Accelerated Primal-Dual Algorithms for Distributed Smooth Convex Optimization over Networks

Jinming Xu[†,‡]
State Key Lab of Industrial Control Technology
[†]Zhejiang University

Ye Tian[‡], **Ying Sun**[‡], **and Gesualdo Scutari**[‡]
School of Industrial Engineering
[‡]Purdue University

Abstract

This paper proposes a novel family of primal-dual-based distributed algorithms for smooth, convex, multi-agent optimization over networks that uses only gradient information and gossip communications. The algorithms can also employ acceleration on the computation and communications. We provide a unified analysis of their convergence rate, measured in terms of the Bregman distance associated to the saddle point reformation of the distributed optimization problem. When acceleration is employed, the rate is shown to be optimal, in the sense that it matches (under the proposed metric) existing complexity lower bounds of distributed algorithms applicable to such a class of problem and using only gradient information and gossip communications. Preliminary numerical results on distributed least-square regression problems show that the proposed algorithm compares favorably on existing distributed schemes.

1 Introduction

We study distributed (smooth) convex optimization over multi-agent networks, modeled as a fixed, undirected graph. Agents aim to cooperatively solve

$$\min_{x\in\mathbb{R}^d} F(x) := \sum_{i=1}^{m} f_i(x), \tag{1}$$

where $x \in \mathbb{R}^d$ is the vector of optimization variables, shared among the m agents; and $f_i : \mathbb{R}^d \to \mathbb{R}$ is the cost function of agent i, assumed to be smooth, convex and known only to that agent. We are interested in network architectures that do not have any centralized (master) node handling the entire optimization process or able to gather information from all the other agents in the system (such as master/slave architectures); each agent instead controls a local estimate of the common vector x, which is iteratively updated based upon its local gradient and information received from its immediate neighbors. This scenario arises naturally from several large-scale machine learning applications wherein the sheer volume and spatial/temporal disparity of scattered data render centralized processing and storage infeasible or inefficient.

The focus of this paper is on *optimal rate* decentralized algorithms for Problem (1) that use only gradient information and gossip communications. By *optimal* we mean that these algorithms provably achieve lower complexity bounds for such a class of problems and oracle decentralized algorithms. Primal (Duchi et al., 2012; Yuan et al., 2016; Jakovetic et al., 2014; Nedic and Olshevsky, 2014; Di Lorenzo and Scutari, 2016; Nedich et al., 2017; Qu and Li, 2017b; Xu et al., 2015; Sun et al., 2019) and primal-dual distributed methods (Shi et al., 2015, 2014; Ling et al., 2015; Wei and Ozdaglar, 2012; Chang et al., 2015) applicable to Problem (1) have been extensively studied in the literature, enjoying different convergence rates. In general, these rates are not optimal for several reasons: i) the schemes do not employ any acceleration on the local optimization step and/or communications; or ii) they do not balance optimally the number of optimization and communications steps. Optimal rates of first-order distributed algorithms have been recently studied in Scaman et al. (2017, 2018); Sun and Hong (2018); Uribe et al. (2018); Lan et al. (2017); Shamir (2014); Arjevani and Shamir (2015) for different classes of optimization problems and network topologies; they however are not optimal or applicable to the formulation considered in this paper.

Related works. Optimal lower complexity bounds and matching distributed algorithms have been recently investigated in Scaman et al. (2017) for smooth strongly convex functions, in Scaman et al. (2018) for nons-

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

mooth convex functions, and in Sun and Hong (2018) for smooth nonconvex functions. Fully connected networks have been considered in Shamir (2014); Arjevani and Shamir (2015). However, to our knowledge, no first-order gossip algorithm is known that achieves *both* computation *and* communication lower complexity bound for the minimization of *smooth convex* functions over graphs. Attempts of designing accelerated distributed algorithms for Problem (1) can be found in Li et al. (2018); Qu and Li (2017a); Uribe et al. (2018) and are briefly discussed next. The scheme in Qu and Li (2017a) combines the technique of gradient tracking (Di Lorenzo and Scutari, 2016; Xu et al., 2015; Nedich et al., 2017) with Nesterov acceleration of local computations and achieves an $\epsilon > 0$ solution in $O\left(1/\epsilon^{5/7}\right)$ gradient and communication steps, under the assumption that the solution set of the optimization problem (1) is compact. Algorithm 7 in Uribe et al. (2018) is designed for general smooth convex objectives; it reaches an ϵ solution in $O\left(\sqrt{L_f/(\eta\,\epsilon)}\,\log 1/\epsilon\right)$ outer loops of communications and $O\left(\sqrt{L_f/\epsilon}\log 1/\epsilon\right)$ inner loops of computations (per communication), resulting in an overall gradient evaluations of $O\left(L_f/(\epsilon\sqrt{\eta})\,\log^2 1/\epsilon\right)$, which do not match existing lower bounds. The subsequent work (Li et al., 2018) proposes an accelerated penalty-based method with increasing penalty values; the algorithm achieves the lower bound of $O\left(\sqrt{L_f/\epsilon}\right)$ gradient evaluations but at the cost of an *increasing* number of communications per gradient evaluation (iteration)–namely: $O\left(\sqrt{L_f/\left(\eta\epsilon\right)}\log 1/\epsilon\right)$, making it not optimal in terms of communication steps.

Summary of the contributions. We propose a novel family of primal-dual-based distributed algorithms for Problem (1) that use *only gradient* information and gossip communications. The algorithms can also employ acceleration on the computation and communications. We provide a unified analysis of their convergence rate, measured in terms of the Bregman distance associated to the saddle point reformation of (1). When acceleration on both computation and communications is properly designed, the proposed algorithms are shown to be optimal, in the sense that they match existing complexity lower bounds (Li et al., 2018), rewritten in terms of the Bregman distance metric. Furthermore, differently from Scaman et al. (2017); Uribe et al. (2018), our algorithms do not require any information on the Fenchel conjugate of the agents' functions, which significantly enlarge the class of functions to which provably optimal rate algorithms can be applied to. Hence, we termed our algorithms OPTRA (*optimal conjugate-free distributed primal-dual methods*). Our preliminary numerical results show that OPTRA compares favorably with existing distributed accelerated methods (Li et al., 2018; Qu and Li, 2017a; Uribe et al., 2018) proposed for Problem (1), which supports our theoretical findings.

Technical novelties. While the genesis of OPTRA finds routs in the primal-dual algorithm (Chambolle and Pock, 2011) and employs Nesterov acceleration similarly to Chen et al. (2014) (which also builds on Chambolle and Pock (2011)), there are some substantial differences between the proposed distributed algorithms and the aforementioned schemes (Chambolle and Pock, 2011; Chen et al., 2014), which are briefly discussed next. The scheme in Chambolle and Pock (2011) is meant for abstract saddle-point problems and so Chen et al. (2014) does; the focus therein is not on distributed optimization. Hence communications over networks are not explicitly accounted. Furthermore, Chambolle and Pock (2011) does not employ any acceleration while Chen et al. (2014) accelerates the computation but lacks of the communication (networking) component (no gossip-based updates are present in (Chen et al., 2014, Alg. 2)). On the other hand, OPTRA adopts Nesterov *and* Chebyshev acceleration to balance computation and communication, so that lower complexity bounds on both are achieved (in terms of Bregman distance). This is a major novelty with respect to Chambolle and Pock (2011); Chen et al. (2014). Because of these differences, the convergence analysis of OPTRA can not be deducted by that of Chambolle and Pock (2011); Chen et al. (2014); a novel convergence proof is provided, which shows an explicit dependence of the rate on key network parameters.

Notations: We use $\mathbf{null}(\cdot)$ (resp. $\mathbf{span}(\cdot)$) to denote the null space (resp. range space) of the matrix argument. The vector or matrix (with proper dimension) of all ones (resp. all zeros) is denoted by $\mathbf{1}$ (resp. $\mathbf{0}$); e_i denotes the i-th canonical vector; and the identity matrix is denoted by $\mathbf{I}$; the dimensions of these vector and matrices will be clear from the context. The inner product between two matrices $\mathbf{x}, \mathbf{y}$ is defined as $\langle \mathbf{x}, \mathbf{y}\rangle := \text{trace}(\mathbf{x}, \mathbf{y})$ while the induced norm is $\|\mathbf{x}\| := \|\mathbf{x}\|_F$; we will use the same notation for vectors, treated as special cases. Given a positive semidefinite matrix $\mathbf{G}$, we define $\langle \mathbf{x}, \mathbf{x}'\rangle_{\mathbf{G}} = \langle \mathbf{G}\mathbf{x}, \mathbf{x}'\rangle$ and $\|\mathbf{x}\|_{\mathbf{G}} = \sqrt{\langle \mathbf{G}\mathbf{x}, \mathbf{x}\rangle}$.

2 Problem formulation

2.1 Distributed optimization over networks

We study Problem (1) under the following assumptions.

Assumption 1. (i) Each cost function $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex and L_{f_i}-smooth and (ii) Problem (1) has a solution. Define $L_f := \max_{i=1}^m L_{f_i}$

Network model Agents are embedded in a communication network, modeled as an undirected graph

$\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{V}$ is the set of vertices–the agents– and $\mathcal{E}$ is the set of edges; $\{i, j\} \in \mathcal{E}$ if there is a communication link between agent i and agent j. We assume that the graph has no self-loops, that is, $\{i, i\} \notin \mathcal{E}$. We use $\mathcal{N}_i := \{j | \{i, j\} \in \mathcal{E}\}$ to denote the set of neighbors of agent i.

Definition 1 (Graph Induced Matrix)**.** The symmetric matrix $\mathbf{S} = [s_{ij}] \in \mathbb{R}^{m\times m}$ is said to be induced by the graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ if $s_{ij} \neq 0$ only if $i = j$ or $\{i, j\} \in \mathcal{E}$. The set of such matrices is denoted by $\mathcal{W}_{\mathcal{G}}$.

Since we are interested in optimization over networks with no centralized nodes, we will focus on distributed algorithms whereby agents communicate with their neighbors using a suitably designed gossip matrix. Standard assumptions on such matrices are the following.

Assumption 2. Given the graph $\mathcal{G}$, the gossip matrix $\mathbf{L} \in \mathbb{R}^{m\times m}$ satisfies:

(i) $\mathbf{L} \in \mathcal{W}_{\mathcal{G}}$;

(ii) Positive semi-definiteness: $\mathbf{L} \succeq 0$, with $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_m$;

(iii) Connectivity: $\mathbf{null}(\mathbf{L}) = \mathbf{span}(\mathbf{1})$;

where $\{\lambda_i\}_{i=1}^m$ are the eigenvalues of $\mathbf{L}$.

It is not difficult to check a gossip matrix satisfying Assumption 2 always exists if the associated graph is connected; see, e.g., Olfati-Saber et al. (2007). Several gossip matrices have been considered in the literature; we refer the reader to Xiao and Boyd (2004); Nedić et al. (2018) and references therein for specific examples.

2.2 Saddle-point reformulation

A standard approach for solving (1) consists in rewriting the optimization problem in the so-called consensus optimization form, that is

$$\min_{\mathbf{x}\in\mathbb{R}^{m\times d}} f(\mathbf{x}) + \iota_{\mathcal{C}}(\mathbf{x}), \tag{2}$$

where $\mathbf{x} = [x_1, x_2, ..., x_m]^\top \in \mathbb{R}^{m\times d}$, with x_i being the local estimate of x owned by agent i; $f(\mathbf{x}) := \sum_{i=1}^m f_i(x_i)$; and $\iota_{\mathcal{C}}(\cdot)$ is the indicator function on the consensus space $\mathcal{C} := \{\mathbf{1}_m x^\top \,|\, x \in \mathbb{R}^d\}$. Note that $\nabla f(\mathbf{x}) = [\nabla f_1(x_1), \nabla f_2(x_2), ..., \nabla f_m(x_m)]^\top \in \mathbb{R}^{m\times d}$.

To solve Problem (2), we consider the following closely related saddle point formulation

$$\max_{\mathbf{y}\in\mathbb{R}^{m\times d}} \min_{\mathbf{x}\in\mathbb{R}^{m\times d}} \Phi(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) + \langle \mathbf{y}, \mathbf{x}\rangle - \iota_{\mathcal{C}^\perp}(\mathbf{y}), \tag{3}$$

where $\mathcal{C}^\perp$ is the space orthogonal to $\mathcal{C}$ and $\Phi(\mathbf{x}, \mathbf{y})$ is the Lagrangian associated to problem (2). By Assumption 1, strong duality holds for (3); hence, (3) admits a primal-dual optimal solution pair $(\mathbf{x}^\star, \mathbf{y}^\star) \in \mathcal{D} := \mathbb{R}^{m\times d} \times \mathcal{C}^\perp$ that satisfies the following KKT conditions

$$\text{(Lagrangian Optimality)} \quad \mathbf{y}^\star = -\nabla f(\mathbf{x}^\star), \tag{4a}$$

$$\text{(Primal Feasibility)} \quad \mathbf{x}^\star \in \mathcal{C}, \tag{4b}$$

and the saddle-point property $\Phi(\mathbf{x}^\star, \mathbf{y}) \leq \Phi(\mathbf{x}^\star, \mathbf{y}^\star) \leq \Phi(\mathbf{x}, \mathbf{y}^\star)$, for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$. Note that $\mathbf{x}^\star$ solves Problem (2) and thus it is also a solution of the original formulation (1) (Bertsekas et al., 2003).

Using (3) and (4), one can write

$$\begin{aligned}\Phi(\mathbf{x}, \mathbf{y}^\star) - \Phi(\mathbf{x}^\star, \mathbf{y}) &= f(\mathbf{x}) + \langle \mathbf{y}^\star, \mathbf{x}\rangle - f(\mathbf{x}^\star) - \langle \mathbf{y}, \mathbf{x}^\star\rangle \\ &\overset{(4)}{=} f(\mathbf{x}) - f(\mathbf{x}^\star) - \langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star\rangle \triangleq G(\mathbf{x}, \mathbf{x}^\star) \geq 0.\end{aligned} \tag{5}$$

where $G(\mathbf{x}, \mathbf{x}^\star)$ is the Bregman distance. The following properties of G are instrumental for our develoments (the proof is provided in the supporting material).

Proposition 1. *Let $\mathbf{x}^\star$ be any optimal solution of (2); the following hold for G defined in (5):*

(a) $\bar{\mathbf{x}}$ is an optimal solution of (2) if and only if $\bar{\mathbf{x}} \in \mathcal{C}$ and $G(\bar{\mathbf{x}}, \mathbf{x}^\star) = 0$;

(b) $G(\mathbf{x}, \bullet)$ is constant over the solution set of (2).

Due to (b), for notational simplicity, in what follows, we will write $G(\mathbf{x})$ for $G(\mathbf{x}, \mathbf{x}^\star)$.

Remark 1. In this paper we will use G as metric to assess the (worst-case) convergence rate of the proposed algorithms as well as to state lower complexity bounds. Note that, since f is not assumed to be strictly convex, $G(\mathbf{x}) = 0$ does not imply $\mathbf{x} = \mathbf{x}^\star$, but it is only a necessary condition for $\mathbf{x}$ to be optimal (cf. Proposition 1(a)). Still, G is a valid merit function for both purposes above, as explained next. First, $G(\mathbf{x}) > \epsilon$ implies that $\mathbf{x}$ is ϵ "far" away (in the G-measure) from *any* optimal solution of (2); hence, a lower bound in terms of G is an informative measure. Furthermore, when it comes to the convergence rate analysis of distributed algorithms, Proposition 1-(a) legitimates the use of (the decay rate of) G along the agents' iterates $\{\mathbf{x}^k\}_{k=0}^\infty$, as the distance of $\mathbf{x}^k$ from $\mathcal{C}$ is proved to be vanishing–see Sec. 4.

3 Lower Complexity Bounds

We recall here existing lower complexity bounds for decentralized first-order schemes belonging to the same oracle class of the distributed algorithms we are going to introduce. The difference from the literature is that we will write such bounds in terms of the Bregman distance G. We begin introducing the distributed oracle model (cf. Sec. 3.1), followed by the lower complexity bound (cf. Sec. 3.2).

3.1 Decentralized first-order oracle

Given Problem (1) over the graph $\mathcal{G}$, we consider distributed algorithms wherein each agent i controls a

local variable $x_i \in \mathbb{R}^d$, which is an estimate of the shared optimization variable x in (1). The value of x_i at (continuous) time $t \in \mathbb{R}_+$ is denoted by $x_i^{(t)}$. To update its own variable, each agent i: 1) has access to the gradient of its own function–we assume that the time to inquire such a gradient is normalized to one; and 2) can communicate values (vectors in $\mathbb{R}^d$) to (some of) its neighbors $j \in \mathcal{N}_i$–this communication requires a time $\tau_c \in \mathbb{R}_+$ (which may be smaller or greater than one). Each update $x_i^{(t)}$ is generated according to the following general *black-box procedure*.

Distributed first-order oracle $\mathcal{A}$: A distributed first order iterative method generates a sequence $\{\mathbf{x}^{(t)}\}_{t\geq 0}$, with $\mathbf{x}^{(t)} \triangleq [x_1^{(t)}, \ldots, x_m^{(t)}]$, such that

$$x_i^{(t)} \in \underbrace{\mathbf{span}(x_j^{(s)} \,|\, j \in \mathcal{N}_i \text{ and } 0 \leq s < t - \tau_c)}_{\textbf{local communication}} + \underbrace{\mathbf{span}(x_i^{(s)}, \nabla f_i(x_i^{(s)}) \,|\, 0 \leq s < t-1)}_{\textbf{local computation}}, \tag{6}$$

for all $i \in \mathcal{V}$. We made the blanket assumption that each $x_i^0 = 0$, without loss of generality.

The oracle (6) allows each agent to use all the historical values of its local gradients (local computations) as well as the historical values of the decision variables received from its neighbors (local communications). Furthermore, (6) also captures algorithms employing multiple rounds of communications (resp. gradient computations) per gradient evaluation (resp. communication). In the supporting material (Appendix A), we show that, in fact, the above oracle accounts for most existing distributed algorithms, such as primal-dual methods (Shi et al., 2015) as well as gradient tracking methods (Di Lorenzo and Scutari, 2016; Nedich et al., 2017; Qu and Li, 2017b; Xu et al., 2015).

A similar black-box procedure has been introduced in Scaman et al. (2017) for strongly convex instances of (1). The difference here is that the oracle in (6) cannot return the gradient of the conjugate of the f_i's. The reason of considering such "less powerful" methods is that, in practice, it is hard to compute the gradient of conjugate functions. This means that the gossip (dual-based) methods in Scaman et al. (2017) do not belong to the oracle considered in this paper.

3.2 Lower complexity bounds

We state now lower complexity bounds in the G-metric for the class of algorithms $\mathcal{A}$ applied to Problem (2) [and thus (1)] over a connected graph $\mathcal{G}$. In Section 4 we will introduce a primal-dual distributed algorithm that indeed converges to an optimal solution of (2) driving G to zero at a rate that matches the lower complexity bound. Proofs of the results are available as supporting material.

Theorem 2. *Consider Problem* (1) *under Assumption 1 and let $\mathcal{G}$ be a connected graph. For any given $\eta \in (0,1]$ and $L_f > 0$, there exists a gossip matrix $\mathbf{L} \in \mathcal{W}_\mathcal{G}$ with eigengap $\eta \triangleq \frac{\lambda_2(\mathbf{L})}{\lambda_m(\mathbf{L})}$, and a set of local cost functions $\{f_i\}_{i=0}^m$, $f_i : \mathbb{R}^d \to \mathbb{R}$, with $f(\mathbf{x}) = \sum_i f_i(x_i)$ being L_f-smooth such that, for any first-order gossip algorithm in $\mathcal{A}$ using $\mathbf{L}$, we have*

$$G\big(\mathbf{x}^{(t)}\big) = \Omega\left(\frac{L_f R^2}{\left(\frac{t}{1+\left\lceil \frac{1}{5\sqrt{\eta}}\right\rceil \tau_c} + 2\right)^2} + \frac{R\,\|\nabla f(\mathbf{x}^\star)\|}{\frac{t}{1+\left\lceil \frac{1}{5\sqrt{\eta}}\right\rceil \tau_c} + 2}\right), \tag{7}$$

for all $t \in \left[0, \frac{d-1}{2}\left(1 + \left\lceil \frac{1}{5\sqrt{\eta}}\right\rceil \tau_c\right)\right]$, where $R \triangleq \|\mathbf{x}^0 - \mathbf{x}^\star\|$. Furthermore,

$$\frac{L_f R^2}{t \,/\, \left(1 + \left\lceil \frac{1}{5\sqrt{\eta}}\right\rceil \tau_c\right)} = \Theta\left(R\,\|\nabla f(\mathbf{x}^\star)\|\right). \tag{8}$$

Corollary 3. *In the setting of Theorem 2, the overall time needed by any first-order algorithm in $\mathcal{A}$ using the gossip matrix $\mathbf{L}$ to drive G below $\epsilon > 0$, with f given in Theorem 2, is*

$$\Omega\left(\left(1 + \frac{1}{\sqrt{\eta}}\tau_c\right)\left(\sqrt{\frac{L_f R^2}{\epsilon}} + \frac{R\,\|\nabla f(\mathbf{x}^\star)\|}{\epsilon}\right)\right). \tag{9}$$

Notice that, because of (8), the lower bound (9) can be equivalently stated as

$$\Omega\left(\left(1 + \frac{1}{\sqrt{\eta}}\tau_c\right)\sqrt{\frac{L_f R^2}{\epsilon}}\right). \tag{10}$$

It is not difficult to check that the lower bound in terms of the traditional function-error-based metric (FEM):

$$\max_{i\in\mathcal{V}}(F(x_i) - \min_{x\in\mathbb{R}^d} F(x)) \tag{11}$$

has the same expression as (7) [and thus (9) and (10)] up to some constants. This observation is also reported in Li et al. (2018) without proof, and stated formally below for completeness (the proof can be found in the supporting material).

Theorem 4 (Lower bound on the FEM-metric)**.** *In the setting of Theorem 2, the overall time needed by any first-order algorithm in $\mathcal{A}$ using the gossip matrix $\mathbf{L}$ to drive the function-error-based metric, $\max_{i\in\mathcal{V}}(F(x_i) - \min_{x\in\mathbb{R}^d} F(x))$, below $\epsilon > 0$, with f given in Theorem 2, is bounded by (9) [or, equivalently, by (10)].*

Remark 2 (Balancing computations & communications)**.** The above lower bounds tell us that one cannot reach an ϵ-solution of (2) (measured either

in terms of the G or FEM-metrics) in less than $O\left(\sqrt{L_f R^2/\epsilon} + R\|\nabla f(\mathbf{x}^\star)\|/\epsilon\right)$ computing time and $O\left(\tau_c/\sqrt{\eta}\cdot\left(\sqrt{L_f R^2/\epsilon} + R\|\nabla f(\mathbf{x}^\star)\|/\epsilon\right)\right)$ communication time for the worst-case problem as stated in Theorem 2 (see Eq. (28) in the supporting material for a concrete example). Since the time for a single gradient evaluation has been normalized to one, the former lower bound corresponds also to the overall number of gradient evaluations while the overall communication steps read $\Omega\left(1/\sqrt{\eta}\cdot\left(\sqrt{L_f R^2/\epsilon} + R\|\nabla f(\mathbf{x}^\star)\|/\epsilon\right)\right)$. This sheds light also on the optimal balance between computation and communication: the optimal number of communication steps per gradient evaluations is $\lceil 1/\sqrt{\eta}\rceil$ (in the worst case). In the next section, we introduce a distributed, gossip-based algorithm that achieves lower complexity bounds in the G-metric.

4 Distributed primal-dual algorithms

4.1 A general primal-dual scheme

A gamut of primal-dual algorithms has been proposed in the literature to solve Problem (2) in a centralized setting; see, e.g., Condat (2013); Chambolle and Pock (2011) and references therein for details. Building on Condat (2013); Chambolle and Pock (2011), here, we propose a general primal-dual algorithm to solve the saddle point problem (3) in a *distributed* manner. The algorithm reads: given $\mathbf{x}^k$ and $\mathbf{y}^k$ at iteration k,

$$\mathbf{x}^{k+1} = \mathbf{A}(\mathbf{x}^k - \gamma(\nabla f(\mathbf{x}^k) + \hat{\mathbf{y}}^k)), \tag{12a}$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \tau\mathbf{B}\mathbf{x}^{k+1}, \tag{12b}$$
$$\hat{\mathbf{y}}^{k+1} = \mathbf{y}^{k+1} + (\mathbf{y}^{k+1} - \mathbf{y}^k), \tag{12c}$$

where $\mathbf{y}^k$ is the dual vector variable; γ and τ are the primal and dual step-sizes common to all the agents; and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$ satisfy the following assumption.

Assumption 3. The weight matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$ in (12) are such that

(i) $\mathbf{A} = \mathbf{A}^\top$, $\mathbf{0} \preceq \mathbf{A} \preceq \mathbf{I}$, and $\mathbf{null}(\mathbf{I}-\mathbf{A}) \supseteq \mathbf{span}(\mathbf{1})$;

(ii) $\mathbf{B} = \mathbf{B}^\top$, $\mathbf{B} \succeq \mathbf{0}$, and $\mathbf{null}(\mathbf{B}) = \mathbf{span}(\mathbf{1})$.

Remark 3. Several choices for $\mathbf{A}$ and $\mathbf{B}$ satisfying Assumption 3 are possible, resulting in a gamut of specific algorithms, obtained as instances of (12). Note that, when $\mathbf{A}$ and $\mathbf{B}$ satisfy also Assumption 2, all these algorithms are implementable over the graph $\mathcal{G}$. Several examples of such distributed algorithms are discussed in details in Appendix A. Here, we only mention that the gradient tracking methods (Di Lorenzo and Scutari, 2016; Nedich et al., 2017; Qu and Li, 2017b; Xu et al., 2015) and primal-dual methods, such as EXTRA (Shi et al., 2015), are all special cases of (12); the former schemes are obtained setting $\mathbf{A} = \mathbf{W}^2$ and $\mathbf{B} = (\mathbf{I}-\mathbf{W})^2$, where $\mathbf{W} \in \mathcal{W}_\mathcal{G}$ is the weight matrix used by the agents to employ the consensus step; and EXTRA is obtained setting $\mathbf{A} = \mathbf{W}$ and $\mathbf{B} = \mathbf{I} - \mathbf{W}$.

We begin studying convergence of the general primal-dual algorithm (12), under the following tuning of the free parameters:

$$\gamma = \frac{\nu}{\nu L_f + 1}, \quad \tau = \frac{1}{\nu\lambda_m(\mathbf{B})}, \quad (1-\gamma L_f)\mathbf{I} - \gamma\tau\mathbf{B} \succeq \mathbf{0}, \tag{13}$$

where $\lambda_m(\mathbf{B})$ is the largest eigenvalue of $\mathbf{B}$.

Theorem 5. *Consider Problem* (1) *under Assumption 1. Given* $(\mathbf{x}^1, \mathbf{y}^1)$*, let* $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k=1}^\infty$ *be the sequence generated by Algorithm* (12)*, under Assumption 3 and the setting in* (13)*. Define* $\bar{\mathbf{x}}^k := \frac{1}{k-1}\sum_{t=2}^k \mathbf{x}_t$ *and* $R \triangleq \|\mathbf{x}^1 - \mathbf{x}^\star\|$*. Then, the following hold: (i)* $\{\mathbf{x}^k\}_{k=0}^\infty$ *converges to an optimal solution* $\mathbf{x}^\star$ *of* (2) *[thus* $\mathbf{x}^\star = \mathbf{1}x^\star$*, for some solution* $x^\star$ *of* (1)*]; therefore* $\lim_{k\to\infty} G(\mathbf{x}^k) = 0$*; and (ii) the number of iterations needed for* $G(\bar{\mathbf{x}}^k)$ *to go below* $\epsilon > 0$ *is*[1]

$$O\left(\frac{L_f R^2}{\epsilon} + \frac{1}{\sqrt{\eta(\mathbf{B})}}\frac{R\|\nabla f(\mathbf{x}^\star)\|}{\epsilon}\right). \tag{14}$$

The proof of the theorem can be found in the supporting material. Note that the convergence rate (14) does not match the lower bound given in Theorem 2. For instance, consider as concrete example the choice $\mathbf{A} = \mathbf{I} - \mathbf{L}$ and $\mathbf{B} = \mathbf{L}$; and let $\tau_c \in \mathbb{R}_+$ (resp. 1) be the time for each agent to perform a single communication to its neighbors (resp. gradient evaluation). The time complexity of the primal-dual algorithm (12) becomes

$$O\left((1+\tau_c)\left(\frac{L_f R^2}{\epsilon} + \frac{1}{\sqrt{\eta}}\frac{R\|\nabla f(\mathbf{x}^\star)\|}{\epsilon}\right)\right).$$

To match the lower lower bound given in Theorem 2, our next step is accelerating the algorithm, both the computational part and the communication step; we leverage Nesterov acceleration (Nesterov, 2013) for the optimization step while employ Chebyshev polynomials (Wien, 2011) to accelerate communications. To provide some insight of our construction, we begin with the former acceleration; the latter is added in Section 4.3.

4.2 Accelerated primal-dual algorithms

We accelerate the primal-dual algorithm (12) as follows:

$$\mathbf{u}^{k+1} = \mathbf{A}(\mathbf{x}^k - \gamma(\nabla f(\mathbf{x}^k) + \hat{\mathbf{y}}^k)), \tag{15a}$$
$$\mathbf{x}^{k+1} = \mathbf{u}^{k+1} + \alpha_k(\mathbf{u}^{k+1} - \mathbf{u}^k), \tag{15b}$$
$$\hat{\mathbf{x}}^{k+1} = \sigma_k\mathbf{x}^{k+1} + (1-\sigma_k)\mathbf{u}^{k+1} \tag{15c}$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \tau_k\mathbf{B}\hat{\mathbf{x}}^{k+1}, \tag{15d}$$
$$\hat{\mathbf{y}}^{k+1} = \mathbf{y}^{k+1} + \beta_k(\mathbf{y}^{k+1} - \mathbf{y}^k), \tag{15e}$$

[1] We use $\eta(\mathbf{B})$ to denote the eigengap of $\mathbf{B}$.

where $\mathbf{u}^k$, $\hat{\mathbf{x}}^k$, and $\hat{\mathbf{y}}^k$ are auxiliary variables and $\alpha_k, \sigma_k, \tau_k, \beta_k$ are parameters to be properly chosen. Roughly speaking, (15a), (15d) and (15e) are the standard primal-dual steps while (15b) and (15c) are the extra steps meant for the acceleration, with (15b) being the standard Nesterov momentum step and (15c) being a correction step. Note that setting $\alpha_k \equiv 0, \sigma_k \equiv 1, \tau_k \equiv \tau, \beta_k \equiv 1$, the algorithm reduces to the primal-dual method (12). We provide next an instance of (15) that is suitable for a distributed implementation.

Let T be the overall number of iterations being carried out. The free parameters in (15) is chosen as follows:

$$\mathbf{A} = \mathbf{I} - \mathbf{L}/\lambda_m(\mathbf{L}),\ \mathbf{B} = \mathbf{L}/\lambda_m(\mathbf{L}),\ \gamma = \frac{\nu}{\nu L_f + T},$$
$$\tau = \frac{1}{\nu T\, \lambda_m(\mathbf{B})}\ , \frac{1}{\theta_k} = \frac{1+\sqrt{1+4(\frac{1}{\theta_{k-1}})^2}}{2} \text{ with } \theta_1 = 1,$$
$$\sigma_k = \frac{1}{\theta_{k+1}},\ \alpha_k = \frac{\theta_{k+1}}{\theta_k} - \theta_{k+1},\ \beta_k = \frac{\tau_{k+1}}{\tau_k},\ \tau_k = \frac{\tau}{\theta_k}. \tag{16}$$

The resulting scheme is summarized in Algorithm 1, and its convergence properties are stated in Theorem 6. We point out that Theorem 6, although stated for Algorithm 1, can be readily extended to the more general accelerated primal-dual scheme (15), with other choices of $\mathbf{A}$ and $\mathbf{B}$ just satisfying Assumption 3.

Algorithm 1 OPTRA-N

Input: number of iterations T, Laplacian matrix $\mathbf{L}$, $\nu > 0$
Output: $(\mathbf{u}^T, \mathbf{y}^T)$
Initialization: $y_i^1 = 0, \forall i \in \mathcal{V}$ and $\theta_1 = 1$

1: $\hat{\mathbf{y}}^1 = \tau_1 \mathbf{B}\mathbf{x}^1$, $\mathbf{u}^1 = \mathbf{x}^1$
2: **for** $k = 1, 2, ..., T$ **do**
3: compute θ_k according to (16),
4: **for** $\forall i \in \mathcal{V}$ **do** in parallel
5: compute the next iterate according to (15), using the tuning as in (16),
6: **end for**
7: **end for**
8: **Return** $(\mathbf{u}^T, \mathbf{y}^T)$

Theorem 6. *Consider Problem* (1) *under Assumption 1; let* $\mathbf{u}^{(t)}$ *be the value of the* $\mathbf{u}$*-vector generated by Algorithm 1 at time* $t \in \mathbb{R}_+$*, under Assumptions 2 and 3, and the parameter setting in* (16)*. If* $\nu = \sqrt{\eta}$*, then*

$$G(\mathbf{u}^{(t)}) = O\left(\frac{L_f R^2}{\left(\frac{t}{1+\tau_c}\right)^2} + \frac{R^2 + \|\nabla f(\mathbf{x}^\star)\|^2}{\sqrt{\eta}\frac{t}{1+\tau_c}}\right).$$

If one can set $\nu = O\left(\sqrt{\eta} R / \|\nabla f(\mathbf{x}^\star)\|\right)$*, the above bound can be improved to*

$$G(\mathbf{u}^{(t)}) = O\left(\frac{L_f R^2}{\left(\frac{t}{1+\tau_c}\right)^2} + \frac{R\|\nabla f(\mathbf{x}^\star)\|}{\sqrt{\eta}\frac{t}{1+\tau_c}}\right). \tag{17}$$

Furthermore, the consensus error decays

$$\left\|\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)\mathbf{u}^{(t)}\right\| = O\left(\frac{L_f R^2}{\|\nabla f(\mathbf{x}^\star)\|\left(\frac{t}{1+\tau_c}\right)^2} + \frac{R^2 + \|\nabla f(\mathbf{x}^\star)\|^2}{\|\nabla f(\mathbf{x}^\star)\|\sqrt{\eta}\frac{t}{1+\tau_c}}\right). \tag{18}$$

While the convergence time of Algorithm 1 benefits from the Nesterov acceleration of the computation step, it is not optimal in terms of communications (optimal dependence on η). In fact, when the network is poorly connected, the second term on the RHS of (17) becomes dominant with respect to the first one, and (17) overall will be larger than (7). This is due to the fact that Algorithm 1 performs a one-consensus-one-gradient update while the lower bound shows an optimal ratio of $\lceil 1/\sqrt{\eta} \rceil$ (cf. Remark 2). This optimal ratio can be achieved accelerating also the communication step, as described in the next section.

4.3 Optimal primal-dual algorithms with Chebyshev acceleration

We employ the acceleration of the communication step in Algorithm 1 by replacing the gossip matrix $\mathbf{L}$ with $P_K(\mathbf{L})$, where $P_K(\cdot)$ is a polynomial of at most K degree that maximizes the eigengap of $P_K(\mathbf{L})$. This leads to a widely used acceleration scheme known as Chebyshev acceleration and the choice $P_K(x) = 1 - T_K(c_1(1-x))/T_K(c_1)$, with $c_1 = (1+\eta(\mathbf{L}))/(1-\eta(\mathbf{L}))$ and $T_K(\cdot)$, are the Chebyshev polynomials (Wien, 2011). It is not difficult to check that such a $P_K(\mathbf{L})$ is still a gossip matrix. Using in (15) the following setting:

$$\mathbf{A} = \mathbf{I} - c_2 P_K(\mathbf{L}),\ \mathbf{B} = P_K(\mathbf{L}),\ K = \left\lceil 1/\sqrt{\eta(\mathbf{L})} \right\rceil, \text{ with}$$
$$c_2 = \left(1 + 2\frac{c_0^K}{(1+c_0^{2K})}\right)^{-1},\ c_0 = \frac{1-\sqrt{\eta(\mathbf{L})}}{1+\sqrt{\eta(\mathbf{L})}}, \tag{19}$$

leads to the distributed scheme described in Algorithm 2, whose convergence rate achieves the lower bound (9), as proved in Theorem 7 below. Note that, although the idea of using Chebyshev polynomial has been already used in some (centralized and distributed) algorithms in the literature (Wien, 2011; Scaman et al., 2017), Algorithm 2 substantially differs from that of Scaman et al. (2017), which assumes strongly-convex cost functions and is not rate-optimal in the setting considered in this paper (cf. Sec. G in the supporting material for more details).

Algorithm 2 OPTRA

Input: number of iterations T, Laplacian matrix $\widetilde{\mathbf{L}}$, number of inner consensus $K = \left\lceil \frac{1}{\sqrt{\eta(\mathbf{L})}} \right\rceil$, $c_0 = \frac{1-\sqrt{\eta(\widetilde{\mathbf{L}})}}{1+\sqrt{\eta(\widetilde{\mathbf{L}})}}$, $c_1 = \frac{1+\eta(\widetilde{\mathbf{L}})}{1-\eta(\widetilde{\mathbf{L}})}$, $c_2 = 1/\left(1+2\frac{c_0^K}{1+c_0^{2K}}\right)$, $\tau = \frac{c_2}{\nu T}$, $\gamma = \frac{\nu}{\nu L_f + T}$, $\nu > 0$.

Initialization: $\mathbf{y}^1 = \mathbf{0}$; **Preprocessing**: $\mathbf{L} = \frac{2}{\lambda_2(\widetilde{\mathbf{L}})+\lambda_n(\widetilde{\mathbf{L}})}\widetilde{\mathbf{L}}$.

Output: $(\mathbf{u}^T, \mathbf{y}^T)$

$\hat{\mathbf{y}}^1 = \tau_1 \cdot \text{AccGossip}(\mathbf{x}^1, \mathbf{L}, K)$, $\mathbf{u}^1 = \mathbf{x}^1$
for $k = 1, 2, ..., T$ **do**
  $\mathbf{u}^{k+\frac{1}{2}} = \mathbf{x}^k - \gamma\left(\nabla f(\mathbf{x}^k) + \hat{\mathbf{y}}^k\right)$,
  $\mathbf{u}^{k+1} = \mathbf{u}^{k+\frac{1}{2}} - c_2 \cdot \text{AccGossip}(\mathbf{u}^{k+\frac{1}{2}}, \mathbf{L}, K)$,
  $\mathbf{x}^{k+1} = \mathbf{u}^{k+1} + \left(\frac{\theta_{k+1}}{\theta_k} - \theta_{k+1}\right)(\mathbf{u}^{k+1} - \mathbf{u}^k)$,
  $\hat{\mathbf{x}}^{k+1} = \frac{1}{\theta_{k+1}}\mathbf{x}^{k+1} + \left(1 - \frac{1}{\theta_{k+1}}\right)\mathbf{u}^{k+1}$,
  $\mathbf{y}^{k+1} = \mathbf{y}^k + \frac{\tau}{\theta_k}\text{AccGossip}(\hat{\mathbf{x}}^{k+1}, \mathbf{L}, K)$,
  $\hat{\mathbf{y}}^{k+1} = \mathbf{y}^{k+1} + \frac{\theta_k}{\theta_{k+1}}(\mathbf{y}^{k+1} - \mathbf{y}^k)$,
end for
Return $(\mathbf{u}^T, \mathbf{y}^T)$.

procedure AccGossip$(\mathbf{x}, \mathbf{L}, K)$
  $a_0 = 1, a_1 = c_1$
  $\mathbf{z}_0 = \mathbf{x}, \mathbf{z}_1 = c_1(\mathbf{I} - \mathbf{L})\mathbf{x}$
  for $k = 1$ to $K - 1$ **do**
    $a_{k+1} = 2c_1 a_k - a_{k-1}$
    $\mathbf{z}_{k+1} = 2c_1(\mathbf{I} - \mathbf{L})\mathbf{z}_k - \mathbf{z}_{k-1}$
  end for
  return $\mathbf{z}_0 - \frac{\mathbf{z}_K}{a_K}$
end procedure

Theorem 7. *Consider Problem* (1) *under Assumption 1; let* $\mathbf{u}^{(t)}$ *be the value of the u-vector generated by Algorithm 2 at time* $t \in \mathbb{R}_+$*, under Assumptions 2 and 3, the parameter setting in* (16)*, and employing the Chebyshev acceleration* (19)*. If* $\nu = 1$*, then*

$$G(\mathbf{u}^{(t)}) = O\left(\frac{L_f R^2}{\left(\frac{t}{1+\lceil\frac{1}{\sqrt{\eta}}\rceil\tau_c}\right)^2} + \frac{R^2 + \|\nabla f(\mathbf{x}^\star)\|^2}{\frac{t}{1+\lceil\frac{1}{\sqrt{\eta}}\rceil\tau_c}}\right).$$

If one can set $\nu = O\left(R/\|\nabla f(\mathbf{x}^\star)\|\right)$*, the above bound can be improved to*

$$G(\mathbf{u}^{(t)}) = O\left(\frac{L_f R^2}{\left(\frac{t}{1+\lceil\frac{1}{\sqrt{\eta}}\rceil\tau_c}\right)^2} + \frac{R\|\nabla f(\mathbf{x}^\star)\|}{\frac{t}{1+\lceil\frac{1}{\sqrt{\eta}}\rceil\tau_c}}\right).$$

Furthermore, the consensus error $\left\|(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{u}^{(t)}\right\|$ *decays as*

$$O\left(\frac{L_f R^2}{\|\nabla f(\mathbf{x}^\star)\|\left(\frac{t}{1+\lceil\frac{1}{\sqrt{\eta}}\rceil\tau_c}\right)^2} + \frac{R^2 + \|\nabla f(\mathbf{x}^\star)\|^2}{\|\nabla f(\mathbf{x}^\star)\|\frac{t}{1+\lceil\frac{1}{\sqrt{\eta}}\rceil\tau_c}}\right).$$

According to Theorem 7, given $\epsilon > 0$, the time needed by the algorithm to drive G below $\epsilon > 0$ is

$$O\left(\left(1 + \frac{1}{\sqrt{\eta}}\tau_c\right)\left(\sqrt{\frac{L_f R^2}{\epsilon}} + \frac{R\|\nabla f(\mathbf{x}^\star)\|}{\epsilon}\right)\right),$$

matching the lower complexity bound given in (9).

Note that the optimality is stated in terms of the G-metric and does not imply that the algorithm is rate optimal also in the FEM-metric (11), which to date remains an open question. In our experiments (cf. Sec. 5) we observed i) the same behavior of the two errors as a function of the total number of computations and communications; and ii) that Algorithm 2 in fact outperforms existing distributed schemes.

5 Numerical Results

We report here some preliminary numerical results[2] validating our theoretical findings. We compare the proposed rate-optimal algorithm–OPTRA–with existing accelerated ones designed for convex smooth problems, namely: Acc-DNGD-NSC (Qu and Li, 2017a) and APM-C (Li et al., 2018). We also included non-accelerated schemes that perform quite well in practice, namely: i) the gradient tracking method, NEXT/DIGing (Di Lorenzo and Scutari, 2016; Nedich et al., 2017); ii) the primal-dual method, EXTRA (Shi et al., 2015); and iii) the decentralized stochastic gradient method, DPSGD (Lian et al., 2017).

We tested the above algorithms on a decentralized linear regression problem, in the form $\min_{x\in\mathbb{R}^d}\|\mathbf{A}x - \mathbf{b}\|^2$, where $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2; \cdots; \mathbf{A}_m] \in \mathbb{R}^{mr\times d}$ and $\mathbf{b} = [\mathbf{b}_1; \mathbf{b}_2; \cdots; \mathbf{b}_m] \in \mathbb{R}^{mr\times 1}$, with $\mathbf{A}_i \in \mathbb{R}^{r\times d}$ and $\mathbf{b}_i \in \mathbb{R}^{r\times 1}$, $r = 10$, $d = 500$, and $m = 20$. Note that each agent i can only access the data $(\mathbf{A}_i, \mathbf{b}_i)$. We generated the matrix $\mathbf{A}$ of the feature vectors according to the following procedure, proposed in Agarwal et al. (2010). We first generate a random matrix $\mathbf{Z}$ with each entry i.i.d. drawn from $\mathcal{N}(0, 1)$. Using a control parameter $\omega \in [0, 1)$, we generate columns of $\mathbf{A}$ ($\mathbf{M}_{:,i}$ and $\mathbf{M}_{i,:}$ denote the i-th column and i-th row of a matrix $\mathbf{M}$, respectively) so that the first column is $\mathbf{A}_{:,1} = \mathbf{Z}_{:,1}/\sqrt{1-\omega^2}$ and the rest are recursively set as $\mathbf{A}_{:,i} = \omega\mathbf{A}_{:,i-1} + Z_{:,i}$, for $i = 2, \ldots, d$. As result, each row $\mathbf{A}_{i,:} \in \mathbb{R}^d$ is a Gaussian random vector and its covariance matrix $\Sigma = \text{cov}(\mathbf{A}_{:,i})$ is the identity matrix if

[2] Code: https://github.com/YeTian-93/OPTRA.

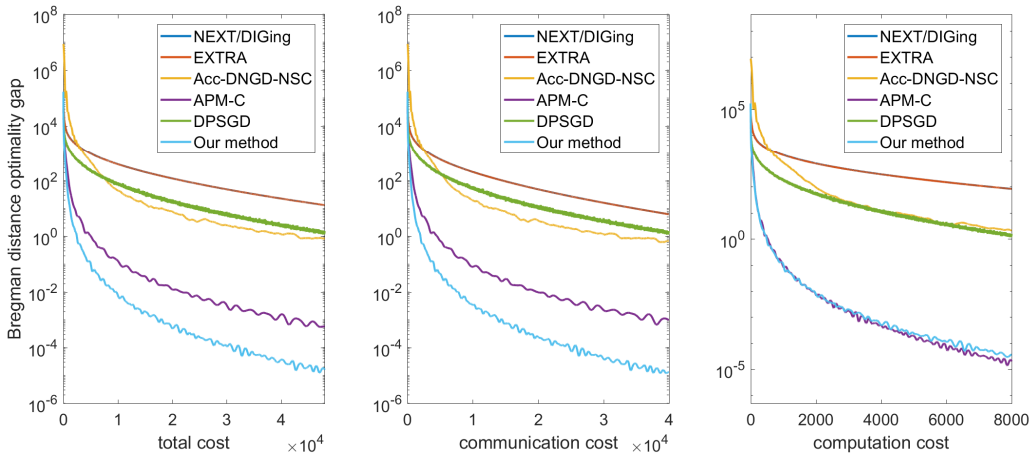

Figure 1: Comparison of distributed first-order gradient algorithms: Bregman distance versus the total cost (left panel), the communication cost (middle panel), and the gradient computation cost (right panel). The curves of DIGing/NEXT overlap with that of EXTRA.

$\omega = 0$ and becomes extremely ill-conditioned as $\omega \to 1$; we set $\omega = 0.95$. Finally we generate $x_0 \in \mathbb{R}^d$ with each entry i.i.d. drawn from $\mathcal{N}(0,1)$, and set $\mathbf{b} = \mathbf{A}x_0 + \boldsymbol{\xi}$, where each component of the noise $\boldsymbol{\xi}$ is i.i.d. drawn from $\mathcal{N}(0, 0.25)$. We simulated a network of $m = 20$ agents, connected throughout a communication graph, generated using the Erdös-RéTyi model; the probability of having an edge between any two nodes is set to 0.1. We calculated L_f from the generated data and used the exact value whenever this parameter is needed. We tuned the free parameters of the simulated algorithms manually to achieve the best practical performance for each algorithm. This leads to the following choices: **i)** the step size of DIGing/NEXT and EXTRA is set to 10^{-5}; **ii)** for Acc-DNGD-NSC, we used the fixed step-size rule, with $\eta = 0.005/L_f$ (the one provided in (Qu and Li, 2017a, Th. 5) is too conservative, resulting in poor practical performance); **iii)** for APM-C, we set (see notation therein) $T_k = \lceil c \cdot (\log k/\sqrt{1-\sigma_2(\mathbf{W})}) \rceil$, with $c = 0.2$ and $\beta_0 = 10^4$; **iv)** for DPSGD, we set the step size to 10^{-5}; at each iteration, the gradient of each agent was computed using 20% of the samples in the local data set; and **v)** for our algorithm, we set $\nu = 100$ and $K = 2$.

Our experiments are reported in Figure 1, where we plot the Bregman distance versus the overall number of communications and computations performed by each agent (left panel), the number of communications (middle panel), and the number of computations (right panel). The time for local communications and gradient computations using all the local data samples is normalized to one; for DPSGD, the computation time unit is scaled proportionally to the size of the local mini-batch. The plots in terms of the more traditional FEM-metric are reported in the supporting material, the behavior is consistent with the results in Figure 1.

The following comments are in order. The accelerated schemes and the stochastic algorithm–DPSGD–converge faster than the non-accelerated schemes–NEXT/DIGing, EXTRA (the curves of EXTRA and NEXT/DIGing coincide in all the panels). In our experiments (including those not reported), we observed that this gap is quite evident when problems are ill-conditioned. From the right panel, one can see that APM-C performs better than OPTRA and Acc-DNGD-NSC in terms of the number of gradient evaluations, which is expected since APM-C employs an increasing number of communication steps per gradient evaluation. On the other hand, APM-C suffers from high communication cost (which is evident from the middle panel), making it not competitive with respect to the proposed OPTRA in terms of communications. When both communication and computation costs are considered (left panel), OPTRA outperforms all the other simulated schemes, which support our theoretical findings.

6 Conclusion

We studied distributed gossip first-order methods for smooth convex optimization over networks. We provided a novel primal-dual distributed algorithm that employs Nesterov acceleration on the optimization step and acceleration of the communication step via Chebyshev polynomials, balancing thus computation and communication. We also proved that the algorithm achieves the lower complexity bound in the Bregman distance-metric. Preliminary numerical results showed that the proposed scheme outperforms existing distributed algorithms proposed for the same class of problems. An open question, currently under investigation, is whether the proposed distributed algorithms are rate optimal also in terms of the FEM metric. No such an algorithm is known so far in the literature.

Acknowledgments

This work has been supported by the following grants: NSF of USA under Grants CIF 1719205 and CMMI 1832688; in part by the Army Research Office under Grant W911NF1810238; and in part by NSF of China under Grants U1909207 and 61922058.

References

Agarwal, A., Negahban, S., and Wainwright, M. J. (2010). Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45.

Arjevani, Y. and Shamir, O. (2015). Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pages 1756–1764.

Bertsekas, D. P., Nedić, A., and Ozdaglar, A. E. (2003). *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. of Mathematical Imaging and Vision*, 40(1):120–145.

Chambolle, A. and Pock, T. (2016). On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287.

Chang, T.-H., Hong, M., and Wang, X. (2015). Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Trans. Signal Process*, 63(2):482–497.

Chen, J. and Sayed, A. H. (2012). Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process*, 60(8):4289–4305.

Chen, Y., Lan, G., and Ouyang, Y. (2014). Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814.

Condat, L. (2013). A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2):460–479.

Di Lorenzo, P. and Scutari, G. (2016). Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136.

Duchi, J., Agarwal, A., and Wainwright, M. (2012). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Autom. Control*, 57(3):592–606.

Jakovetic, D., Xavier, J., and Moura, J. (2014). Fast distributed gradient methods. *IEEE Trans. Autom. Control*, 59(5):1131–1146.

Lan, G., Lee, S., and Zhou, Y. (2017). Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pages 1–48.

Li, H., Fang, C., Yin, W., and Lin, Z. (2018). A sharp convergence rate analysis for distributed accelerated gradient methods. *arXiv preprint arXiv:1810.01053*.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. (2017). Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340.

Ling, Q., Shi, W., Wu, G., and Ribeiro, A. (2015). DLM: Decentralized linearized alternating direction method of multipliers. *IEEE Trans. Signal Process*, 63(15):4051–4064.

Nedic, A. and Olshevsky, A. (2014). Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *arXiv:1406.2075*.

Nedić, A., Olshevsky, A., and Rabbat, M. G. (2018). Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61.

Nedich, A., Olshevsky, A., and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. on Optimization*, 27(4):2597–2633.

Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.

Ouyang, Y. and Xu, Y. (2018). Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *arXiv preprint arXiv:1808.02901*.

Qu, G. and Li, N. (2017a). Accelerated distributed nesterov gradient descent. *arXiv preprint arXiv:1705.07176*.

Qu, G. and Li, N. (2017b). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3027–3036.

Scaman, K., Bach, F., Bubeck, S., Massoulié, L., and Lee, Y. T. (2018). Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749.

Shamir, O. (2014). Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems*, pages 163–171.

Shi, W., Ling, Q., Wu, G., and Yin, W. (2015). EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966.

Shi, W., Ling, Q., Yuan, K., Wu, G., and Yin, W. (2014). On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Trans. Signal Process*, 62(7):1750–1761.

Sun, H. and Hong, M. (2018). Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 38–42. IEEE.

Sun, Y., Daneshmand, A., and Scutari, G. (2019). Convergence rate of distributed optimization algorithms based on gradient tracking. *arXiv:1905.02637*.

Uribe, C. A., Lee, S., Gasnikov, A., and Nedić, A. (2018). A dual approach for optimal algorithms in distributed optimization over networks. *arXiv preprint arXiv:1809.00710*.

Wei, E. and Ozdaglar, A. E. (2012). Distributed alternating direction method of multipliers. In *Proceedings of IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 5445–5450.

Wien, A. (2011). *Iterative solution of large linear systems*. Lecture Notes, TU Wien.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78.

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

Yuan, K., Ling, Q., and Yin, W. (2016). On the convergence of decentralized gradient descent. *SIAM J. on Optim.*, 26(3):1835–1854.