Distributed Algorithms for Composite Optimization: Unified Framework and Convergence Analysis

Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari, *Fellow, IEEE*

***Abstract*—We study distributed composite optimization over networks: agents minimize a sum of smooth (strongly) convex functions–the agents' sum-utility–plus a nonsmooth (extended-valued) convex one. We propose a general *unified* algorithmic framework for such a class of problems and provide a convergence analysis leveraging the theory of operator splitting. Distinguishing features of our scheme are: (i) When each of the agent's functions is strongly convex, the algorithm converges at a *linear* rate, whose dependence on the agents' functions and network topology is *decoupled*; (ii) When the objective function is convex (but not strongly convex), similar decoupling as in (i) is established for the coefficient of the proved sublinear rate. This also reveals the role of function heterogeneity on the convergence rate. (iii) The algorithm can adjust the ratio between the number of communications and computations to achieve a rate (in terms of computations) independent on the network connectivity; and (iv) A by-product of our analysis is a tuning recommendation for several existing (non-accelerated) distributed algorithms yielding provably faster (worst-case) convergence rate for the class of problems under consideration.**

***Index Terms*—Distributed optimization, linear convergence.**

I. Introduction

We STUDY distributed optimization over networks, modeled as undirected static graphs. Agents aim at solving

$$\min_{x\in\mathbb{R}^d} F(x)+G(x), \quad F(x)\triangleq\frac{1}{m}\sum_{i=1}^{m} f_i(x), \qquad \text{(P)}$$

where $f_i:\mathbb{R}^d\to\mathbb{R}$ is the cost function of agent i, assumed to be L-smooth, μ-strongly convex (with $\mu\geq 0$), and known only to the agent; and $G:\mathbb{R}^d\to\mathbb{R}\cup\{-\infty,\infty\}$ is a nonsmooth, convex (extended-value) function, which can be used to force shared constraints or some structure on the solution (e.g., sparsity). This setting is fairly general and finds applications in several areas, including network information processing, telecommunications, multi-agent control, and machine learning; see Section V-D for a case-study in machine learning.

The focus of this paper is the design of a unified (first-order) algorithmic framework for Problem (P) over undirected graphs with provably convergence rate. When $G=0$ and $\mu>0$, several distributed schemes have been proposed in the literature that enjoy *linear* rate; examples include EXTRA [3], AugDGM [4], [5], NEXT [6], Harnessing [7], SONATA [8], [9], DIGing [10], NIDS [11], Exact Diffusion [12], MSDA [13], and the distributed algorithms in [14], [15]. When $\mu=0$ and still $G=0$, a sublinear rate of $O(1/k)$ (k counts the number of gradient evaluations) has been certified for some of the above methods [4], [6], [7] and other primal-dual schemes, including D-ADMM [16]. Results for $G\neq 0$ are relatively scarce; to our knowledge, the only two schemes achieving linear rate for strongly convex (P) are SONATA [9] and the one in [17]. Sublinear rate of $O(1/k)$ has been proved for a variety of schemes, including PG-EXTRA [18], D-FBBS [19] and DPGA [20]. Notice that convergence of some of these algorithms have been studied under weaker assumptions on F and network topology than those considered in this paper. For instance, linear rate of [3], [6], [9], [10], [12], [17] is established for F strongly convex (rather than each f_i to be so); [8]–[10], [21], [22] are applicable also to directed graphs, with [8]–[10] considering also time-varying topologies.

Even restricted to the setting of this paper, none of the above studies provide a *unified* algorithmic design and convergence analysis. Furthermore, for most of the schemes, there is a gap between theory and practice: tuning recommendations and rate bounds provided by the analysis are showed numerically being too conservative. To make these algorithms work in practice, practitioners often use manual, ad-hoc tunings. This however makes the comparison of different schemes hard. These issues suggest the following questions:

- **Q1)** Can one unify the design and analysis of distributed algorithms for Problem (P)?
- **Q2)** How do provable rates of such schemes compare each other and with that of the centralized proximal-gradient algorithm applied to (P)?

On (Q1): Recent efforts toward a better understanding of the taxonomy of distributed algorithms are the following: [14]

Manuscript received March 4, 2020; revised November 15, 2020, March 7, 2021, and May 6, 2021; accepted May 19, 2021. Date of publication June 7, 2021; date of current version June 29, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antti Tölli. The work of Jinming Xu was supported in part by NSF of China under Grants 62003302, 62088101, 61922058, U1909207, and the Fundamental Research Funds for the Central Universities (2020QNA5014). The work of Ye Tian, Ying Sun, and Gesualdo Scutari was supported by the USA NSF Grant CIF 1719205, the ARO Grant W911NF1810238, and the ONR Grant. This work was presented in part at IEEE CAMSAP 2019 [1], and IEEE no. 1317827 CDC 2020 [2]. *(Corresponding author: Gesualdo Scutari.)*

Jinming Xu is with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: jimmyxu@zju.edu.cn).

Ye Tian and Gesualdo Scutari are with the School of Industrial Engineering, Purdue University, West-Lafayette, IN 47907 USA (e-mail: tian110@purdue.edu; gscutari@purdue.edu).

Ying Sun is with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, State College, PA 16802 USA (e-mail: sun578@purdue.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2021.3086579, provided by the authors.

Digital Object Identifier 10.1109/TSP.2021.3086579

TABLE I
CONVERGENCE PROPERTIES OF DISTRIBUTED ALGORITHMS FOR L-SMOOTH AND μ-STRONGLY CONVEX $\{f_i\}$ $(\mu > 0)$

$\rho \triangleq \sigma_{\max}(W - J)$ with $J = \frac{1}{m}1_m 1_m^\top$, $\check{\lambda} \triangleq \lambda_{\min}(W)$, and $\mathbb{W}^m \triangleq \{W | W 1_m = 1_m, 1_m^\top W = 1_m^\top \text{ and } \rho < 1\}$, and $\mathbb{S}^m \triangleq \{W | W = W^\top\}$.

Algorithm	**Original assumption**		**Stepsize**		**Rate:** $\mathcal{O}\left(\delta \log(\frac{1}{\epsilon})\right)$	
	$W \in$	$F, \{f_i\}$	**literature (upper bound)**	**this paper (Corollary 19)**	δ**, literature**	δ**, this paper**
EXTRA [3]	$\mathbb{S}^m \cap \mathbb{W}^m$	F scvx	$\mathcal{O}\left(\frac{\mu(1-\rho)}{L^2}\right)$	$\frac{2}{2L/(1+\check{\lambda})+\mu} \geq \frac{1-\rho}{L+\mu}$	$\frac{\kappa^2}{1-\rho}$	$\frac{\kappa}{1-\rho}$
NEXT [6] AugDGM [4], [5]	$\mathbb{W}^m$	F scvx	$\min\{\frac{(1-\rho)^2}{10L\rho\sqrt{n}\sqrt{\kappa}}, \frac{1}{2L}\}$	$\frac{2}{L+\mu}$	$\max\left\{\frac{1}{\gamma\mu}, \frac{1}{1-\rho-\sqrt{10L\rho\sqrt{n}\sqrt{\kappa}\gamma}}\right\}$	$\max\left\{\kappa, \frac{1}{(1-\rho)^2}\right\}$
DIGing [10]	$\mathbb{W}^m$	F scvx	$\mathcal{O}\left(\frac{(1-\rho)^2}{\mu\kappa^{1.5}\sqrt{n}}\right)$	$\frac{2}{L/\lambda_{\min}(W^2)+\mu} \geq \frac{2\lambda_{\min}(W^2)}{L+\mu}$	$\frac{\kappa^{1.5}}{(1-\rho)^2}$	$\max\left\{\frac{\kappa}{\lambda_{\min}(W^2)}, \frac{1}{(1-\rho)^2}\right\}$
Exact Diffusion [12]	$\mathbb{W}^m$	F scvx	$\mathcal{O}\left(\frac{\mu}{L^2}\right)$	$\frac{2}{L+\mu}$	$\frac{\kappa^2}{1-\rho}$	$\max\{\kappa, \frac{1}{1-\rho}\}$
Harnessing [7]	$\mathbb{W}^m$	$\{f_i\}$ scvx	$\mathcal{O}\left(\frac{(1-\rho)^2}{\kappa L}\right)$	$\frac{2}{L/\lambda_{\min}(W^2)+\mu} \geq \frac{2\lambda_{\min}(W^2)}{L+\mu}$	$\frac{\kappa^2}{(1-\rho)^2}$	$\max\left\{\frac{\kappa}{\lambda_{\min}(W^2)}, \frac{1}{(1-\rho)^2}\right\}$
NIDS [11]	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_i\}$ scvx	$\frac{2}{L}$	$\frac{2}{L+\mu}$	$\max\{\kappa, \frac{1}{1-\rho}\}$	$\max\{\kappa, \frac{1}{1-\rho}\}$
[14] $(b = 0)$	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_i\}$ scvx	$\mathcal{O}\left(\frac{(1-\rho)^2}{\kappa L}\right)$	$\frac{2}{L/\lambda_{\min}(W^2)+\mu} \geq \frac{2\lambda_{\min}(W^2)}{L+\mu}$	$\frac{\kappa^2}{(1-\rho)^2}$	$\max\left\{\frac{\kappa}{\lambda_{\min}(W^2)}, \frac{1}{(1-\rho)^2}\right\}$
[14] $(b = \frac{1}{\gamma}W)$	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_i\}$ scvx	N.A.	$\frac{2}{2L/(1+\check{\lambda})+\mu} \geq \frac{1-\rho}{L+\mu}$	N.A.	$\frac{\kappa}{1-\rho}$
[15]	$\mathbb{S}_{++}^m \cap \mathbb{W}^m$	$\{f_i\}$ scvx	(14) in the paper	$\frac{2}{\mu+L/((1-\check{\lambda})\check{\lambda}^K)} \geq \frac{2(1-\check{\lambda})\check{\lambda}^K}{L+\mu}$	N.A.	$\max\left\{\frac{1}{1-\rho^K}, \frac{\kappa}{(1-\check{\lambda})\check{\lambda}^K}\right\}$
[17]	$\mathbb{S}_{++}^m \cap \mathbb{W}^m$	F scvx	$< \frac{\check{\lambda}}{L}$	$\frac{2\check{\lambda}}{L+\mu\check{\lambda}} > \frac{\check{\lambda}}{L}$	$> \max\{\frac{\kappa}{\check{\lambda}}, \frac{1}{\alpha(1-\rho)}\}$	$\max\{\frac{\kappa}{\check{\lambda}}, \frac{1}{\alpha(1-\rho)}\}$
this paper	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_i\}$ scvx	$\frac{2}{L+\mu}$		$\max\left\{\kappa, \frac{1}{1-\rho}\right\}$	

Postilla: Not all the algorithms above were studied under the same setting; the different assumptions on F and W are listed above. The expressions of the stepsize as reported above for DIGing, Exact Diffusion, Harnessing and NIDS (resp. AugDGM and NEXT) are obtained under the extra assumption that W is invertible (resp. $W \succeq 0$).

provides a connection between EXTRA and DIGing; [23] provides a canonical representation of some of the distributed algorithms above–NIDS and Exact-Diffusion are proved to be equivalent; and [24] provides an automatic (numerical) procedure to prove linear rate of some classes of distributed algorithms. These efforts model only first-order distributed algorithms applicable to Problem (P) *with* $G = 0$ and employing a *single* round of communication and gradient computation. Despite these connections, convergence of these schemes has been established by ad-hoc analysis, resulting in different rate expressions and stepsize bounds–Table I summarizes these results within the setting of our paper. For instance, a direct comparison between NIDS [11] and Exact Diffusion [12] shows that, although equivalent [14], [23], they exhibit different theretical rate bounds and admissible stepsize values.

On (Q2): Question (Q2) has been only partially addressed in the literature. For instance, MSDA [13] uses multiple communication steps to achieve the lower complexity bound of (P) when $\mu > 0$ and $G = 0$; the OPTRA algorithm [25] achieves the lower bound when $\mu = 0$ (still and $G = 0$); and the algorithms in [26] and [11] achieve linear rate and can adjust the number of communications performed at each iteration to match the rate of the centralized gradient descent. However it is not clear how to extend (if possible) these methods and their convergence analysis to the more general composite ($G \neq 0$) setting (P). Furthermore, even when $G = 0$, the rate results of existing algorithms are not theoretically comparable with each other–see Table I; they have been obtained under different stepsize range values and problem assumptions (e.g., on the weight matrices). Similarly, when $\mu = 0$, EXTRA [3], DIGing [7], [10] D-ADMM [16], and PG-EXTRA [18], D-FBBS [19], DPGA [20] achieve a sublinear rate of $O(1/k)$ for $G = 0$ and $G \neq 0$, respectively. However, the rate expression therein lacks of insight on the dependence of the rate on the key design parameters (e.g., the stepsize).

This paper aims at addressing Q1 and Q2 in the setting (P), over undirected graphs. Our major contributions are discussed next.

1) Unified framework and rate analysis: We propose a general primal-dual distributed algorithmic framework that unifies both ATC (Adapt-Then-Combine)- *and* CTA (Combine-Then-Adapt)-based distributed algorithms, solving either smooth ($G = 0$) or *composite* optimization problems ($G \neq 0$). Most of existing ATC and CTA schemes are special cases of the proposed framework–see Table II. By product of our unified framework and convergence conditions, several existing schemes, proposed only to solve smooth instances of (P) [3], [4], [6], [7], [11], [12], gain now their "proximal" extension and thus become applicable also to composite optimization while enjoying the same convergence rate (as derived in this paper) of their "non-proximal" counterparts.

2) Improving upon existing results and tuning recommendations: Under the setting of this work, our results improve on existing convergence conditions and rate bounds, such as [3], [4], [6], [7], [11], [12]–Table I shows the improvement achieved by our analysis in terms of stepsize bounds and rate expression (see Section V-C for more details). The tightness of our rates as well as the established ranking of the algorithms based on the new rate expressions are supported by numerical results.

3) Rate separation when $G \neq 0$*:* For ATC-based schemes, when $\mu > 0$, the dependency of the linear rate on the agents' functions and the network topology are *decoupled*, matching the rate of the proximal gradient algorithm applied to (P). Furthermore, the optimal stepsize value is independent on the network and matches the optimal choice for the centralized proximal gradient algorithm. When $\mu = 0$, we provide an

TABLE II
SPECIAL CASES OF ALGORITHM (4) FOR SPECIFIC CHOICES OF A, B, C MATRICES AND GIVEN GOSSIP MATRIX $-I \prec W \preceq I$.

Algorithm	**Problem**	**Choice of the A, B, C**	**# communications**
EXTRA [3]	F	$A = \frac{I+W}{2} \quad B = I \quad C = \frac{I-W}{2}$	1
NEXT [6]/AugDGM [4], [5]	F	$A = W^2 \quad B = W^2 \quad C = (I-W)^2$	2
DIGing [10]/Harnessing [7]	F	$A = W^2 \quad B = I \quad C = (I-W)^2$	2
NIDS [11]/Exact Diffusion [12]	F	$A = \frac{I+W}{2} \quad B = \frac{I+W}{2} \quad C = \frac{I-W}{2}$	1
[14] ($B' = bI$)	F	$A = W^2 + \gamma b(I-W) \quad B = I \quad C = (I-W)^2 + \gamma b(I-W)$	2
[15]	F	$A = W^K \quad B = \sum_{i=0}^{K-1} W^i \quad C = I - W^K$	K
[17]	$F+G$	$A = W \quad B = I \quad C = \alpha(I-W)$ with $0 \prec W \preceq I$ and $\alpha \leq 1$	1

explicit expression of the sublinear rate (beyond the "Big-O" decay) revealing a similar decoupling between optimization and network parameters. This expression sheds also light on the choice of the stepsize minimizing the rate bound, which is not necessarily $1/L$ but instead depends on the network parameters as well as the degree of heterogeneity of the agents' functions (cf. Section VI). This shows that one can achieve faster rates when the agents' functions are similar, a fact that happens often in machine learning applications, as discussed in details in Section V-D. These results are a major departure from existing analyses, which do not show such a clear separation, and complements those in [11] applicable only to smooth and strongly convex instances of (P).

4) Balancing computation and communication: When $\mu > 0$, the proposed scheme can adjust the ratio between the number of communication and computation steps to improve the overall rate. We show that Chebyshev acceleration can also be employed to further reduce the number of communication steps per computation.

The results of this work have been partially presented in [1], [2]. While preparing the final version of this manuscript, we noticed the arxiv submission [27], which is an independent and parallel work (cf. [1]). There are some substantial differences between our findings and [27]: i) our algorithmic framework unifies ATC and CTA schemes while [27] can cover only ATC ones; our analysis is based on an operator contraction-based analysis, which is of independent interest; and ii) we study convergence also when F is convex but not strongly convex while [27] focuses only on strongly convex problems.

Due to space limits, some of the proofs are omitted and reported in the on-line supplementary material of this paper. The full version of the paper is also available online [28].

II. PROBLEM STATEMENT

We study Problem (P) under the following assumption, capturing either strongly convex or just convex objectives.

Assumption 1: (i) Each $f_i : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex, $\mu \geq 0$, and L-smooth; (ii) and $G : \mathbb{R}^d \to \mathbb{R} \cup \{\pm\infty\}$ is proper, closed and convex. When $\mu > 0$, define $\kappa \triangleq L/\mu$.

Network model: Agents are embedded in a network, modeled as an undirected, static graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (agents) and $\{i, j\} \in \mathcal{E}$ if there is an edge (communication link) between node i and j. We make the blanket assumption that $\mathcal{G}$ is connected. We introduce the following matrices associated with $\mathcal{G}$, which will be used to build the proposed distributed algorithms.

Definition 2 (Gossip matrix): A matrix $W \triangleq [W_{ij}] \in \mathbb{R}^{m\times m}$ is said to be compliant to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $W_{ij} \neq 0$ for $\{i, j\} \in \mathcal{E}$, and $W_{ij} = 0$ otherwise. The set of such matrices is denoted by $\mathcal{W}_{\mathcal{G}}$.

Definition 3 (K-hop gossip matrix): Given $K \in \mathbb{N}_+$, a matrix $W' \in \mathbb{R}^{m\times m}$ is said to be a K-hop gossip matrix associated to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $W' = P_K(W)$, for some $W \in \mathcal{W}_{\mathcal{G}}$, where $P_K(\cdot)$ is a monic polynomial of order K.

Note that, if $W \in \mathcal{W}_{\mathcal{G}}$, using W_{ij} to linearly combine information between two immediate neighbor agents i and j corresponds to performing a single communication round. Using a K-hop matrix $W' = P_K(W)$ requires instead K consecutive rounds of communications. K-hop gossip matrices are crucial to employ acceleration of the communication step, which will be a key ingredient to exploit the tradeoff between communications and computations (cf. Section V-C3).

A saddle-point reformulation: Our path to design distributed solution methods for (P) is to solve a saddle-point reformulation of (P) via general proximal splitting algorithms that are implementable over $\mathcal{G}$. Following a standard path in the literature, we introduce local copies $x_i \in \mathbb{R}^d$ (the i-th one is owned by agent i) of x and functions

$$f(X) \triangleq \sum_{i=1}^{m} f_i(x_i) \quad \text{and} \quad g(X) \triangleq \sum_{i=1}^{m} G(x_i), \tag{1}$$

with $X \triangleq [x_1, \ldots, x_m]^\top \in \mathbb{R}^{m\times d}$; (P) can be rewritten as

$$\min_{X\in\mathbb{R}^{m\times d}} f(X) + g(X), \ \text{s.t.} \ \sqrt{C}X = 0, \tag{2}$$

where C satisfies the following assumption ($\mathtt{span}(\bullet)$ and $\mathtt{null}(\bullet)$ denote the range space and null space of the argument vector/matrix, respectively):

Assumption 4: $C \in \mathbb{S}_+^m$ and $\mathtt{null}(C) = \mathtt{span}(1_m)$.

Under this condition, the constraint $\sqrt{C}X = 0$ enforces a consensus among x_i's and thus (2) is equivalent to (P).

The set of points satisfying the KKT conditions of (2) reads:

$$\mathcal{S}_{\text{KKT}} \triangleq \Big\{ X \in \mathbb{R}^{m\times d} \,\Big|\, \exists Y \in \mathbb{R}^{m\times d} \text{ such that } \sqrt{C}X = 0, \quad \nabla f(X) + \sqrt{C}Y \in -\partial g(X) \Big\}, \tag{3}$$

where $\nabla f(X) \triangleq [\nabla f_1(x_1), \nabla f_2(x_2), \ldots, \nabla f_m(x_m)]^\top$ and $\partial g(X)$ denotes the subdifferential of g at X. Then we have the following standard result.

Lemma 5: Under Assumption 1, $x^\star \in \mathbb{R}^d$ is an optimal solution of Problem (P) if and only if $1_m x^{\star\top} \in \mathcal{S}_{\mathrm{KKT}}$.

Building on Lemma 5, in the next section, we propose a general distributed algorithm for (P) based on a suitably defined operator splitting solving the KKT system (3).

III. A General Primal-Dual Proximal Algorithm

The proposed general primal-dual proximal algorithm, termed $ABC-$Algorithm, reads

$$X^k = \mathrm{prox}_{\gamma g}\left(Z^k\right), \tag{4a}$$

$$Z^{k+1} = AX^k - \gamma B \nabla f(X^k) - Y^k, \tag{4b}$$

$$Y^{k+1} = Y^k + CZ^{k+1}, \tag{4c}$$

with $Z^0 \in \mathbb{R}^{m\times d}$ and $Y^0 = 0$. In (4a), $\mathrm{prox}_{\gamma g}(X) \triangleq \arg\min_Y g(Y) + \frac{1}{2\gamma}\|X - Y\|^2$ is the standard proximal operator. Eq. (4a) and (4b) represent the update of the primal variables, where $A, B \in \mathbb{R}^{m\times m}$ are suitably chosen weight matrices, and $\gamma > 0$ is the stepsize. Eq. (4c) represents the update of the dual variables.

Define the set

$$\mathcal{S}_{\mathtt{Fix}} \triangleq \big\{X \in \mathbb{R}^{m\times d} \,\big|\, CX = 0 \text{ and } 1_m^\top (I - A)X + \gamma\, 1_m^\top B \nabla f(X) \in -\gamma\, 1_m^\top \partial g(X)\big\}. \tag{5}$$

Since all agents share the same G, it is not difficult to check that any fixed point $(X^\star, Z^\star, Y^\star)$ of Algorithm (4) is such that $X^\star \in \mathcal{S}_{\mathtt{Fix}}$. The following are *necessary* and *sufficient* conditions on A, B for $X^\star \in \mathcal{S}_{\mathtt{Fix}}$ to be a solution of (2).

Assumption 6: The weight matrices $A, B \in \mathbb{R}^{m\times m}$ satisfy: $1_m^\top A\, 1_m = m$, and $1_m^\top B = 1_m^\top$.

Lemma 7: Under Assumption 4, $\mathcal{S}_{\mathrm{KKT}} = \mathcal{S}_{\mathtt{Fix}}$ if and only if A, B satisfy Assumption 6.

Proof: See the details in the supplementary material. ■

A. Connections With Existing Distributed Algorithms

Algorithm (4) contains a gamut of distributed (and centralized) schemes, corresponding to different choices of the weight matrices A, B and C; any $A, B, C \in \mathcal{W}_{\mathcal{G}}$ leads to distributed implementations. The use of general matrices A and B (rather the more classical choices $A = B$ or $B = I$) permits a unification of both ATC- and CTA-based updates; this includes several existing distributed algorithms proposed for special cases of (P), as discussed next.

We begin rewriting (4) in the following equivalent form by subtracting (4b) at iteration $k+1$ from (4b) at iteration k:

$$\begin{aligned} Z^{k+2} = (I - C)Z^{k+1} &+ A(X^{k+1} - X^k) \\ &- \gamma B(\nabla f(X^{k+1}) - \nabla f(X^k)), \end{aligned} \tag{6}$$

where $X^k = \mathrm{prox}_{\gamma g}(Z^k)$.

When $G = 0$, (6) reduces to

$$\begin{aligned} X^{k+2} = (I - C + A)&X^{k+1} \\ &- AX^k - \gamma B(\nabla f(X^{k+1}) - \nabla f(X^k)). \end{aligned} \tag{7}$$

We show next that the schemes in [3], [4], [6], [7], [10]–[12], [14], [15], [17] are all special cases of Algorithm (4). Table II summarizes the specific choices of A, B and C in (4) yielding the desired equivalence, where $W \in \mathcal{W}_{\mathcal{G}}$ is the weight matrix used in the target distributed algorithms. Notice that all these choices satisfy Assumptions 4 and 6.

1) EXTRA [3]: EXTRA solves (P) with $G = 0$, and reads

$$\begin{aligned} X^{k+2} = (I + W)&X^{k+1} - \tilde{W}X^k \\ &- \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)), \end{aligned} \tag{8}$$

where $W, \tilde{W}$ are two design weight matrices satisfying $(I + W)/2 \succeq \tilde{W} \succeq W$ and $\tilde{W} \succ 0$. Clearly, (8) is an instance of (7) [and thus (4)], with $A = \tilde{W}$, $B = I$, and $C = \tilde{W} - W$.

2) NIDS [11] / Exact diffusion [12], [29]: The NIDS (Exact Diffusion) algorithm applies to (P) with $G = 0$, and reads

$$\begin{aligned} X^{k+2} = \frac{I + W}{2}(2&X^{k+1} - X^k \\ &- \gamma(\nabla f(X^{k+1}) - \nabla f(X^k))), \end{aligned}$$

which is an instance of our general scheme, with $A = B = (I + W)/2$ and $C = (I - W)/2$.

3) NEXT [6] & AugDGM [4]: The gradient tracking-based algorithms NEXT/AugDGM applied to (P) with $G = 0$, are:

$$X^{k+1} = W(X^k - \gamma Y^k), \tag{9a}$$

$$Y^{k+1} = W(Y^k + \nabla f(X^{k+1}) - \nabla f(X^k)). \tag{9b}$$

Eliminating the Y-variable, (9) can be rewritten as:

$$\begin{aligned} X^{k+2} = 2W&X^{k+1} - W^2 X^k \\ &- \gamma W^2(\nabla f(X^{k+1}) - \nabla f(X^k)), \end{aligned}$$

which is clearly an instance of our general scheme (4), with $A = B = W^2, C = (I - W)^2$. Notice that distributed gradient tracking schemes in the so-called CTA form are also special cases of Algorithm (4). For instance, one can show that the DIGing algorithm [10] corresponds to the setting $A = W^2, B = I$, and $C = (I - W)^2$.

4) General primal-dual scheme [14], [15]: A general distributed primal-dual algorithm was proposed in [14] for (P) with $G = 0$ as follows

$$X^{k+1} = WX^k - \gamma(\nabla f(X^k) + Y^k), \tag{10a}$$

$$Y^{k+1} = Y^k - (I - W)(\nabla f(X^k) + Y^k - B'X^k), \tag{10b}$$

where B' can be bI or bW for some positive constant $b > 0$ therein. Eliminating the Y-variable, (10) reduces to

$$\begin{aligned} X^{k+2} = 2WX^{k+1} &- (W^2 + \gamma(I - W)B')X^k \\ &- \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)), \end{aligned}$$

which corresponds to the proposed algorithm, with $A = W^2 + \gamma(I - W)B,' B = I, C = (I - W)^2 + \gamma(I - W)B'$.

Similarly, building on a general augmented Lagrangian, another general primal-dual algorithm was proposed in [15] for (P) with $G = 0$, which reads

$$X^{k+1} = (I - \alpha B')^K X^k - \alpha C'(\nabla f(X^k) + A'^\top Y^k), \tag{11a}$$

$$Y^{k+1} = Y^k + \beta A' X^{k+1}, \tag{11b}$$

where $A,' B,' C'$ are certain weight matrices therein and $C' = \sum_{i=0}^{K-1}(I - \alpha B')^i$, with K being the number of communication steps performed at each iteration. Eliminating Y yields

$$\begin{aligned} X^{k+2} = {} & (I + (I - \alpha B')^K - \alpha\beta C' A'^\top A')X^{k+1} \\ & - (I - \alpha B')^K X^k - \alpha C'(\nabla f(X^{k+1}) - \nabla f(X^k)), \end{aligned}$$

which corresponds to Algorithm (4) with $A = (I - \alpha B')^K, B = C,' C = \alpha\beta C' A'^\top A'$. Notice that, letting $W = I - \alpha B'$ and $B' = \beta A'^\top A'$, we have $A = W^K, B = \sum_{i=0}^{K-1} W^i$ and $C = (I - W)\sum_{i=0}^{K-1} W^i = I - W^K$, which satisfy Assumption 6.

6) Decentralized proximal algorithm [17]: A proximal algorithm is proposed to solve (P) with $G \neq 0$, which reads

$$\begin{aligned} Z^{k+2} = {} & (I - \alpha B')Z^{k+1} + (I - B')(X^{k+1} - X^k) \\ & - \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)), \end{aligned}$$

where $X^k = \text{prox}_{\gamma g}(Z^k)$ and $0 \preceq B' \prec I$ is some matrix ensuring consensus. It is easy to show that the above algorithm corresponds to Algorithm (4) with $A = I - B,' B = I, C = \alpha B'$. Choosing $W = I - B'$, we have $A = W, B = I$ and $C = \alpha(I - W)$, which clearly satisfy Assumption 6. Note that, since $B = I$, this algorithm (and thus [17]) is of CTA form and cannot model ATC-based schemes, such as NEXT/AugDGM and NIDS/Exact Diffusion listed in Table II.

IV. An Operator Splitting Interpretation

Our convergence analysis builds on an equivalent fixed-point reformulation of Algorithm (4), whose mapping enjoys a favorable decomposition in terms of contractive and nonexpansive operators. We begin introducing the following assumptions.

Assumption 8: The weight matrices satisfy:

i) $A = BD$;

ii) B and C commute.

Under the above assumption, the following lemma provides an operator splitting form for Algorithm (4).

Proposition 9: Given the sequence $\{(Z^k, X^k, Y^k)\}_{k \in \mathbb{N}_+}$ generated by Algorithm (4), define $U^k \triangleq [(Z^k)^\top, (Y^k)^\top]^\top$. Under Assumption 8, the following hold:

1)

$$U^k = \begin{bmatrix} B & 0 \\ 0 & B\sqrt{C} \end{bmatrix} \widetilde{U}^k, \quad \text{with} \quad \widetilde{U}^k \triangleq \begin{bmatrix} \widetilde{Z}^k \\ \sqrt{C}\widetilde{Y}^k \end{bmatrix}; \tag{12}$$

and $\{\widetilde{U}^k\}_k$ satisfies the following dynamics

$$\widetilde{U}^{k+1} = \underbrace{\begin{bmatrix} (D - \gamma\nabla f) \circ \text{prox}_{\gamma g} \circ B & -\sqrt{C} \\ \sqrt{C}(D - \gamma\nabla f) \circ \text{prox}_{\gamma g} \circ B & I - C \end{bmatrix}}_{T} \widetilde{U}^k, \quad k \geq 1, \tag{13}$$

with initialization $\widetilde{Z}^1 = \widetilde{Y}^1 = (D - \gamma\nabla f)(X^0)$;

2) The operator T can be decomposed as

$$T = \underbrace{\begin{bmatrix} I & -\sqrt{C} \\ \sqrt{C} & I - C \end{bmatrix}}_{\triangleq T_C} \underbrace{\begin{bmatrix} D - \gamma\nabla f & 0 \\ 0 & I \end{bmatrix}}_{\triangleq T_f} \underbrace{\begin{bmatrix} \text{prox}_{\gamma g} & 0 \\ 0 & I \end{bmatrix}}_{\triangleq T_g} \underbrace{\begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}}_{\triangleq T_B}, \tag{14}$$

where T_C and T_B are the operators associated with communications while T_f and T_g are the gradient and proximal operators, respectively;

3) Every fixed point $\widetilde{U}^\star \triangleq [\widetilde{Z}^\star, \sqrt{C}\widetilde{Y}^\star]$ of T is such that $X^\star \triangleq \text{prox}_{\gamma g}(B\widetilde{Z}^\star) \in \mathcal{S}_{\texttt{Fix}}$. Therefore, $X^\star = 1_m x^{\star\top}$, where $x^\star$ is an optimal solution of (P).

Proof: From (4), we have $Z^{k+1} = (I - C)Z^k + A(X^k - X^{k-1}) - \gamma B(\nabla f(X^k) - \nabla f(X^{k-1}))$, which applied recursively yields

$$\begin{aligned} Z^{k+1} = {} & \sum_{t=1}^{k}(I - C)^{k-t}\Big(A(X^t - X^{t-1}) \\ & - \gamma B(\nabla f(X^t) - \nabla f(X^{t-1}))\Big) \\ & + (I - C)^k \left(AX^0 - \gamma B\nabla f(X^0)\right) \\ \overset{(*)}{=} {} & B\Bigg(\sum_{t=1}^{k}(I - C)^{k-t}(D(X^t - X^{t-1}) \\ & - \gamma(\nabla f(X^t) - \nabla f(X^{t-1})) \\ & + (I - C)^k \left(DX^0 - \gamma\nabla f(X^0)\right)\Bigg) \\ = {} & B\sum_{t=0}^{k}(I - C)^{k-t}(D - \gamma\nabla f)(X^t) \\ & - B\sum_{t=0}^{k-1}(I - C)^{k-1-t}(D - \gamma\nabla f)(X^t), \end{aligned}$$

where in $(*)$ we used Assumption 17 i) and 17 iv).

Define $\widetilde{Z}^k$ such that $Z^k = B\widetilde{Z}^k$, $k \geq 1$; and let

$$\widetilde{Y}^{k+1} \triangleq \sum_{t=1}^{k+1}\widetilde{Z}^t = \sum_{t=0}^{k}(I - C)^{k-t}(D - \gamma\nabla f)(X^t), \tag{15}$$

for $k \geq 0$. It is clear from the definition of $\widetilde{Z}$ and $\widetilde{Y}$ that

$$\begin{bmatrix} \widetilde{Z}^{k+1} \\ \widetilde{Y}^{k+1} \end{bmatrix} = \begin{bmatrix} (D - \gamma\nabla f) \circ \text{prox}_{\gamma g} \circ B & -C \\ (D - \gamma\nabla f) \circ \text{prox}_{\gamma g} \circ B & I - C \end{bmatrix} \begin{bmatrix} \widetilde{Z}^k \\ \widetilde{Y}^k \end{bmatrix}. \tag{16}$$

Introducing $\widetilde{U}^k$ as defined in (12), it follows from (16) that $\widetilde{U}^k$ obeys the dynamics (13). The equation $Y^k = BC\widetilde{Y}^k$ follows readily from (4c) and (15). Finally, the decomposition of the transition matrix T can be checked by inspection.

We prove now the last statement of the theorem. For every fixed point $\widetilde{U}^\star \triangleq [\widetilde{Z}^\star, \sqrt{C}\widetilde{Y}^\star]$ of T, we have $\texttt{span}(\widetilde{Z}^\star) \subset$

$\mathtt{span}(1)$ and

$$-1^\top\left(B(D-\gamma\nabla f)\circ\text{prox}_{\gamma g}\circ B\left(\widetilde{Z}^\star\right)\right)+1^\top B\widetilde{Z}^\star=0. \tag{17}$$

For $X^\star \triangleq \text{prox}_{\gamma g}(B\widetilde{Z}^\star)$, it holds $\mathtt{span}(X^\star)\subset\mathtt{span}(1)$ and

$$B\widetilde{Z}^\star\in X^\star+\gamma\partial g(X^\star). \tag{18}$$

Combining (17) and (18) leads to $1^\top(I-A)X^\star+\gamma\,1^\top B\nabla f(X^\star)\in-\gamma\,1^\top\partial g(X^\star)$, which is equivalent to $X^\star\in\mathcal{S}_{\mathtt{Fix}}$. The proof follows from Lemma 5 and 7. ■

We summarize next the main properties of the operators T_C, T_f, T_g, and T_B, which will be instrumental to establish linear convergence rate of the proposed algorithm. We will use the following notation: given $X\in\mathbb{R}^{2m\times d}$, we denote by $(X)_u$ and $(X)_\ell$ its upper and lower $m\times d$ matrix-block; for any matrix $A\in\mathbb{R}^{m\times m}$, we denote $\Lambda_A=\mathtt{diag}(A,I)\in\mathbb{R}^{2\,m\times 2\,m}$ and $V_A=\mathtt{diag}(I,A)\in\mathbb{R}^{2\,m\times 2\,m}$.

Lemma 10 (Contraction of T_C): The operator T_C satisfies

$$\|T_C\,X-T_C\,Y\|_{\Lambda_{I-C}}=\|X-Y\|_{V_{I-C}},\quad\forall X,Y\in\mathbb{R}^{2m\times d}.$$

Proof: The result comes readily from the definition of T_C and the fact that $T_C^\top\Lambda_{I-C}T_C=V_{I-C}$. ■

Lemma 11 (Contraction of T_f): Consider the operator T_f under Assumption 1, with $\mu>0$, and $0\prec D\preceq I$. If $0<\gamma\leq\gamma^\star(D)$ with

$$\gamma^\star(D)\triangleq\frac{2\lambda_{\min}(D)}{L+\mu\cdot\lambda_{\min}(D)}, \tag{19}$$

then

$$\|(T_fX)_u-(T_fY)_u\|^2\leq q(D,\gamma)\,\|(X)_u-(Y)_u\|_D^2,$$

$\forall X,Y\in\mathbb{R}^{2m\times d}$, where

$$q(D,\gamma)=1-\frac{2\gamma L}{\kappa+\lambda_{\min}(D)}. \tag{20}$$

The stepsize minimizing the contraction factor is $\gamma=\gamma^\star(D)$, resulting in the smallest achievable $q(D,\gamma)$, given by

$$q^\star(D)\triangleq\left(\frac{\kappa-\lambda_{\min}(D)}{\kappa+\lambda_{\min}(D)}\right)^2. \tag{21}$$

Proof: See Section A in Appendix. ■

We conclude with the properties of T_g and T_B, which follow readily from the non-expansive property of the proximal operator and the linear nature of T_B, respectively.

Lemma 12 (Non-expansiveness of T_g): The operator T_g satisfies: $\forall X,Y\in\mathbb{R}^{2m\times d}$,

$$\|(T_g\,X)_u-(T_g\,Y)_u\|^2\leq\|(X)_u-(Y)_u\|^2$$
$$(T_g\,X)_\ell=(X)_\ell.$$

Lemma 13 (Non-expansiveness of T_B): The operator T_B satisfies: $\forall X\in\mathbb{R}^{2m\times d}$,

$$\|(T_B\,X)_u\|^2=\|(X)_u\|_{B^2}^2,\quad(T_g\,X)_\ell=(X)_\ell.$$

V. Linear Convergence

In this section we prove linear convergence of Algorithm (4), under strong convexity of each f_i. Since most of the algorithms in the literature considered only the case $G=0$, we begin with that setting (cf. Section V-A). Sec. V-B extends our analysis to $G\neq 0$. Finally, we comment our results in Section V-C.

A. Convergence Under $G=0$

Consider Problem (P) with $G=0$. Algorithm (4) reduces to

$$X^{k+1}=AX^k-\gamma B\nabla f(X^k)-Y^k, \tag{22a}$$

$$Y^{k+1}=Y^k+CX^{k+1}, \tag{22b}$$

with $X^0\in\mathbb{R}^{m\times d}$ and $Y^0=0$.

Theorem 15 below establishes linear convergence of Algorithm (22) under the following assumption on A,B and C.

Assumption 14: The weight matrices $A\in\mathbb{R}^{m\times m}$, $B,C\in\mathbb{S}^m$ and the stepsize γ satisfy:

i) $A=BD$ with $D\in\mathbb{S}^m$ and $0\prec D\preceq I$;
ii) $1_m^\top D1_m=m$ and $1_m^\top B=1_m^\top$;
iii) $0\preceq C\prec I$ and $\mathtt{null}(C)=\mathtt{span}(1_m)$;
iv) B and C commute;
v) $q(D,\gamma)AB\prec(I-C)$ and $0<\gamma\leq\gamma^\star(D)$,

where $q(D,\gamma)$ and $\gamma^\star(D)$ are defined in (20) and (19), respectively.

Assumption 14 is quite mild and satisfied by a variety of algorithms. For instance, all the algorithms in Table II can satisfy it with proper choices of W. The commuting property of B and C is trivially satisfied when $B,C\in P_K(W)$, for some given $W\in\mathcal{W}_{\mathcal{G}}$.

Theorem 15 (Linear rate for $T_CT_fT_B$): Consider Problem (P) under Assumption 1, $\mu>0$, and $G=0$, with solution $x^\star$. Let $\{(X^k,Y^k)\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm (22) under Assumption 14. Then, $\|X^k-1_mx^{\star\top}\|^2=\mathcal{O}(\delta^k)$, with

$$\delta\triangleq\max\left(q(D,\gamma)\,\lambda_{\max}(AB(I-C)^{-1}),1-\lambda_2(C)\right), \tag{23}$$

where $q(D,\gamma)$ is defined in (20).

Proof: Since (22) corresponds to Algorithm (4) with $G=0$, by Assumption 14 and Prop. 9, (22) can be equivalently rewritten in the form (13), with $T_g=I$; and thus the Z- and X-variables coincide. Define $X^\star=Z^\star\triangleq 1_mx^{\star\top}$. Let $\widetilde{U}^k=[(\widetilde{Z}^k)^\top,(\sqrt{C}\widetilde{Y}^k)^\top]^\top$ be the auxiliary sequence defined in (12) with $\widetilde{U}^\star\triangleq[\widetilde{Z}^\star,\sqrt{C}\widetilde{Y}^\star]$ the fixed point of $T=T_CT_fT_B$. Then, we have

$$\begin{aligned}\left\|X^k-X^\star\right\|^2=\left\|Z^k-Z^\star\right\|^2&\overset{(12)}{\leq}\left\|\widetilde{Z}^k-\widetilde{Z}^\star\right\|_{B^2}^2\\&\leq\frac{\lambda_{\max}(B^2)}{\lambda_{\min}(I-C)}\left\|\widetilde{Z}^k-\widetilde{Z}^\star\right\|_{I-C}^2\\&\leq\frac{\lambda_{\max}(B^2)}{\lambda_{\min}(I-C)}\left\|\widetilde{U}^k-\widetilde{U}^\star\right\|_{\Lambda_{I-C}}^2.\end{aligned} \tag{24}$$

Using (13) in (24), it is sufficient to prove that T is contractive w.r.t. the norm $\|\cdot\|_{\Lambda_{I-C}}$. To this end, consider the following chain of inequalities: $\forall\, X, Y \in \mathbb{R}^{2m\times d}$, $X_\ell, Y_\ell \in \mathtt{span}(\sqrt{C})$,

$$\begin{aligned}
&\|T\,X - T\,Y\|^2_{\Lambda_{I-C}}\\
&= \|T_C \circ T_f \circ T_B\,(X) - T_C \circ T_f \circ T_B\,(Y)\|^2_{\Lambda_{I-C}}\\
&\overset{Lem.\ 10}{=} \|T_f \circ T_B\,(X) - T_f \circ T_B\,(Y)\|^2_{V_{I-C}}\\
&\overset{Lem.\ 11}{\leq} \|T_B\,(X) - T_B\,(Y)\|^2_{\mathtt{diag}(q(D,\gamma)\,D,\,I-C)}\\
&\overset{Lem.\ 13}{=} \|X - Y\|^2_{\mathtt{diag}(q(D,\gamma)\,BDB,\,I-C)}\,.
\end{aligned} \tag{25}$$

Note that: i) for all $(Z)_u \in \mathbb{R}^{m\times d}$,

$$\begin{aligned}
&\|(Z)_u\|^2_{BDB} = \|(I-C)^{\frac{1}{2}}(Z)_u\|^2_{(I-C)^{-1/2}BDB(I-C)^{-1/2}}\\
&\leq \lambda_{\max}(AB(I-C)^{-1})\|(I-C)^{\frac{1}{2}}(Z)_u\|^2\\
&= \lambda_{\max}(AB(I-C)^{-1})\,\|(Z)_u\|^2_{I-C}\,;
\end{aligned}$$

and ii) $X_\ell, Y_\ell \in \mathtt{span}(\sqrt{C})$. The upper block term of the RHS of (25) can be upper bounded by $q(D,\gamma)\lambda_{\max}(AB(I-C)^{-1})\|X_u - Y_u\|^2_{\Lambda_{I-C}}$; and the lower block term of that can be upper bounded by $(1-\lambda_2(C))\|X_\ell - Y_\ell\|^2$. Together we have $\|T\,X - T\,Y\|^2_{\Lambda_{I-C}} \leq \delta\|X-Y\|^2_{\Lambda_{I-C}}$. ■

Note that Theorem 15 is the first unified convergence result stating linear rate for ATC (corresponding to $D=I$) *and* CTA (corresponding to $B=I$) schemes. Because of this generality and consistency with existing conditions for the convergence of CTA-based schemes, the choice of the stepsize satisfying Assumption 14 might depend on some network parameters. This is due to the fact that $\lambda_{\max}(AB(I-C)^{-1}) \geq 1$, since $(I-C)^{-1/2}AB(I-C)^{-1/2}1_m = 1_m$. Hence, when $\lambda_{\max}(AB(I-C)^{-1}) > 1$, the stepsize needs to be leveraged to guarantee that $q(D,\gamma)\,\lambda_{\max}(AB(I-C)^{-1}) < 1$, reducing the range of feasible values. For instance, this happens for i) CTA schemes ($B=I$) such that $D \preceq I-C$ does not hold; of ii) for ATC schemes ($D=I$) that do not satisfy the condition $B^2 \preceq I-C$.

Corollary 16 below provides a condition on the weight matrices enlarging the range of the stepsize to $[0, \gamma^\star(D)]$. Furthermore, the tuning minimizing the contraction factor δ in (23) is derived.

Corollary 16: Consider the setting of Theorem 15, and further assume $AB \preceq I-C$. Then, $\|X^k - 1_m x^{\star\top}\|^2 = \mathcal{O}(\delta^k)$, with

$$\delta = \max\left(q(D,\gamma),\ 1-\lambda_2(C)\right). \tag{26}$$

The stepsize that minimizes (26) is $\gamma = \gamma^\star(D) = \frac{2\lambda_{\min}(D)}{L+\mu\cdot\lambda_{\min}(D)}$, resulting in the contraction factor

$$\delta = \max\left(\left(\frac{\kappa-\lambda_{\min}(D)}{\kappa+\lambda_{\min}(D)}\right)^2, 1-\lambda_2(C)\right). \tag{27}$$

The smallest δ is achieved choosing $D=I$, which yields $\gamma = \gamma^\star \triangleq \frac{2}{\mu+L}$ and

$$\delta^\star = \max\left\{\left(\frac{\kappa-1}{\kappa+1}\right)^2,\ 1-\lambda_2(C)\right\}. \tag{28}$$

Proof: Since $(I-C)^{-1/2}AB(I-C)^{-1/2}1_m = 1_m$ and $AB \preceq I-C$, we have $\lambda_{\max}(AB(I-C)^{-1}) = 1$, which together with (23) yield (26). Eq. (27) follows readily from the decreasing property of $q(D,\gamma)$ on $\gamma \in (0, \gamma^\star(D)]$, for any given $0 \prec D \preceq I$. Finally, (28) is the result of the following optimization problem: $\max_{D\in\mathbb{S}^m} \lambda_{\min}(D)$, subject to $0 \prec D \preceq I$ [Assumption 14(i)] and $1_m^\top D 1_m = m$ [Assumption 14(ii)], whose solution is $D=I$. ■

B. The General Case $G \neq 0$

We establish now linear convergence of Algorithm (4) applied to Problem (P), with $G \neq 0$. We introduce the following assumption similar to Assumption 14 for $G=0$.

Assumption 17: The weight matrices $A \in \mathbb{R}^{m\times m}$, $B, C \in \mathbb{S}^m$ and the stepsize γ satisfy:

i) $A = BD$ with $D \in \mathbb{S}^m$ and $0 \prec D \preceq I$;
ii) $1_m^\top D 1_m = m$ and $1_m^\top B = 1_m^\top$;
iii) $0 \preceq C \prec I$ and $\mathtt{null}(C) = \mathtt{span}(1_m)$;
iv) B and C commute;
v) $q(D,\gamma)\,B^2 \prec (I-C)$ and $0 < \gamma \leq \gamma^\star(D)$,

where $q(D,\gamma)$ and $\gamma^\star(D)$ are defined in (20) and (19), respectively.

Condition v) in Assumption 17 is slightly stronger than its counterpart in Assumption 14 (as $BDB \prec B^2$). This is due to the complication of dealing with the nonsmooth function G (the presence of the proximal operator T_g). However, as shown in Corollary 19 below, this does not affect the smallest achievable contraction rate, which coincides with the one attainable when $G=0$. Note that Assumption 17 is satisfied by all the algorithms in Table II.

Theorem 18 (Linear rate for $T = T_C T_f T_g T_B$): Consider Problem (P) under Assumption 1 with $\mu > 0$, whose optimal solution is $x^\star$. Let $\{(X^k, Z^k, Y^k)\}_{k\geq 0}$ be the sequence generated by Algorithm (4) under Assumption 17. Then $\|X^k - 1_m x^{\star\top}\|^2 = \mathcal{O}(\delta^k)$, with

$$\delta \triangleq \max\left(q(D,\gamma)\,\lambda_{\max}(B^2(I-C)^{-1}),\ 1-\lambda_2(C)\right), \tag{29}$$

where $q(D,\gamma)$ is defined in (20).

The proof of Theorem 18 is similar to that of Theorem 15 and can be found in the supplementary material.

Corollary 19: Consider the setting of Theorem 18, and further assume $B^2 \preceq I-C$. Then, the same conclusions as in Corollary 16 hold for Algorithm (4).

Remark: We point out that linear convergence of Algorithm (4) can be established requiring that only F is strongly convex (rather than all f_i's). The proof of this result can be found in the supplementary material. However, differently from (29), the proved convergence rate does show a coupling between optimization and network parameters. This is consistent with existing results in the literature.

C. Discussion

1) Unified Convergence Conditions: Theorems 15 and 18 offer a unified platform for the analysis and design of a gamut of linearly convergent algorithms–all the schemes, new and old, that can be written in the form (22) and (4) satisfying Assumption

14 and 17, respectively–e.g., all the algorithms listed in Table I. In particular, our convergence results embrace *both* ATC and CTA algorithms, solving either smooth ($G = 0$) or *composite* ($G \neq 0$) optimization problems. This improves the results in [17] and [27].

2) On the Rate Expression: We comment the expression of the rate focusing on Theorem 18 and Corollary 19 ($G \neq 0$); same conclusions can be drawn for Algorithm (22) (Theorem 15 and Corollary 16). Theorem 18 provides the explicit expression of the linear rate provably achievable by Algorithm (4), for a given choice of the weight matrices A, B and C and stepsize γ (satisfying Assumption 17). In general, this rate depends on both optimization parameters (L and μ) and network-related quantities (A, B and C); furthermore, feasible stepsize values and network parameters are coupled by Assumption 17v). **CTA-based schemes:** This is consistent with existing convergence results of CTA-based algorithms (known only for $G = 0$), which are special cases of Algorithm (22). For instance, consider EXTRA [3] and DIGing [10] (corresponding to Algorithm (22) with $B = I$, cf. Table I): γ, C and D are coupled via the condition $q(D, \gamma) \prec (I - C)$, instrumental to achieve linear rate. **ATC-based schemes:** For algorithms in the ATC form, i.e., $A = B$, less restrictive conditions are required. For instance, when Assumption 17 v) is satisfied by $B^2 \prec I - C$–a condition that is met by several algorithms in Table I–the stepsize can be chosen in the larger region $[0, \gamma^\star(D)]$, resulting in the smaller rate $\max(q(D, \gamma), 1 - \lambda_2(C)) \geq \max(q^\star(D), 1 - \lambda_2(C))$ (recall that, in such a case, $\lambda_{\max}(B^2(I - C)^{-1}) = 1$), where the lower bound is achieved when $\gamma = \gamma^\star(D)$ (cf. Corollary 19).

On the other hand, when the algorithm parameters can be freely designed, Corollary 16 offers the "optimal" choice, resulting in the smallest contraction factor, as in (28). This instance enjoys two desirable properties, namely:

i) Network-independent stepsize: The stepsize $\gamma^\star$ in Corollary 16 does not depend on the network parameters but only on the optimization and its value coincides with the optimal stepsize of the centralized proximal-gradient algorithm. This is a major advantage over current distributed schemes applicable to (P) (but with $G \neq 0$) and complements the results in [11], whose algorithm however cannot deal with the non-smooth term G and use more stringent stepsize.

ii) Rate-separation: The rate (28) is determined by the worst rate between the one due to the communication $1 - \lambda_2(C)$ and that of the optimization $((\kappa - 1)/(\kappa + 1))^2$. This separation is the key enabler for our distributed scheme to achieve the convergence rate of the centralized proximal gradient algorithm-we elaborate on this property next.

3) Balancing Computation and Communications: Note that $\rho_{\texttt{opt}} \triangleq (\kappa - 1)/(\kappa + 1)$ is the rate of the centralized proximal-gradient algorithm applied to (P), under Assumption 1. This means that if the network is "sufficiently connected," specifically $1 - \lambda_2(C) \leq \rho_{\texttt{opt}}^2$, the proposed algorithm converges at the *desired* linear rate $\rho_{\texttt{opt}}$. On the other hand, when $1 - \lambda_2(C) > \rho_{\texttt{opt}}^2$, one can still achieve the centralized rate $\rho_{\texttt{opt}}$ by enabling multiple (finite) rounds of communications per proximal gradient evaluations. Two strategies are: 1) performing multiple rounds of consensus using each time the same weight matrix; or 2) employing acceleration via Chebyshev polynomials. **1) Multiple rounds of consensus:** Given a weight matrix $W \in \mathcal{W}_{\mathcal{G}}$, as concrete example, consider the case $W \in \mathbb{S}_{++}^m$ and $A = B = I - C = W^K$, with $K \geq 1$, which implies $B^2 \preceq I - C$ (cf. Corollary 16). The resulting algorithm will require K rounds of communications (each of them using W) per gradient evaluation. Denote $\rho_{\texttt{com}} \triangleq \lambda_{\max}(W - J)$; we have $1 - \lambda_2(C) = \lambda_{\max}(W^K - J) = \rho_{\texttt{com}}^K$. The value of K is chosen to minimize the resulting rate λ [cf. (28)], i.e., such that $\rho_{\texttt{com}}^K \leq \rho_{\texttt{opt}}^2$, which leads to $K = \lceil \log_{\rho_{\texttt{com}}}(\rho_{\texttt{opt}}^2) \rceil$.

2) Chebyshev acceleration: To further reduce the communication cost, we can leverage Chebyshev acceleration [30]. As specific example, consider the case $W \in \mathbb{S}^m$ is invertible; we set $A = P_K(W)$ and $P_K(1) = 1$ (the latter is to ensure the double stochasticity of A), with $P_K \in \mathbb{P}_K$, where $\mathbb{P}_K$ denotes the set of polynomials with degree less than or equal than K. This leads to $1 - \lambda_2(C) = \lambda_{\max}(A^2 - J)$. The idea of Chebyshev acceleration is to find the "optimal" polynomial P_K such that $\lambda_{\max}(A^2 - J)$ is minimized, i.e., $\rho_C \triangleq \min_{P_K \in \mathbb{P}_K, P_K(1)=1} \max_{t \in [-\rho_{\texttt{com}}, \rho_{\texttt{com}}]} |P_K(t)|$. The optimal solution of this problem is $P_K(x) = T_K(\frac{x}{\rho_{\texttt{com}}})/T_K(\frac{1}{\rho_{\texttt{com}}})$ [30, Theorem 6.2], with $\alpha' = -\rho_{\texttt{com}}, \beta' = \rho_{\texttt{com}}, \gamma' = 1$ (which are certain parameters therein), where T_K is the K-order Chebyshev polynomials that can be computed in a distributed manner via the following iterates [13], [30]: $T_{k+1}(\xi) = 2\xi\, T_k(\xi) - T_{k-1}(\xi)$, $k \geq 1$, with $T_0(\xi) = 1, T_1(\xi) = \xi$. Also, invoking [30, Corollary 6.3], we have $\rho_C = \frac{2c^K}{1+c^{2\,K}}$, where $c = \frac{\sqrt{\vartheta}-1}{\sqrt{\vartheta}+1}, \vartheta = \frac{1+\rho_{\texttt{com}}}{1-\rho_{\texttt{com}}}$. Thus, the minimum value of K that leads to $\rho_C \leq \rho_{\texttt{opt}}^2$ can be obtained as $K = \left\lceil \log_c(1/\rho_{\texttt{opt}}^2 + \sqrt{1/\rho_{\texttt{opt}}^4 - 1}) \right\rceil$. Note that to be used, A must be returned as nonsingular. More details of Chebyshev acceleration applied to the ABC-Algorithm along with some numerical results can be found in [1]

4) Improvement Upon Existing Results and Tuning Recommendations: Theorems 15 and 18 improve upon existing convergence conditions and rate bounds (when restricted to our setting, cf. Assumptions 1 and 4). A comparison with notable distributed algorithms in the literature is presented in Table I. Since all the schemes therein are special cases of Algorithm (22) [with the exception of [17] that is an instance of Algorithm (4)] (cf. Table II) and satisfy Assumption 14 (or Assumption 17), one can readily apply Theorem 15 (or Theorem 18) and determine, for each of them, a new stepsize range and achievable rate: the column "Stepsize/this paper (optimal, Corollary 16)" reports the stepsize value $\gamma^\star(D)$ for the different algorithms (i.e., given B, C and D) while the column "Rate/δ this paper" shows the resulting provably rate, as given in (27). A direct comparison with the columns "Stepsize/literature (upper bound)" and "Rate/δ, literature" respectively, shows that our theorems provide strictly larger ranges for the stepsize of EXTRA [3] NEXT [6]/AugDGM [4], [31] and Exact Diffusion [12], and faster linear rates for *all* the algorithms in the table.

Table I also serves as comparison of the convergence rates *provably achievable* by the different algorithms. For instance, we notice that, although EXTRA and NIDS both require one

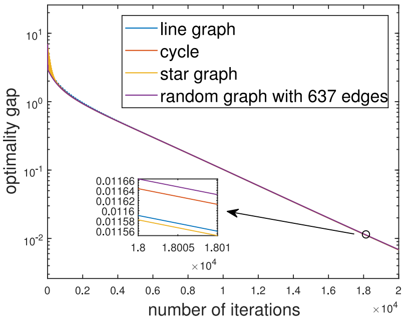


Fig. 1. Instance of the ABC algorithm on problems of the same ill-conditioned optimization data, but over different graph topologies.

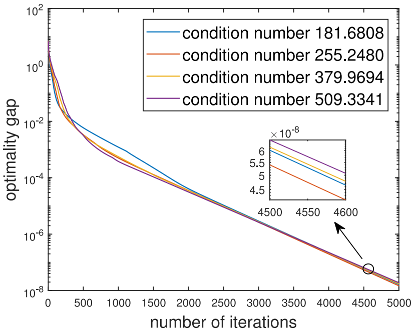


Fig. 2. Instance of the ABC algorithm on problems over the same line graph, but with optimization data of different condition numbers.

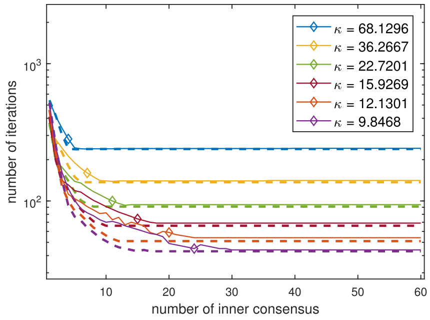


Fig. 3. Elastic net problem: Number of iterations (gradient evaluations) needed to reach an accuracy of 10^{-8} by Algorithm 4 employing Chebyshev acceleration (dashed lines) and multiple rounds of consensus (solid lines).

communication per gradient evaluation, NIDS is provably faster, achieving a linear rate of $\delta^{\star}\log(1/\epsilon)$, with $\delta^{\star}$ defined in (28), versus the linear rate $(\kappa/(1-\rho))\log(1/\epsilon)$ of EXTRA. In Section VII-A we show that the ranking based on our theoretical findings in Table I is reflected by our numerical experiments–see Fig. 3. For the sake of fairness, we remark one more time that, the stepsize and rate expressions of some of the algorithms listed in Table I were obtained under weaker conditions on F and W than Assumptions 1 and 4.

5) Generalizing Existing Algorithms to the Case $G \neq 0$: All the algorithms listed in Table I but [6] and [17] are designed for Problem (P) with $G = 0$. Since they are special cases of our general framework and Algorithm (4) can deal with the case $G \neq 0$, they inherit the same feature. Their "proximal" extension is given by (6), with the matrices A, B and C as in original algorithm (cf. Table II). Theorem 18 and Corollary 19 show that these new algorithms enjoy the same convergence rates of their "no-proximal" counterpart. For instance, consider AugDGM, corresponding to Algorithm (22) with $A = B = W^2$, $D = I$, $C = (I - W)^2$; it clearly satisfies Assumption 17 for $W \succ 0$. Its extension to the general optimization with $G \neq 0$ comes readily substituting these choices of A, B, C into (6) (or Algorithm 22), yielding

$$\begin{aligned} X^{k+1} &= \mathrm{prox}_{\gamma g}\left(Z^{k+1}\right), \\ Z^{k+2} &= (2W - W^2)Z^{k+1} + W^2(X^{k+1} - X^k) \\ &\quad - \gamma\, W^2(\nabla f(X^{k+1}) - \nabla f(X^k)). \end{aligned}$$

As second example, consider the primal-dual scheme such as NIDS and Exact Diffusion; they correspond to Algorithm (22) with $A = B = \frac{I+W}{2}, C = \frac{I-W}{2}$. Similarly, we can introduce their "proximal" version as follows:

$$\begin{aligned} X^k &= \mathrm{prox}_{\gamma g}\left(Z^k\right), \\ Z^{k+2} &= \frac{I+W}{2}\left(Z^{k+1} + X^{k+1} - X^k\right. \\ &\quad \left. -\gamma(\nabla f(X^{k+1}) - \nabla f(X^k))\right). \end{aligned}$$

D. Application to Statistical Learning

We customize our rate results to the instance of (P) modeling statistical learning tasks over networks. This is an example where the local strong convexity and smoothness constants of the agent functions are different; still, we will show that, when the data sets across the agents are sufficiently similar, the rate achieved by the proposed algorithm is within a range of $\widetilde{\mathcal{O}}(1/\sqrt{n})$ of that of the centralized counterpart.

Suppose each agent i has access to n $i.i.d.$ samples $\{z_j\}_{j\in\mathcal{D}_i}$ following the distribution $\mathcal{P}$. The goal is to learn a model parameter x using the samples from all the agents; mathematically, we aim at solving the following empirical risk minimization problem: $\min_{x\in\mathbb{R}^d}\sum_{i\in[m]}\sum_{j\in\mathcal{D}_i}\ell(x;z_j)$, where $\ell(x;z_j)$ is the loss function measuring the fitness of the statistical model parameterized by x to sample z_j; we assume each $\ell(x;z_j)$ to be quadratic in x and satisfy $\widetilde{\mu}I \preceq \nabla^2\ell(x;z) \preceq \widetilde{L}I$, for all z. This problem is an instance of (P) with $f_i(x) \triangleq \sum_{j\in\mathcal{D}_i}\ell(x;z_j)$. Denote the largest and the smallest eigenvalues of $\nabla^2 f_i(x)$ (resp. $\nabla^2 F(x)$) as L_i and μ_i (resp. $\bar{L}$ and $\bar{\mu}$). Then, each $f_i(x)$ is $\mu \triangleq \min_{i\in[m]}\mu_i$-strongly convex and $L \triangleq \max_{i\in[m]} L_i$-smooth. Recalling $\kappa = L/\mu$, the rate in (28) reduces to $((\kappa-1)/(\kappa+1))^2$, when $1 - \lambda_2(C) \leq ((\kappa-1)/(\kappa+1))^2$ (possibly using multiple rounds of communications), resulting in $\mathcal{O}(\kappa\log(1/\epsilon))$ overall number of gradient evaluations. On the other hand, the complexity of the centralized gradient descent algorithm reads $\mathcal{O}(\frac{\bar{L}}{\bar{\mu}}\log(\frac{1}{\epsilon}))$. To compare these two quantities, compute

$$\begin{aligned} \left|\frac{L}{\mu} - \frac{\bar{L}}{\bar{\mu}}\right| &= \frac{\left|L\bar{\mu} - \bar{L}\mu\right|}{\mu\bar{\mu}} \leq \frac{\left|L - \bar{L}\right|\bar{\mu} + \bar{L}\left|\bar{\mu} - \mu\right|}{\widetilde{\mu}^2} \\ &\leq \frac{1}{\widetilde{\mu}^2}\left(\bar{\mu}\max_{i\in[m]}\left|L_i - \bar{L}\right| + \bar{L}\max_{i\in[m]}\left|\mu_i - \bar{\mu}\right|\right) \end{aligned}$$

$$\overset{(a)}{\leq} \frac{\bar{\mu}+\bar{L}}{\widetilde{\mu}^2}\sqrt{\frac{32\widetilde{L}^2\log(dm/\delta)}{n}}, \quad \text{with probability } 1-\delta$$
$$\leq 8\sqrt{2}\,\frac{\widetilde{L}^2}{\widetilde{\mu}^2}\sqrt{\frac{\log(dm/\delta)}{n}},$$

where in (a) we used [32, Corollary 6.3.8]

$$\max_{i\in[m]}\left(|\mu_i-\bar{\mu}|\,,\ \left|L_i-\bar{L}\right|\right) \leq \left\|\nabla^2 f_i(x)-\nabla^2 f(x)\right\|, \tag{30}$$

and [33, Lemma 2]

$$\max_{i\in[m]}\left\|\nabla^2 f_i(x)-\nabla^2 f(x)\right\| \leq \sqrt{\frac{32\widetilde{L}^2\log(dm/\delta)}{n}} \tag{31}$$

with probability at least $1-\delta$. Therefore, the complexity of our algorithm becomes $\mathcal{O}((\frac{\bar{L}}{\bar{\mu}}+\widetilde{\mathcal{O}}(\frac{\bar{L}^2}{\bar{\mu}^2}\frac{1}{\sqrt{n}}))\cdot\log(\frac{1}{\epsilon}))$, with $\widetilde{\mathcal{O}}$ hiding the factor $\log(dm/\delta)$. This shows that when agents have enough data locally (n is large), the above rate is of the same order of that of the centralized gradient descent algorithm.

VI. Sublinear Convergence (Convex Case)

We consider now Problem (P) when f_i's are assumed to be convex ($\mu=0$) but not strongly-convex. We study the sublinear convergence for two splitting schemes, namely: i) $T=T_CT_fT_B$ applied to (P) with $G=0$; and ii) $T=T_CT_gT_fT_B$ applied to (P) with $G\neq 0$.

A. Convergence Under $G=0$

We establish sublinear convergence of Algorithm (22) (corresponding to $T=T_CT_fT_B$) under the following assumption.

Assumption 20: The weight matrices $A\in\mathbb{R}^{m\times m}$, B, $C\in\mathbb{S}^m$ satisfy:

i) $A=BD$, with $B\succeq 0$, $D\in\mathbb{S}^m$ and $D\succ 0$;
ii) $D1_m=1_m$ and $1_m^\top B=1_m^\top$;
iii) $C\succeq 0$ and $\texttt{null}(C)=\texttt{span}(1_m)$;
iv) B and C commute;
v) $I-\frac{1}{2}C-\sqrt{B}D\sqrt{B}\succeq 0$ ($\Leftrightarrow I-\frac{1}{2}C-A\succeq 0$, if B commutes with D).

We quantify the progress of algorithms towards optimality in this setting using the following merit function:

$$M(X)\triangleq\max\left\{\|(I-J)X\|\,\|\nabla f(X^\star)\|\,,|f(X)-f(X^\star)|\right\},$$

where $J\triangleq\frac{1}{m}1_m1_m^\top$ and $X^\star\triangleq 1_m(x^\star)^\top$; the first term encodes consensus errors while the second term measures the optimality gap.

We begin by rewriting Algorithm (22) in an equivalent form given in Lemma 21, which does not have a mixing matrix multiplied to the gradient term.

Lemma 21: Suppose Assumption 8 holds. Then, Algorithm (22) can be rewritten as (with $\underline{Y}^0\triangleq 0$):

$$X^k=B\underline{X}^k, \tag{32a}$$
$$\underline{X}^{k+1}=DX^k-\gamma(\nabla f(X^k)+\underline{Y}^k), \tag{32b}$$
$$\underline{Y}^{k+1}=\underline{Y}^k+\frac{1}{\gamma}C\underline{X}^{k+1}. \tag{32c}$$

Proof: since $Y^0=0$, we know $\texttt{span}(X^1),\texttt{span}(Y^1)\subset\texttt{span}(B)$. It is easy then to deduce from induction that $\texttt{span}(X^k)$, $\texttt{span}(Y^k)\subset\texttt{span}(B)$, $\forall k$. Setting $Y^k=\gamma B\underline{Y}^k$ and $X^k=B\underline{X}^k$ leads to this equivalent form. ■

Define $\phi(X,Y)=f(X)+\langle Y,X\rangle$. In Lemma 22 and 23 below, we establish two fundamental inequalities on $\phi(X^k,Y)$ and $\phi(X,Y)$ for $X\in\texttt{span}(1_m)$ and $Y\in\texttt{span}(C)$, instrumental to prove the sublinear rate; the proofs are reported in Section B in Appendix.

Lemma 22: Consider the setting of Theorem 24, let $\{X^k,\underline{X}^k,\underline{Y}^k\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm (32) under Assumption 20. Then, it holds:

$$\begin{aligned}&\phi(X^{k+1},Y)\\&\leq\phi(X,Y)-\frac{1}{\gamma}\left\|\underline{X}^{k+1}\right\|^2_{B-BC-AB}\\&\quad-\frac{1}{\gamma}\left\langle X^{k+1}-X^k,X^{k+1}-X\right\rangle_D\\&\quad-\gamma\left\langle\underline{Y}^{k+1}-Y,\underline{Y}^{k+1}-\underline{Y}^k\right\rangle_{B'}+\frac{L}{2}\left\|X^{k+1}-X^k\right\|^2,\end{aligned} \tag{33}$$

for all $X\in\texttt{span}(1_m)$ and $Y\in\texttt{span}(C)$, where $B'=(C+bJ)^{-1}B$, $b\geq 2$.

Lemma 23: Under the same conditions as in Lemma 22, if $\gamma\leq\frac{\lambda_{\min}(D)}{L}$, then

$$\begin{aligned}&\phi(\widehat{X}^k,Y)-\phi(X,Y)\\&\leq\frac{1}{2\,k}\left(\frac{1}{\gamma}\left\|X^0-X\right\|^2_D+\gamma\frac{\rho(B-J)}{\lambda_2(C)}\|Y\|^2\right),\end{aligned} \tag{34}$$

for all $X\in\texttt{span}(1_m)$ and $Y\in\texttt{span}(C)$, where $\widehat{X}^k\triangleq\frac{1}{k}\sum_{t=1}^k X^t$.

We now prove the sublinear convergence rate.

Theorem 24 (Sublinear rate for $T_CT_fT_B$): Consider Problem (P) under Assumption 1 with $\mu=0$ and $G=0$; and let $x^\star$ be an optimal solution. Let $\{(X^k,Y^k)\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm (22) under Assumptions 20. Then, if $0<\gamma\leq\frac{\lambda_{\min}(D)}{L}$, we have

$$\begin{aligned}M(\widehat{X}^k)\leq\frac{1}{k}\Bigg(&\frac{1}{2\gamma}\left\|X^0-X^\star\right\|^2_D\\&+2\gamma\frac{\rho(B-J)}{\lambda_2(C)}\|\nabla f(X^\star)\|^2\Bigg),\end{aligned} \tag{35}$$

where $\widehat{X}^k=\frac{1}{k}\sum_{t=1}^k X^t$.

Proof: Setting $X=X^\star$ in (33), it holds

$$\phi(\widehat{X}^k,Y)-\phi(X^\star,Y)=f(\widehat{X}^k)-f(X^\star)-\left\langle\widehat{X}^k-X^\star,Y\right\rangle$$
$$=f(\widehat{X}^k)-f(X^\star)-\left\langle\widehat{X}^k,Y\right\rangle\leq h(\|Y\|),$$

for $Y\in\texttt{span}(C)$, where $h(\cdot)=\frac{1}{2\,k}(\frac{1}{\gamma}\|X^0-X^\star\|^2_D+\gamma\frac{\rho(B-J)}{\lambda_2(C)}(\cdot)^2)$. Setting $Y=-2\frac{(I-J)\widehat{X}^k}{\|(I-J)\widehat{X}^k\|}\|Y^\star\|$, with

$Y^\star = -\nabla f(X^\star)$, we have

$$f(\widehat{X}^k) - f(X^\star) + 2\left\|Y^\star\right\| \left\|(I-J)\widehat{X}^k\right\| \le h(2\left\|Y^\star\right\|).$$

By the convexity of f, $f(\widehat{X}^k) - f(X^\star) + \left\langle (I-J)\widehat{X}^k, Y^\star \right\rangle = f(\widehat{X}^k) - f(X^\star) + \left\langle \widehat{X}^k, Y^\star \right\rangle \ge 0$, we have $f(\widehat{X}^k) - f(X^\star) \ge -\|Y^\star\|\|(I-J)\widehat{X}^k\|$. Combining the above two relations, we have $M(\widehat{X}^k) \le h(2\|Y^\star\|)$. This completes the proof. ■

Finally, we leverage Young inequality to provide the choice of γ that optimizes the rate given in Theorem 24.

Corollary 25: Consider the setting of Theorem 24. The stepsize that minimizes the right hand side of (35) is

$$\gamma = \min\left(\frac{\lambda_{\min}(D)}{L}, \frac{1}{2}\sqrt{\frac{\lambda_2(C)}{\rho(B-J)}}\frac{\left\|X^0 - X^\star\right\|_D}{\|\nabla f(X^\star)\|}\right), \quad (36)$$

leading to a sublinear rate

$$M(\widehat{X}^k) \le \frac{1}{k}\max\left\{\frac{L\left\|X^0 - X^\star\right\|_D^2}{\lambda_{\min}(D)},\right. \\ \left. 2\sqrt{\frac{\rho(B-J)}{\lambda_2(C)}}\left\|X^0 - X^\star\right\|_D \left\|\nabla f(X^\star)\right\|\right\}. \quad (37)$$

Note that the stepsize in (36) depends on $\|X^0 - X^\star\|_D / \|\nabla f(X^\star)\|$, an information that is not generally available; we discuss this issue in Sec. VI-C.

B. Convergence Under $G \neq 0$

We consider now Problem (P) with $G \neq 0$ and $\mu = 0$. We study convergence of a variation of the general scheme (4), where the proximal operator is employed before T_f and $B = I$, yielding the operator decomposition $T_C T_g T_f$.[1] This scheme reads

$$\begin{aligned} \underline{X}^{k+1} &= DX^k - \gamma(\nabla f(X^k) + \underline{Y}^k), \\ X^{k+1} &= \mathrm{prox}_{\gamma g}\left(\underline{X}^{k+1}\right), \\ \underline{Y}^{k+1} &= \underline{Y}^k + \frac{1}{\gamma}CX^{k+1}, \end{aligned} \quad (38)$$

with $\underline{Y}^0 \triangleq 0$. Note that a key difference between (4) and the above algorithm is that the former uses $\underline{X}$ in the update of the dual variable Y, the variable before the operator $\mathrm{prox}_{\gamma g}(\cdot)$, while the latter uses the variable X, i.e., the variable after the operator $\mathrm{prox}_{\gamma g}(\cdot)$. It is not difficult to check that (38) subsumes many existing proximal-gradient methods, such as PG-EXTRA [18] or ID-FBBS [19] (with $D = W, C = I - W$). We present a unified result of the sublinear convergence for the algorithm (38), under the following assumption.

Assumption 26: The weight matrices $C,\, D \in \mathbb{S}^m$ satisfy:

i) $1_m^\top D 1_m = m$;

ii) $C \succeq 0$ and $\texttt{null}(C) = \texttt{span}(1_m)$;

iii) $0 \prec D \preceq I - \frac{C}{2}$.

Note that the above assumption is, indeed, a customization of Assumption 20. We study convergence of Algorithm (38) using the following merit function measuring the progresses of the algorithms from consensus and optimality. Define

$$M(X) \triangleq \max\left\{\|(I-J)X\|\left\|Y^\star\right\|,\, \left|(f+g)(X) \right.\right. \\ \left.\left. - (f+g)(X^\star)\right|\right\}.$$

where $Y^\star = -(\nabla f(X^\star) + 1_m(\xi^\star)^\top)$, for some $\xi^\star \in \partial G(x^\star)$ such that $\xi^\star + \nabla F(x^\star) = \xi^\star + \frac{1}{m}\sum_{i=1}^m \nabla f_i(x^\star) = 0$. Note that, since $1_m^\top Y^\star = 0$, we have $Y^\star \in \texttt{span}(C)$.

We are now ready to state our convergence result, whose proof is left to the supplementary material due to its similarity to that of Theorem 24.

Theorem 27 (Sublinear rate for $T = T_C T_g T_f$): Consider Problem (P) under Assumption 1 with $\mu = 0$; and let $x^\star$ be an optimal solution. Let $\{(X^k, Y^k)\}_{k\ge 0}$ be the sequence generated by Algorithm (38) under Assumptions 26. Then, if $\gamma < \frac{\lambda_{\min}(D)}{L}$, we have

$$M(\widehat{X}^k) \le \frac{1}{k}\left(\frac{1}{2\gamma}\left\|X^0 - X^\star\right\|_D^2 + 2\gamma\frac{1}{\lambda_2(C)}\left\|\nabla f(X^\star)\right\|^2\right), \quad (39)$$

where $\widehat{X}^k = \frac{1}{k}\sum_{t=1}^k X^t$.

Corollary 28: Consider the setting of Theorem 27. The stepsize that minimizes the right hand side of (39) is

$$\gamma = \min\left(\frac{\lambda_{\min}(D)}{L}, \frac{1}{2}\sqrt{\lambda_2(C)}\frac{\left\|X^0 - X^\star\right\|_D}{\|\nabla f(X^\star)\|}\right), \quad (40)$$

leading to a sublinear rate

$$M(\widehat{X}^k) \le \frac{1}{k}\max\left\{\frac{L\left\|X^0 - X^\star\right\|_D^2}{\lambda_{\min}(D)},\right. \\ \left. 2\sqrt{\frac{1}{\lambda_2(C)}}\left\|X^0 - X^\star\right\|_D \left\|\nabla f(X^\star)\right\|\right\}. \quad (41)$$

C. Discussion

1) On Rate Seperation: Differently from most of the existing works, such as [3], [7], [20], the above convergence results (Corollary 25 and 28) establish the explicit dependency of the rate on the network parameter as well as the properties of the cost functions. Specifically, the rate coefficients in (37) and (41) show an explicit dependence on the network and optimization parameters, with the first term on the RHS corresponding to the rate of the centralized optimization algorithm while the second term related to both the communication network and the heterogeneity of the cost functions of the agents (i.e., $\|\nabla f(x^\star)\|$). The smaller $\|\nabla f(x^\star)\|$, the more similar the objective functions agents have. For instance, when f_i's share a common minimizer, i.e., $\|\nabla f(x^\star)\| = 0$, the rate will reduce to the centralized one. The term $\sqrt{\frac{\rho(B-J)}{\lambda_2(C)}}$ accounts for the network effect on the rate. For instance, set $C = I - B$, so that $\lambda_2(C) = 1 - \rho(B - J)$. If $\rho(B - J) \to 0$ (meaning a network tending to a fully connected

[1] It is not difficult to check that any fixed point of $T_C T_g T_f$ has the same fixed-points of the operator in (14)

graph), $\sqrt{\frac{\rho(B-J)}{\lambda_2(C)}} \to 0$, leading to the rate of the centralized gradient algorithm [cf. (37)]. On the other hand, if $\rho(B-J) \to 1$ (poorly connected network), $\sqrt{\frac{\rho(B-J)}{\lambda_2(C)}} \to +\infty$, deteriorating the overall rate. As a result, when the agents have similar cost functions (i.e., small value of $\|\nabla f(x^\star)\|$) or the network is well connected, the first term will dominate the second, leading to the centralized performance. The impact of the heterogeneity quantity $\|\nabla f(x^\star)\|$ on the convergence behavior is validated by our numerical results–see Section VII-B1.

2) On the Choice of Stepsize: The optimal stepsize, as indicated in (36) (resp. (40)), is such that the two terms in (35) (resp. (39)) are balanced. Albeit (36) and (40) generally are not implementable, due to the unknown quantity $\|X^0 - X^\star\|_D / \|\nabla f(X^\star)\|$, the result is interesting on the theoretical side, showing that the "optimal" stepsize is not necessarily $1/L$ but depends on the the network and the degree of heterogeneity of the cost functions as well. In particular, the optimal choice is $1/L$ when the network is well connected and agents share similar "interests," i.e., $\|\nabla f(x^\star)\|$ is small. On the other hand, as the connectivity of the network becomes worse and/or the heterogeneity of local cost functions becomes larger, stepsize values smaller than $1/L$ ensure better performance. This observation provides recommendations on stepsize tuning and it is validated by our numerical experiments as well.

VII. Numerical Results

We report some numerical results on strongly convex and convex instances of (P), supporting our theoretical findings. The obtained stepsize bounds and rates are shown to predict well the practical behavior of the algorithms. For instance, the ATC-based schemes exhibit a clear rate separation [as predicted by (28)]: the convergence rate cannot be continuously improved by unilaterally decreasing the condition number of the f_i's or increasing the connectivity of the network.

A. Strongly Convex Problems

We consider a regularized least squares problem over an undirected graph consisting of 50 nodes, generated through the Erdos-Renyi model with activating probability of 0.05 for each edge. The problem reads

$$\min_{x\in\mathbb{R}^d} \left(\frac{1}{50}\sum_{i=1}^{50} \|U_i x - v_i\|^2\right) + \rho\,\|x\|_2^2 + \lambda\,\|x\|_1 . \qquad (42)$$

where $U_i \in \mathbb{R}^{r\times d}$ and $v_i \in \mathbb{R}^{r\times 1}$ are the feature vector and labels, respectively, only accessible by node i. For brevity, we denote $U = [U_1; U_2; \cdots; U_{50}] \in \mathbb{R}^{50r\times d}$ and $v = [v_1; v_2; \cdots; v_{50}] \in \mathbb{R}^{50r\times 1}$ and use $M_{:,i}$ (resp. $M_{i,:}$) to denote the i-th column (resp. row) of a matrix M. In the simulation, we set $r = 20$, $d = 40$, $\rho = 20$ and $\lambda = 1$. We generate the matrix U of the feature vectors according to the following procedure, proposed in [34]: we first generate an innovation matrix Z with each entry i.i.d. drawn from $\mathcal{N}(0,1)$. Using a control parameter $\omega \in [0,1)$, we then generate columns of U such that the first column is $U_{:,1} = Z_{:,1}/\sqrt{1-\omega^2}$ and the rest are recursively set as $U_{:,i} = \omega U_{:,i-1} + Z_{:,i}$, for $i = 2,\ldots,d$. As a result, each row $U_{i,:} \in \mathbb{R}^d$ is a Gaussian random vector and its covariance matrix $\Sigma = \text{cov}(U_{:,i})$ is the identity matrix if $\omega = 0$ and becomes extremely ill-conditioned as $\omega \to 1$. Finally, we generate $x_0 \in \mathbb{R}^d$ with sparsity level 0.3 and each nonzero entry i.i.d. drawn from $\mathcal{N}(0,1)$, and set $v = Ux_0 + \xi$, where each component of the noise ξ is i.i.d. drawn from $\mathcal{N}(0, 0.04)$. By changing ω one can control the conditional number κ of the smooth objective in (42).

1) Validating the Rate Separation: We validate here the rate results predicted by Corollary 16 and 19. We consider Algorithm (4), with $A = B = \frac{I+W}{2}$ and $C = I - B$, and run two experiments. **1)** We simulated problem (42), with $\rho = 10$ and $\omega = 0.999$–this leads to an extremely large condition number, $((\kappa-1)/(\kappa+1))^2 \approx 0.9999$)–and run the algorithm over different graphs, namely: a line, a cycle, a star, and a random graph with 637 edges, with $1-\lambda_2(C)$ being 0.9993, 0.9974, 0.9900 and 0.6948 respectively; Fig. 1 plots the optimality gap $\frac{1}{\sqrt{50}}\|X^k - 1_m x^{\star\top}\|$ versus the number of iterations, achieved over the different graph topologies. **2)** On the other extreme, in the second experiment, we considered a poorly connected line graph with $1-\lambda_2(C) \approx 0.9993$ and run the algorithm for different instances of the optimization problem–specifically, $\rho = 5$ and $\omega = \{0.75, 0.8, 0.85, 0.88\}$–resulting in $((\kappa-1)/(\kappa+1))^2$ being 0.9782, 0.9845, 0.9895 and 0.9922 respectively; Fig. 2 plots the optimality gap (defined as in Fig. 1) versus the number of iterations, achieved for the different optimization problems. These experiments clearly support the rate separation predicted by our theory: the rate is determined by the bottleneck between the network and optimization. Fig. 1: For ill-conditioned problems–meaning $((\kappa-1)/(\kappa+1))^2 > 1-\lambda_2(C)$–the algorithm exhibits almost identical rates, irrespectively of the specific graph instances. On the other hand, Fig. 2 shows that, on poorly connected networks, the convergence rate of the algorithm is not affected by the condition number of the optimization problem, as long as $((\kappa-1)/(\kappa+1))^2 < 1-\lambda_2(C)$.

2) More on the Rate Separation (28): We simulated the following instances of Algorithm 4. We set $A = B = (\frac{I+W}{2})^K$ and $C = I - B$, where W is a weight matrix generated using the Metropolis-Hastings rule [35], and $K \geq 1$ is the number of inner consensus steps. When Chebyshev acceleration is employed in the inner consensus steps, we instead used $A = B = (I + P_K(\widetilde{W}))/2$ and $C = I - B$ (condition of Corollary 19 is satisfied). In Fig. 3, we plot the number of iterations (gradient evaluations) needed by the algorithm to reach an accuracy of 10^{-8}, versus the number of inner consensus K, for different values of κ; solid (resp. dashed) line-curves refer to non-accelerated (Chebyshev) consensus steps. The markers (diamond symbol) correspond to the number of iterations predicted by (28) for the max in (28) to achieve the minimum value, that is, $\left\lceil 2\log(\frac{\kappa-1}{\kappa+1})/\log(\frac{1+\lambda_{m-1}(W)}{2})\right\rceil$. The following comments are in order. **(i)** As K increases, the number of iterations needed to reach the desired solution accuracy decreases till it reaches a plateau; further communication rounds do not improve the performance, as the optimization component becomes the bottleneck [as predicted by (28)]. **(ii)** Less number of iterations are

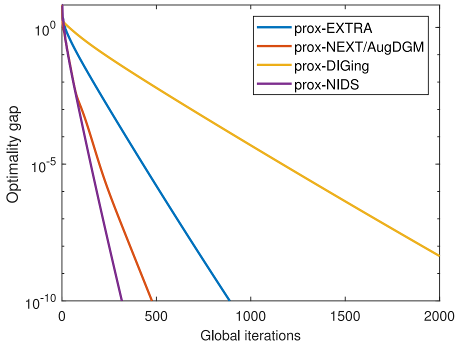


Fig. 4. Performance comparison of the proximal extensions of some existing algorithms–these extension schemes are all new and are instances of (4).

needed when κ becomes smaller (simpler problem). Finally, **(iii)** Chebyshev acceleration further reduces the number of iterations. These were all predicted by our theoretical findings.

3) Validating Table I. Comparison of the "prox"-Versions of Existing Algorithms: In Fig. 4 we compare the "prox" version of several existing algorithms, applied to (42): we plot the optimality gap $\|X^k - 1_m x^{\star\top}\|$ versus the overall number of iterations (gradient evaluations). The setting is the same as in the previous example, except that now we set $\omega = 0.8$. The stepsize of each algorithm is chosen according to (19). The network is the Erdos-Renyi model with connection probability of 0.25; in this setting, the max in (28) is achieved at $(\kappa-1)/(\kappa+1)$. It follows from the figure that ATC-based schemes, such as Prox-NEXT/AugDGM, Prox-NIDS, outperform non-ATC ones, such as Prox-EXTRA and Prox-DIGing, validating the ranking established in (the last column of) Table I.

B. Non-Strongly-Convex Problems

To illustrate the results for non-strongly convex problems, we report here a logistic regression problem using the Ionosphere Data Set as follows [36]:

$$\min_{x\in\mathbb{R}^{34}} \frac{1}{50}\sum_{i=1}^{50}\sum_{k=7(i-1)+1}^{7i} \log(1+\exp(-v_k u_k^\top x)),$$

where $u_k \in \mathbb{R}^{34}$ and $v_k \in \{-1,1\}$ are respectively the feature vector and label of the k-th sample. We use $U = [u_1, u_2, \ldots, u_{350}]^\top$ to denote the feature matrix. We construct several problems with different Lipschitz constant by multiplying the feature matrix U with different scaling factors. In particular, given the original problem with an L-smooth objective function f, one can multiply U by a scalar $0 < \alpha < 1$ to construct a new α^2 L-smooth objective function $f_\alpha(\cdot)$. In the simulation, we consider the polynomial method and thus set $A = B = \widetilde{W}^K$ and $C = I - B$. The stepsize of the algorithm is chosen[2] according to (36). Figure 5 plots the number of iterations (gradient evaluations) needed by the algorithm to reach an accuracy of 10^{-4} in solving different problems with different difficulty versus the number of inner loop of consensus.

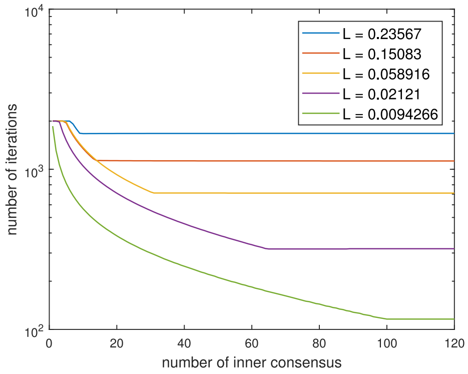


Fig. 5. Logistic regression problem: Number of iterations (gradient evaluations) needed to reach an accuracy of 10^{-4} by Algorithm 22 (equivalently Algorithm 32) employing multiple rounds of consensus.

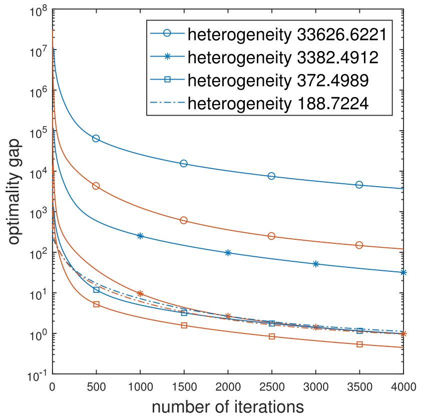


Fig. 6. Convergence behavior of the ABC algorithm and the centralized gradient descent for problems with different level of heterogeneity (measured by $\|\nabla f(X^\star)\|$). The blue curves are associated with the ABC algorithm while the red ones with the centralized gradient descent.

It follows from the figure that, similar as with the strongly convex case, the number of iterations needed is decreasing with the number of inner loops of consensus, until it reaches to a turning point which appears later as the Lipschitz constant L decreases. This observation verifies the result as shown in (37) where the two quantities is to be properly balanced with multiple communication steps.

1) On the Heterogeneity of f_i's: We exemplify the role of the heterogeneity measure $\|\nabla f(X^\star)\|$ on the convergence rate, stated in Corollary 25 and Corollary 28. We consider the distributed least squares problem (42), with $\rho = \lambda = 0$. Each row of U is now drawn i.i.d. from a multivariate normal distribution $\mathcal{N}(\mu, \alpha\Sigma)$. Each element of the mean vector μ is generated i.i.d. from $Unif(0,1)$ and $\Sigma \triangleq BB^\top$ with each entry of B generated i.i.d. from the standard normal. We then generate v as $v = U * \hat{x} + \xi$, wherein each element of $\hat{x}$ and ξ is drawn i.i.d. from the standard Normal. The local observation matrices U_i's become more similar to each other (thus $\|\nabla f(X^\star)\|$ becoming smaller), when we decrease the positive scalar α. We generate a graph via the Erdos-Renyi model with a connection probability 0.05 and a conforming weight matrix W. We set

[2] This choice is not implementable in practice but only for illustration.

$A = B = \frac{I+W}{2}$ and $C = \frac{I-W}{2}$, and compare the convergence of the ABC with that of the centralized gradient descent algorithm, for $\alpha = \{100, 1, 10^{-2}, 10^{-3}\}$ respectively. Note that all the above generated problems are ill-conditioned. For fair comparison, we rescale the metric $M(X)$ by $1/50$ for the ABC and use the metric $F(x) - F^\star$ for the centralized algorithm. As shown in Fig. 6, when the agents' cost functions become more similar (i.e. $\|\nabla f(X^\star)\|$ becomes smaller), the performance of the ABC algorithm (the red lines) become closer to its centralized counterpart, as predicted by Corollary 25 and 28.

VIII. Conclusion

We proposed a unified distributed algorithmic framework for composite optimization problems over networks; the framework subsumes many existing schemes. When the agents' functions are strongly convex, linear convergence is proved leveraging an operator contraction-based analysis. With a proper choice of the design parameters, the rate dependency on the network and cost functions can be decoupled, which permits to achieve the rate of the centralized (proximal)-gradient method (applied in the same setting) using a finite number of communications per gradient evaluations. Our convergence conditions and rate bounds improve on existing ones. When the functions of the agents are (not strongly) convex, a sublinear convergence rate was established, shedding light on the dependency of the convergence on the connectivity of the network and the heterogeneity of the cost functions.

Appendix

A. Proof of Lemma 11

Since $0 \prec D \preceq I$, we have

$$\begin{aligned}
&\|DX - \gamma\nabla f(X) - DY + \gamma\nabla f(Y)\|^2 \\
&\leq \|DX - \gamma\nabla f(X) - DY + \gamma\nabla f(Y)\|_{D^{-1}}^2 \\
&= \|X - Y\|_D^2 - 2\gamma\left\langle X - Y, \nabla f(X) - \nabla f(Y)\right\rangle \\
&\quad + \gamma^2 \|\nabla f(X) - \nabla f(Y)\|_{D^{-1}}^2 .
\end{aligned} \tag{43}$$

Then we proceed to lower bound $\langle X - Y, \nabla f(X) - \nabla f(Y)\rangle$. Let $X' = \sqrt{D}X$, $\tilde{f}(X) = f(\sqrt{D^{-1}}X)$. Given any two points $X, Y \in \mathbb{R}^{m\times d}$, we have

$$\begin{aligned}
&\langle X - Y, \nabla f(X) - \nabla f(Y)\rangle \\
&= \left\langle \sqrt{D^{-1}}X' - \sqrt{D^{-1}}Y', \nabla f(\sqrt{D^{-1}}X') - \nabla f(\sqrt{D^{-1}}Y')\right\rangle \\
&= \left\langle X' - Y', \nabla\tilde{f}(X') - \nabla\tilde{f}(Y')\right\rangle \\
&\overset{(*)}{\geq} \frac{L'\mu'}{L' + \mu'}\|X' - Y'\|^2 + \frac{1}{L' + \mu'}\left\|\nabla\tilde{f}(X') - \nabla\tilde{f}(Y')\right\|^2 \\
&= \frac{L'\mu'}{L' + \mu'}\|X - Y\|_D^2 + \frac{1}{L' + \mu'}\|\nabla f(X) - \nabla f(Y)\|_{D^{-1}}^2
\end{aligned}$$

where $(*)$ is due to [37, Theorem 2.1.12], with $L' = \frac{L}{\lambda_{\min}(D)}$ and $\mu' = \frac{\mu}{\lambda_{\max}(D)}$. Thus, knowing that $0 < \gamma \leq \frac{2\lambda_{\min}(D)}{L + \mu\cdot\eta(D)} = \frac{2}{L' + \mu'}$ and continuing from (43), we have

$$\begin{aligned}
&\|DX - \gamma\nabla f(X) - DY + \gamma\nabla f(Y)\|^2 \\
&\leq \left(1 - 2\gamma\frac{L'\mu'}{L' + \mu'}\right)\|X - Y\|_D^2 \\
&\quad - \left(\frac{2\gamma}{L' + \mu'} - \gamma^2\right)\|\nabla f(X) - \nabla f(Y)\|_{D^{-1}}^2 \\
&\leq \left(1 - 2\gamma\frac{L'\mu'}{L' + \mu'}\right)\|X - Y\|_D^2 .
\end{aligned}$$

In particular, if we set $\gamma = \gamma^\star$, we have $1 - 2\gamma^\star\frac{L'\mu'}{L'+\mu'} = (\frac{L'-\mu'}{L'+\mu'})^2 = (\frac{\kappa-\eta(D)}{\kappa+\eta(D)})^2$.

B. Proof of Lemma 22

Since f is L-smooth, we have

$$\begin{aligned}
&f(X^{k+1}) \\
&\leq f(X^k) + \left\langle \nabla f(X^k), X^{k+1} - X^k\right\rangle + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2 \\
&\overset{(a)}{\leq} f(X) + \left\langle \nabla f(X^k), X^k - X\right\rangle + \left\langle \nabla f(X^k), X^{k+1} - X^k\right\rangle \\
&\quad + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2 \\
&= f(X) + \left\langle \nabla f(X^k), X^{k+1} - X\right\rangle + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2 .
\end{aligned} \tag{44}$$

where (a) is due to the fact that $f(X) \geq f(X^k) + \left\langle \nabla f(X^k), X - X^k\right\rangle$ from the convexity of f.

Then, we relate the gradient term $\nabla f(X^k)$ to other quantities using (32b) as follows

$$\begin{aligned}
&\left\langle \nabla f(X^k), X^{k+1} - X\right\rangle = -\frac{1}{\gamma}\left\langle \underline{X}^{k+1}, X^{k+1} - X\right\rangle \\
&\quad + \frac{1}{\gamma}\left\langle DX^k - \gamma\underline{Y}^k, X^{k+1} - X\right\rangle \\
&= -\frac{1}{\gamma}\left\langle (I - C)\underline{X}^{k+1}, X^{k+1} - X\right\rangle \\
&\quad + \frac{1}{\gamma}\left\langle DX^k - \gamma\underline{Y}^{k+1}, X^{k+1} - X\right\rangle ,
\end{aligned}$$

where we have used (32c) to obtain the last relation. Now, substituting the above relation into (44), we further have

$$\begin{aligned}
f(X^{k+1}) &\leq f(X) - \frac{1}{\gamma}\left\langle (I - C)\underline{X}^{k+1}, X^{k+1} - X\right\rangle \\
&\quad + \frac{1}{\gamma}\left\langle DX^k, X^{k+1} - X\right\rangle - \left\langle \underline{Y}^{k+1}, X^{k+1} - X\right\rangle \\
&\quad + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2
\end{aligned} \tag{45}$$

Adding $\left\langle Y, X^{k+1} - X\right\rangle$, with $X \in \mathtt{span}(1_m)$ and $Y \in \mathtt{span}(C)$, to both sides of the above equation and noticing $(C + bJ)^{-1}C = I - J$ yields

$$\phi(X^{k+1}, Y) \leq \phi(X, Y) - \frac{1}{\gamma}\left\langle (I - C)\underline{X}^{k+1}, B(\underline{X}^{k+1} - X)\right\rangle$$

$$
\begin{aligned}
&+ \frac{1}{\gamma}\left\langle DB\underline{X}^k, B(\underline{X}^{k+1}-X)\right\rangle + \frac{L}{2}\left\|X^{k+1}-X^k\right\|^2 \\
&- \left\langle (C+2\,J)^{-1}C(\underline{Y}^{k+1}-Y), B(\underline{X}^{k+1}-X)\right\rangle \\
&= \phi(X,Y) - \frac{1}{\gamma}\left\langle (I-C)\underline{X}^{k+1}, B(\underline{X}^{k+1}-X)\right\rangle \\
&+ \frac{1}{\gamma}\left\langle DB\underline{X}^k, B(\underline{X}^{k+1}-X)\right\rangle + \frac{L}{2}\left\|X^{k+1}-X^k\right\|^2 \\
&- \left\langle \underline{Y}^{k+1}-Y, C\underline{X}^{k+1}\right\rangle_{B'} \\
&= \phi(X,Y) - \frac{1}{\gamma}\left\langle (I-C-DB)\underline{X}^{k+1}, B(\underline{X}^{k+1}-X)\right\rangle \\
&+ \frac{1}{\gamma}\left\langle DB(\underline{X}^k-\underline{X}^{k+1}), B(\underline{X}^{k+1}-X)\right\rangle \\
&- \gamma\left\langle \underline{Y}^{k+1}-Y, \underline{Y}^{k+1}-\underline{Y}^k\right\rangle_{B'} + \frac{L}{2}\left\|X^{k+1}-X^k\right\|^2,
\end{aligned}
$$

where we have used (32c) to obtain the last relation. Knowing that $X = B\underline{X}$ from (32a), we complete the proof.

C. Proof of Lemma 23

Invoking Lemma 22 and using the identity

$$
2\langle a-b, a-c\rangle = \|a-b\|^2 - \|b-c\|^2 + \|a-c\|^2,
$$

we have that

$$
\begin{aligned}
&\phi(X^{k+1},Y) \\
&\le \phi(X,Y) - \frac{1}{2\gamma}\left(\left\|X^{k+1}-X\right\|_D^2 - \left\|X^k-X\right\|_D^2\right) \\
&\quad - \frac{1}{\gamma}\left\|\underline{X}^{k+1}\right\|_{B-BC-AB}^2 - \left\|X^{k+1}-X^k\right\|_{\frac{1}{2\gamma}D-\frac{L}{2}I}^2 \\
&\quad - \frac{\gamma}{2}(\left\|\underline{Y}^{k+1}-Y\right\|_{B'}^2 - \left\|\underline{Y}^k-Y\right\|_{B'}^2 + \left\|\underline{Y}^{k+1}-\underline{Y}^k\right\|_{B'}^2) \\
&\overset{(a)}{=} \phi(X,Y) - \frac{1}{2\gamma}\left(\left\|X^{k+1}-X\right\|_D^2 - \left\|X^k-X\right\|_D^2\right) \\
&\quad - \frac{1}{\gamma}\left\|\underline{X}^{k+1}\right\|_{B-\frac{1}{2}BC-AB}^2 - \left\|X^{k+1}-X^k\right\|_{\frac{1}{2\gamma}D-\frac{L}{2}I}^2 \\
&\quad - \frac{\gamma}{2}\left(\left\|\underline{Y}^{k+1}-Y\right\|_{B'}^2 - \left\|\underline{Y}^k-Y\right\|_{B'}^2\right) \\
&\overset{(b)}{\le} \phi(X,Y) - \frac{1}{2\gamma}\left(\left\|X^{k+1}-X\right\|_D^2 - \left\|X^k-X\right\|_D^2\right) \\
&\quad - \frac{\gamma}{2}\left(\left\|\underline{Y}^{k+1}-Y\right\|_{B'}^2 - \left\|\underline{Y}^k-Y\right\|_{B'}^2\right)
\end{aligned}
\tag{46}
$$

where (a) is due to the fact that $\|\underline{Y}^{k+1}-\underline{Y}^k\|_{B'}^2 = \frac{1}{\gamma^2}\|\underline{X}^{k+1}\|_{BC}^2$ since $\underline{Y}^{k+1}-\underline{Y}^k = 1/\gamma C\underline{X}^{k+1}$ and $B'C^2 = (C+bJ)^{-1}C^2B = CB$; (b) comes from that $\gamma \le \frac{\lambda_{\min}(D)}{L}$ and $B - \frac{1}{2}BC - AB = \sqrt{B}(I-\frac{1}{2}C-\sqrt{B}D\sqrt{B})\sqrt{B} \succeq 0$.

Then, averaging (46) over k from 0 to $t-1$, we have

$$
\begin{aligned}
&\frac{1}{t}\sum_{k=0}^{t-1}\left(\phi(X^{k+1},Y)-\phi(X,Y)\right) \\
&\le -\frac{1}{2\gamma t}\left(\left\|X^t-X\right\|_D^2 - \left\|X^0-X\right\|_D^2\right) \\
&\quad - \frac{\gamma}{2t}\left(\left\|\underline{Y}^t-Y\right\|_{B'}^2 - \left\|\underline{Y}^0-Y\right\|_{B'}^2\right) \\
&\overset{(a)}{\le} \frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0-X\right\|_D^2 + \gamma\frac{1}{\lambda_2(C)}\|Y\|_B^2\right) \\
&\overset{(b)}{=} \frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0-X\right\|_D^2 + \gamma\frac{1}{\lambda_2(C)}\|Y\|_{B-J}^2\right) \\
&\le \frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0-X\right\|_D^2 + \gamma\frac{\rho(B-J)}{\lambda_2(C)}\|Y\|^2\right)
\end{aligned}
\tag{47}
$$

where we used: (a) $Y^0 = 0$ and $\lambda_{\max}((C+bJ)^{-1}) = 1/\lambda_{\min}(C+bJ) = 1/\lambda_2(C)$ due to $C \preceq 2I$; (b) $Y \in \mathtt{span}(1_m)^\perp$. Using the convexity of ϕ we complete the proof.

References

[1] J. Xu, Y. Sun, Y. Tian, and G. Scutari, “A unified contraction analysis of a class of distributed algorithms for composite optimization,” in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2019, pp. 485–489.

[2] J. Xu, Y. Tian, Y. Sun, and G. Scutari, “A unified algorithmic framework for distributed composite optimization,” in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 2309–2316.

[3] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[4] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, “Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes,” in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 2055–2060.

[5] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, “Geometrically convergent distributed optimization with uncoordinated step-sizes,” in *Proc. Amer. Control Conf.*, 2017, pp. 3950–3955.

[6] P. D. Lorenzo and G. Scutari, “Next: In-network nonconvex optimization,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[7] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *IEEE Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[8] G. Scutari and Y. Sun, “Distributed nonconvex constrained optimization over time-varying digraphs,” *Math. Program.*, vol. 176, no. 1-2, pp. 497–544, Jul. 2019.

[9] Y. Sun, A. Daneshmand, and G. Scutari, “Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation,” 2019, *arXiv:1905.02637*.

[10] A. Nedich, A. Olshevsky, and W. Shi, “Achieving geometric convergence for distributed optimization over time-varying graphs,” *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[11] Z. Li, W. Shi, and M. Yan, “A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates,” *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4494–4506, Sep. 2019.

[12] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning—Part I: Algorithm development,” *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 708–723, Feb. 2019.

[13] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, “Optimal algorithms for smooth and strongly convex distributed optimization in networks,” in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3027–3036.

[14] D. Jakovetić, “A unification and generalization of exact distributed first-order methods,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 31–46, Mar. 2019.

[15] F. Mansoori and E. Wei, "A general framework of exact primal-dual first order algorithms for distributed optimization," *Proc. 58th IEEE Conf. Decis. Control*, 2019, pp. 6386–6391, *arXiv:1903.12601*.

[16] E. Wei and A. E. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE 51st Annu. Conf. Decis. Control*, 2012, pp. 5445–5450.

[17] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "A linearly convergent proximal gradient algorithm for decentralized optimization," *Proc. Adv. Neural Inf. Process. Syst.*, 2019, *arXiv:1905.07996*.

[18] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6013–6023, Nov. 2015.

[19] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "A bregman splitting scheme for distributed optimization over networks," *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 3809–3824, Nov. 2018.

[20] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 5–20, Jan. 2018.

[21] S. Pu, W. Shi, J. Xu, and A. Nedic, "Push-pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, Jan. 2021.

[22] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Contr. Syst. Lett.*, vol. 2, no. 3, pp. 315–320, Jul. 2018.

[23] A. Sundararajan, B. Van Scoy, and L. Lessard, "A canonical form for first-order distributed optimization algorithms," in *Proc. Amer. Control Conf. IEEE*, 2019, pp. 4075–4080.

[24] A. Sundararajan, B. Hu, and L. Lessard, "Robust convergence analysis of distributed optimization algorithms," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput.*, 2017, pp. 2740–2749.

[25] J. Xu, Y. Tian, Y. Sun, and G. Scutari, "Accelerated primal-dual algorithms for distributed smooth convex optimization over networks," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2381–2391.

[26] B. Van Scoy and L. Lessard, "A distributed optimization algorithm over time-varying graphs with efficient gradient evaluations," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 357–362, 2019.

[27] S. A. Alghunaim, E. K. Ryu, K. Yuan, and A. H. Sayed, "Decentralized proximal gradient algorithms with linear convergence rates," *IEEE Trans. Autom. Control*, vol. 66, no. 6, Jun. 2021.

[28] J. Xu, Y. Tian, Y. Sun, and G. Scutari, "Distributed algorithms for composite optimization: unified framework and convergence analysis," 2020, *arXiv:2002.11534*.

[29] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—Part II: Convergence analysis," *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 724–739, Feb. 2019.

[30] W. Auzinger and J.M. Melenk, Iterative solution of large linear systems, *Lecture Notes*, TU Wien, 2011.

[31] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, Feb. 2018.

[32] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[33] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1000–1008.

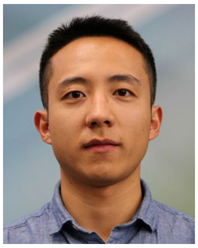
[34] A. Agarwal, S. Negahban, and M. J. Wainwright, "Fast global convergence rates of gradient methods for high-dimensional statistical recovery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 37–45.

[35] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[36] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://archive.ics.uci.edu/ml

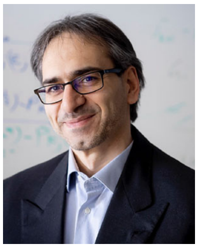
[37] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Boston, MA, USA: Kluwer, 2004.

Jinming Xu received the B.S. degree in mechanical engineering from Shandong University, China, in 2009 and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2016. He was a Research Fellow of the EXQUITUS center with NTU from 2016 to 2017, he also received postdoctoral training in the Ira A. Fulton Schools of Engineering, Arizona State University, from 2017 to 2018, and School of Industrial Engineering, Purdue University, from 2018 to 2019, respectively. He is currently an Assistant Professor with the College of Control Science and Engineering, Zhejiang University, China. His research interests include distributed optimization and control, machine learning and network science.



Ye Tian received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 2016. He is currently working toward the Ph.D. degree with the School of Industrial Engineering, Purdue University. His research interests include optimization algorithms and their applications in machine learning.

Ying Sun received the B.E. degree in electronic information from the Huazhong University of Science and Technology, Wuhan, China, in 2011, and the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology in 2016. She was a Postdoc Researcher with the School of Industrial Engineering, Purdue University from 2016 to 2020. She is currently an Assistant Professor with the Department of Electrical Engineering, The Pennsylvania State University. Her research interests include statistical signal processing, optimization algorithms and machine learning. She is the co-recipient of a Student Best Paper at IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) 2017, and the recipient of the 2020 IEEE Signal Processing Society Young Author Best Paper Award.



Gesualdo Scutari (Fellow, IEEE) received the Electrical Engineering and Ph.D. degrees (both with Hons.) from the University of Rome "La Sapienza," Rome, Italy, in 2001 and 2005, respectively. He is a Professor with the School of Industrial Engineering, Purdue University, West Lafayette, IN, USA. His research interests include continuous and distributed optimization, equilibrium programming, and their applications to signal processing and machine learning. He is a Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and an Associate Editor of *SIAM Journal on Optimization*, Among others, he was the recipient of the 2013 NSF CAREER Award, the 2015 IEEE Signal Processing Society Young Author Best Paper Award, and the 2020 IEEE Signal Processing Society Best Paper Award.