

## EXTRAPUSH FOR CONVEX SMOOTH DECENTRALIZED OPTIMIZATION OVER DIRECTED NETWORKS\*

Jinshan Zeng

*College of Computer Information Engineering, Jiangxi Normal University, Nanchang,  
Jiangxi 330022, China*

*Email: jinshanzeng@jxnu.edu.cn*

Wotao Yin

*Department of Mathematics, University of California, Los Angeles, CA 90095, USA*

*Email: wotaoyin@math.ucla.edu*

### Abstract

In this note, we extend the algorithms Extra [13] and subgradient-push [10] to a new algorithm *ExtraPush* for consensus optimization with convex differentiable objective functions over a *directed* network. When the stationary distribution of the network can be computed in advance, we propose a simplified algorithm called *Normalized ExtraPush*. Just like Extra, both ExtraPush and Normalized ExtraPush can iterate with a fixed step size. But unlike Extra, they can take a column-stochastic mixing matrix, which is not necessarily doubly stochastic. Therefore, they remove the undirected-network restriction of Extra. Subgradient-push, while also works for *directed* networks, is slower on the same type of problem because it must use a sequence of diminishing step sizes.

We present preliminary analysis for ExtraPush under a bounded sequence assumption. For Normalized ExtraPush, we show that it naturally produces a bounded, linearly convergent sequence provided that the objective function is strongly convex.

In our numerical experiments, ExtraPush and Normalized ExtraPush performed similarly well. They are significantly faster than subgradient-push, even when we hand-optimize the step sizes for the latter.

*Mathematics subject classification:* 90C25, 90C30.

*Key words:* Decentralized optimization, Directed graph, Consensus, Non-doubly stochastic, Extra.

## 1. Introduction

We consider the following consensus optimization problem defined on a directed, strongly connected network of  $n$  agents:

$$\underset{x \in \mathbf{R}^p}{\text{minimize}} f(x) \triangleq \sum_{i=1}^n f_i(x), \quad (1.1)$$

where  $f_i$  is a proper, closed, convex, differentiable function only known to the agent  $i$ .

The model (1.1) finds applications in decentralized averaging, learning, estimation, and control. For a stationary network with *bi-directional* communication, the existing algorithms include the (sub)gradient methods [2, 5, 8, 9, 13, 19], and the primal-dual domain methods such as the decentralized alternating direction method of multipliers (DADMM) [11, 12].

---

\* Received November 12, 2015 / Revised version received June 2, 2016 / Accepted June 27, 2016 /  
Published online June 1, 2017 /

This note focuses on a *directed* network (with *directional* communication), where the research of decentralized optimization is pioneered by the works [15–17]. When communication is bi-directional, algorithms can use a symmetric and doubly-stochastic mixing matrix to obtain a consensual solution; however, once the communication is directional, the mixing matrix becomes generally asymmetric and only column-stochastic. Also consider the setting where each agent broadcasts its information to its neighbors, yet an agent may not receive the information from a neighbor. An agent can weigh its information (both from itself and received from its neighbors) so that the total weights add up to 1, but an agent cannot ensure that its broadcasted information receives weights that precisely add up to exactly 1. Therefore, only each column of the mixing matrix sums to 1. In the column-stochastic setting, the push-sum protocol [6] can be used to obtain a stationary distribution for the mixing matrix.

In the symmetric and doubly-stochastic setting, if the objective is Lipschitz-differentiable, the gradient-based algorithm Extra [13] converges at the rate of  $O(1/t)$ , where  $t$  is the iteration number. In the column-stochastic setting, the best rate is  $O(\ln t/\sqrt{t})$  from the subgradient-based algorithm [10]. We address the open question of how to take advantage of the gradient of a Lipschitz-differentiable objective. We make an attempt in this note to combine ideas in [10,13] and present our preliminary results.

Specifically, we propose *ExtraPush*, which is a two-step iteration like Extra and incorporates the push-sum protocol. At each iteration, the Extra variables are approximately normalized by the current push-sum variables. When the stationary distribution of the network can be easily computed, we propose to first apply the push-sum protocol to obtain the stationary distribution and then run the two-step iteration Normalized ExtraPush. At each iteration, its running variables are normalized by the stationary distribution.

Our algorithms are essentially the same as found in the recent work by Xi and Khan [18]. They attempted to prove convergence for a strongly convex objective function. They noticed that a certain matrix that is important to the analysis (as a part of their convergence metric) is positive semi-definite. Our analysis also uses this property. However, their analysis breaks down due to incorrect assumptions. More specifically, each function  $f_i$  is assumed in [18] to be strongly convex and also has a bounded and Lipschitz gradient (i.e., its gradient is bounded and Lipschitz continuous). However, no function can satisfy these assumptions simultaneously since gradients of a strongly convex are strictly increasing and unbounded.

It is worth noting that our algorithm can be applied to a time-varying directed network after a straightforward modification; our convergence proof, however, will need a significant change.

The rest of this note is organized as follows. Section 2 introduces the problem setup and preliminaries. Section 3 develops ExtraPush and Normalized ExtraPush. Section 4 establishes the optimality conditions for ExtraPush and shows its convergence under the boundedness assumption. Section 5 assumes that the objective is strongly convex and shows that Normalized ExtraPush produces a bounded sequence that converges linearly. Section 6 presents our numerical simulation results. We conclude this paper in Section 7.

**Notation:** Let  $\mathbf{I}_n$  denote an identity matrix with the size  $n \times n$ , and  $\mathbf{1}_{n \times p} \in \mathbf{R}^{n \times p}$  denote the *matrix* with all entries equal to 1. We also use  $\mathbf{1}_n \in \mathbf{R}^n$  as a vector of all 1's. For any *vector*  $x$ , we let  $x_i$  denote its  $i$ th component and  $\mathbf{diag}(x)$  denote the diagonal matrix generated by  $x$ . For any matrix  $X$ ,  $X^T$  denotes its transpose,  $X_{ij}$  denotes its  $(i, j)$ th component, and  $\|X\| \triangleq \sqrt{\langle X, X \rangle} = \sqrt{\sum_{i,j} X_{ij}^2}$  denotes its Frobenius norm. The largest and smallest eigenvalues of matrix  $X$  are denoted as  $\lambda_{\max}(X)$  and  $\lambda_{\min}(X)$ , respectively. For any matrix  $B \in \mathbf{R}^{m \times n}$ ,  $\mathbf{null}(B) \triangleq \{x \in \mathbf{R}^n | Bx = 0\}$  is the null space of  $B$ . Given a matrix  $B \in \mathbf{R}^{m \times n}$ , by  $Z \in \mathbf{null}(B)$ ,

we mean that each column of  $Z$  lies in  $\mathbf{null}(B)$ . The smallest *nonzero* eigenvalue of a symmetric positive semidefinite matrix  $X \neq \mathbf{0}$  is denoted as  $\tilde{\lambda}_{\min}(X)$ , which is strictly positive. For any positive semidefinite matrix  $G \in \mathbf{R}^{n \times n}$  (not necessarily symmetric in this paper), we use the notion  $\|X\|_G^2 \triangleq \langle X, GX \rangle$  for a matrix  $X \in \mathbf{R}^{n \times p}$ .

## 2. Problem Reformulation

### 2.1. Network

Consider a *directed* network  $\mathcal{G} = \{V, E\}$ , where  $V$  is the vertex set and  $E$  is the edge set. Any edge  $(i, j) \in E$  represents a directed arc from node  $i$  to node  $j$ . The sets of in-neighbors and out-neighbors of node  $i$  are

$$\mathcal{N}_i^{\text{in}} \triangleq \{j : (j, i) \in E\} \cup \{i\}, \quad \mathcal{N}_i^{\text{out}} \triangleq \{j : (i, j) \in E\} \cup \{i\},$$

respectively. Let  $d_i \triangleq |\mathcal{N}_i^{\text{out}}|$  be the out-degree of node  $i$ . In  $\mathcal{G}$ , each node  $i$  can only send information to its out-neighbors, *not* vice versa.

To illustrate a mixing matrix for a directed network, consider  $A \in \mathbf{R}^{n \times n}$  where

$$\begin{cases} A_{ij} > 0, & \text{if } j \in \mathcal{N}_i^{\text{in}} \\ A_{ij} = 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

The entries  $A_{ij}$  satisfy that, for each node  $j$ ,  $\sum_{i \in V} A_{ij} = 1$ . An example is the following mixing matrix

$$A_{ij} = \begin{cases} 1/d_j, & \text{if } j \in \mathcal{N}_i^{\text{in}} \\ 0, & \text{otherwise,} \end{cases} \quad (2.2)$$

$i, j = 1, \dots, n$ , which is used in the subgradient-push method [10]. See Fig. 2.1 for a directed graph  $\mathcal{G}$  and an example of its mixing matrix  $A$ . The matrix  $A$  is column stochastic and asymmetric in general.

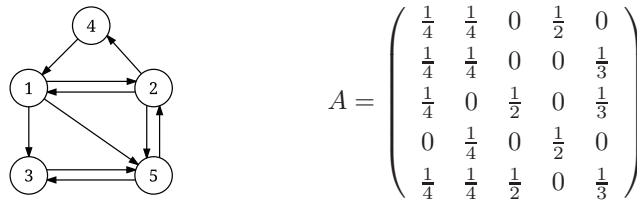


Fig. 2.1. A directed graph  $\mathcal{G}$  (left) and its mixing matrix  $A$  (right).

**Assumption 1.** *The graph  $\mathcal{G}$  is strongly connected.*

**Property 1.** *Under Assumption 1, the followings hold (parts (i) and (iv) are results in [10, Corollary 2]):*

(i) Let  $A^t = \overbrace{A \times A \cdots A}^t$  for any  $t \in \mathbf{N}$ . Then

$$A^t \rightarrow \phi \mathbf{1}_n^T \text{ geometrically fast as } t \rightarrow \infty, \quad (2.3)$$

for some stationary distribution vector  $\phi$ , i.e.,  $\phi_i \geq 0$  and  $\sum_i^n \phi_i = 1$ .

(ii)  $\text{null}(\mathbf{I}_n - \phi \mathbf{1}_n^T) = \text{null}(\mathbf{I}_n - A)$ .

(iii)  $A\phi = \phi$ .

(iv) The quantity  $\xi \triangleq \inf_t \min_{1 \leq i \leq n} (A^t \mathbf{1}_n)_i \geq \frac{1}{n^n} > 0$ .

*Proof.* Part (iii) is obvious from (ii) since  $\phi \in \text{null}(\mathbf{I}_n - \phi \mathbf{1}_n^T)$  and  $\sum_i \phi_i = 1$ . Next, we show part (ii). First, let  $z \in \text{null}(\mathbf{I}_n - \phi \mathbf{1}_n^T)$ , which means  $z = \phi \mathbf{1}_n^T z$  and thus  $Az = A\phi \mathbf{1}_n^T z$ . By (2.3), it is obvious that  $A\phi \mathbf{1}_n^T = \phi \mathbf{1}_n^T$ . Therefore,  $Az = \phi \mathbf{1}_n^T z = z$  and hence  $\text{null}(\mathbf{I}_n - \phi \mathbf{1}_n^T) \subseteq \text{null}(\mathbf{I}_n - A)$ . On the other hand, any  $z \in \text{null}(\mathbf{I}_n - A)$ , equivalently,  $z = Az$ , obeys  $z = A^t z$  for any  $t \geq 1$ . Letting  $t \rightarrow \infty$ , it holds that  $z = \phi \mathbf{1}_n^T z$ , that is,  $z \in \text{null}(\mathbf{I}_n - \phi \mathbf{1}_n^T)$ . Therefore, part (ii) holds.  $\square$

**2.2. Problem Given in the Matrix Notation**

Let  $x_{(i)} \in \mathbf{R}^p$  denote the *local copy* of  $x$  at node  $i$ , and  $x_{(i)}^t$  denote its value at the  $t$ th iteration. Throughout the note, we use the following equivalent form of the problem (1.1) using local copies of the variable  $x$ :

$$\text{minimize}_{\mathbf{x}} \mathbf{1}_n^T \mathbf{f}(\mathbf{x}) \triangleq \sum_{i=1}^n f_i(x_{(i)}), \quad \text{subject to } x_{(i)} = x_{(j)}, \forall i, j \in E, \tag{2.4}$$

where  $\mathbf{1}_n \in \mathbf{R}^n$  denotes the vector with all its entries equal to 1,

$$\mathbf{x} \triangleq \begin{pmatrix} - & x_{(1)}^T & - \\ - & x_{(2)}^T & - \\ & \vdots & \\ - & x_{(n)}^T & - \end{pmatrix} \in \mathbf{R}^{n \times p}, \quad \mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} f_1(x_{(1)}) \\ f_2(x_{(2)}) \\ \vdots \\ f_n(x_{(n)}) \end{pmatrix} \in \mathbf{R}^n.$$

In addition, the gradient of  $\mathbf{f}(\mathbf{x})$  is

$$\nabla \mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} - & \nabla f_1(x_{(1)})^T & - \\ - & \nabla f_2(x_{(2)})^T & - \\ & \vdots & \\ - & \nabla f_n(x_{(n)})^T & - \end{pmatrix} \in \mathbf{R}^{n \times p}.$$

The  $i$ th rows of the above matrices  $\mathbf{x}$  and  $\nabla \mathbf{f}(\mathbf{x})$ , and vector  $\mathbf{f}(\mathbf{x})$ , correspond to agent  $i$ . For simplicity, one can treat  $p = 1$  throughout this paper.

For a vector  $\bar{x} \in \mathbf{R}^n$ , let  $\bar{x}^{\text{ave}} \triangleq \frac{1}{n}(\sum_{i=1}^n \bar{x}_i) \in \mathbf{R}$ . A special case of (2.4) is the well-known average consensus problem, where  $f_i(x_{(i)}) = \frac{1}{2}(x_{(i)} - \bar{x}_i)^2$  for each node  $i$  and the solution is  $x_{(i)} = \bar{x}^{\text{ave}}$  for all  $i$ .

**3. Proposed Algorithms**

**3.1. Reviews of Extra and subgradient-push**

Extra [13] is a ‘‘two-step’’ iterative algorithm for solving (2.4) over an undirected network. Let  $W \in \mathbf{R}^{n \times n}$  be a symmetric and doubly stochastic mixing matrix, and  $\bar{W} \triangleq \frac{\mathbf{I}_n + W}{2}$ . The Extra iteration is

$$\mathbf{x}^{t+2} = (\mathbf{I}_n + W)\mathbf{x}^{t+1} - \bar{W}\mathbf{x}^t - \alpha(\nabla \mathbf{f}(\mathbf{x}^{t+1}) - \nabla \mathbf{f}(\mathbf{x}^t)), \quad t = 0, 1, \dots, \tag{3.1}$$

which starts with  $\mathbf{x}^1 = \mathbf{x}^0 - \alpha \nabla \mathbf{f}(\mathbf{x}^0)$ , any  $\mathbf{x}^0 \in \mathbf{R}^{n \times p}$  and uses a properly bounded step size  $\alpha > 0$ . Extra converges at a rate  $o(\frac{1}{t})$ , measured by the best running violation to the first-order optimality condition, provided that  $f$  is Lipschitz differentiable. It improves to a linear rate of convergence if  $f$  is also (restricted) strongly convex.

The subgradient-push algorithm [10] is proposed to solve the decentralized optimization problem (1.1) over a time-varying directed graph. It is a combination of the subgradient method and the push-sum protocol [1, 6, 7]. Let  $A(t)$  be the mixing matrix at the  $t$ th iteration as defined in (2.2) for a time-varying directed network. The iteration of subgradient-push is

$$\begin{cases} \mathbf{z}^{t+1} = A(t)\mathbf{z}^t - \alpha_t \nabla \mathbf{f}(\mathbf{x}^t), \\ \mathbf{w}^{t+1} = A(t)\mathbf{w}^t, \\ \mathbf{x}^{t+1} = \mathbf{diag}(\mathbf{w}^{t+1})^{-1} \mathbf{z}^{t+1}, \end{cases} \quad (3.2)$$

where  $\alpha_t > 0$  is the step size at the  $t$ th iteration that decays as follows:  $\sum_{t=1}^{\infty} \alpha_t = \infty$ ,  $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$ , and  $\alpha_t \leq \alpha_s$  for all  $t > s \geq 1$ . It is shown in [10] that the convergence rate of subgradient-push algorithm is  $O(\ln t / \sqrt{t})$ .

### 3.2. Proposed: ExtraPush

ExtraPush combines the above two algorithms. Specifically, set arbitrary  $\mathbf{z}^0$  and  $\mathbf{w}^0 = \mathbf{1}_n$ ; set  $\mathbf{x}^0 = \mathbf{z}^0$ ; for  $t = 1$ , set  $\mathbf{w}^1 = A\mathbf{w}^0$ ,  $\mathbf{z}^1 = A\mathbf{z}^0 - \alpha \nabla \mathbf{f}(\mathbf{x}^0)$ , and  $\mathbf{x}^1 = \mathbf{diag}(\mathbf{w}^1)^{-1} \mathbf{z}^1$ . Letting  $\bar{A} \triangleq \frac{\mathbf{I}_n + A}{2}$ , for  $t = 2, 3, \dots$ , perform

$$\begin{cases} \mathbf{z}^t = (A + \mathbf{I}_n)\mathbf{z}^{t-1} - \bar{A}\mathbf{z}^{t-2} - \alpha(\nabla \mathbf{f}(\mathbf{x}^{t-1}) - \nabla \mathbf{f}(\mathbf{x}^{t-2})), \\ \mathbf{w}^t = A\mathbf{w}^{t-1}, \\ \mathbf{x}^t = \mathbf{diag}(\mathbf{w}^t)^{-1} \mathbf{z}^t. \end{cases} \quad (3.3)$$

By the structure of  $A$ , each node  $i$  broadcasts its  $z_{(i)}$  to its out-neighbors at each ExtraPush iteration. The step size  $\alpha > 0$  needs to be properly set. The iteration (3.3) of ExtraPush can be implemented at each agent  $i$  as follows:

$$\begin{cases} z_{(i)}^t = z_{(i)}^{t-1} + \sum_{j \in \mathcal{N}_i^{\text{in}}} A_{ij} z_{(j)}^{t-1} - \sum_{j \in \mathcal{N}_i^{\text{in}}} \bar{A}_{ij} z_{(j)}^{t-2} - \alpha(\nabla f_i(x_{(i)}^{t-1}) - \nabla f_i(x_{(i)}^{t-2})), \\ w_i^t = \sum_{j \in \mathcal{N}_i^{\text{in}}} A_{ij} w_j^{t-1}, \\ x_{(i)}^t = \frac{z_{(i)}^t}{w_i^t}, \end{cases}$$

where  $\bar{A}_{ij}$  is the  $(i, j)$ th component of  $\bar{A}$ , and  $w_i^t$  is the  $i$ th component of  $\mathbf{w}^t$ , for all  $i, j$ .

### 3.3. Proposed: Normalized ExtraPush

Normalized ExtraPush first computes the stationary distribution  $\phi$  of  $A$  and saves each  $\phi_i$  at node  $i$ . Next, in the main iteration, the  $\mathbf{w}$ -step from (3.3) is removed, and  $n \cdot \phi$  instead of  $\mathbf{w}^t$  is used to obtain  $\mathbf{x}^t$ . As such, the main iteration of Normalized ExtraPush simplifies (3.3). Letting,

$$D \triangleq n \mathbf{diag}(\phi).$$

the iteration of Normalized ExtraPush proceeds as follows: set arbitrary  $\mathbf{z}^0$  and  $\mathbf{x}^0 = D^{-1} \mathbf{z}^0$ ; for  $t = 1$ , set  $\mathbf{z}^1 = A\mathbf{z}^0 - \alpha \nabla \mathbf{f}(\mathbf{x}^0)$  and  $\mathbf{x}^1 = D^{-1} \mathbf{z}^1$ . For  $t = 2, 3, \dots$ , perform

$$\begin{cases} \mathbf{z}^t = (A + \mathbf{I}_n)\mathbf{z}^{t-1} - \bar{A}\mathbf{z}^{t-2} - \alpha(\nabla \mathbf{f}(\mathbf{x}^{t-1}) - \nabla \mathbf{f}(\mathbf{x}^{t-2})), \\ \mathbf{x}^t = D^{-1} \mathbf{z}^t. \end{cases} \quad (3.4)$$

At each agent  $i$ , the iterate (3.4) of Normalized ExtraPush can be implemented as follows:

$$\begin{cases} z_{(i)}^t = z_{(i)}^{t-1} + \sum_{j \in \mathcal{N}_i^{\text{in}}} A_{ij} z_{(j)}^{t-1} - \sum_{j \in \mathcal{N}_i^{\text{in}}} \bar{A}_{ij} z_{(j)}^{t-2} - \alpha(\nabla f_i(x_{(i)}^{t-1}) - \nabla f_i(x_{(i)}^{t-2})), \\ x_{(i)}^t = \frac{z_{(i)}^t}{n\phi_i}. \end{cases}$$

Next, we present two equivalent forms of Normalized ExtraPush. Letting  $\mathbf{f}_\phi(\mathbf{z}) \triangleq D\mathbf{f}(D^{-1}\mathbf{z})$ , we have  $\nabla \mathbf{f}_\phi(\mathbf{z}) = \nabla \mathbf{f}(D^{-1}\mathbf{z})$ . Substituting the  $\mathbf{x}$ -step of (3.4) into its  $\mathbf{z}$ -step yields the single-value iteration:

$$\mathbf{z}^t = (A + \mathbf{I}_n)\mathbf{z}^{t-1} - \bar{A}\mathbf{z}^{t-2} - \alpha(\nabla \mathbf{f}_\phi(\mathbf{z}^{t-1}) - \nabla \mathbf{f}_\phi(\mathbf{z}^{t-2})). \tag{3.5}$$

Upon stopping, one shall return  $\mathbf{x}^t = D^{-1}\mathbf{z}^t$ . The iteration (3.5) is nearly identical to the Extra iteration (3.1) except that (3.1) must use a doubly-stochastic matrix.

Letting  $A_\phi \triangleq D^{-1}AD$  and  $\bar{A}_\phi \triangleq \frac{1}{2}(\mathbf{I}_n + A_\phi)$ , which are row stochastic matrices, gives another equivalent form of Normalized ExtraPush

$$\mathbf{x}^t = (A_\phi + \mathbf{I}_n)\mathbf{x}^{t-1} - \bar{A}_\phi\mathbf{x}^{t-2} - \alpha D^{-1}(\nabla \mathbf{f}(\mathbf{x}^{t-1}) - \nabla \mathbf{f}(\mathbf{x}^{t-2})), \tag{3.6}$$

which, compared to the Extra iteration (3.1), has the extra diagonal matrix  $D^{-1}$ . Indeed, this iteration generalizes Extra to use row-stochastic matrices  $A_\phi$  and  $\bar{A}$ .

### 4. Preliminary Analysis of ExtraPush

In this section, we first develop the first-order optimality conditions for the problem (2.4) and then provide the convergence of ExtraPush under the boundedness assumption.

**Theorem 1. (first-order optimality conditions)** *Suppose that graph  $\mathcal{G}$  is strongly connected. Then  $\mathbf{x}^*$  is consensual and  $x_{(1)}^* \equiv x_{(2)}^* \equiv \dots \equiv x_{(n)}^*$  is an optimal solution of (1.1) if and only if, for some  $\alpha > 0$ , there exist  $\mathbf{z}^* \in \text{null}(\mathbf{I}_n - A)$  and  $\mathbf{y}^* \in \text{null}(\mathbf{1}_n^T)$  such that the following conditions hold*

$$\begin{cases} \mathbf{y}^* + \alpha \nabla \mathbf{f}(\mathbf{x}^*) = 0, \\ \mathbf{x}^* = D^{-1}\mathbf{z}^*. \end{cases} \tag{4.1}$$

(We let  $\mathcal{L}^*$  denote the set of triples  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*)$  satisfying the above conditions.)

*Proof.* Assume that  $\mathbf{x}^*$  is consensual and  $x_{(1)}^* \equiv x_{(2)}^* \equiv \dots \equiv x_{(n)}^*$  is optimal. Let  $\mathbf{z}^* = n\text{diag}(\phi)\mathbf{x}^* = n(\phi x_{(1)}^{*T})$ . Then  $\phi \mathbf{1}_n^T \mathbf{z}^* = \phi \mathbf{1}_n^T n \phi x_{(1)}^{*T} = n \phi x_{(1)}^{*T} = \mathbf{z}^*$ . It implies that  $\mathbf{z}^* \in \text{null}(\mathbf{I} - \phi \mathbf{1}_n^T)$ . By Property 1 (ii), it follows that  $\mathbf{z}^* \in \text{null}(\mathbf{I}_n - A)$ . Moreover, letting  $\mathbf{y}^* = -\alpha \nabla \mathbf{f}(\mathbf{x}^*)$ , it holds that  $\mathbf{1}_n^T \mathbf{y}^* = -\alpha \mathbf{1}_n^T \nabla \mathbf{f}(\mathbf{x}^*) = 0$ , that is,  $\mathbf{y}^* \in \text{null}(\mathbf{1}_n^T)$ .

On the other hand, assume (4.1) holds. By Property 1 (ii), it follows that  $\mathbf{z}^* = \phi \mathbf{1}_n^T \mathbf{z}^*$ . Plugging  $\mathbf{x}^* = D^{-1}\mathbf{z}^*$  gives  $\mathbf{x}^* = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \mathbf{z}^*$ , which implies that  $\mathbf{x}^*$  is consensual. Moreover, by  $\mathbf{y}^* + \alpha \nabla \mathbf{f}(\mathbf{x}^*) = 0$  and  $\mathbf{y}^* \in \text{null}(\mathbf{1}_n^T)$ , it holds  $\mathbf{1}_n^T \nabla \mathbf{f}(\mathbf{x}^*) = -\frac{1}{\alpha} \mathbf{1}_n^T \mathbf{y}^* = 0$ , which implies that  $\mathbf{x}^*$  is optimal.  $\square$

Introducing the sequence

$$\mathbf{y}^t \triangleq \sum_{k=0}^t (\bar{A} - A)\mathbf{z}^k, \tag{4.2}$$

the iteration (3.3) of ExtraPush can be rewritten as

$$\begin{cases} \bar{A}\mathbf{z}^{t+1} = \bar{A}\mathbf{z}^t - \alpha\nabla\mathbf{f}(\mathbf{x}^t) - \mathbf{y}^{t+1}, \\ \mathbf{y}^{t+1} = \mathbf{y}^t + (\bar{A} - A)\mathbf{z}^{t+1}, \\ \mathbf{w}^{t+1} = A\mathbf{w}^t, \\ \mathbf{x}^{t+1} = \text{diag}(\mathbf{w}^{t+1})^{-1}\mathbf{z}^{t+1}. \end{cases} \quad (4.3)$$

**Theorem 2.** *Suppose that the sequence  $\{\mathbf{x}^t\}$  generated by ExtraPush (3.3) and the sequence  $\{\mathbf{y}^t\}$  defined in (4.2) are bounded. Then, any limit point of  $\{(\mathbf{z}^t, \mathbf{y}^t, \mathbf{x}^t)\}$ , denoted by  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*)$ , satisfies the optimality conditions (4.1).*

*Proof.* By Property 1,  $\{\mathbf{w}^t\}$  is bounded. By the last update of (4.3) and the boundedness of both  $\{\mathbf{x}^t\}$  and  $\{\mathbf{w}^t\}$ ,  $\{\mathbf{z}^t\}$  is bounded. Hence, there exists a convergent subsequence  $\{(\mathbf{z}, \mathbf{y}, \mathbf{w}, \mathbf{x})^{t_j}\}_{j=1}^\infty$ . Let  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{w}^*, \mathbf{x}^*)$  be its limit. By (2.3), we know that  $\mathbf{w}^* = n\phi$  and thus that  $\mathbf{x}^* = D^{-1}\mathbf{z}^*$ . Letting  $t \rightarrow \infty$  in the second equation of (4.3) gives  $\mathbf{z}^* = A\mathbf{z}^*$ , or equivalently  $\mathbf{z}^* \in \text{null}(\mathbf{I}_n - A)$ . Similarly, letting  $t \rightarrow \infty$  in the first equation of (4.3) yields  $\mathbf{y}^* + \alpha\nabla\mathbf{f}(\mathbf{x}^*) = 0$ . Moreover, from the definition (4.2) of  $\mathbf{y}^t$  and the facts that both  $A$  and  $\bar{A}$  are column stochastic, it follows that  $\mathbf{1}_n^T \mathbf{y}^* = 0$  and  $\mathbf{1}_n^T \nabla\mathbf{f}(\mathbf{x}^*) = 0$ . Therefore,  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*)$  satisfies the optimality conditions (4.1).  $\square$

## 5. Convergence of Normalized ExtraPush

In this section, we show the linear convergence of Normalized ExtraPush under the smoothness and strong convexity assumptions of the objective function. Similar to (4.3), introducing a new sequence  $\mathbf{y}^t = \sum_{k=0}^t (\bar{A} - A)\mathbf{z}^k$ , the iterative formula (3.4) of Normalized ExtraPush implies

$$\begin{cases} \bar{A}\mathbf{z}^{t+1} = \bar{A}\mathbf{z}^t - \alpha\nabla\mathbf{f}(\mathbf{x}^t) - \mathbf{y}^{t+1}, \\ \mathbf{y}^{t+1} = \mathbf{y}^t + (\bar{A} - A)\mathbf{z}^{t+1}, \\ \mathbf{x}^{t+1} = D^{-1}\mathbf{z}^{t+1}. \end{cases} \quad (5.1)$$

**Theorem 3.** *Suppose that the sequence  $\{\mathbf{x}^t\}$  generated by Normalized ExtraPush (3.4) is bounded, and that the sequence  $\{\mathbf{y}^t\}$  is also bounded. Then, any limit point of  $\{(\mathbf{z}^t, \mathbf{y}^t, \mathbf{x}^t)\}_{t=0}^\infty$ , denoted by  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*)$ , satisfies the first-order optimality conditions (4.1).*

The proof is very similar to that of Theorem 2. It only needs to replace the sequence  $\{\mathbf{w}^t\}$  with its limitation  $n\phi$  in the proof procedure, thus we omit it here. From Theorem 3, it shows that Normalized ExtraPush has subsequence convergence to an optimal solution of the considered optimization problem under the boundedness assumption. To obtain the linear convergence of Normalized ExtraPush, we still need the following assumptions.

**Assumption 2. (existence of solution)** *Let  $\mathcal{X}^*$  be the optimal solution set of problem (1.1), and assume that  $\mathcal{X}^*$  is nonempty.*

**Assumption 3.** *For each agent  $i$ , its objective function  $f_i$  satisfies the following:*

- (i) **(Lipschitz differentiability)**  $f_i$  is differentiable, and its gradient  $\nabla f_i$  is  $L_i$ -Lipschitz continuous, i.e.,  $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i\|x - y\|, \forall x, y \in \mathbf{R}^p$ ;

- (ii) (**quasi-strong convexity**)  $f_i$  is quasi-strongly convex, and there exists a positive constant  $S_i$  such that  $S_i \|x^* - y\|^2 \leq \langle \nabla f_i(x^*) - \nabla f_i(y), x^* - y \rangle$  for any  $y \in \mathbf{R}^p$  and some optimal value  $x^* \in \mathcal{X}^*$ .

Following Assumption 3, there hold for any  $\mathbf{x}, \mathbf{y} \in \mathbf{R}^{n \times p}$  and some  $\mathbf{x}^* \equiv \mathbf{1}_n(x^*)^T$

$$\|\nabla \mathbf{f}(\mathbf{x}) - \nabla \mathbf{f}(\mathbf{y})\| \leq L_f \|\mathbf{x} - \mathbf{y}\|, \quad (5.2)$$

$$S_f \|\mathbf{x}^* - \mathbf{y}\|^2 \leq \langle \nabla \mathbf{f}(\mathbf{x}^*) - \nabla \mathbf{f}(\mathbf{y}), \mathbf{x}^* - \mathbf{y} \rangle, \quad (5.3)$$

where the constants  $L_f \triangleq \max_i L_i$  and  $S_f \triangleq \min_i S_i$ .

**Assumption 4. (positive definiteness)**  $D^{-1}\bar{A} + \bar{A}^T D^{-1} \succ 0$ .

By noticing  $D^{-1}\bar{A} + \bar{A}^T D^{-1} = D^{-1/2}(D^{-1/2}\bar{A}D^{1/2} + D^{1/2}\bar{A}^T D^{-1/2})D^{-1/2}$ , we can guarantee the positive definiteness of  $D^{-1}\bar{A} + \bar{A}^T D^{-1}$  by ensuring the matrix  $\bar{A} + \bar{A}^T$  to be positive definite. Note that  $\bar{A}_{ii} > \sum_{j \neq i} \bar{A}_{ij}$  for each  $i$ , which means that  $\bar{A}$  is strictly column-diagonal dominant. To ensure the positive definiteness of  $\bar{A} + \bar{A}^T$ , each node  $j$  can be “selfish” and take a sufficiently large  $A_{jj}$ .

Before presenting the main result, we introduce the following notation. For each  $t$ , introducing  $\mathbf{u}^t = \sum_{k=0}^t \mathbf{z}^k$ , then the Normalized ExtraPush iteration (3.4) reduces to

$$\begin{cases} \bar{A}\mathbf{z}^{t+1} = \bar{A}\mathbf{z}^t - \alpha \nabla \mathbf{f}(\mathbf{x}^t) - (\bar{A} - A)\mathbf{u}^{t+1} \\ \mathbf{u}^{t+1} = \mathbf{u}^t + \mathbf{z}^{t+1} \\ \mathbf{x}^{t+1} = D^{-1}\mathbf{z}^{t+1}. \end{cases} \quad (5.4)$$

Let  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*) \in \mathcal{L}^*$ , where  $\mathbf{x}^*$  has been specified in (5.3). Let  $\mathbf{u}^*$  be any matrix that satisfies  $(\bar{A} - A)\mathbf{u}^* = \mathbf{y}^*$ . For simplicity, we introduce

$$\mathbf{v}^t = \begin{pmatrix} \mathbf{z}^t \\ \mathbf{u}^t \end{pmatrix}, \quad \mathbf{v}^* = \begin{pmatrix} \mathbf{z}^* \\ \mathbf{u}^* \end{pmatrix}, \quad G = \begin{pmatrix} N^T & \mathbf{0} \\ \mathbf{0} & M \end{pmatrix}, \quad S = \begin{pmatrix} \mathbf{0} & M \\ -M^T & \mathbf{0} \end{pmatrix}, \quad (5.5)$$

where  $N = D^{-1}\bar{A}$ ,  $M = D^{-1}(\bar{A} - A)$ . Let  $\mathbf{f}_D(\mathbf{z}) \triangleq \mathbf{f}(D^{-1}\mathbf{z})$  and  $\bar{\mathbf{f}}(\mathbf{v}) \triangleq \mathbf{f}_D(\mathbf{z})$ . Then  $\nabla \bar{\mathbf{f}}(\mathbf{v}) = [\nabla \mathbf{f}_D(\mathbf{z}), 0]$ . By (5.2) and (5.3), there hold

$$\|\nabla \bar{\mathbf{f}}(\mathbf{v}_1) - \nabla \bar{\mathbf{f}}(\mathbf{v}_2)\| = \|\nabla \mathbf{f}_D(\mathbf{z}_1) - \nabla \mathbf{f}_D(\mathbf{z}_2)\| \leq \bar{L} \|\mathbf{z}_1 - \mathbf{z}_2\|, \quad (5.6)$$

$$\bar{\mu} \|\mathbf{z}^* - \mathbf{z}\|^2 \leq \langle \nabla \mathbf{f}_D(\mathbf{z}^*) - \nabla \mathbf{f}_D(\mathbf{z}), \mathbf{z}^* - \mathbf{z} \rangle = \langle \nabla \bar{\mathbf{f}}(\mathbf{v}^*) - \nabla \bar{\mathbf{f}}(\mathbf{v}), \mathbf{v}^* - \mathbf{v} \rangle, \quad (5.7)$$

where  $\bar{L} \triangleq \frac{L_f}{\sigma_{\min}^2(D)}$  and  $\bar{\mu} \triangleq \frac{S_f}{\sigma_{\max}^2(D)}$ . By (5.4) and (5.5), the Normalized ExtraPush iteration (3.4) implies

$$G^T(\mathbf{v}^{t+1} - \mathbf{v}^t) = -S\mathbf{v}^{t+1} - \alpha \nabla \bar{\mathbf{f}}(\mathbf{v}^t). \quad (5.8)$$

Next, we will show that  $G + G^T$  is positive semidefinite, which by Assumption 4 implies that  $N + N^T$  is positive definite. It is sufficient to show that  $M + M^T$  is positive semidefinite. Note that

$$\begin{aligned} M + M^T &= \frac{D^{-1}(\mathbf{I}_n - A)}{2} + \frac{(\mathbf{I}_n - A)^T D^{-1}}{2} \\ &= D^{-1/2} \left( \mathbf{I}_n - \frac{D^{1/2} A^T D^{-1/2} + D^{-1/2} A D^{1/2}}{2} \right) D^{-1/2} \triangleq D^{-1/2} \Lambda D^{-1/2}, \end{aligned}$$



and by Property 1 (iii),  $n\phi^T$  is the left eigenvector of  $A^T$  corresponding to eigenvalue 1, and thus,  $\Lambda$  is the Laplacian of a certain directed graph  $\mathcal{G}'$  with  $A^T$  being its corresponding transition probability matrix [3]. It follows that  $0 = \lambda_1(\Lambda) \leq \lambda_2(\Lambda) \leq \dots \leq \lambda_n(\Lambda)$ , where  $\lambda_i(\Lambda)$  denotes the  $i$ th eigenvalue of  $\Lambda$ . Therefore,  $M + M^T$  is positive semidefinite, and the following property holds

$$\|x\|_G^2 = \frac{1}{2}\|x\|_{G+G^T}^2 \geq 0, \quad \forall x \in \mathbf{R}^n.$$

Let

$$c_1 = \frac{\lambda_{\max}(MM^T)}{\lambda_{\min}(M^TM)}, \quad c_2 = \frac{\lambda_{\max}(\frac{M+M^T}{2})}{\lambda_{\min}(M^TM)}, \quad c_3 = \lambda_{\max}(NN^T) + 3c_1\lambda_{\max}(N^TN),$$

and let

$$\Delta_1 = \left(\bar{\mu} - \frac{\eta}{2}\right)^2 - 6c_1\bar{L}^2, \quad \Delta_2 = \frac{\bar{L}^4}{4\eta^2} - 3c_1\bar{L}^2\sigma\left(c_3\sigma - \lambda_{\min}(N^T + N)\right)$$

for some appropriate tunable parameters  $\eta$  and  $\sigma$ . Then we describe our main result as follows.

**Theorem 4.** *Under Assumptions 1-4, if the step size parameter  $\alpha$  satisfies*

$$\frac{\bar{\mu} - \frac{\eta}{2} - \sqrt{\Delta_1}}{3c_1\bar{L}^2\sigma} < \alpha < \min\left\{\frac{\bar{\mu} - \frac{\eta}{2} + \sqrt{\Delta_1}}{3c_1\bar{L}^2\sigma}, \frac{-\frac{\bar{L}^2}{2\eta} + \sqrt{\Delta_2}}{3c_1\bar{L}^2\sigma}\right\} \quad (5.9)$$

for some appropriate  $\eta$  and  $\sigma$  as specified in (5.25) and (5.26), respectively, then the sequence  $\{\mathbf{v}^t\}$  defined in (5.5) satisfies

$$\|\mathbf{v}^t - \mathbf{v}^*\|_G^2 \geq (1 + \delta)\|\mathbf{v}^{t+1} - \mathbf{v}^*\|_G^2, \quad (5.10)$$

for  $\delta > 0$  obeying

$$0 < \delta \leq \min\left\{\frac{-\frac{1}{\sigma} + (\bar{\mu} - \frac{\eta}{2})\alpha - \frac{3}{2}c_1\bar{L}^2\sigma\alpha^2}{\lambda_{\max}(\frac{N+N^T}{2}) + 3c_2\alpha^2\bar{L}^2}, \frac{\lambda_{\min}(\frac{N^T+N}{2}) - \frac{c_3\sigma}{2} - \frac{\bar{L}^2\alpha}{2\eta} - \frac{3}{2}c_1\bar{L}^2\sigma\alpha^2}{3c_2(\lambda_{\max}(N^TN) + \alpha^2\bar{L}^2)}\right\}.$$

From this theorem, the sequence  $\{\mathbf{v}^t\}$  converges to  $\mathbf{v}^*$  at a linear rate in the sense of “ $G$ -norm”. By the definition of  $\mathbf{v}^*$  in (5.5),  $\mathbf{v}^*$  is indeed defined by some optimal value  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*)$ . Roughly speaking, bigger  $\delta$  means faster convergence rate. As specified in Theorem 4,  $\delta$  is affected by many factors. Generally,  $\delta$  decreases with respect to both  $\lambda_{\max}(\frac{N+N^T}{2})$  and  $\lambda_{\max}(N^TN)$ , which potentially implies that if all nodes are more “selfish”, that is, they hold more information for themselves than sending to their out-neighbors. Consequently, the information mixing speed of the network will get smaller, and thus the convergence of Normalized ExtraPush becomes slower. Therefore, we suggest a more democratic rule (such as the matrix  $A$  specified in (2.2)) for faster convergence in practice. To ensure  $\delta > 0$ , it requires that the step size  $\alpha$  lie in an appropriate interval. It should be pointed out that the condition (5.9) on  $\alpha$  is sufficiently, not necessary, for the linear convergence of Normalized ExtraPush. Normalized ExtraPush algorithm may not diverge if a small  $\alpha$  is set. In fact, in the next section, it can be observed that both ExtraPush and Normalized ExtraPush algorithms converge under small values of  $\alpha$ . In general, a smaller  $\alpha$  implies a slower rate of convergence.

To prove Theorem 4, we need the following lemmas.

**Lemma 1.** For any  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*) \in \mathcal{L}^*$ , let  $\mathbf{u}^*$  satisfy  $(\bar{A} - A)\mathbf{u}^* = \mathbf{y}^*$ . Then there hold

$$M\mathbf{z}^* = \mathbf{0}, \quad (5.11)$$

$$M^T\mathbf{z}^* = \mathbf{0}, \quad (5.12)$$

$$S\mathbf{v}^* + \alpha\nabla\bar{\mathbf{f}}(\mathbf{v}^*) = \mathbf{0}. \quad (5.13)$$

*Proof.* By the optimality of  $(\mathbf{z}^*, \mathbf{y}^*, \mathbf{x}^*)$ , the followings hold: (i)  $(\bar{A} - A)\mathbf{z}^* = \mathbf{0}$ , and thus  $M\mathbf{z}^* = \mathbf{0}$ ; (ii)  $D^{-1}\mathbf{z}^* = \mathbf{x}^*$  is consensual; from the column stochasticity of both  $A$  and  $\bar{A}$ , it follows  $M^T\mathbf{z}^* = (\bar{A} - A)^T\mathbf{x}^* = \mathbf{0}$ ; (iii)  $M\mathbf{u}^* + \alpha\nabla\mathbf{f}_D(\mathbf{z}^*) = D^{-1}\mathbf{y}^* + \alpha D^{-1}\nabla\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$ , with  $M^T\mathbf{z}^* = \mathbf{0}$ , which imply  $S\mathbf{v}^* + \alpha\nabla\bar{\mathbf{f}}(\mathbf{v}^*) = \mathbf{0}$ .  $\square$

**Lemma 2.** For any  $t \in \mathbf{N}$ , it holds

$$N(\mathbf{z}^{t+1} - \mathbf{z}^t) = -M(\mathbf{u}^{t+1} - \mathbf{u}^*) - \alpha[\nabla\mathbf{f}_D(\mathbf{z}^t) - \nabla\mathbf{f}_D(\mathbf{z}^*)]. \quad (5.14)$$

This lemma follows from (5.4) and the fact  $M\mathbf{u}^* + \alpha\nabla\mathbf{f}_D(\mathbf{z}^*) = \mathbf{0}$  in the last lemma.

**Lemma 3.** Let  $\{\mathbf{v}^t\}$  be a sequence generated by the iteration (5.8) and  $\mathbf{v}^*$  be defined in (5.5). Then, it holds

$$\begin{aligned} \|\mathbf{v}^{t+1} - \mathbf{v}^*\|_G^2 - \|\mathbf{v}^t - \mathbf{v}^*\|_G^2 &\leq -\|\mathbf{v}^{t+1} - \mathbf{v}^t\|_G^2 + \|\mathbf{z}^{t+1} - \mathbf{z}^t\|_{\frac{\sigma}{2}NN^T + \frac{\alpha\bar{L}^2}{2\eta}\mathbf{I}_n}^2 \\ &\quad - \|\mathbf{z}^* - \mathbf{z}^{t+1}\|_{(\alpha\bar{\mu} - \frac{\alpha\eta}{2} - \frac{1}{\sigma})\mathbf{I}_n}^2 + \frac{\sigma}{2}\|\mathbf{u}^* - \mathbf{u}^{t+1}\|_{MM^T}^2, \end{aligned} \quad (5.15)$$

where  $\sigma, \eta > 0$  are two tunable parameters.

*Proof.* Note that

$$\begin{aligned} \|\mathbf{v}^{t+1} - \mathbf{v}^*\|_G^2 - \|\mathbf{v}^t - \mathbf{v}^*\|_G^2 &= -\|\mathbf{v}^{t+1} - \mathbf{v}^t\|_G^2 + \langle \mathbf{v}^* - \mathbf{v}^{t+1}, G(\mathbf{v}^t - \mathbf{v}^{t+1}) \rangle \\ &\quad + \langle \mathbf{v}^* - \mathbf{v}^{t+1}, G^T(\mathbf{v}^t - \mathbf{v}^{t+1}) \rangle. \end{aligned} \quad (5.16)$$

In the following, we analyze the two inner-product terms:

$$\begin{aligned} \langle \mathbf{v}^* - \mathbf{v}^{t+1}, G(\mathbf{v}^t - \mathbf{v}^{t+1}) \rangle &= \langle \mathbf{z}^* - \mathbf{z}^{t+1}, N^T(\mathbf{z}^t - \mathbf{z}^{t+1}) \rangle + \langle M^T(\mathbf{u}^* - \mathbf{u}^{t+1}), \mathbf{u}^t - \mathbf{u}^{t+1} \rangle \\ (\because (5.11), M\mathbf{z}^* = \mathbf{0}) &= \langle \mathbf{z}^* - \mathbf{z}^{t+1}, N^T(\mathbf{z}^t - \mathbf{z}^{t+1}) \rangle + \langle M^T(\mathbf{u}^* - \mathbf{u}^{t+1}), \mathbf{z}^* - \mathbf{z}^{t+1} \rangle \\ &\leq \frac{\sigma}{2}\|\mathbf{z}^t - \mathbf{z}^{t+1}\|_{NN^T}^2 + \frac{1}{\sigma}\|\mathbf{z}^* - \mathbf{z}^{t+1}\|^2 + \frac{\sigma}{2}\|\mathbf{u}^* - \mathbf{u}^{t+1}\|_{MM^T}^2, \end{aligned} \quad (5.17)$$

and

$$\begin{aligned} \langle \mathbf{v}^* - \mathbf{v}^{t+1}, G^T(\mathbf{v}^t - \mathbf{v}^{t+1}) \rangle &= \langle \mathbf{v}^* - \mathbf{v}^{t+1}, S\mathbf{v}^{t+1} + \alpha\nabla\bar{\mathbf{f}}(\mathbf{v}^t) \rangle (\because (5.5)) \\ &= \langle \mathbf{v}^* - \mathbf{v}^{t+1}, S(\mathbf{v}^{t+1} - \mathbf{v}^*) + \alpha(\nabla\bar{\mathbf{f}}(\mathbf{v}^t) - \nabla\bar{\mathbf{f}}(\mathbf{v}^*)) \rangle (\because (5.13)) \\ (\because S = -S^T) &= \alpha\langle \mathbf{v}^* - \mathbf{v}^{t+1}, \nabla\bar{\mathbf{f}}(\mathbf{v}^t) - \nabla\bar{\mathbf{f}}(\mathbf{v}^*) \rangle \\ &= \alpha\langle \mathbf{v}^* - \mathbf{v}^{t+1}, \nabla\bar{\mathbf{f}}(\mathbf{v}^{t+1}) - \nabla\bar{\mathbf{f}}(\mathbf{v}^*) \rangle \\ &\quad + \alpha\langle \mathbf{v}^* - \mathbf{v}^{t+1}, \nabla\bar{\mathbf{f}}(\mathbf{v}^t) - \nabla\bar{\mathbf{f}}(\mathbf{v}^{t+1}) \rangle \\ &\leq -\alpha\bar{\mu}\|\mathbf{z}^{t+1} - \mathbf{z}^*\|^2 + \frac{\alpha\eta}{2}\|\mathbf{z}^* - \mathbf{z}^{t+1}\|^2 + \frac{\alpha\bar{L}^2}{2\eta}\|\mathbf{z}^t - \mathbf{z}^{t+1}\|^2. \end{aligned} \quad (5.18)$$

Substituting (5.17) and (5.18) into (5.16), then we can conclude the lemma.  $\square$

*Proof.* (for Theorem 4) In order to establish (5.10) for some constant  $\delta > 0$ , in light of Lemma 3, it is sufficient to show that the right-hand side of (5.15) is no more than  $-\delta\|\mathbf{v}^{t+1} - \mathbf{v}^*\|_G^2$ , which implies

$$\|\mathbf{z}^{t+1} - \mathbf{z}^*\|_P^2 + \|\mathbf{z}^{t+1} - \mathbf{z}^t\|_Q^2 \geq \|\mathbf{u}^{t+1} - \mathbf{u}^*\|_R^2, \quad (5.19)$$

where

$$P = \left( \alpha\bar{\mu} - \frac{\alpha\eta}{2} - \frac{1}{\sigma} \right) \mathbf{I}_n - \delta \frac{N + N^T}{2}, \quad Q = \frac{N^T + N}{2} - \frac{\sigma}{2} NN^T - \frac{\alpha\bar{L}^2}{2\eta} \mathbf{I}_n,$$

$$R = \frac{\sigma}{2} MM^T + \delta \left( \frac{M + M^T}{2} \right).$$

*Establishing (5.19): Step 1.* From Lemma 2, there holds

$$\begin{aligned} \|\mathbf{u}^* - \mathbf{u}^{t+1}\|_{M^T M}^2 &= \|M(\mathbf{u}^* - \mathbf{u}^{t+1})\|^2 \\ &= \|N(\mathbf{z}^{t+1} - \mathbf{z}^t) + \alpha[\nabla \mathbf{f}_D(\mathbf{z}^{t+1}) - \nabla \mathbf{f}_D(\mathbf{z}^*)] + \alpha[\nabla \mathbf{f}_D(\mathbf{z}^t) - \nabla \mathbf{f}_D(\mathbf{z}^{t+1})]\|^2 \\ &\leq 3\|\mathbf{z}^{t+1} - \mathbf{z}^t\|_{N^T N}^2 + 3\alpha^2 \bar{L}^2 \|\mathbf{z}^{t+1} - \mathbf{z}^*\|^2 + 3\alpha^2 \bar{L}^2 \|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2 \\ &= \|\mathbf{z}^{t+1} - \mathbf{z}^*\|_{3\alpha^2 \bar{L}^2 \mathbf{I}_n}^2 + \|\mathbf{z}^{t+1} - \mathbf{z}^t\|_{3(N^T N + \alpha^2 \bar{L}^2 \mathbf{I}_n)}^2. \end{aligned} \quad (5.20)$$

Note that

$$\frac{\|\mathbf{u}^* - \mathbf{u}^{t+1}\|_{\frac{\sigma}{2} MM^T + \delta M}^2}{\frac{\sigma \lambda_{\max}(MM^T)}{2} + \delta \lambda_{\max}\left(\frac{M+M^T}{2}\right)} \leq \|\mathbf{u}^* - \mathbf{u}^{t+1}\|^2 \leq \frac{\|\mathbf{u}^* - \mathbf{u}^{t+1}\|_{M^T M}^2}{\tilde{\lambda}_{\min}(M^T M)}.$$

If the following conditions hold

$$\begin{cases} P \succeq 3\left(\frac{1}{2}c_1\sigma + c_2\delta\right)\alpha^2 \bar{L}^2 \mathbf{I}_n, \\ Q \succeq 3\left(\frac{1}{2}c_1\sigma + c_2\delta\right)(N^T N + \alpha^2 \bar{L}^2 \mathbf{I}_n), \end{cases} \quad (5.21)$$

then (5.19) holds. To show (5.21), it is sufficient to prove

$$\begin{cases} (\lambda_{\max}\left(\frac{N+N^T}{2}\right) + 3c_2\alpha^2 \bar{L}^2)\delta \leq -\frac{1}{\sigma} + \left(\bar{\mu} - \frac{\eta}{2}\right)\alpha - \frac{3}{2}c_1 \bar{L}^2 \sigma \alpha^2, \\ 3c_2(\lambda_{\max}(N^T N) + \alpha^2 \bar{L}^2)\delta \leq \lambda_{\min}\left(\frac{N^T+N}{2}\right) - \frac{c_3\sigma}{2} - \frac{\bar{L}^2\alpha}{2\eta} - \frac{3}{2}c_1 \bar{L}^2 \sigma \alpha^2. \end{cases} \quad (5.22)$$

Let  $c_4 \triangleq \left(\bar{\mu} - \frac{\eta}{2}\right) + \sqrt{\Delta_1}$ ,  $c_5 \triangleq \frac{\bar{L}^2}{\eta}$ ,  $c_6 \triangleq \frac{2c_4c_5 + 12c_1\bar{L}^2}{c_4^2}$ ,  $c_7 \triangleq \frac{\lambda_{\min}(N^T+N)}{4c_3}$ ,  $c_8 \triangleq a(c_7 + 2) - (2 - c_7)$  for some positive constant  $a \in (0, 1)$ ,  $\Delta_3 \triangleq \lambda_{\min}^2(N^T + N) - 4c_3c_6$ . After reduction, we claim that if the following conditions hold

$$\frac{2 - c_7}{2 + c_7} < a < 1, \quad (5.23)$$

$$\bar{\mu} > \left( \sqrt{\frac{6c_1}{1-a^2}} + \frac{1}{c_8} \sqrt{\frac{1-a^2}{6c_1}} \right) \bar{L}, \quad (5.24)$$

$$\bar{\mu} \left( 1 - \sqrt{1 - \frac{4\bar{L}^2}{c_8\bar{\mu}^2}} \right) < \eta < \min \left\{ \bar{\mu} \left( 1 + \sqrt{1 - \frac{4\bar{L}^2}{c_8\bar{\mu}^2}} \right), 2 \left( \bar{\mu} - \sqrt{\frac{6c_1}{1-a^2}} \bar{L} \right) \right\}, \quad (5.25)$$

$$\frac{\lambda_{\min}(N^T + N) - \sqrt{\Delta_3}}{2c_3} < \sigma < \frac{\lambda_{\min}(N^T + N) + \sqrt{\Delta_3}}{2c_3}, \quad (5.26)$$

$$\frac{\bar{\mu} - \frac{\eta}{2} - \sqrt{\Delta_1}}{3c_1\bar{L}^2\sigma} < \alpha < \min \left\{ \frac{\bar{\mu} - \frac{\eta}{2} + \sqrt{\Delta_1}}{3c_1\bar{L}^2\sigma}, \frac{-\frac{\bar{L}^2}{2\eta} + \sqrt{\Delta_2}}{3c_1\bar{L}^2\sigma} \right\}, \quad (5.27)$$

then (5.22) holds for some positive constant  $\delta$ . We then end the proof of this theorem.  $\square$

### 6. Numerical Experiments

In this section, we present the results of a series of numerical experiments that demonstrate the effectiveness of the proposed algorithms relative to the subgradient-push algorithm. The used network and its corresponding mixing matrix  $A$  are depicted in Fig. 2.1.

#### 6.1. Decentralized Least Squares

Consider the following decentralized least squares problem:

$$x^* \leftarrow \operatorname{argmin}_{x \in \mathbf{R}^p} f(x) = \sum_{i=1}^n f_i(x), \tag{6.1}$$

where  $f_i(x) = \frac{1}{2} \|B_{(i)}x - b_{(i)}\|_2^2$ ,  $B_{(i)} \in \mathbf{R}^{m_i \times p}$ ,  $b_{(i)} \in \mathbf{R}^{m_i}$  for  $i = 1, \dots, n$ . The solution  $x^*$  is  $B^\dagger b$ , where  $B = \sum_{i=1}^n B_{(i)}^T B_{(i)}$ ,  $b = \sum_{i=1}^n B_{(i)}^T b_{(i)}$ , and  $B^\dagger$  is the pseudo-inverse of  $B$ . In this experiment, we take  $n = 5$ ,  $p = 256$ , and  $m_i = 100$  for  $i = 1, \dots, 5$ . For both ExtraPush and Normalized ExtraPush, we first choose an  $\alpha$  in the hand-optimized manner (in this case,  $\alpha = 0.1$ ) and then take a smaller one like  $\alpha = 0.02$  to show the difference due to a smaller step size. The step size of the subgradient-push algorithm is handed optimized to  $\alpha_t = \frac{0.8}{\sqrt{t}}$ . The experiment results are illustrated in Fig. 6.1.

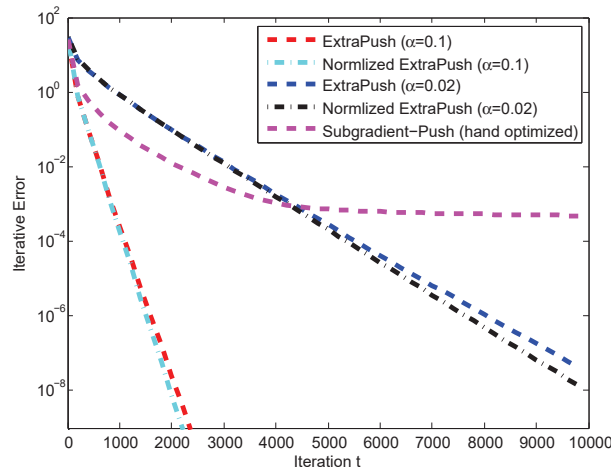


Fig. 6.1. Experiment results for decentralized least squares regression. History of  $\|\mathbf{x}^t - \mathbf{x}^*\|_2$ , where  $\mathbf{x}^*$  is the exact solution. The performances of ExtraPush and Normalized ExtraPush are very similar.

As illustrated in Fig. 6.1, the performances of ExtraPush and Normalized ExtraPush are almost identical. Their linear convergence rates are affected by different step sizes; a smaller  $\alpha$  leads to a slower rate, as one would expect.

#### 6.2. Decentralized Huber-like Regression

Instead of least squares, this experiment minimizes the Huber loss function, which is known to be robust to outliers:

$$x^* \leftarrow \operatorname{argmin}_{x \in \mathbf{R}^p} f(x) = \sum_{i=1}^n f_i(x), \tag{6.2}$$

where  $f_i(x) = \sum_{j=1}^{m_i} H_\xi(B_{(i)j}x - b_{(i)j})$ ,  $B_{(i)j}$  is  $j$ th row of matrix  $B_{(i)} \in \mathbf{R}^{m_i \times p}$  and  $b_{(i)j}$  is the  $j$ th entry of vector  $b_{(i)} \in \mathbf{R}^{m_i}$  for  $i = 1, \dots, n$ . The Huber loss function is defined as

$$H_\xi(a) = \begin{cases} \frac{1}{2}a^2, & \text{for } |a| \leq \xi \quad (\ell_2^2 \text{ zone}), \\ \xi(|a| - \frac{1}{2}\xi), & \text{otherwise} \quad (\ell_1 \text{ zone}). \end{cases} \quad (6.3)$$

Similar to the experimental setting in [13], we let  $\xi = 2$  and set the solution  $x^*$  in the  $\ell_2^2$  zone while initializing  $x_{(i)}^0$  in the  $\ell_1$  zone for all agents  $i$ . Similar to the last experiment, we test two different step sizes,  $\alpha = 0.1$  and  $0.02$ , where  $\alpha = 0.1$  is hand-optimized. The step size of the subgradient-push algorithm is hand optimized to  $\alpha_t = \frac{5}{\sqrt{t+100}}$ . The numerical results are depicted in Fig. 6.2.

As shown by Fig. 6.2, when  $\alpha = 0.1$ , both ExtraPush and Normalized ExtraPush algorithms have the sublinear convergence in their first 500 iterations and then show linear convergence, as  $x_{(i)}^t$  for most  $i$  have entered the  $\ell_2^2$  zone. While for  $\alpha = 0.02$ , more iterations (about 2500) are needed before both algorithms start decaying linearly.

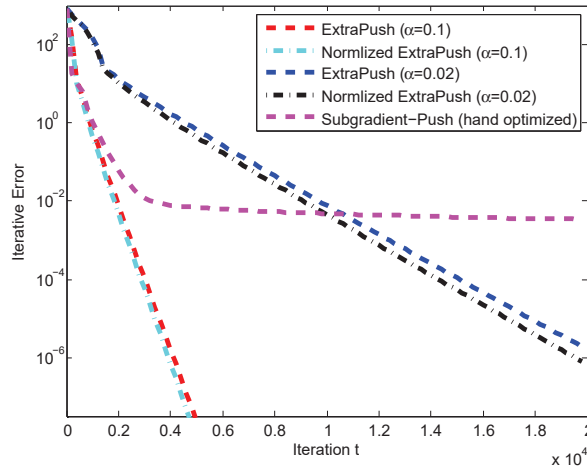


Fig. 6.2. Experiment results for decentralized Huber regression. History of  $\|\mathbf{x}^t - \mathbf{x}^*\|_2$ , where  $\mathbf{x}^*$  is the exact solution. The performances of ExtraPush and Normalized ExtraPush are very similar.

### 7. Conclusion

In this note, we propose a decentralized algorithm called ExtraPush, as well as its simplified version called Normalized ExtraPush, for solving distributed consensus optimization problems over directed graphs. The algorithms use column-stochastic mixing matrices. We show that Normalized ExtraPush converges at a linear rate if the objective function is smooth and strongly convex. In addition, we develop the first-order optimality conditions and provide the convergence of ExtraPush under the boundedness assumption. The convergence as well as the rate of convergence of ExtraPush should be justified in the future. Moreover, when applied to a directed time-varying network, the performance of the proposed algorithms will also be studied. Another line of future research is to generalize ExtraPush to handle the sum of smooth and proximable (possibly nonsmooth) functions as done in [14] that has generalized Extra this way.

**Acknowledgments.** The work of W. Yin has been supported in part by the NSF grants DMS-1317602 and ECCS-1462398. The work of J. Zeng has been supported in part by the NSF grant 11501440.

## References

- [1] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis and M. Vetterli, Weighted Gossip: distributed averaging using non-doubly stochastic matrices, in *Proc. IEEE Conf. Decision Control*, (2010), 1753-1757.
- [2] I. Chen, Fast distributed first-order methods, Masters thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 2012.
- [3] F. Chung, Laplacians and the cheeger inequality for directed graphs, *Annals of Combinatorics*, **2** (2005), 1-19.
- [4] J. Duchi, A. Agarwal and M. Wainwright, Dual averaging for distributed optimization: Convergence analysis and network scaling, *IEEE Trans. Automat. Contr.*, **57** (2012) 592-606.
- [5] D. Jakovetic, J. Xavier and J. Moura, Fast distributed gradient methods, *IEEE Trans. Automat. Contr.*, **59** (2014) 1131-1146.
- [6] D. Kempe, A. Dobra and J. Gehrke, Gossip-based computation of aggregate information, in *44th Annual IEEE Symposium on Foundations of Computer Science*, (2003), 482-491.
- [7] F. Iutzeler, P. Ciblat and W. Hachem, Analysis of sum-weight-like algorithms for averaging in wireless sensor networks, *IEEE Trans. Signal Process.*, **61**(2013) 2802-2814.
- [8] I. Matei and J. Baras, Performance evaluation of the consensus-based distributed subgradient method under random communication topologies, *IEEE J. Sel. Top. Signal Process.*, **5** (2011) 754-771.
- [9] A. Nedic and A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, *IEEE Trans. Automat. Contr.*, **54** (2009) 48-61.
- [10] A. Nedic and A. Olshevsky, Distributed optimization over time-varying directed graphs, *IEEE Trans. Automat. Contr.*, **60** (2015) 601-615.
- [11] I. Schizas, A. Ribeiro, and G. Giannakis, Consensus in ad hoc WSNs with noisy links-part I: Distributed estimation of deterministic signals, *IEEE Trans. Signal Process.*, **56**:1 (2008) 350-364.
- [12] W. Shi, Q. Ling, K. Yuan, G. Wu and W. Yin, On the linear convergence of the ADMM in decentralized consensus optimization, *IEEE Trans. Signal Process.*, **62** (2014) 1750-1761.
- [13] W. Shi, Q. Ling, G. Wu and W. Yin, EXTRA: an exact first-order algorithm for decentralized consensus optimization, *SIAM J. Optimiz.*, **25**:2 (2015) 944-966.
- [14] W. Shi, Q. Ling, G. Wu and W. Yin, A Proximal Gradient Algorithm for Decentralized Composite Optimization, *IEEE Trans. Signal Process.*, **63**:22 (2015) 6013-6023.
- [15] K. I. Tsianos, S. Lawlor and M. G. Rabbat, Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning, in *Proc. 50th Allerton Conf. Commun., Control, Comp.*, (2012), 1543-1550.
- [16] K. I. Tsianos, S. Lawlor and M. G. Rabbat, Push-sum distributed dual averaging for convex optimization, in *Proc. IEEE Conf. Decision Control*, (2012), 5453-5458.
- [17] K. I. Tsianos, The role of the Network in Distributed Optimization Algorithms: Convergence Rates, Scalability, Communication/Computation Tradeoffs and Communication Delays, PhD thesis, Dept. Elect. Comp. Eng., McGill Univ., Montreal, QC, Canada, 2013.
- [18] C. Xi and U.A. Khan, On the linear convergence of distributed optimization over directed graphs, preprint, arXiv:1510.02149, 2015.
- [19] K. Yuan, Q. Ling and W. Yin. On the convergence of decentralized gradient descent. *SIAM J. Optimiz.*, **26**:3 (2016), 1835-1854.