mathbf{t}_{k-1} + H_{k-1}\mathbf{s}_{k-1}. \tag{12}$$

Note that Eq. (12) provides a linear iterative relation between $\mathbf{t}_k$ and $\mathbf{t}_{k-1}$ with matrix G and H_k. Thus, the convergence of $\mathbf{t}_k$ is fully determined by G and H_k. More specifically, if we want to prove a linear convergence rate of $\|\mathbf{t}_k\|$ to zero, it is sufficient to show that $\rho(G) < 1$ and the linear decaying of $\|H_k\|$. In Lemma 3, we first show that with appropriate step-size, $\rho(G) < 1$. Following Lemma 3, we show the linear convergence of $\|G^k\|$ and $\|H_k\|$ in Lemma 4.

Lemma 3. *Consider the matrix, G_α, defined in Eq. (11) as a function of the step-size, α. It follows that $\rho(G_\alpha) < 1$ if the step-size, $\alpha \in (0, \overline{\alpha})$, where*

$$\overline{\alpha} = \frac{\sqrt{(\epsilon\tau s)^2 + 4\epsilon y(l+s)s(1-\sigma)^2} - \epsilon\tau s}{2\epsilon lyy_-(l+s)}. \tag{13}$$

Proof. It is easy to verify that $\overline{\alpha} \leq \frac{\sqrt{4\epsilon y(l+s)s(1-\sigma)^2}}{2\epsilon lyy_-(l+s)} < \frac{1}{l}$. As a result, we have $\eta = 1 - \alpha s$. When $\alpha = 0$, we have that

$$G_0 = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & 1 & 0 \\ \epsilon l\tau y_- & 0 & \sigma \end{bmatrix}, \tag{14}$$

whose eigenvalues are σ and 1. Therefore, $\rho(G_0) = 1$. We now consider how the eigenvalue 1 is changed if we slightly increase α from 0. We denote $\mathcal{P}_{\mathcal{G}_\alpha}(q) = \det(qI_n - G_\alpha)$ the characteristic polynomial of G_α. By letting $\det(qI_n - G_\alpha) = 0$, we get the following equation,

$$\begin{aligned}((q-\sigma)^2 - \alpha\epsilon ly_-(q-\sigma))(q - 1 + \alpha s) - \alpha^3 l^3\epsilon yy_-^2 \\ -\alpha(q - 1 + \alpha s)(\epsilon l\tau y_- + \alpha(\epsilon l^2 yy_-^2)) = 0.\end{aligned} \tag{15}$$

Since we have already shown that 1 is one of the eigenvalues of G_0, Eq. (15) is valid when $q = 1$ and $\alpha = 0$. Take the derivative on both sides of Eq. (15), and let $q = 1$ and $\alpha = 0$, we obtain that $\frac{dq}{d\alpha}|_{\alpha=0,q=1} = -s < 0$. This is saying that when α increases from 0 slightly, $\rho(G_\alpha)$ will decrease first.

We now calculate all possible values of α for $\lambda(G_\alpha) = 1$. Let $q = 1$ in Eq. (15), and solve the step-size, α, we obtain that, $\alpha_1 = 0$, $\alpha_2 < 0$, and

$$\alpha_3 = \overline{\alpha} = \frac{\sqrt{(\epsilon\tau s)^2 + 4\epsilon y(l+s)s(1-\sigma)^2} - \epsilon\tau s}{2\epsilon lyy_-(l+s)}.$$

Since α has no other value for $\lambda(G_\alpha) = 1$, we know that $\lambda(G_\alpha) < 1$ for $\alpha \in (0, \overline{\alpha})$ by considering the fact that eigenvalues are continuous functions of matrix. □

Lemma 4. *With the step-size, $\alpha \in (0, \overline{\alpha})$, where $\overline{\alpha}$ is defined in Eq. (13), the following statements hold for all k,*

(a) there exist $0 < \gamma_1 < 1$ and $0 < \Gamma_1 < \infty$, where γ_1 is defined in Eq. (9), such that $\|H_k\| = \Gamma_1\gamma_1^k$;

(b) there exist $0 < \gamma_2 < 1$ and $0 < \Gamma_2 < \infty$, such that $\left\|G^k\right\| \leq \Gamma_2\gamma_2^k$;

(c) there exist $\gamma = \max\{\gamma_1, \gamma_2\}$ and $\Gamma = \Gamma_1\Gamma_2/\gamma$, such that for all $0 \leq r \leq k$, $\left\|G^{k-r-1}H_r\right\| \leq \Gamma\gamma^k$.

We now present the main result of this paper in Theorem 2, which shows the linear convergence rate of the algorithm.

Theorem 2. *With $\alpha \in (0, \overline{\alpha})$, where $\overline{\alpha}$ is defined in Eq. (13), the sequence, $\{\mathbf{z}_k\}$, generated by Eq. (4), converges exactly to the optimal solution, $\mathbf{z}^*$, at a linear rate, i.e., there exist some bounded constants $M > 0$ and $\gamma < \mu < 1$, where γ is used in Lemma 4(c), such that for any k,*

$$\|\mathbf{z}_k - \mathbf{z}^*\| \leq M\mu^k. \tag{16}$$

Proof. We write Eq. (12) recursively, which results

$$\mathbf{t}_k \leq G^k \mathbf{t}_0 + \sum_{r=0}^{k-1} G^{k-r-1} H_r \mathbf{s}_r. \tag{17}$$

By taking the norm on both sides of Eq. (17), and considering Lemma 4, we obtain that

$$\|\mathbf{t}_k\| \leq \Gamma_2 \gamma_2^k \|\mathbf{t}_0\| + \sum_{r=0}^{k-1} \Gamma \gamma^k \|\mathbf{s}_r\|, \tag{18}$$

in which we can bound $\|\mathbf{s}_r\|$ as

$$\begin{aligned}\|\mathbf{s}_r\| \leq& \|\mathbf{x}_r - Y_\infty \overline{\mathbf{x}}_r\| + \|Y_\infty\| \|\overline{\mathbf{x}}_r - \mathbf{z}^*\| + \|Y_\infty\| \|\mathbf{z}^*\| \\ \leq& (1+y) \|\mathbf{t}_r\| + y \|\mathbf{z}^*\|. \end{aligned} \tag{19}$$

Therefore, we have that for all k

$$\|\mathbf{t}_k\| \leq \Big(\Gamma_2 \|\mathbf{t}_0\| + \Gamma(1+y) \sum_{r=0}^{k-1} \|\mathbf{t}_r\| + \Gamma y k \|\mathbf{z}^*\| \Big) \gamma^k. \tag{20}$$

Our first step is to show that $\|\mathbf{t}_k\|$ is bounded for all k. It is true that there exists some bounded $K > 0$ such that for all $k > K$ it satisfies that

$$(\Gamma_2 + \Gamma(1+2y)k)\, \gamma^k \leq 1. \tag{21}$$

Define $\Phi = \max_{0 \leq k \leq K} (\|\mathbf{t}_k\|, \|\mathbf{z}_*\|)$, which is bounded since K is bounded. It is true that $\|\mathbf{t}_k\| \leq \Phi$ for $0 \leq k \leq K$. Consider the case when $k = K+1$. By combining Eqs. (20) and (21), we have that

$$\|\mathbf{t}_{K+1}\| \leq \Phi \Big(\Gamma_2 + \Gamma(1+2y)(K+1) \Big) \gamma^{K+1} \leq \Phi. \tag{22}$$

We repeat the procedures to show that $\|\mathbf{t}_k\| \leq \Phi$ for all k.

The next step is to show that $\|\mathbf{t}_k\|$ decays linearly. For any μ satisfying $\gamma < \mu < 1$, there exist a constant U such that $(\frac{\mu}{\gamma})^k > \frac{k}{U}$ for all k. Therefore, by bounding all $\|\mathbf{t}_k\|$ and $\|\mathbf{z}^*\|$ by Φ in Eq. (20), we obtain that for all k

$$\begin{aligned}\|\mathbf{t}_k\| \leq& \Phi \left(\Gamma_2 + \Gamma(1+2y) U \frac{k}{U} \left(\frac{\gamma}{\mu} \right)^k \right) \mu^k \\ \leq& \Phi \Big(\Gamma_2 + \Gamma(1+2y) U \Big) \mu^k. \end{aligned} \tag{23}$$

It follows that $\|\mathbf{z}_k - \mathbf{z}^*\|$ and $\|\mathbf{t}_k\|$ satisfy the relation that

$$\begin{aligned}\|\mathbf{z}_k - \mathbf{z}^*\| \leq& \left\| Y_k^{-1} \mathbf{x}_k - Y_k^{-1} Y_\infty \overline{\mathbf{x}}_k \right\| + \left\| Y_k^{-1} Y_\infty \mathbf{z}^* - \mathbf{z}^* \right\| \\ &+ \left\| Y_k^{-1} Y_\infty \overline{\mathbf{x}}_k - Y_k^{-1} Y_\infty \mathbf{z}^* \right\| \\ \leq& y_- (1+y) \|\mathbf{t}_k\| + y_- T \gamma_1^k \|\mathbf{z}^*\|, \end{aligned} \tag{24}$$

where in the second inequality we use the relation $\|Y_k^{-1} Y_\infty - I_n\| \leq \|Y_k^{-1}\| \|Y_\infty - Y_k\| \leq y_- T \gamma_1^k$ achieved from Eq. (9). By combining Eqs. (23) and (24), we obtain that

$$\|\mathbf{z}_k - \mathbf{z}^*\| \leq y_- \Phi \left[(1+y)(\Gamma_2 + \Gamma(1+2y)U) + T \right] \mu^k.$$

The desired result is obtained by letting $M = y_- \Phi[(1+y)(\Gamma_2 + \Gamma(1+2y)U) + T]$. □

4. NUMERICAL EXPERIMENTS

In this section, we compare the performances of algorithms for distributed optimization over directed graphs. Our numerical experiments are based on the distributed logistic regression problem over a directed graph:

$$\mathbf{z}^* = \underset{\mathbf{z} \in \mathbb{R}^p}{\operatorname{argmin}} \frac{\beta}{2} \|\mathbf{z}\|^2 + \sum_{i=1}^{n} \sum_{j=1}^{m_i} \ln \left[1 + \exp\left(- \left(\mathbf{c}_{ij}^\top \mathbf{z} \right) b_{ij} \right) \right],$$

where for any agent i, it is accessible to m_i training examples, $(\mathbf{c}_{ij}, b_{ij}) \in \mathbb{R}^p \times \{-1, +1\}$, where $\mathbf{c}_{ij}$ includes the p features of the jth training example of agent i, and b_{ij} is the corresponding label. In our setting, we have $n = 10$, $m_i = 10$, for all i, and $p = 3$. The first simulation, see Fig. 1 (Left), compares the convergence rates between the proposed algorithm and other methods that designed for directed graphs. We apply the same local degree weighting strategy to all methods. The step-size used in SP [13], D-DSD [20], and WB-DSD [23] is $\alpha_k = 1/\sqrt{k}$. The constant step-size used in DEXTRA [26] and our algorithm is $\alpha = 1$. It can be found that the proposed algorithm and DEXTRA has a fast linear convergence rate, while other methods are sub-linear. The second experiment compares the proposed algorithm and DEXTRA in terms of their step-size ranges. We stick to the same local degree weighting strategy for both algorithms. It is shown in Fig. 1 (Right) that the greatest lower bound of DEXTRA is round $\underline{\alpha} = 0.2$. In contrast, our algorithm can pick whatever small values to ensure the convergence.

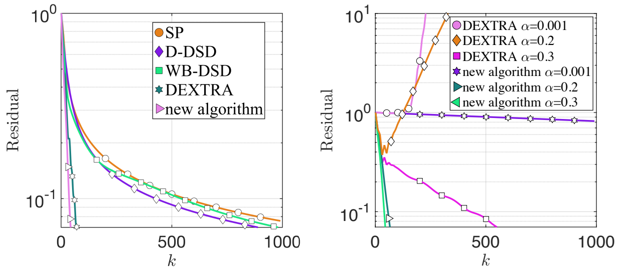

Fig. 1: (Left) Convergence for related algorithms over directed networks. (Right) Comparison with DEXTRA in terms of step-size ranges.

5. CONCLUSIONS

We focus on solving the distributed optimization problem over directed graphs. The proposed algorithm converges at a linear rate $O(\mu^k)$ for $0 < \mu < 1$ given the assumption that the objective functions are strongly-convex. Our algorithm supports a more realistic range of step-sizes, i.e., the greatest lower bound of step-size for the proposed algorithm is zero. This guarantees the convergence of our algorithm in the distributed implementation as long as agents picking some arbitrary small step-size.

References

[1] A. Ribeiro, “Optimal resource allocation in wireless communication and networking,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–19, 2012.

[2] T. M. Kim, H. J. Yang, and A. J. Paulraj, “Distributed sum-rate optimization for full-duplex mimo system under limited dynamic range,” *IEEE Signal Processing Letters*, vol. 20, no. 6, pp. 555–558, June 2013.

[3] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, April 2004, pp. 20–27.

[4] Q. Ling and Z. Tian, “Decentralized sparse signal recovery for compressive sleeping wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3816–3827, July 2010.

[5] 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, Sep. 2014.

[6] J. B. Predd, S. B. Kulkarni, and H. V. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, July 2006.

[7] 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, Apr. 2010.

[8] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” *arXiv preprint arXiv:1310.7063*, 2013.

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

[10] 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.

[11] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *arXiv preprint arXiv:1605.07112*, 2016.

[12] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, “D-ADMM: A communication-efficient distributed algorithm for separable optimization,” *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, May 2013.

[13] A. Nedic and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2014.

[14] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, “Push-sum distributed dual averaging for convex optimization,” in *51st IEEE Annual Conference on Decision and Control*, Maui, Hawaii, Dec. 2012, pp. 5453–5458.

[15] K. I. Tsianos, *The role of the Network in Distributed Optimization Algorithms: Convergence Rates, Scalability, Communication/Computation Tradeoffs and Communication Delays*, Ph.D. thesis, Dept. Elect. Comp. Eng. McGill University, 2013.

[16] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, “Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning,” in *50th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Oct. 2012, pp. 1543–1550.

[17] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

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

[19] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, “Weighted gossip: Distributed averaging using non-doubly stochastic matrices,” in *IEEE International Symposium on Information Theory*, Jun. 2010, pp. 1753–1757.

[20] C. Xi, Q. Wu, and U. A. Khan, “Distributed gradient descent over directed graphs,” *arXiv preprint arXiv:1510.02146*, 2015.

[21] C. Xi and U. A. Khan, “Distributed subgradient projection algorithm over directed graphs,” *arXiv preprint arXiv:1602.00653*, 2016.

[22] K. Cai and H. Ishii, “Average consensus on general strongly connected digraphs,” *Automatica*, vol. 48, no. 11, pp. 2750 – 2761, 2012.

[23] A. Makhdoumi and A. Ozdaglar, “Graph balancing for distributed subgradient methods over directed graphs,” *to appear in 54th IEEE Annual Conference on Decision and Control*, 2015.

[24] L. Hooi-Tong, “On a class of directed graphswith an application to traffic-flow problems,” *Operations Research*, vol. 18, no. 1, pp. 87–94, 1970.

[25] A. Nedic and A. Olshevsky, “Distributed optimization of strongly convex functions on directed time-varying graphs,” in *IEEE Global Conference on Signal and Information Processing*, Dec. 2013, pp. 329–332.

[26] C. Xi and U. A. Khan, “On the linear convergence of distributed optimization over directed graphs,” *arXiv preprint arXiv:1510.02149*, 2015.

[27] A. Olshevsky and J. N. Tsitsiklis, “Convergence speed in distributed consensus and averaging,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[28] C. Xi and U. A. Khan, “ADD-OPT: Accelerated distributed directed optimization,” *arXiv preprint arXiv:1607.04757*, 2016.