This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

Fenchel Dual Gradient Methods for Distributed Convex Optimization over Time-varying Networks

Xuyang Wu and Jie Lu

***Abstract*—We develop a family of Fenchel dual gradient methods for solving constrained, strongly convex but not necessarily smooth multi-agent optimization problems over time-varying networks. The proposed algorithms are constructed based on weighted Fenchel dual gradients and can be implemented in a fully decentralized fashion. We show that the proposed algorithms drive all the agents to both primal and dual optimality at sublinear rates under a standard connectivity condition. Compared with the existing distributed optimization methods that also have convergence rate guarantees over time-varying networks, our algorithms are able to address constrained problems and have better scalability with respect to network size and time for reaching connectivity. The competent performance of the Fenchel dual gradient methods is demonstrated via simulations.**

***Index Terms*—distributed optimization, multi-agent optimization, Fenchel duality.**

I. Introduction

In many engineering scenarios, a network of agents often need to jointly make a decision so that the sum of their local costs is minimized and certain global constraints are satisfied. Such a multi-agent optimization problem has found a considerable number of applications, such as estimation by sensor networks [1], network resource allocation [2], and cooperative control [3].

To address convex multi-agent optimization in an efficient, robust, and scalable way, distributed optimization algorithms have been substantially exploited, which allow each agent to reach an optimal or suboptimal decision by repeatedly exchanging its own information with neighbors. One typical approach is to let the agents perform consensus operations so as to mix their decisions that are updated using first-order information of their local objectives (e.g., [4]–[14]). Another standard approach is to utilize dual decomposition techniques, which often lead to a dual problem with a decomposable structure, so that it can be solved in a distributed fashion by classic optimization methods including the gradient projection method, the accelerated gradient methods, the method of multipliers, and their variants (e.g., [2], [3], [15]–[20]). Other lines of research on distributed optimization include incremental methods (e.g., [21]), Newton-like methods (e.g., [22]), continuous-time algorithms (e.g., [23]), etc.

Among the existing distributed optimization algorithms, only few of them provide guaranteed convergence rates on time-varying networks. The Subgradient-Push method [10] is able to converge to optimality at an $O(\ln k/\sqrt{k})$ rate on such networks for nonsmooth objective functions with uniformly bounded subgradients [10]. When it comes to problems with strongly convex and smooth objective functions, an $O(\ln k/k)$ rate is established for the Gradient-Push method [11], and linear rates $O(q^k)$, $0<q<1$ are established for the DIGing and Push-DIGing methods [12]. Nevertheless, these algorithms all require the problem to be *unconstrained*.

Motivated by this, we develop a family of distributed Fenchel dual gradient methods that address *constrained* multi-agent optimization problems at a guaranteed *convergence rate* over *time-varying* undirected networks. The main contributions are highlighted as follows:

1) We require the local objectives of the agents to be strongly convex but not necessarily differentiable, which is less restrictive than [11], [12] and different from [10]. Also, we allow for a global constraint set which is the intersection of the distinct local constraints of the agents, while [10]–[12] admit no constraints.
2) The proposed distributed Fenchel dual gradient methods are constructed by deriving a class of weighted gradient methods to solve the Fenchel dual of multi-agent optimization, instead of the conventional Lagrange dual. This allows the agents to evaluate the exact dual gradient in parallel over time-varying networks. Such weighted gradient methods consistently ensure dual feasibility and generalize the algorithms in [24], [25].
3) We provide an $O(1/k)$ rate of convergence to dual optimality under the standard B-connectivity, which is also a new convergence result for weighted gradient methods.
4) We derive $O(1/\sqrt{k})$ rates of convergence to primal optimality and feasibility. The convergence rates lead to an evaluable iteration complexity for attaining any given accuracy, which has better scalability with respect to network size and time B to reach connectivity than the algorithms in [10]–[12].
5) The efficacy of the Fenchel dual gradient algorithms is demonstrated via simulations.

The outline of the paper is as follows: Section II formulates the problem, and Section III develops the algorithms. Section IV establishes the convergence results, and Section V presents the simulation results. Finally, Section VI concludes the paper. A preliminary conference version of this paper can be found in [26], which contains no proofs. In this paper, we significantly improve all the convergence results in [26] and add iteration complexity analysis, comparative discussions, new simulation results, as well as all the proofs.

Notations: We use $\|\cdot\|$ to represent the Euclidean norm. For any set $X\subseteq\mathbb{R}^d$, $\operatorname{int} X$ is its interior, $\operatorname{rel\,int} X$ is its relative interior, $|X|$ is its cardinality, and $P_X(x)=\arg\min_{y\in X}\|x-y\|$

X. Wu and J. Lu are with the School of Information Science and Technology, ShanghaiTech University, 201210 Shanghai, China. Email: {wuxy, lujie}@shanghaitech.edu.cn.

This work has been supported by the National Natural Science Foundation of China under grant 61603254 and the Natural Science Foundation of Shanghai under grant 16ZR1422500.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

is the projection of $x \in \mathbb{R}^d$ onto X. The ball centered at $x \in \mathbb{R}^d$ with radius $r > 0$ is denoted by $B(x, r) := \{y \in \mathbb{R}^d : \|y - x\| \leq r\}$. The floor of a real number is represented by $\lfloor \cdot \rfloor$. For any $\mathbf{x} \in \mathbb{R}^{nd}$, $\mathbf{x} = (x_1^T, \ldots, x_n^T)^T$ means the even partition of $\mathbf{x}$ into n blocks, i.e., $x_i \in \mathbb{R}^d \ \forall i = 1, \ldots, n$. For any function $f : \mathbb{R}^d \to \mathbb{R}$, $\partial f(x)$ denotes a subgradient of f at $x \in \mathbb{R}^d$, and $\nabla f(x)$ denotes the gradient of f at $x \in \mathbb{R}^d$ if f is differentiable. In addition, I_d is the $d \times d$ identity matrix, O_d is the $d \times d$ zero matrix, $\mathbf{1}_d \in \mathbb{R}^d$ is the all-one vector, $\mathbf{0}_d \in \mathbb{R}^d$ is the all-zero vector, and $\otimes$ is the Kronecker product. For any matrices $M, M' \in \mathbb{R}^{n \times n}$, $M \preceq M'$ and $M' \succeq M$ both mean $M' - M$ is positive semidefinite. Also, $[M]_{ij}$ represents the (i, j)-entry of M, $\mathcal{R}(M)$ the range of M, and $\mathrm{Null}(M)$ the null space of M. If M is a block diagonal matrix with diagonal blocks $M_1, \ldots, M_m$, we write it as $M = \mathrm{diag}(M_1, \ldots, M_m)$. If $M = M^T$ is positive semidefinite, $\lambda_i^{\downarrow}(M) \geq 0$ denotes its ith largest eigenvalue and $M^{\dagger}$ its Moore-Penrose pseudoinverse.

II. Problem Formulation

Consider a set $\mathcal{V} = \{1, 2, \ldots, n\}$ of agents, where each agent $i \in \mathcal{V}$ possesses a local objective function $f_i : \mathbb{R}^d \to \mathbb{R}$ and a local constraint set $X_i \subseteq \mathbb{R}^d$. All of the $n \geq 2$ agents attempt to jointly solve the constrained optimization problem

$$\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & \sum_{i \in \mathcal{V}} f_i(x) \\ \text{subject to} & x \in \bigcap_{i \in \mathcal{V}} X_i, \end{array} \tag{1}$$

which satisfies the following assumption:

Assumption 1. *Problem* (1) *satisfies the following:*

(a) Each f_i, $i \in \mathcal{V}$ is strongly convex over X_i with convexity parameter $\theta_i > 0$, i.e., for any $x, y \in X_i$ and any subgradient $\partial f_i(x)$ of f_i at x, $f_i(y) - f_i(x) - \partial f_i(x)^T (y - x) \geq \frac{\theta_i}{2} \|y - x\|^2$.

(b) $\mathrm{rel\,int} \bigcap_{i \in \mathcal{V}} X_i \neq \emptyset$.

Many engineering problems occurring in control, estimation, and machine learning on networked systems can be cast in the form of problem (1) satisfying Assumption 1, including LASSO regression [15], logistic regression [22], distributed model predictive control [3], robust estimation using pseudo Huber loss functions [27], maximum-likelihood parameter estimation [28], etc. Notice that Assumption 1(a) is a typical assumption for distributed optimization methods with convergence rate guarantees (e.g., [2], [3], [11], [12], [18], [22]). In addition, unlike many existing works that require each f_i to be continuously differentiable (e.g., [7], [8], [11]–[14], [17], [18], [22], [23]), here each f_i is not necessarily differentiable. Assumption 1(b) is an indispensable assumption to guarantee zero duality gap, which, along with Assumption 1(a), ensures a unique optimal solution $x^\star \in \bigcap_{i \in \mathcal{V}} X_i$ to problem (1).

We model the n agents and their interactions as an undirected graph $\mathcal{G}^k = (\mathcal{V}, \mathcal{E}^k)$ with time-varying topologies, where $k \in \{0, 1, \ldots\}$ represents time, $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes (i.e., the agents), and $\mathcal{E}^k \subseteq \{\{i, j\} : i, j \in \mathcal{V}, i \neq j\}$ is the set of links (i.e., the agent interactions) at time k. We assume that $\mathcal{E}^k \neq \emptyset \ \forall k \geq 0$. In addition, for each node $i \in \mathcal{V}$, we use $\mathcal{N}_i^k = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}^k\}$ to denote the set of its neighbors at time k.

To make the nodes cooperate, we impose the following connectivity condition on the time-varying graph $\mathcal{G}^k$:

Assumption 2 (B-connectivity)**.** *There exists a positive integer B such that for any $k \in \{0, 1, \ldots\}$, the graph $(\mathcal{V}, \bigcup_{t=kB}^{(k+1)B-1} \mathcal{E}^t)$ is connected.*

Assumption 2 says that each node must have an impact on the others during every B iterations, which is prevalent in the literature (e.g., [4], [6], [9]–[11], [16], [17], [20]).

III. Fenchel Dual Gradient Algorithms

In this section, we develop a family of distributed algorithms to solve problem (1) based on Fenchel duality.

A. Fenchel Dual Problem

We first transform (1) into the following equivalent problem:

$$\begin{array}{ll} \underset{\mathbf{x} \in \mathbb{R}^{nd}}{\text{minimize}} & F(\mathbf{x}) := \sum_{i \in \mathcal{V}} f_i(x_i) \\ \text{subject to} & x_i \in X_i, \quad \forall i \in \mathcal{V}, \\ & \mathbf{x} \in S, \end{array} \tag{2}$$

where $\mathbf{x} = (x_1^T, \ldots, x_n^T)^T$ and $S := \{\mathbf{x} \in \mathbb{R}^{nd} : x_1 = x_2 = \cdots = x_n\}$. Note that problem (2) has a unique optimal solution $\mathbf{x}^\star = ((x^\star)^T, \ldots, (x^\star)^T)^T$, where $x^\star \in \bigcap_{i \in \mathcal{V}} X_i$ is the unique optimum of problem (1). In addition, its optimal value $F^\star$ is equal to that of problem (1).

Next, we construct the Fenchel dual problem of (2). To this end, we introduce a function $q_i : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ for each $i \in \mathcal{V}$:

$$q_i(x_i, w_i) = w_i^T x_i - f_i(x_i).$$

The conjugate convex function $d_i : \mathbb{R}^d \to \mathbb{R}$ is then given by

$$d_i(w_i) = \sup_{x_i \in X_i} q_i(x_i, w_i).$$

Then, the Fenchel dual problem of (2) can be described as

$$\begin{array}{ll} \underset{\mathbf{w} \in \mathbb{R}^{nd}}{\text{maximize}} & -D(\mathbf{w}) := -\sum_{i \in \mathcal{V}} d_i(w_i) \\ \text{subject to} & \mathbf{w} \in S^{\perp}, \end{array} \tag{3}$$

where $\mathbf{w} = (w_1^T, \ldots, w_n^T)^T$ and $S^{\perp} := \{\mathbf{w} \in \mathbb{R}^{nd} : w_1 + w_2 + \cdots + w_n = \mathbf{0}_d\}$ is the orthogonal complement of S (cf. [29]). Note that (3) is a convex optimization problem. Also, with Assumption 1, it can be shown that strong duality between (2) and (3) holds, i.e., the optimal value $-D^\star$ of (3) equals $F^\star$, and that the optimal set of (3) is nonempty [29]. Moreover, $\mathbf{w}^\star = ((w_1^\star)^T, \ldots, (w_n^\star)^T)^T \in S^{\perp}$ is an optimal solution to (3) if and only if $\nabla d_i(w_i^\star) = \nabla d_j(w_j^\star) \ \forall i, j \in \mathcal{V}$, i.e., $\nabla D(\mathbf{w}^\star) \in S$ [25, Lemma 3.1].

Below we acquire a couple of properties regarding the Fenchel dual problem (3). Notice from Assumption 1(a) that for each $i \in \mathcal{V}$ and each $w_i \in \mathbb{R}^d$, there uniquely exists

$$\tilde{x}_i(w_i) := \arg\max_{x \in X_i} q_i(x, w_i). \tag{4}$$

Thus, from Danskin's theorem [29], d_i is differentiable and

$$\nabla d_i(w_i) = \tilde{x}_i(w_i).$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

The following proposition shows that d_i is smooth, i.e., ∇d_i is Lipschitz.

Proposition 1. *[2, Lemma II.1] Suppose Assumption 1 holds. Then, for each $i \in \mathcal{V}$, ∇d_i is Lipschitz continuous with Lipschitz constant $L_i = 1/\theta_i$, i.e., $\|\nabla d_i(u_i) - \nabla d_i(v_i)\| \leq L_i\|u_i - v_i\|$ $\forall u_i, v_i \in \mathbb{R}^d$.*

Likewise, we can see that $D(\mathbf{w})$ is differentiable and

$$\nabla D(\mathbf{w}) = \tilde{\mathbf{x}}(\mathbf{w}) := (\tilde{x}_1(w_1)^T, \ldots, \tilde{x}_n(w_n)^T)^T. \quad (5)$$

According to (4) and (5), if each w_i is known to node i, then the dual gradient $\nabla D(\mathbf{w})$ can be exactly evaluated in parallel by the nodes, while the Lagrange dual of problem (2) or its equivalent forms do not have such a favorable feature when the network is time-varying and not necessarily connected at each time instance. Further, notice that $F(\mathbf{x})$ in problem (2) is strongly convex over $X_1 \times \cdots \times X_n$ with convexity parameter $\theta_{\min} := \min_{i\in\mathcal{V}} \theta_i$. Also note that $D(\mathbf{w}) = \sup_{\mathbf{x}\in X_1\times\cdots\times X_n} \mathbf{w}^T\mathbf{x} - F(\mathbf{x})$. Like Proposition 1, we can establish the Lipschitz continuity of ∇D.

Corollary 1. *Suppose Assumption 1 holds. Then, ∇D is Lipschitz continuous with Lipschitz constant $L = 1/\theta_{\min}$.*

B. Algorithms

To solve the Fenchel dual problem (3), consider a class of weighted gradient methods as follows: Starting from an arbitrary $\mathbf{w}^0 \in S^\perp$, the subsequent iterates are generated by

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \alpha^k(H_{\mathcal{G}^k} \otimes I_d)\nabla D(\mathbf{w}^k), \quad \forall k \geq 0, \quad (6)$$

where $\alpha^k > 0$ is the step-size and $H_{\mathcal{G}^k} \in \mathbb{R}^{n\times n}$ is the weight matrix that depends on the topology of $\mathcal{G}^k$ and is defined as

$$[H_{\mathcal{G}^k}]_{ij} = \begin{cases} \sum_{s\in\mathcal{N}_i^k} h_{is}^k, & \text{if } i = j, \\ -h_{ij}^k, & \text{if } \{i,j\} \in \mathcal{E}^k, \\ 0, & \text{otherwise,} \end{cases} \quad \forall i, j \in \mathcal{V}. \quad (7)$$

We require $h_{ij}^k = h_{ji}^k > 0$ $\forall\{i,j\} \in \mathcal{E}^k$ $\forall k \geq 0$. We also assume that there exists a finite interval $[\underline{h}, \bar{h}]$ such that

$$h_{ij}^k \in [\underline{h}, \bar{h}] \subset (0, \infty), \quad \forall k \geq 0,\ \forall i \in \mathcal{V},\ \forall j \in \mathcal{N}_i^k. \quad (8)$$

Since $\mathcal{E}^k \neq \emptyset$, $H_{\mathcal{G}^k} \neq O_n$ for any $k \geq 0$. Moreover, $H_{\mathcal{G}^k}$ is symmetric positive semidefinite and $H_{\mathcal{G}^k}\mathbf{1}_n = \mathbf{0}_n$. Thus, using the same rationale as [24], [25], the proposition below shows that as long as $\mathbf{w}^0$ is feasible, so are $\mathbf{w}^k$ $\forall k \geq 1$.

Proposition 2. *Let $(\mathbf{w}^k)_{k=0}^\infty$ be the iterates generated by* (6)*. If $\mathbf{w}^0 \in S^\perp$, then $(\mathbf{w}^k)_{k=0}^\infty \subseteq S^\perp$.*

Remark 1. *The weighted gradient method* (6) *can be tuned to solve problems of minimizing $\sum_{i\in\mathcal{V}} d_i(w_i)$ subject to $\sum_{i\in\mathcal{V}} w_i = c$, $\forall c \in \mathbb{R}^d$. To do so, we can simply replace the initial condition $\mathbf{w}^0 \in S^\perp$ with $\sum_{i\in\mathcal{V}} w_i^0 = c$.*

Next, we introduce primal iterates to the weighted gradient method (6) that is intended for the Fenchel dual problem (3). Note from (7) and (5) that (6) can be written as

$$x_i^k = \tilde{x}_i(w_i^k), \quad \forall i \in \mathcal{V},$$

$$w_i^{k+1} = w_i^k - \alpha^k \sum_{j\in\mathcal{N}_i^k} h_{ij}^k(x_i^k - x_j^k), \quad \forall i \in \mathcal{V},$$

where $w_i^k \in \mathbb{R}^d$ is the ith d-dimensional block of $\mathbf{w}^k$ and $\tilde{x}_i(w_i^k)$ is defined in (4). We assign each w_i^k and x_i^k to node i as its dual and primal iterates, with x_i^k being node i's estimate on the optimal solution $x^\star$ of problem (1). Thus, the above algorithm with both dual and primal iterates can be implemented in a distributed and possibly asynchronous way on the time-varying network, as is shown in Algorithm 1.

Algorithm 1 Fenchel Dual Gradient Method

1: **Initialization:** Each node $i \in \mathcal{V}$ selects $w_i^0 \in \mathbb{R}^d$ so that $\sum_{j\in\mathcal{V}} w_j^0 = \mathbf{0}_d$ (or simply sets $w_i^0 = \mathbf{0}_d$), and sets $x_i^0 = \arg\max_{x\in X_i}(w_i^0)^T x - f_i(x)$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Each node $i \in \mathcal{V}$ with $\mathcal{N}_i^k \neq \emptyset$ sends its x_i^k to all $j \in \mathcal{N}_i^k$.
4: Upon receiving x_j^k $\forall j \in \mathcal{N}_i^k$, each node $i \in \mathcal{V}$ with $\mathcal{N}_i^k \neq \emptyset$ updates $w_i^{k+1} = w_i^k - \alpha^k \sum_{j\in\mathcal{N}_i^k} h_{ij}^k(x_i^k - x_j^k)$.
5: Each node $i \in \mathcal{V}$ with $\mathcal{N}_i^k \neq \emptyset$ computes $x_i^{k+1} = \arg\max_{x\in X_i}(w_i^{k+1})^T x - f_i(x)$.
6: Each node $i \in \mathcal{V}$ with $\mathcal{N}_i^k = \emptyset$ takes no action, i.e., $w_i^{k+1} = w_i^k$ and $x_i^{k+1} = x_i^k$.
7: **end for**

To implement Algorithm 1, each node i needs to select the weights h_{ij}^k $\forall j \in \mathcal{N}_i^k$ that satisfy $h_{ij}^k = h_{ji}^k$ in a predetermined interval $[\underline{h}, \bar{h}] \subset (0, \infty)$, where $\underline{h}$ and $\bar{h}$ may or may not be related with $\mathcal{G}^k$ $\forall k \geq 0$. This can be done through inexpensive interactions between neighboring nodes.

The remaining parameter to be determined is the step-size α^k. Later in Section IV, we will show that the following step-size condition is sufficient to guarantee the convergence of Algorithm 1: Suppose there is a finite interval $[\underline{\alpha}, \bar{\alpha}]$ such that

$$\alpha^k \in [\underline{\alpha}, \bar{\alpha}] \subset (0, 2/\delta), \quad \forall k \geq 0, \quad (9)$$

where $\delta > 0$ can be any positive constant satisfying

$$H_{\mathcal{G}^k} \preceq \delta\Lambda_L^{-1}, \quad \forall k \geq 0, \quad (10)$$

with $\Lambda_L := \mathrm{diag}(L_1, \ldots, L_n)$. Note that such δ always exists because Λ_L^{-1} is positive definite and $H_{\mathcal{G}^k}$ is positive semidefinite. For example, we may choose $\delta = L\sup_{k\geq 0}\lambda_1^\downarrow(H_{\mathcal{G}^k})$, where $L = 1/\theta_{\min} = \max_{i\in\mathcal{V}} L_i$. More conservatively, we can always let $\delta = L\bar{h}n$ and thus

$$[\underline{\alpha}, \bar{\alpha}] \subset (0, 2/(L\bar{h}n)).$$

Since $\bar{h}$ can be predetermined and known to all the nodes, the above condition only requires the nodes to obtain the global quantities n and $L = \max_{i\in\mathcal{V}} L_i$, which can be computed decentralizedly by some consensus schemes (e.g., [30]). Below, we provide two typical examples of $H_{\mathcal{G}^k}$ and the corresponding less conservative step-size ranges[1].

[1]We refer the readers to [26], the conference version of this paper, for detailed derivations in Examples 1 and 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

Example 1. *We may let $H_{\mathcal{G}^k}$ be the graph Laplacian matrix $L_{\mathcal{G}^k}$ of $\mathcal{G}^k$, i.e., $h^k_{ij}=h^k_{ji}=1\ \forall\{i,j\}\in\mathcal{E}^k$. Then, α^k can be selected in the interval $[\underline{\alpha},\bar{\alpha}]$ satisfying*

$$0<\underline{\alpha}\leq\bar{\alpha}<\frac{1}{\min\{\frac{L}{2}\sup_{k\geq 0}\lambda_1^{\downarrow}(L_{\mathcal{G}^k}),\sup_{k\geq 0}\max_{i\in\mathcal{V}}|\mathcal{N}_i^k|L_i\}}.$$

If the nodes interact in a gossiping pattern, i.e., each $\mathcal{E}^k$ contains only one link, then we may let $0<\underline{\alpha}\leq\bar{\alpha}<1/L$.

Example 2. *We may also let $H_{\mathcal{G}^k}$ be the Metropolis weight matrix [24], i.e., $h^k_{ij}=h^k_{ji}=1/\max\{|\mathcal{N}_i^k|L_i,|\mathcal{N}_j^k|L_j\}$ $\forall\{i,j\}\in\mathcal{E}^k$. Then, the step-sizes can be selected as*

$$0<\underline{\alpha}\leq\alpha^k\leq\bar{\alpha}<1,\quad\forall k\geq 0.$$

The underlying weighted gradient method (6) in Algorithm 1 can be viewed as a generalization of the distributed weighted gradient methods in [24], [25]. In particular, [24] proposes a class of weighted gradient methods in the form of (6) but with a constant weight matrix. The distributed implementation of such methods is enabled for undirected, fixed networks. Examples 1 and 2 extend the methods in [24] as well as their step-size conditions to handle time-varying networks. On the other hand, by setting $H_{\mathcal{G}^k}$ to the graph Laplacian matrix and $\alpha^k=1/(2nL)\ \forall k\geq 0$, (6) reduces to the algorithm in [25], which considers time-varying undirected networks under Assumption 2. Note that Example 1 provides a much broader step-size range for this particular weight matrix.

IV. Convergence Analysis

This section is dedicated to analyzing the convergence of the Fenchel dual gradient methods in Algorithm 1.

We first show that $(D(\mathbf{w}^k))_{k=0}^{\infty}$ is non-increasing.

Lemma 1. *Suppose Assumption 1 holds. Let $(\mathbf{w}^k)_{k=0}^{\infty}$ be the dual iterates generated by Algorithm 1. If the step-sizes $(\alpha^k)_{k=0}^{\infty}$ satisfy* (9)*, then for each $k\geq 0$,*

$$D(\mathbf{w}^{k+1})-D(\mathbf{w}^k)\leq-\rho\nabla D(\mathbf{w}^k)^T(H_{\mathcal{G}^k}\otimes I_d)\nabla D(\mathbf{w}^k),$$

where $\rho:=\min\{\underline{\alpha}-\frac{\underline{\alpha}^2\delta}{2},\bar{\alpha}-\frac{\bar{\alpha}^2\delta}{2}\}\in(0,\infty)$, with $\underline{\alpha},\bar{\alpha}>0$ in (9) *and $\delta>0$ in* (10)*.*

Proof. See Appendix A. □

Next, we attempt to bound the accumulative drop in D during every B iterations. To this end, for each $k\geq 0$, let $\tilde{\mathcal{G}}^k=(\mathcal{V},\tilde{\mathcal{E}}^k)$ be *any* spanning subgraph of $(\mathcal{V},\bigcup_{t=k}^{k+B-1}\mathcal{E}^t)$, which, owing to Assumption 2, is chosen to be connected at $k\in\{0,B,2B,\ldots\}$. Also let ϖ^k be the maximum degree of $\tilde{\mathcal{G}}^k$ and $\bar{\varpi}:=\sup_{t\in\{0,1,\ldots\}}\varpi^{tB}$. Clearly, $1\leq\bar{\varpi}\leq n-1$.

Lemma 2. *Suppose Assumptions 1 and 2 hold. Let $(\mathbf{w}^k)_{k=0}^{\infty}$ be the dual iterates generated by Algorithm 1. If the step-sizes $(\alpha^k)_{k=0}^{\infty}$ satisfy* (9)*, then for each $k\in\{0,B,2B,\ldots\}$,*

$$\sum_{t=k}^{k+B-1}\nabla D(\mathbf{w}^t)^T(H_{\mathcal{G}^t}\otimes I_d)\nabla D(\mathbf{w}^t)\geq\nabla D(\mathbf{w}^k)^T(L_{\tilde{\mathcal{G}}^k}\otimes I_d)\nabla D(\mathbf{w}^k)/\eta,\tag{11}$$

where $L_{\tilde{\mathcal{G}}^k}$ is the graph Laplacian matrix of $\tilde{\mathcal{G}}^k$ and $\eta:=3B\bar{\varpi}\bar{\alpha}^2\delta L+3/\underline{h}\in(0,\infty)$, with $\bar{\alpha}>0$ in (9)*, $\delta>0$ in* (10)*, $L>0$ in Corollary 1, and $\underline{h}>0$ in* (8)*.*

Proof. See Appendix B. □

For the particular choice $H_{\mathcal{G}^k}=L_{\mathcal{G}^k}$ and $\alpha^k=1/(2nL)$, [25, Lemma A.9] provides a similar bound to (11) with η replaced by $3B/2$ and $\tilde{\mathcal{G}}^k$ being a spanning tree. Lemma 2 improves this bound since $\eta\leq 3B/4+3$ in this case. It also sheds light on how the network topologies come into play for more general selections of $H_{\mathcal{G}^k}$ and α^k.

Lemmas 1 and 2 together bound the decrease in the value of D during every B iterations, with which we are able to derive both dual and primal convergence rates. Prior to doing that, we define a sequence $(\tilde{M}_k)_{k=0}^{\infty}$ as $\tilde{M}_0=\tilde{M}_1$ and

$$\tilde{M}_k=\max_{t=0,\ldots,k-1}\min_{\mathbf{w}^\star\in S^\perp:D(\mathbf{w}^\star)=D^\star}\|\mathbf{w}^{tB}-\mathbf{w}^\star\|,\ \forall k\geq 1.\tag{12}$$

Theorem 1. *Suppose Assumptions 1 and 2 hold. Let $(\mathbf{w}^k)_{k=0}^{\infty}$ and $(\mathbf{x}^k)_{k=0}^{\infty}$ be the dual and primal iterates generated by Algorithm 1, respectively. If the step-sizes $(\alpha^k)_{k=0}^{\infty}$ satisfy* (9)*, then for each $k\geq 0$,*

$$D(\mathbf{w}^k)-D^\star\leq\frac{\eta\tilde{M}^2_{\lfloor k/B\rfloor}(D(\mathbf{w}^0)-D^\star)}{\eta\tilde{M}^2_{\lfloor k/B\rfloor}+\rho\underline{\lambda}(D(\mathbf{w}^0)-D^\star)\lfloor k/B\rfloor},\tag{13}$$

$$\|P_{S^\perp}(\mathbf{x}^k)\|\leq\|\mathbf{x}^k-\mathbf{x}^\star\|\leq\sqrt{\frac{2L\eta\tilde{M}^2_{\lfloor k/B\rfloor}(D(\mathbf{w}^0)-D^\star)}{\eta\tilde{M}^2_{\lfloor k/B\rfloor}+\rho\underline{\lambda}(D(\mathbf{w}^0)-D^\star)\lfloor k/B\rfloor}},\tag{14}$$

$$F(\mathbf{x}^k)-F^\star\leq\|\mathbf{w}^k\|\sqrt{\frac{2L\eta\tilde{M}^2_{\lfloor k/B\rfloor}(D(\mathbf{w}^0)-D^\star)}{\eta\tilde{M}^2_{\lfloor k/B\rfloor}+\rho\underline{\lambda}(D(\mathbf{w}^0)-D^\star)\lfloor k/B\rfloor}},$$

$$F(\mathbf{x}^k)-F^\star\geq-\|\mathbf{w}^\star\|\sqrt{\frac{2L\eta\tilde{M}^2_{\lfloor k/B\rfloor}(D(\mathbf{w}^0)-D^\star)}{\eta\tilde{M}^2_{\lfloor k/B\rfloor}+\rho\underline{\lambda}(D(\mathbf{w}^0)-D^\star)\lfloor k/B\rfloor}},$$

where $\tilde{M}_{\lfloor k/B\rfloor}\geq 0$ is defined in (12)*, $\underline{\lambda}:=\inf_{t\in\{0,1,\ldots\}}\lambda_{n-1}^{\downarrow}(L_{\tilde{\mathcal{G}}^{tB}})\in(0,\infty)$, $\eta>0$ is given in Lemma 2, $\rho>0$ is given in Lemma 1, $\mathbf{w}^\star$ is any optimal solution of problem* (3)*, and L is given in Corollary 1.*

Proof. See Appendix C. □

Theorem 1 states that the error $D(\mathbf{w}^k)-D^\star$ in dual optimality converges to zero at an $O(1/k)$ rate, and the errors $\|\mathbf{x}^k-\mathbf{x}^\star\|$ and $|F(\mathbf{x}^k)-F^\star|$ in primal optimality as well as the primal infeasibility $\|P_{S^\perp}(\mathbf{x}^k)\|$ all converge to zero at $O(1/\sqrt{k})$ rates, provided that the dual iterate $\mathbf{w}^k$ is uniformly bounded. From Lemma 1 and Proposition 2, the compactness of the level sets $S_0(\mathbf{w}):=\{\mathbf{w}'\in S^\perp:D(\mathbf{w}')\leq D(\mathbf{w})\}$ $\forall\mathbf{w}\in S^\perp$ suffices to guarantee the boundedness of $\mathbf{w}^k$. As is shown in the following proposition, such level sets are compact if the global constraint set $\cap_{i\in\mathcal{V}}X_i$ has a nonempty interior, which is commonly assumed in existing works on constrained distributed optimization (e.g., [4], [6], [9], [20]).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

Proposition 3. *Suppose Assumption 1 holds. Also suppose* $\operatorname{int}\cap_{i\in\mathcal{V}}X_i \neq \emptyset$. *Then, the level sets* $S_0(\mathbf{w})$ $\forall \mathbf{w} \in S^\perp$ *are compact. In addition, for any dual optimum* $\mathbf{w}^\star \in S^\perp$,

$$\|\mathbf{w}^\star\| \leq \frac{(\sum_{i\in\mathcal{V}} \max_{x_i\in B(x',r_c)} f_i(x_i)) - F^\star}{r_c} < \infty, \quad (15)$$

where $x' \in \mathbb{R}^d$ *is an arbitrary vector in* $\operatorname{int}\cap_{i\in\mathcal{V}}X_i$ *and* $r_c \in (0,\infty)$ *is such that* $B(x',r_c) \subseteq \cap_{i\in\mathcal{V}}X_i$.

Proof. See Appendix D. □

Remark 2. *Inequality* (13) *is a new convergence rate result for weighted gradient methods in the form of* (6). *It eliminates the assumption on the strong convexity of* D *in [24]. Also, it is stronger than the convergence results in [25]. In particular, [25] only proves asymptotic convergence of* $D(\mathbf{w}^k)$ *to* $D^\star$ *and* $\min_{t=1,\ldots,T}\|P_{S^\perp}(\nabla D(\mathbf{w}^{tB}))\|^2 \leq C\cdot n^3B/T$ *for some* $C>0$. *Note that* (13) *provides a rate for* $D(\mathbf{w}^k)\to D^\star$. *Moreover, since* $\nabla D(\mathbf{w}^k)=\mathbf{x}^k$, (14) *is comparable to and slightly stronger than the above rate result in [25]. Further, the* $O(1/\sqrt{k})$ *primal convergence rates in Theorem 1 commensurate with the convergence rate of the classic (centralized) subgradient projection method [31].*

Based on Theorem 1, below we derive a bound on the number of iterations needed to guarantee $\|\mathbf{x}^k-\mathbf{x}^\star\|\leq\epsilon$ for any given accuracy $\epsilon>0$.

Proposition 4. *Suppose all the conditions in Theorem 1 hold. Also suppose* $\operatorname{int}\cap_{i\in\mathcal{V}}X_i\neq\emptyset$. *Let* $\alpha^k=1/\delta$ $\forall k\geq 0$, *where* $\delta>0$ *is given by* (10). *Then, for any* $\epsilon>0$, $\|\mathbf{x}^k-\mathbf{x}^\star\|\leq\epsilon$ *if*

$$k\geq\frac{3}{2}n(n-1)B\Big((n-1)BL+\frac{\delta}{\underline{h}}\Big)\Big(\frac{2L}{\epsilon^2}-\frac{1}{D(\mathbf{w}^0)+F(\mathbf{x}')}\Big)\cdot\left(\max_{t=0,\ldots,\lfloor\frac{k}{B}\rfloor}\|\mathbf{w}^{tB}\|+\sum_{i\in\mathcal{V}}\frac{\max_{x_i\in B(x',r_i)}f_i(x_i)+d_i(w_i^0)}{\min_{j\in\mathcal{V}}r_j}\right)^2, \quad (16)$$

where $\mathbf{x}'=((x')^T,\ldots,(x')^T)^T$ *is an arbitrary vector in* $S\cap(\operatorname{int}X_1\times\cdots\times\operatorname{int}X_n)$ *and each* $r_i>0$, $i\in\mathcal{V}$ *is such that* $B(x',r_i)\subseteq X_i$.

Proof. See Appendix E. □

Since $\|\mathbf{w}^k\|$ is bounded, (16) is guaranteed to hold for sufficiently large k. Also, as $\delta/\underline{h}\leq O(n)$ (cf. Section III-B), (16) implies that the worst-case iteration complexity for Algorithm 1 to reach ϵ-accuracy in primal optimality is $O(n^3B^2/\epsilon^2)$. If more about the union of $\mathcal{G}^k$ over every B iterations is revealed, the iteration complexity may be improved with lower order of n. Furthermore, Proposition 4 considers a constant step-size $1/\delta$ for better presenting the result. Indeed, similar iteration complexities can be derived for more general step-sizes satisfying (9).

A *stopping criterion* for every node can be obtained through (16): Upon completing iteration $k=0,B,2B,\ldots$, each node checks whether (16) holds and stops updating as soon as (16) is satisfied. To do so, the nodes need to keep track of the largest $\|\mathbf{w}^{tB}\|$, $t\in\{0,1,\ldots\}$ in history. This can be realized in a decentralized way by letting the nodes aggregate $\|w_i^{tB}\|$ at each $k=tB$ via consensus schemes (e.g., [30]). Decentralized computation/estimation of the remaining global quantities appearing in (16) can also be done using consensus, once and for all.

A. Comparison with related algorithms

Finally, we compare the Fenchel dual gradient methods in Algorithm 1 with the existing distributed optimization algorithms that also have guaranteed *convergence rates* over *time-varying* networks, including Subgradient-Push [10], Gradient-Push [11], DIGing [12], and Push-DIGing [12]. Table I lists their assumptions, convergence rate in primal optimality, and scalability of iteration complexity with respect to n and B, leading to the following observations:

1) Only Algorithm 1 addresses problems with different local constraints of the nodes, while the existing algorithms all require the problem to be unconstrained and their extensions to constrained problems are still open challenges.
2) Gradient-Push, DIGing, and Push-DIGing require both strong convexity and smoothness of the f_i's, while Algorithm 1 and Subgradient-Push allow the f_i's to be nonsmooth. Naturally, the former have better convergence rates than the latter. In fact, when each f_i is both strongly convex and smooth, Algorithm 1 can also achieve a linear convergence rate like DIGing and Push-DIGing. This result is omitted due to space limitation.
3) Subgradient-Push does not need strong convexity of each f_i, but its convergence rate is slower than that of Algorithm 1. Note that Algorithm 1 does not necessarily require stronger assumptions than Subgradient-Push, as Subgradient-Push requires the subgradients of each f_i to be uniformly bounded over $\mathbb{R}^d$ but Algorithm 1 does not.
4) The iteration complexities of Subgradient-Push, Gradient-Push, and Push-DIGing are exponential in n and B, although they allow directed links. The iteration complexities of DIGing and Algorithm 1 are polynomial in n and B, with that of Algorithm 1 the best.

V. Numerical Example

In addition to the theoretical comparison in Section IV-A, we further compare Algorithm 1 with the existing methods in Table I via numerical examples.

Consider the following logistic regression problem that can be used to estimate parameters of a logistic model or learn a linear classifier [22]:

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad \frac{\lambda}{2}\|x\|^2+\sum_{i\in\mathcal{V}}\sum_{j=1}^{J}\log\left(1+e^{-(u_{ij}^Tx)v_{ij}}\right), \quad (17)$$

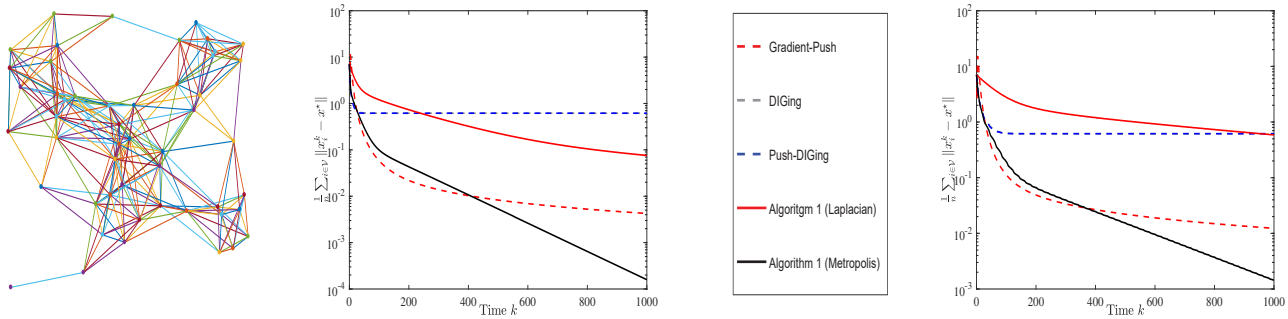
where $\lambda>0$, $u_{ij}\in\mathbb{R}^d$ is the feature with the dth element equal to 1, and $v_{ij}\in\{-1,1\}$ is the label. Observe that (17) is in the form of (1) with each $f_i(x)=\frac{\lambda}{2n}\|x\|^2+\sum_{j=1}^{J}\log\left(1+e^{-(u_{ij}^Tx)v_{ij}}\right)$ and it satisfies not only Assumption 1 but the more restrictive assumptions for Gradient-Push [11], DIGing [12], and Push-DIGing [12] as well.

In the simulations, we consider a 50-node network, with B set to be 5 and 20, respectively. To create a B-connected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

Algorithm	unconstrained problem	strongly convex	Lipschitz gradient	bounded subgradient	undirected links	convergence rate	scalability w.r.t. n and B
Subgradient-Push [10]	$\surd$			$\surd$		$O(\ln k/\sqrt{k})$	$O(n^{2nB})$
Gradient-Push [11]	$\surd$	$\surd$	$\surd$			$O(\ln k/k)$	$O(n^{2nB})$
DIGing [12]	$\surd$	$\surd$	$\surd$		$\surd$	$O(q^k)$, $0<q<1$	$O(n^{4.5}B^3)$
Push-DIGing [12]	$\surd$	$\surd$	$\surd$			$O(q^k)$, $0<q<1$	$O(n^{n^2B^2})$
Algorithm 1		$\surd$			$\surd$	$O(1/\sqrt{k})$	$O(n^3B^2)$

TABLE I: Comparison of Algorithm 1 and related methods in assumptions, convergence rate, and scalability. Here, $\surd$ means the assumption is required.



(a) A 50-node network with all possible 284 links

(b) $B = 5$

(c) $B = 20$

Fig. 1: Convergence performance of Gradient-Push, DIGing, Push-DIGing, and Algorithm 1.

time-varying graph $\mathcal{G}^k = (\mathcal{V}, \mathcal{E}^k)$, we first generate a connected random geometric graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$, as is shown in Figure 1(a). Then, we divide $\mathcal{E}'$ into B subsets, and each subset is cyclically selected to be $\mathcal{E}^k$, $k = 0, 1, \ldots$. The problem data are generated as follows: Let $\lambda = 2$, $d = 5$, and $J = 6$. For each $i \in \mathcal{V}$, we let $v_{ij} = 1$ for $j = 1, \ldots, J/2$ and $v_{ij} = -1$ otherwise. Then, the first $d-1$ elements of u_{ij} are drawn from normal distribution with mean v_{ij} and variance 0.5, and the dth element is fixed to 1.

With the above settings, we simulate Algorithm 1, Gradient-Push [11], DIGing [12], and Push-DIGing [12], which all have convergence rate guarantees on time-varying networks. Note that Subgradient-Push [10] reduces to Gradient-Push in this case and is therefore omitted. For Algorithm 1, we consider two weight matrices, i.e., the graph Laplacian weight and the Metropolis weight given in Examples 1 and 2, with step-size $\alpha^k = 1/(nL)$ and $\alpha^k = 1/2$, respectively. For the remaining algorithms, we fine-tune every parameter within the range that guarantees the theoretical convergence results.

Figure 1(b) and 1(c) plot the average distances between the optimum and the primal iterates generated by the aforementioned algorithms. Observe that Algorithm 1 with the Metropolis weight prominently outperforms the remaining methods. Gradient-Push is the second best. DIGing and Push-DIGing exhibit almost the same performance and converge much slower. In the case where $B = 20$, although Algorithm 1 with graph Laplacian weight has a larger primal error than DIGing and Push-DIGing on the figure, it tends to reach higher accuracy eventually. The reason for the competitive performance of Algorithm 1 may be that exact dual gradients are calculated in parallel by the nodes at each iteration, while the other methods only approximate the global gradients. Further, by comparing Figure 1(b) with 1(c), it can be seen that smaller B leads to faster convergence of Algorithm 1, which is consistent with Theorem 1.

We also compare Algorithm 1 with another two methods in [4], [20] for solving constrained, nonsmooth optimization problems. Due to space limitation, we refer the readers to [26].

VI. Conclusion

We have constructed a family of distributed Fenchel dual gradient methods for solving multi-agent optimization problems with strongly convex but nonsmooth local objectives and nonidentical local constraints over time-varying networks. The proposed algorithms have an $O(1/\sqrt{k})$ convergence rate under a standard connectivity condition. Thorough comparisons with related methods in both theoretical and numerical results are provided, which demonstrate the competitive performance of the proposed algorithms. In future, this work may be extended in a few directions such as problems with general convex objective functions and networks with directed links.

Appendix

A. Proof of Lemma 1

For convenience, let $\mathbf{y}^k = (H_{\mathcal{G}^k} \otimes I_d)\nabla D(\mathbf{w}^k)$. Due to the Descent Lemma [29] and (6), we have

$$\begin{aligned} D(\mathbf{w}^{k+1}) - D(\mathbf{w}^k) &\le \langle \nabla D(\mathbf{w}^k), \mathbf{w}^{k+1} - \mathbf{w}^k \rangle \\ &\quad + (\mathbf{w}^{k+1} - \mathbf{w}^k)^T \frac{\Lambda_L \otimes I_d}{2} (\mathbf{w}^{k+1} - \mathbf{w}^k) \\ &= -\alpha^k \langle \nabla D(\mathbf{w}^k), \mathbf{y}^k \rangle + (\alpha^k)^2 (\mathbf{y}^k)^T \frac{\Lambda_L \otimes I_d}{2} \mathbf{y}^k. \end{aligned} \quad (18)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

Then, consider the following lemma.

Lemma 3. *Suppose $M, \bar{M} \in \mathbb{R}^{n\times n}$ are symmetric positive semidefinite and $M \preceq \bar{M}$. Then, for any $\mathbf{x} \in \mathbb{R}^{nd}$ and any $\mathbf{y} \in \mathcal{R}(M \otimes I_d)$,*

$$\langle \mathbf{x}, (M \otimes I_d)\mathbf{x}\rangle \geq \langle (M \otimes I_d)\mathbf{x}, (\bar{M}^\dagger \otimes I_d)(M \otimes I_d)\mathbf{x}\rangle.$$

Proof. Let $\mathbf{x} \in \mathbb{R}^{nd}$. Then,

$$\begin{aligned}&\langle \mathbf{x}, (M \otimes I_d)\mathbf{x}\rangle - \langle (M \otimes I_d)\mathbf{x}, (\bar{M}^\dagger \otimes I_d)(M \otimes I_d)\mathbf{x}\rangle\\ &\quad = \mathbf{x}^T[(M - M\bar{M}^\dagger M) \otimes I_d]\mathbf{x}. \end{aligned} \tag{19}$$

In addition, by Schur complement condition, $M \succeq O_n$ and $\bar{M} \succeq M$ imply

$$\begin{pmatrix} M & M \\ M & \bar{M} \end{pmatrix} \succeq O_{2n}$$

and the inequality above leads to $M - M\bar{M}^\dagger M \succeq O_n$. Combining this with (19), we complete the proof. □

From Lemma 3, $(\mathbf{y}^k)^T(\Lambda_L \otimes I_d)\mathbf{y}^k \leq \delta\langle \nabla D(\mathbf{w}^k), \mathbf{y}^k\rangle$. It follows from (18) that $D(\mathbf{w}^{k+1}) - D(\mathbf{w}^k) \leq ((\alpha^k)^2\delta/2 - \alpha^k)\langle \nabla D(\mathbf{w}^k), \mathbf{y}^k\rangle$. This and (9) then complete the proof.

B. Proof of Lemma 2

Let $k \in \{0, B, 2B, \ldots\}$. For each $\{i,j\} \in \tilde{\mathcal{E}}^k$, let $t^k_{\{i,j\}} \in \{k, \ldots, k+B-1\}$ be such that $\{i,j\} \in \mathcal{E}^{t^k_{\{i,j\}}}$. Then, note from Proposition 1 that $\|\nabla d_i(w_i^k) - \nabla d_i(w_i^{t^k_{\{i,j\}}})\|^2 = \|\sum_{t=k}^{t^k_{\{i,j\}}-1}(\nabla d_i(w_i^{t+1}) - \nabla d_i(w_i^t))\|^2 \leq B\sum_{t=k}^{k+B-1}\|\nabla d_i(w_i^{t+1}) - \nabla d_i(w_i^t)\|^2 \leq L_i^2 B\sum_{t=k}^{k+B-1}\|w_i^{t+1} - w_i^t\|^2$. Thus,

$$\begin{aligned}&\sum_{\{i,j\}\in\tilde{\mathcal{E}}^k}(\|\nabla d_i(w_i^k) - \nabla d_i(w_i^{t^k_{\{i,j\}}})\|^2 + \|\nabla d_j(w_j^{t^k_{\{i,j\}}}) - \nabla d_j(w_j^k)\|^2)\\ &\leq B\sum_{\{i,j\}\in\tilde{\mathcal{E}}^k}\sum_{t=k}^{k+B-1}(L_i^2\|w_i^{t+1} - w_i^t\|^2 + L_j^2\|w_j^{t+1} - w_j^t\|^2)\\ &\leq B\bar{\varpi}\sum_{t=k}^{k+B-1}\sum_{i\in\mathcal{V}}L_i^2\|w_i^{t+1} - w_i^t\|^2\\ &\leq B\bar{\varpi}\bar{\alpha}^2\sum_{t=k}^{k+B-1}\langle \nabla D(\mathbf{w}^t), ((H_{\mathcal{G}^t}\Lambda_L^2 H_{\mathcal{G}^t}) \otimes I_d)\nabla D(\mathbf{w}^t)\rangle.\end{aligned}$$

Note that $H_{\mathcal{G}^t}\Lambda_L^2 H_{\mathcal{G}^t} \preceq LH_{\mathcal{G}^t}\Lambda_L H_{\mathcal{G}^t}$. Also, from (10) and Lemma 3, $H_{\mathcal{G}^t}\Lambda_L H_{\mathcal{G}^t} \preceq \delta H_{\mathcal{G}^t}$. As a result,

$$\begin{aligned}&\sum_{\{i,j\}\in\tilde{\mathcal{E}}^k}(\|\nabla d_i(w_i^k) - \nabla d_i(w_i^{t^k_{\{i,j\}}})\|^2 + \|\nabla d_j(w_j^{t^k_{\{i,j\}}}) - \nabla d_j(w_j^k)\|^2)\\ &\leq B\bar{\varpi}\bar{\alpha}^2\delta L\sum_{t=k}^{k+B-1}\nabla D(\mathbf{w}^t)^T(H^t_{\mathcal{G}} \otimes I_d)\nabla D(\mathbf{w}^t).\end{aligned} \tag{20}$$

In addition,

$$\begin{aligned}&\textstyle\sum_{\{i,j\}\in\tilde{\mathcal{E}}^k}\|\nabla d_i(w_i^{t^k_{\{i,j\}}}) - \nabla d_j(w_j^{t^k_{\{i,j\}}})\|^2\\ &\leq \frac{1}{\underline{h}}\textstyle\sum_{t=k}^{k+B-1}\sum_{\{i,j\}\in\mathcal{E}^t}h_{ij}^k\|\nabla d_i(w_i^t) - \nabla d_j(w_j^t)\|^2\\ &\leq \frac{1}{\underline{h}}\textstyle\sum_{t=k}^{k+B-1}\nabla D(\mathbf{w}^t)^T(H^t_{\mathcal{G}} \otimes I_d)\nabla D(\mathbf{w}^t).\end{aligned}$$

This, along with (20), implies $\nabla D(\mathbf{w}^k)^T(L_{\tilde{\mathcal{G}}^k} \otimes I_d)\nabla D(\mathbf{w}^k) = \sum_{\{i,j\}\in\tilde{\mathcal{E}}^k}\|\nabla d_i(w_i^k) - \nabla d_j(w_j^k)\|^2 \leq 3\sum_{\{i,j\}\in\tilde{\mathcal{E}}^k}(\|\nabla d_i(w_i^k) - \nabla d_i(w_i^{t^k_{\{i,j\}}})\|^2 + \|\nabla d_j(w_j^{t^k_{\{i,j\}}}) - \nabla d_j(w_j^k)\|^2 + \|\nabla d_i(w_i^{t^k_{\{i,j\}}}) - \nabla d_j(w_j^{t^k_{\{i,j\}}})\|^2) \leq \eta\sum_{t=k}^{k+B-1}\nabla D(\mathbf{w}^t)^T(H^t_{\mathcal{G}} \otimes I_d)\nabla D(\mathbf{w}^t)$.

C. Proof of Theorem 1

Let $k \geq 0$. By Lemmas 1 and 2,

$$\begin{aligned}&\big(D(\mathbf{w}^{(k+1)B}) - D^\star\big) - \big(D(\mathbf{w}^{kB}) - D^\star\big)\\ &= \textstyle\sum_{t=kB}^{(k+1)B-1}(D(\mathbf{w}^{t+1}) - D(\mathbf{w}^t))\\ &\leq -\rho\textstyle\sum_{t=kB}^{(k+1)B-1}\nabla D(\mathbf{w}^t)^T(H_{\mathcal{G}^t} \otimes I_d)\nabla D(\mathbf{w}^t)\\ &\leq -\rho\nabla D(\mathbf{w}^{kB})^T(L_{\tilde{\mathcal{G}}^{kB}} \otimes I_d)\nabla D(\mathbf{w}^{kB})/\eta\\ &\leq -\rho\underline{\lambda}\|P_{S^\perp}(\nabla D(\mathbf{w}^{kB}))\|^2/\eta,\end{aligned} \tag{21}$$

where the last inequality is because $\tilde{\mathcal{G}}^{kB}$ is connected and thus $\mathrm{Null}(L_{\tilde{\mathcal{G}}^{kB}} \otimes I_d) = S$. Also, since $\tilde{\mathcal{G}}^{tB}$ $\forall t = 0, 1, \ldots$ are connected, we have $\underline{\lambda} > 0$. From Proposition 2, we know that $\mathbf{w}^{kB} \in S^\perp$. Also, for any optimal solution $\mathbf{w}^\star$ to (3), because $\mathbf{w}^\star \in S^\perp$, we have $\mathbf{w}^{kB} - \mathbf{w}^\star \in S^\perp$. Then,

$$\begin{aligned}D(\mathbf{w}^{kB}) - D^\star &\leq \langle \nabla D(\mathbf{w}^{kB}), \mathbf{w}^{kB} - \mathbf{w}^\star\rangle\\ &= \langle P_{S^\perp}(\nabla D(\mathbf{w}^{kB})), \mathbf{w}^{kB} - \mathbf{w}^\star\rangle\\ &\leq \|P_{S^\perp}(\nabla D(\mathbf{w}^{kB}))\| \cdot \|\mathbf{w}^{kB} - \mathbf{w}^\star\|.\end{aligned}$$

This, along with (21), gives $\big(D(\mathbf{w}^{(k+1)B}) - D^\star\big) - \big(D(\mathbf{w}^{kB}) - D^\star\big) \leq -\rho\underline{\lambda}\big(D(\mathbf{w}^{kB}) - D^\star\big)^2/(\eta\min_{\mathbf{w}^\star\in S^\perp : D(\mathbf{w}^\star) = D^\star}\|\mathbf{w}^{kB} - \mathbf{w}^\star\|^2)$. Finally, using Lemma 6 in [32, Sec. 2.2.1], we obtain

$$\begin{aligned}&D(\mathbf{w}^{kB}) - D^\star\\ &\leq \frac{D(\mathbf{w}^0) - D^\star}{1 + \frac{\rho\underline{\lambda}(D(\mathbf{w}^0) - D^\star)}{\eta}\sum_{t=0}^{k-1}(\min_{\mathbf{w}^\star\in S^\perp : D(\mathbf{w}^\star) = D^\star}\|\mathbf{w}^{tB} - \mathbf{w}^\star\|^2)^{-1}}\\ &\leq \frac{D(\mathbf{w}^0) - D^\star}{1 + \rho\underline{\lambda}(D(\mathbf{w}^0) - D^\star)k/(\eta\tilde{M}_k^2)}.\end{aligned}$$

Note that the above inequality is equivalent to (13) since $(D(\mathbf{w}^k))_{k=0}^\infty$ is non-increasing.

Now let $\mathbf{w} \in S^\perp$. Note that $\|P_{S^\perp}(\tilde{\mathbf{x}}(\mathbf{w}))\| = \|\tilde{\mathbf{x}}(\mathbf{w}) - P_S(\tilde{\mathbf{x}}(\mathbf{w}))\| \leq \|\tilde{\mathbf{x}}(\mathbf{w}) - \mathbf{x}^\star\|$. Also, due to Corollary 1, [31, Theorem 2.1.5], and (5), $D(\mathbf{w}) - D^\star \geq \langle \nabla D(\mathbf{w}^\star), \mathbf{w} - \mathbf{w}^\star\rangle + \frac{1}{2L}\|\nabla D(\mathbf{w}) - \nabla D(\mathbf{w}^\star)\|^2 = \frac{1}{2L}\|\tilde{\mathbf{x}}(\mathbf{w}) - \mathbf{x}^\star\|^2$, where the last equality is because $\nabla D(\mathbf{w}^\star) = \mathbf{x}^\star \in S$ and $\mathbf{w}, \mathbf{w}^\star \in S^\perp$. It follows that

$$\|P_{S^\perp}(\tilde{\mathbf{x}}(\mathbf{w}))\| \leq \|\tilde{\mathbf{x}}(\mathbf{w}) - \mathbf{x}^\star\| \leq \sqrt{2L(D(\mathbf{w}) - D^\star)}. \tag{22}$$

Also note that $F(\tilde{\mathbf{x}}(\mathbf{w})) - F^\star = \langle \mathbf{w}, \tilde{\mathbf{x}}(\mathbf{w})\rangle - D(\mathbf{w}) + D^\star \leq \langle \mathbf{w}, \tilde{\mathbf{x}}(\mathbf{w})\rangle = \langle \mathbf{w}, P_{S^\perp}(\tilde{\mathbf{x}}(\mathbf{w}))\rangle$. On the other hand, for any dual optimum $\mathbf{w}^\star \in S^\perp$, we have $-F^\star = D^\star \geq \langle \mathbf{w}^\star, \tilde{\mathbf{x}}(\mathbf{w})\rangle - F(\tilde{\mathbf{x}}(\mathbf{w}))$, which leads to $F(\tilde{\mathbf{x}}(\mathbf{w})) - F^\star \geq \langle \mathbf{w}^\star, P_{S^\perp}(\tilde{\mathbf{x}}(\mathbf{w}))\rangle$. As a result,

$$\begin{aligned}-\|\mathbf{w}^\star\| \cdot \|P_{S^\perp}(\tilde{\mathbf{x}}(\mathbf{w}))\| &\leq F(\tilde{\mathbf{x}}(\mathbf{w})) - F^\star\\ &\leq \|\mathbf{w}\| \cdot \|P_{S^\perp}(\tilde{\mathbf{x}}(\mathbf{w}))\|.\end{aligned} \tag{23}$$

Combining (22) and (23) with Proposition 2 and (13) completes the proof.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2901829, IEEE Transactions on Automatic Control

D. Proof of Proposition 3

Consider the following two cases:

Case I: $\mathbf{0}_d \in \operatorname{int}\bigcap_{i\in\mathcal{V}} X_i$. In this case, there exists $r_c \in (0,\infty)$ such that $B(\mathbf{0}_d, r_c) \subseteq \bigcap_{i\in\mathcal{V}} X_i$. Let $\mathbf{w}^\star = ((w_1^\star)^T, \ldots, (w_n^\star)^T)^T$ be an optimal solution of problem (3). For each $i \in \mathcal{V}$, if $w_i^\star \neq \mathbf{0}_d$, let $y_i = r_c \frac{w_i^\star}{\|w_i^\star\|}$; otherwise let $y_i = \mathbf{0}_d$. Clearly, $y_i \in B(\mathbf{0}_d, r_c)$. Consequently, $D^\star = D(\mathbf{w}^\star) = \sum_{i\in\mathcal{V}}\Big(\sup_{x_i\in X_i}(w_i^\star)^T x_i - f_i(x_i)\Big) \geq \sum_{i\in\mathcal{V}}\Big((w_i^\star)^T y_i - f_i(y_i)\Big) = r_c\sum_{i\in\mathcal{V}}\|w_i^\star\| - \sum_{i\in\mathcal{V}} f_i(y_i)$. This, along with $\|\mathbf{w}^\star\| \leq \sum_{i\in\mathcal{V}}\|w_i^\star\|$ and $D^\star = -F^\star$, implies that $\|\mathbf{w}^\star\| \leq \big((\sum_{i\in\mathcal{V}} f_i(y_i)) - F^\star\big)/r_c$. Note that $\sum_{i\in\mathcal{V}} f_i(y_i) \leq \sum_{i\in\mathcal{V}} \max_{x_i\in B(\mathbf{0}_d,r_c)} f_i(x_i)$, where $F^\star \leq \sum_{i\in\mathcal{V}} \max_{x_i\in B(\mathbf{0}_d,r_c)} f_i(x_i) < \infty$ because $B(\mathbf{0}_d, r_c)$ is compact. Therefore, (15) with $x' = \mathbf{0}_d$ holds.

Case II: $\mathbf{0}_d \notin \operatorname{int}\bigcap_{i\in\mathcal{V}} X_i$. Let $x' \in \operatorname{int}\bigcap_{i\in\mathcal{V}} X_i$ and apply the change of variable $z_i = x_i - x'\ \forall i \in \mathcal{V}$ to problem (2). The resulting optimization problem with variable $z_i\ \forall i \in \mathcal{V}$ has the same optimal value $F^\star$ as (2). Also, its Fenchel dual function is $D'(\mathbf{w}) = D(\mathbf{w}) - \mathbf{w}^T\mathbf{x}'$ with $\mathbf{x}' = ((x')^T, \ldots, (x')^T)^T \in S$, which equals $D(\mathbf{w})$ for any $\mathbf{w} \in S^\perp$. Therefore, the optimal set of maximizing $D'(\mathbf{w})$ on $S^\perp$ is the same as that of (3). Since the interior of each local constraint associated with z_i contains $\mathbf{0}_d$, (15) can be obtained by applying *Case I* to D'.

From the above two cases, the optimal set of problem (3) is compact. Then, due to the convexity of D and $S^\perp$, the level sets $S_0(\mathbf{w})\ \forall \mathbf{w} \in S^\perp$ are also compact [33, proposition 1.4.5].

E. Proof of Proposition 4

From Theorem 1 and Lemma 1, to guarantee $\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \epsilon$, it suffices to have $k \geq \frac{B\eta\tilde{M}^2_{\lfloor k/B\rfloor}}{\rho\underline{\lambda}}(\frac{2L}{\epsilon^2} - \frac{1}{D(\mathbf{w}^0)-D^\star})$. Note that $\eta = 3B\bar{\varpi}L/\delta + 3/\underline{h} \leq 3(n-1)BL/\delta + 3/\underline{h}$, $\rho = 1/(2\delta)$, $\underline{\lambda} \geq 4/(n(n-1))$ [34], and $-D^\star \leq F(\mathbf{x}')$. In addition, $\tilde{M}_{\lfloor k/B\rfloor} \leq \max_{t=0,\ldots,\lfloor\frac{k}{B}\rfloor}\|\mathbf{w}^{tB}\| + \|\mathbf{w}^\star\|$ for any dual optimum $\mathbf{w}^\star$. It then follows from Proposition 3 and $-F^\star \leq D(\mathbf{w})\ \forall \mathbf{w} \in S^\perp$ that (16) ensures $\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \epsilon$.

REFERENCES

[1] M. G. Rabbat and R. D. Nowak, "Distributed optimization in sensor networks," in *Proc. International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, 2004, pp. 20–27.

[2] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, "An $O(1/k)$ gradient method for network resource allocation problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 64–73, 2014.

[3] P. Giselsson, M. D. Doan, T. Keviczky, B. Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

[4] A. Nedić, 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.

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

[6] S. Lee and A. Nedić, "Distributed random projection algorithm for convex optimization," *IEEE Journal of Selected Topics in Signal Processing, a special issue on Adaptation and Learning over Complex Networks*, vol. 7, no. 2, pp. 221–229, 2013.

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

[8] 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.

[9] P. Lin, W. Ren, and Y. Song, "Distributed multi-agent optimization subject to nonidentical constraints and communication delays," *Automatica*, vol. 65, pp. 120–131, 2016.

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

[11] ——, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.

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

[13] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, 2017.

[14] C. Xi and U. Khan, "DEXTRA: A fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, 2017.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] 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, 2012.

[17] T. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[18] I. Necoara and V. Nedelcu, "Rate analysis of inexact dual first-order methods application to dual decomposition," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1232–1243, 2014.

[19] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.

[20] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, "Proximal minimization based distributed convex optimization," in *Proc.American Control Conference*, Boston, MA, 2016, pp. 2466–2471.

[21] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[22] D. Bajović, D. Jakovetić, N. Krejić, and N. K. Jerinikić, "Newton-like method with diagonal correction for distributed optimization," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 1171–1203, 2017.

[23] J. Lu and C. Y. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2348–2354, 2012.

[24] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of Optimization Theory and Applications*, vol. 129, no. 3, pp. 469–488, 2006.

[25] H. Lakshmanan and D. P. de Farias, "Decentralized resource allocation in dynamic networks of agents," *SIAM Journal on Optimization*, vol. 19, no. 2, p. 911–940, 2008.

[26] X. Wu and J. Lu, "Fenchel dual gradient methods for distributed convex optimization over time-varying networks," in *Proc. IEEE Conference on Decision and Control*, Melbourne, Australia, 2017, pp. 2894–2899.

[27] K. Fountoulakis and J. Gondzio, "A second-order method for strongly convex l_1-regularization problems," *Mathematical Programming*, vol. 156, no. 2016, pp. 189–219, 2016.

[28] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. International Symposium on Information Processing in Sensor Networks*, Los Angeles, CA, 2005, pp. 63–70.

[29] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

[30] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: Distributed randomized algorithms and analysis," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 987–1000, 2006.

[31] Y. Nesterov, *Introductory lectures on Convex Optimization: A Basic Course*. Norwell, MA: Kluwer Academic Publishers, 2004.

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

[33] D. P. Bertsekas, *Convex optimization theory*. Belmont, MA: Athena Scientific, 2009.

[34] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, vol. 2, pp. 871–898, 1991.