DISTRIBUTED OPTIMIZATION FOR MACHINE LEARNING: GUARANTEES AND TRADEOFFS

by

Ye Tian

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Industrial Engineering
West Lafayette, Indiana
August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Gesualdo Scutari, Chair

School of Industrial Engineering

Dr. Vaneet Aggarwal

School of Industrial Engineering

Dr. Andrew (Lu) Liu

School of Industrial Engineering

Dr. Shreyas Sundaram

School of Electrical and Computer Engineering

Approved by:

Dr. Abhijit Deshmukh

ACKNOWLEDGMENTS

First of all, I gratefully acknowledge the invaluable guidance, support and kindness from my advisor, Prof. Gesualdo Scutari, for the completion of my PhD. His pursuit of excellence has inspired me throughout. Prof. Scutari has devoted his time to helping me formulate the research problem, address technical details, and present the final results. Being very patient, he helped change my way of thinking from being a student to being a researcher.

I am also very grateful to the senior members – Prof. Ying Sun and Prof. Jinming Xu – in the research group. They have exemplified to me how to solve hard problems and how to become an independent researcher with their academic aptitude and hard-working attitude. I have been really fortunate to collaborate with them to solve some very interesting problems.

I would like to thank the rest of my thesis committee, Dr. Vaneet Aggarwal, Dr. Andrew Liu and Dr. Shreyas Sundaram, for attending my preliminary exam and final exam, and providing their insightful comments on my research work.

I would also like to thank the other group members, Amir Daneshmand, Loris Cannelli, Chang-Shen Lee, Marie Maros, Yao Ji, Tejas Tamboli, and Tianyu Cao, from whom I have learnt a lot.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In the era of big data, the sheer volume and widespread spatial distribution of information has been promoting extensive research on distributed optimization over networks. Each computing unit has access only to a relatively small portion of the entire data and can only communicate with a relatively small number of neighbors. The goal of the system is to reach consensus on the optimal parametric model with respect to the entire data among all computing units. Existing work has provided various decentralized optimization algorithms for the purpose. However, some important questions remain unclear: (I) what is the intrinsic connection among different existing algorithms? (II) what is the min-max lower complexity bound for decentralized algorithms? Can one design an optimal decentralized algorithm in the sense that it achieves the lower complexity bound? and (III) in the presence of asynchrony and imperfect communications, can one design linearly convergent decentralized algorithms?

This thesis aims at addressing the above questions. (I) Abstracting from ad-hoc, specific solution methods, we propose a unified distributed algorithmic framework and analysis for a general class of optimization problems over networks. Our method encapsulates several existing first-order distributed algorithms. Distinguishing features of our scheme are: (a) 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; (b) When the objective function is convex, but not strongly convex, similar decoupling as in (a) is established for the coefficient of the proved sublinear rate. This also reveals the role of function heterogeneity on the convergence rate; (c) 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 (d) 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. (II) Referring to lower complexity bounds, the proposed novel family of algorithms, when equipped with acceleration, are proved to be optimal, that is, they achieve convergence rate lower bounds. (III) Finally, to make the proposed algorithms practical, we break the syn-

chronism in the agents' updates: agents wake up and update without any coordination, using information only from immediate neighbors with unknown, arbitrary but bounded delays. Quite remarkably, even in the presence of asynchrony, the proposed algorithmic framework is proved to converge at a linear rate (resp. sublinear rate) when applied to strongly convex (resp. non strongly convex) optimization problems.

1. INTRODUCTION

In the era of big data, the sheer volume of information renders centralized data processing and storage a formidable task. This challenge has been promoting extensive research on parallel and distributed optimization. Classic parallel and distributed optimization typically subsumes a master-worker computational architecture wherein the master nodes/machines/agents gather the necessary information from the worker nodes and are in charge of updating the optimization variable/model parameter (cf. Fig. 1.1-left panel). However, when a large number of worker nodes are spatially scattered, collecting all this local information and routing them to the master nodes is often infeasible or inefficient, due to energy, privacy constraints and/or link/hardware failures. Furthermore, there are some networks such as surveillance networks or some cyber-physical systems, where a master-worker architecture is not desirable, as it makes the system prone to central entity failure.

Motivated by these practical challenges, in this dissertation, we consider a decentralized computational architecture, modeled as a general directed graph that does not possess a central controller/master node (see Fig. 1.1-right panel). Each computing unit has access only to a relatively small portion of the entire data and can only communicate with a relatively small number of neighbors. The goal of the system is to cooperatively solve the optimization problem under consideration.

This decentralized computational setting arises naturally when data are acquired and/or stored at the nodes' side. Examples include resource allocation, swarm robotic control, network information processing, and multi-agent reinforcement learning [1], [2]. In scenarios where both a master-worker and a decentralized architectures are available, decentralized optimization/learning has the advantage of being robust to single point failures and being communication efficient. For instance, [3] compared the performance of stochastic gradient descent on both architectures; they show that, the two implementations have similar total computational complexity, while the maximal communication cost per node of the algorithm running on the decentralized architecture is $\mathcal{O}$(degree of network), significantly smaller than the $\mathcal{O}(m)$ of the same scheme running on a master-worker system.

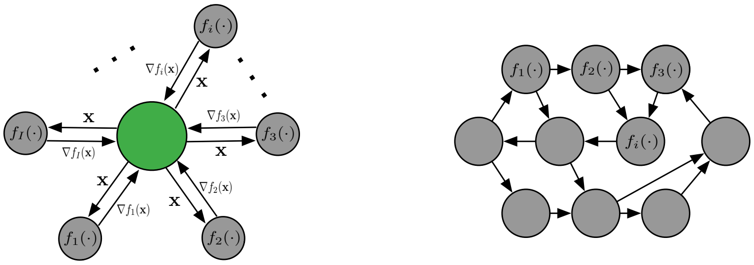


Figure 1.1. Master-worker (left panel) vs. decentralized (right panel) architectures.

As a general model, we consider the following class of (possibly nonconvex) multi-agent *composite* optimization:

$$\min_{x\in\mathcal{K}} U(x) \triangleq \sum_{\mathrm{i}\in[m]} f_{\mathrm{i}}(x) + G(x), \tag{P}$$

where $[m] \triangleq \{1,\ldots,m\}$ is the set of agents in the system; $f_{\mathrm{i}}:\mathbb{R}^d \to \mathbb{R}$ is the cost function of agent i, assumed to be smooth but possibly nonconvex; $G:\mathbb{R}^d \to \mathbb{R}\cup\{-\infty,\infty\}$ is a nonsmooth, convex (extended-value) function; and $\mathcal{K} \subseteq \mathbb{R}^d$ is a closed convex set. We also define the smooth part of Problem (P) as $F(x) \triangleq \sum_{\mathrm{i}\in[m]} f_{\mathrm{i}}(x)$. Each agent has access only to its own objective f_{i} but not F while G and $\mathcal{K}$ are common to all the agents. The communication network of all agents is modeled as a fixed, directed or undirected graph, depending on the application. One important instance of the Problem (P) is decentralized/distributed supervised learning; examples include logistic regression, SVM and LASSO, and deep learning. In these problems, each f_{i} is the empirical risk that measures the mismatch between the model (parameterized by x) to be learnt, and the data set owned *only* by agent i. G and $\mathcal{K}$ plays the role of regularization that restricts the solution space to promote some favorable structure, such as sparsity.

Existing work has provided various decentralized optimization algorithms to solve the Problem (P). However, some important questions remain unclear:

(I) what is the intrinsic connection among different existing algorithms? Can one find a unified algorithmic framework to accommodate most existing algorithms, to enable comparison among them?

(II) what is the min-max lower complexity bound for decentralized algorithms? Can one design an optimal decentralized algorithm in the sense that it achieves the lower complexity bound?

(III) for very large-scale networked system, it becomes inefficient and unrealistic to synchronize the updates of all agents. In addition, imperfect communications happen frequently as link failures and power outage commonly occur. Thus, can one design provably convergent distributed algorithms in the presence of asynchrony and imperfect communications? In particular, when the objective function U is strongly convex, can one still achieve linear convergence?

This dissertation aims at addressing the above questions, and our contributions are summarized next.

1.1 Research contribution

(I) Abstracting from ad-hoc, specific solution methods, we propose a unified distributed algorithmic framework and analysis for a general class of optimization problems over networks. Our method encapsulates several existing first-order distributed algorithms. Distinguishing features of our scheme are: (a) 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; (b) When the objective function is convex, but not strongly convex, similar decoupling as in (a) is established for the coefficient of the proved sublinear rate. This also reveals the role of function heterogeneity on the convergence rate; (c) 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 (d) 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.

(II) We propose an accelerated distributed optimization algorithmic framework, by employing acceleration on both the computations and communications of a novel family of primal-dual-based distributed algorithms. We provide a unified analysis of its convergence rate, measured in terms of the Bregman distance associated to the saddle point reformation of the distributed optimization problem. The rate of the accelerated algorithms is shown to be optimal, in the sense that it matches, under the proposed metric, existing complexity lower bounds of distributed algorithms applicable to such a class of problems and using only gradient information and gossip communications.

(III) Finally, we break the synchronism in the agents' updates: agents wake up and update without any coordination, using information only from immediate neighbors with unknown, arbitrary but bounded delays, and propose asynchronous distributed multi-agent optimization algorithms. Quite remarkably, in the presence of asynchrony, the proposed algorithms converge provably at a linear rate (resp. sublinear rate) when applied to strongly convex (resp. non strongly convex) optimization problems.

1.2 Outline of the Dissertation

In Chapter 2, we discuss a unified distributed algorithmic framework and its convergence analysis. In Chapter 3, we discuss the lower complexity bound of distributed optimization for smooth convex problems with respect to the metric of the Bregman distance, and the accelerated optimal distributed optimization algorithmic framework OPTRA. The remaining of the dissertation focuses on asynchronous decentralized/distributed algorithms; specifically, in Chapter 4, we introduce an asynchronous signal tracking algorithm, which is also the building block of the asynchronous distributed optimization algorithms proposed later; in Chapter 5, we present the asynchronous distributed algorithm, ASY-SONATA, for smooth unconstrained optimization; and in Chapter 6, we discuss the asynchronous distributed algorithm, ASY-DSCA, for general nonsmooth constrained optimization.

1.3 Notation

Throughout this dissertation, we use the following notation. Given the matrix $M \triangleq (m_{\mathrm{ij}})_{\mathrm{i,j}=1}^{m}$, $M_{\mathrm{i},:}$ and $M_{:,\mathrm{j}}$ denote its i-th row vector and j-th column vector. Given the sequence $\{M^t\}_{t=s}^{k}$, with $k \geq s$, we define $M^{k:s} \triangleq M^k M^{k-1} \cdots M^{s+1} M^s$, if $k > s$; and $M^{k:s} \triangleq M^s$ otherwise. Given two matrices (vectors) A and B of same size, by $A \preccurlyeq B$ we mean that

$B - A$ is a nonnegative matrix (vector). The dimensions of the all-one vector $\mathbf{1}$ and the i-th canonical vector e_i will be clear from the context. The indicator function $\mathbb{1}[E]$ of an event E equals to 1 when the event E is true, and 0 otherwise. We use the convention $\sum_{t\in\emptyset} x^t = 0$ and $\prod_{t\in\emptyset} x^t = 1$. We use null$(\cdot)$ (resp. span$(\cdot)$) to denote the null space (resp. range space) of the matrix argument. The inner product between two matrices X, Y is defined as $\langle X, Y\rangle = \mathrm{trace}(X, Y)$, while the induced norm is $\big\|X\big\| = \big\|X\big\|_F$. We use $\big\|\cdot\big\|_2$ to denote the spectral norm of a matrix. Given a positive semidefinite matrix Q, we define $\langle X, X\rangle_Q = \langle QX, X\rangle$ and $\big\|X\big\|_Q = \sqrt{\langle QX, X\rangle}$. Given $G : \mathbb{R}^d \to \mathbb{R}$, the proximal mapping is defined as $\mathrm{prox}_G(x) \triangleq \mathrm{argmin}_{y\in\mathcal{K}}\, G(y) + \frac{1}{2}\big\|y - x\big\|_2^2$. Let $\mathcal{K}^*$ denote the set of stationary solutions of (P), and $\mathrm{dist}(x, \mathcal{K}^*) \triangleq \min_{y\in\mathcal{K}^*} \|x - y\|$.

2. UNIFIED ALGORITHMIC FRAMEWORK FOR COMPOSITE DECENTRALIZED OPTIMIZATION: ABC

In this chapter, we propose a general *unified* algorithmic framework for solving Problem (P) and provide a convergence analysis leveraging the theory of operator splitting. Our results unify and improve several approaches proposed in the literature of distributed optimization. Distinguishing features of our framework 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.

The novel results of this chapter have been published in

- Jinming Xu, Ying Sun, Ye Tian, and Gesualdo Scutari. ”A unified contraction analysis of a class of distributed algorithms for composite optimization.” In 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 485-489. IEEE, 2019.

- Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari. ”A unified algorithmic framework for distributed composite optimization.” In 2020 59th IEEE Conference on Decision and Control (CDC), pp. 2309-2316. IEEE, 2020.

- Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari. ”Distributed algorithms for composite optimization: Unified framework and convergence analysis.” To appear on IEEE Transactions on Signal Processing (TSP), DOI: 10.1109/TSP.2021.3086579, 2021.

2.1 Introduction

The focus of this chapter is to design a unified (first-order) algorithmic framework for Problem (P), over undirected graphs, with provably convergence rate. We assume each $f_{\mathrm{i}} : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and μ-strongly convex, with $\mu \geq 0$. We start from the literature review.

2.1.1 Literature Review

When $G = 0$ and $\mu > 0$, several distributed schemes have been proposed in the literature that enjoy *linear* rate; examples include EXTRA [4], AugDGM [5], [6], NEXT [7], Harnessing [8], SONATA [9], [10], DIGing [11], NIDS [12], Exact Diffusion [13], MSDA [14], and the distributed algorithms in [15], [16]. 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 [5], [7], [8] and other primal-dual schemes, including D-ADMM [17]. Results for $G \neq 0$ are relatively scarce; to our knowledge, the only two schemes achieving linear rate for strongly convex (P) are SONATA [10] and the one in [18]. Sublinear rate of $O(1/k)$ has been proved for a variety of schemes, including PG-EXTRA [19], D-FBBS [20] and DPGA [21]. Notice that convergence of some of these algorithms have been studied under weaker assumptions on F and network topology than those considered here. For instance, linear rate of [4], [7], [10], [11], [13], [18] is established for F strongly convex (rather than each f_{i} to be so); [9]–[11], [22], [23] are applicable also to directed graphs, with [9]–[11] considering also time-varying topologies.

Even restricted to the setting of this chapter, 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: [15] provides a connection between EXTRA and DIGing; [24] provides a canonical representation of some of the distributed algorithms above–NIDS and Exact-Diffusion are proved to be equivalent; and [25] 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 2.1 summarizes these results within the setting of this chapter. For instance, a direct comparison between NIDS [12] and Exact Diffusion [13] shows that, although equivalent [15], [24], 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 [14] uses multiple communication steps to achieve the lower complexity bound of (P) when $\mu > 0$ and $G = 0$; the OPTRA algorithm [26] achieves the lower bound when $\mu = 0$ (still and $G = 0$); and the algorithms in [27] and [12] 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 2.1; they have been obtained under different stepsize range values and problem assumptions (e.g., on the weight matrices). Similarly, when $\mu = 0$, EXTRA [4], DIGing [8], [11] D-ADMM [17], and PG-EXTRA [19], D-FBBS [20], DPGA [21] 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).

Table 2.1. Convergence Properties of Distributed Algorithms for L-Smooth and μ-Strongly Convex $\{f_{\rm i}\}$ $(\mu > 0)$.

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

Algorithm	**Original assumption**		**Stepsize**		**Rate: $\mathcal{O}\left(\delta \log(\frac{1}{\epsilon})\right)$**	
	$W \in$	$F, \{f_{\rm i}\}$	**literature (upper bound)**	**our result (Corollary 2.5.4.1)**	**δ, literature**	**δ, our result**
EXTRA [4]	$\mathbb{S}^m \cap \mathbb{W}^m$	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 [7] AugDGM [5], [6]	$\mathbb{W}^m$	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 [11]	$\mathbb{W}^m$	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 [13]	$\mathbb{W}^m$	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 [8]	$\mathbb{W}^m$	$\{f_{\rm i}\}$ 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 [12]	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_{\rm i}\}$ scvx	$\frac{2}{L}$	$\frac{2}{L+\mu}$	$\max\{\kappa,\ \frac{1}{1-\rho}\}$	$\max\{\kappa,\ \frac{1}{1-\rho}\}$
[15] $(b = 0)$	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_{\rm i}\}$ scvx	$\mathcal{O}\left(\frac{(1-\rho)^2}{\kappa L}\right)$	$\frac{2}{L/\lambda_{\min}(W^2)+\mu} \geq \frac{2\lambda_{\min}(W^2)}{L+\mu}$	$\frac{\kappa^2}{(1-\rho)^2}$	$\max\left\{\frac{\kappa}{\lambda_{\min}(W^2)},\ \frac{1}{(1-\rho)^2}\right\}$
[15] $(b = \frac{1}{\gamma}W)$	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_{\rm i}\}$ scvx	N.A.	$\frac{2}{2L/(1+\check{\lambda})+\mu} \geq \frac{1-\rho}{L+\mu}$	N.A.	$\frac{\kappa}{1-\rho}$
[16]	$\mathbb{S}^m_{++} \cap \mathbb{W}^m$	$\{f_{\rm i}\}$ scvx	(14) in this chapter	$\frac{2}{\mu+L/((1-\check{\lambda})\check{\lambda}^K)} \geq \frac{2(1-\check{\lambda})\check{\lambda}^K}{L+\mu}$	N.A.	$\max\left\{\frac{1}{1-\rho^K}, \frac{\kappa}{(1-\check{\lambda})\check{\lambda}^K}\right\}$
[18]	$\mathbb{S}^m_{++} \cap \mathbb{W}^m$	F scvx	$< \frac{\check{\lambda}}{L}$	$\frac{2\check{\lambda}}{L+\mu\check{\lambda}} > \frac{\check{\lambda}}{L}$	$> \max\{\frac{\kappa}{\check{\lambda}}, \frac{1}{\alpha(1-\rho)}\}$	$\max\{\frac{\kappa}{\check{\lambda}}, \frac{1}{\alpha(1-\rho)}\}$
our result	$\mathbb{S}^m \cap \mathbb{W}^m$	$\{f_{\rm i}\}$ 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. $W \succeq 0$).

2.1.2 Summary of Contributions

This chapter 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 2.2. By product of our unified framework and convergence conditions, several existing schemes, proposed only to solve smooth instances of (P) [4], [5], [7], [8], [12], [13], gain now their "proximal" extension and thus become applicable also to composite optimization while enjoying the same convergence rate (as derived in this chapter) 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 [4], [5], [7], [8], [12], [13]–Table 2.1 shows the improvement achieved by our analysis in terms of stepsize bounds and rate expression (see Sec. 2.5.3 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 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. Sec. 5.3.4). 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 Sec. 2.5.4. These results are a major departure from existing analyses, which do not show such a clear separation, and complements those in [12] 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.

2.2 Problem Statement

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

Assumption 2.2.1. *(i) Each $f_{\mathrm{i}} : \mathbb{R}^d \to \mathbb{R}$ 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 $\{\mathrm{i}, \mathrm{j}\} \in \mathcal{E}$ 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 2.2.1 (Gossip matrix)**.** *A matrix* $W \triangleq [w_{\mathrm{ij}}] \in \mathbb{R}^{m\times m}$ *is said to be compliant to the graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *if* $w_{\mathrm{ij}} \neq 0$ *for* $\{\mathrm{i}, \mathrm{j}\} \in \mathcal{E}$*, and* $w_{\mathrm{ij}} = 0$ *otherwise. The set of such matrices is denoted by* $\mathcal{W}_{\mathcal{G}}$*.*

Definition 2.2.2 (K-hop gossip matrix)**.** *Given* $K \in \mathbb{N}_+$*, a matrix* $\widehat{W} \in \mathbb{R}^{m\times m}$ *is said to be a* K*-hop gossip matrix associated to* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *if* $\widehat{W} = P_K(W)$*, 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. Sec. 2.5.3).

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 $x_{\mathrm{i}} \in \mathbb{R}^d$ (the i-th one is owned by agent i) of x and functions

$$f(X) \triangleq \sum_{\mathrm{i}=1}^{m} f_{\mathrm{i}}(x_{\mathrm{i}}) \quad \text{and} \quad g(X) \triangleq \sum_{\mathrm{i}=1}^{m} \frac{1}{m} G(x_{\mathrm{i}}), \tag{2.1}$$

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

$$\min_{X\in\mathbb{R}^{m\times d}} f(X) + g(X), \ \text{ s.t. } \sqrt{C}X = 0, \tag{2.2}$$

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

Assumption 2.2.2. $C \in \mathbb{S}_+^m$ *and null*$(C) =$ *span*$(\mathbf{1})$*.*

Under this condition, the constraint $\sqrt{C}X = 0$ enforces a consensus among x_{i}'s and thus (2.2) is equivalent to (P). The set of points satisfying the KKT conditions of (2.2) reads:

$$\mathcal{S}_{\texttt{KKT}} \triangleq \left\{ X \in \mathbb{R}^{m\times d} \,\middle|\, \exists Y \in \mathbb{R}^{m\times d} \text{ such that } \sqrt{C}X = 0, \quad \nabla f(X) + \sqrt{C}Y \in -\partial g(X) \right\}, \tag{2.3}$$

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

Lemma 2.2.3. *Under Assumption 2.2.1, $x^\star \in \mathbb{R}^d$ is an optimal solution of Problem (P) if and only if $1_m x^{\star\top} \in \mathcal{S}_{KKT}$.*

Building on Lemma 2.2.3, in the next section, we propose a general distributed algorithm for (P) based on a suitably defined operator splitting solving the KKT system (2.3).

2.3 A General Primal-Dual Proximal Algorithm

Table 2.2. Special cases of Algorithm (2.4) 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 [4]	F	$A = \frac{I+W}{2} \quad B = I \quad C = \frac{I-W}{2}$	1
NEXT [7]/AugDGM [5], [6]	F	$A = W^2 \quad B = W^2 \quad C = (I-W)^2$	2
DIGing [11]/Harnessing [8]	F	$A = W^2 \quad B = I \quad C = (I-W)^2$	2
NIDS [12]/Exact Diffusion [13]	F	$A = \frac{I+W}{2} \quad B = \frac{I+W}{2} \quad C = \frac{I-W}{2}$	1
[15] $(B = bI)$	F	$A = W^2 + \gamma b(I-W) \quad B = I \quad C = (I-W)^2 + \gamma b(I-W)$	2
[16]	F	$A = W^K \quad B = \sum_{\mathrm{i}=0}^{K-1} W^{\mathrm{i}} \quad C = I - W^K$	K
[18]	$F+G$	$A = W \quad B = I \quad C = \alpha(I-W)$ with $0 \prec W \preceq I$ and $\alpha \leq 1$	1

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

$$X^k = \mathrm{prox}_{\gamma g}\left(Z^k\right), \tag{2.4a}$$

$$Z^{k+1} = AX^k - \gamma B\nabla f(X^k) - Y^k, \tag{2.4b}$$

$$Y^{k+1} = Y^k + CZ^{k+1}, \tag{2.4c}$$

with $Z^0 \in \mathbb{R}^{m\times d}$ and $Y^0 = 0$. In (2.4a), $\text{prox}_{\gamma g}(X) \triangleq \arg\min_Y g(Y) + \frac{1}{2\gamma}\left\|X - Y\right\|^2$ is the standard proximal operator. Eq. (2.4a) and (2.4b) 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. (2.4c) represents the update of the dual variables.

Define the set

$$\mathcal{S}_{\texttt{Fix}} \triangleq \left\{ X \in \mathbb{R}^{m\times d} \,\middle|\, CX = 0 \text{ and } \mathbf{1}^\top(I - A)X + \gamma\,\mathbf{1}^\top B\nabla f(X) \in -\gamma\,\mathbf{1}^\top \partial g(X)\right\}. \tag{2.5}$$

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 (2.4) is such that $X^\star \in \mathcal{S}_{\texttt{Fix}}$. The following are *necessary* and *sufficient* conditions on A, B for $X^\star \in \mathcal{S}_{\texttt{Fix}}$ to be a solution of (2.2).

Assumption 2.3.1. *The weight matrices $A, B \in \mathbb{R}^{m\times m}$ satisfy: $\mathbf{1}^\top A\,\mathbf{1} = m$, and $\mathbf{1}^\top B = \mathbf{1}^\top$.*

Lemma 2.3.2. *Under Assumption 2.2.2, $\mathcal{S}_{\texttt{KKT}} = \mathcal{S}_{\texttt{Fix}}$ if and only if A, B satisfy Assumption 2.3.1.*

Proof. $(\Leftarrow)$: Suppose Assumption 2.3.1 hold. First, for any $X \in \mathcal{S}_{\texttt{Fix}}$, we have $\text{span}(X) \subset \text{null}(C) = \text{span}(\mathbf{1})$ and so $\mathbf{1}^\top(I - A)X = 0$. Then we have $\mathbf{1}^\top\nabla f(X) = \mathbf{1}^\top B\nabla f(X) \in -\mathbf{1}^\top\partial g(X)$, i.e., $\exists\,\xi \in \partial g(X)$ such that $\text{span}(\nabla f(X) + \xi) \perp \text{span}(\mathbf{1}) = \text{null}(\sqrt{C})$, which implies that $\text{span}(\nabla f(X) + \xi) \subset \text{span}(\sqrt{C})$. Therefore, $\exists\,Y \in \mathbb{R}^{m\times d}$ such that $\nabla f(X) + \xi = -\sqrt{C}Y$, i.e., $\nabla f(X) + \sqrt{C}Y \in -\partial g(X)$. Hence, $X \in \mathcal{S}_{\texttt{KKT}}$. Secondly, for any $X \in \mathcal{S}_{\texttt{KKT}}$, we have $\text{span}(X) \subset \text{span}(\mathbf{1})$ and so $\mathbf{1}^\top(I - A)X + \gamma\mathbf{1}^\top B\nabla f(X) = \gamma\mathbf{1}^\top\left(\nabla f(X) + \sqrt{C}Y\right) \in -\gamma\mathbf{1}^\top\partial g(X)$, i.e., $X \in \mathcal{S}_{\texttt{Fix}}$.
$(\Rightarrow:)$ $\mathcal{S}_{\texttt{KKT}} = \mathcal{S}_{\texttt{Fix}}$ implies that, for any arbitrarily given f, g and X, if $\text{span}(X) \subset \text{span}(\mathbf{1})$ and $\mathbf{1}^\top\nabla f(X) \in -\mathbf{1}^\top\partial g(X)$, it must be $\mathbf{1}^\top(I - A)X + \gamma\mathbf{1}^\top B\nabla f(X) \in -\gamma\mathbf{1}^\top\partial g(X)$, which, due to the arbitrary nature of f, g, and X, further implies $\mathbf{1}^\top(I - A)\mathbf{1} = 0$ and $\mathbf{1}^\top B = \mathbf{1}^\top$. □

2.3.1 Connections with existing distributed algorithms

Algorithm (2.4) contains a gamut of distributed (and centralized) schemes, corresponding to different choices of the weight matrices A, B and C; any $A, B, C \in \mathcal{W}_G$ 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 (2.4) in the following equivalent form by subtracting (2.4b) at iteration $k+1$ from (2.4b) at iteration k:

$$Z^{k+2} = (I - C)Z^{k+1} + A(X^{k+1} - X^k) - \gamma B(\nabla f(X^{k+1}) - \nabla f(X^k)), \tag{2.6}$$

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

When $G = 0$, (2.6) reduces to

$$X^{k+2} = (I - C + A)X^{k+1} - AX^k - \gamma B(\nabla f(X^{k+1}) - \nabla f(X^k)). \tag{2.7}$$

We show next that the schemes in [4], [5], [7], [8], [11]–[13], [15], [16], [18] are all special cases of Algorithm (2.4). Table 2.2 summarizes the specific choices of A, B and C in (2.4) 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 2.2.2 and 2.3.1.

1) EXTRA [4]: EXTRA solves (P) with $G = 0$, and reads

$$X^{k+2} = (I + W)X^{k+1} - \tilde{W}X^k - \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)), \tag{2.8}$$

where W, $\tilde{W}$ are two design weight matrices satisfying $(I + W)/2 \succeq \tilde{W} \succeq W$ and $\tilde{W} \succ 0$. Clearly, (2.8) is an instance of (2.7) [and thus (2.4)], with $A = \tilde{W}$, $B = I$, and $C = \tilde{W} - W$.

2) NIDS [12] / Exact diffusion [13], [28]: The NIDS (Exact Diffusion) algorithm applies to (P) with $G = 0$, and reads

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

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

3) NEXT [7] **& AugDGM** [5]**:** The gradient tracking-based algorithms NEXT/AugDGM applied to (P) with $G = 0$, are:

$$X^{k+1} = W(X^k - \gamma Y^k), \tag{2.9a}$$

$$Y^{k+1} = W(Y^k + \nabla f(X^{k+1}) - \nabla f(X^k)). \tag{2.9b}$$

Eliminating the Y-variable, (2.9) can be rewritten as:

$$X^{k+2} = 2WX^{k+1} - W^2X^k - \gamma W^2(\nabla f(X^{k+1}) - \nabla f(X^k)),$$

which is clearly an instance of our general scheme (2.4), 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 (2.4). For instance, one can show that the DIGing algorithm [11] corresponds to the setting $A = W^2, B = I$, and $C = (I - W)^2$.

4) General primal-dual scheme [15]**,** [16]: A general distributed primal-dual algorithm was proposed in [15] for (P) with $G = 0$ as follows

$$X^{k+1} = WX^k - \gamma(\nabla f(X^k) + Y^k), \tag{2.10a}$$

$$Y^{k+1} = Y^k - (I - W)(\nabla f(X^k) + Y^k - BX^k), \tag{2.10b}$$

where B can be bI or bW for some positive constant $b > 0$ therein. Eliminating the Y-variable, (2.10) reduces to

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

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 [16] for (P) with $G = 0$, which reads

$$X^{k+1} = (I - \alpha B)^K X^k - \alpha C(\nabla f(X^k) + A^\top Y^k), \tag{2.11a}$$

$$Y^{k+1} = Y^k + \beta A X^{k+1}, \tag{2.11b}$$

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

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

which corresponds to Algorithm (2.4) 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 $A = W^K, B = \sum_{\mathrm{i}=0}^{K-1} W^{\mathrm{i}}$ and $C = (I - W)\sum_{\mathrm{i}=0}^{K-1} W^{\mathrm{i}} = I - W^K$, which satisfy Assumption 2.3.1.

6) Decentralized proximal algorithm [18]: A proximal algorithm is proposed to solve (P) with $G \neq 0$, which reads

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

where $X^k = \mathrm{prox}_{\gamma g}\left(Z^k\right)$ and $0 \preceq B \prec I$ is some matrix ensuring consensus. It is easy to show that the above algorithm corresponds to Algorithm (2.4) 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 2.3.1. Note that, since $B = I$, this algorithm (and thus [18]) is of CTA form and cannot model ATC-based schemes, such as NEXT/AugDGM and NIDS/Exact Diffusion listed in Table 2.2.

2.4 An Operator Splitting Interpretation

Our convergence analysis builds on an equivalent fixed-point reformulation of Algorithm (2.4), whose mapping enjoys a favorable decomposition in terms of contractive and nonexpansive operators. We begin introducing the following assumptions.

Assumption 2.4.1. *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 (2.4).

Proposition 2.4.1. *Given the sequence* $\{(Z^k, X^k, Y^k)\}_{k\in\mathbb{N}_+}$ *generated by Algorithm* (2.4), *define* $U^k \triangleq [(Z^k)^\top, (Y^k)^\top]^\top$. *Under Assumption 2.4.1, the following hold:*

1)

$$U^k = \begin{bmatrix} B & 0 \\ 0 & B\sqrt{C} \end{bmatrix} \widetilde{U}^k, \quad \text{with} \quad \widetilde{U}^k \triangleq \begin{bmatrix} \widetilde{Z}^k \\ \sqrt{C}\widetilde{Y}^k \end{bmatrix}; \tag{2.12}$$

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

$$\widetilde{U}^{k+1} = \underbrace{\begin{bmatrix} (D-\gamma\nabla f)\circ prox_{\gamma g}\circ B & -\sqrt{C} \\ \sqrt{C}(D-\gamma\nabla f)\circ prox_{\gamma g}\circ B & I-C \end{bmatrix}}_{T} \widetilde{U}^k, \quad k \geq 1, \tag{2.13}$$

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

2) *The operator* T *can be decomposed as*

$$T = \underbrace{\begin{bmatrix} I & -\sqrt{C} \\ \sqrt{C} & I-C \end{bmatrix}}_{\triangleq T_C} \underbrace{\begin{bmatrix} D-\gamma\nabla f & 0 \\ 0 & I \end{bmatrix}}_{\triangleq T_f} \underbrace{\begin{bmatrix} prox_{\gamma g} & 0 \\ 0 & I \end{bmatrix}}_{\triangleq T_g} \underbrace{\begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}}_{\triangleq T_B}, \tag{2.14}$$

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* $X^\star \triangleq prox_{\gamma g}(B\widetilde{Z}^\star) \in \mathcal{S}_{\boldsymbol{Fix}}$. *Therefore,* $X^\star = \mathbf{1}x^{\star\top}$, *where* $x^\star$ *is an optimal solution of* (P).

Proof. From (2.4), 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

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

where in $(*)$ we used Assumption 2.5.3i) and 2.5.3iv).

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

$$\widetilde{Y}^{k+1} \triangleq \sum_{t=1}^{k+1}\widetilde{Z}^t = \sum_{t=0}^{k}(I-C)^{k-t}(D-\gamma\nabla f)(X^t), \tag{2.15}$$

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

$$\begin{bmatrix}\widetilde{Z}^{k+1}\\ \widetilde{Y}^{k+1}\end{bmatrix} = \begin{bmatrix}(D-\gamma\nabla f)\circ \operatorname{prox}_{\gamma g}\circ B & -C\\ (D-\gamma\nabla f)\circ \operatorname{prox}_{\gamma g}\circ B & I-C\end{bmatrix}\begin{bmatrix}\widetilde{Z}^{k}\\ \widetilde{Y}^{k}\end{bmatrix}. \tag{2.16}$$

Introducing $\widetilde{U}^k$ as defined in (2.12), it follows from (2.16) that $\widetilde{U}^k$ obeys the dynamics (2.13). The equation $Y^k = BC\widetilde{Y}^k$ follows readily from (2.4c) and (2.15). 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 $\operatorname{span}(\widetilde{Z}^\star) \subset \operatorname{span}(1)$ and

$$-1^\top\left(B(D-\gamma\nabla f)\circ\operatorname{prox}_{\gamma g}\circ B\left(\widetilde{Z}^\star\right)\right) + 1^\top B\widetilde{Z}^\star = 0. \tag{2.17}$$

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

$$B\widetilde{Z}^\star \in X^\star + \gamma\partial g(X^\star). \tag{2.18}$$

Combining (2.17) and (2.18) 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 $X^\star \in \mathcal{S}_{\texttt{Fix}}$. The proof follows from Lemma 2.2.3 and 2.3.2. □

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 $\Lambda_A = \text{diag}(A, I) \in \mathbb{R}^{2m\times 2m}$ and $V_A = \text{diag}(I, A) \in \mathbb{R}^{2m\times 2m}$.

Lemma 2.4.2 (Contraction of T_C)**.** *The operator T_C satisfies*

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

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 2.4.3 (Contraction of T_f)**.** *Consider the operator T_f under Assumption 2.2.1, with $\mu > 0$, and $0 \prec \Sigma \preceq I$. If $0 < \gamma \leq \gamma^\star(D)$ with*

$$\gamma^\star(D) \triangleq \frac{2\lambda_{\min}(D)}{L + \mu \cdot \lambda_{\min}(D)}, \tag{2.19}$$

then

$$\left\|(T_f X)_u - (T_f Y)_u\right\|^2 \leq q(D,\gamma)\left\|(X)_u - (Y)_u\right\|_D^2,$$

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

$$q(D,\gamma) = 1 - \frac{2\gamma L}{\kappa + \lambda_{\min}(D)}. \tag{2.20}$$

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

$$q^\star(D) \triangleq \left(\frac{\kappa - \lambda_{\min}(D)}{\kappa + \lambda_{\min}(D)}\right)^2. \tag{2.21}$$

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

$$\begin{aligned}
&\left\|DX - \gamma\nabla f(X) - DY + \gamma\nabla f(Y)\right\|^2 \\
&\leq \left\|DX - \gamma\nabla f(X) - DY + \gamma\nabla f(Y)\right\|_{D^{-1}}^2 \\
&= \left\|X - Y\right\|_D^2 - 2\gamma\left\langle X - Y, \nabla f(X) - \nabla f(Y)\right\rangle + \gamma^2\left\|\nabla f(X) - \nabla f(Y)\right\|_{D^{-1}}^2.
\end{aligned} \tag{2.22}$$

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

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

where $(*)$ is due to [29, 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 (2.22), we have

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

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

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

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

Lemma 2.4.5 (Non-expansiveness of T_B)**.** *The operator T_B satisfies: $\forall X \in \mathbb{R}^{2m\times d}$,*

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

2.5 Linear Convergence

In this section we prove linear convergence of Algorithm (2.4), 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. Sec. 2.5.1). Sec.2.5.2 extends our analysis to $G \neq 0$. Finally, we comment our results in Sec.2.5.3.

2.5.1 Convergence under $G = 0$

Consider Problem (P) with $G = 0$. Algorithm (2.4) reduces to

$$X^{k+1} = AX^k - \gamma B\nabla f(X^k) - Y^k, \tag{2.23a}$$

$$Y^{k+1} = Y^k + CX^{k+1}, \tag{2.23b}$$

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

Theorem 2.5.2 below establishes linear convergence of Algorithm (2.23) under the following assumption on A, B and C.

Assumption 2.5.1. *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) $\mathbf{1}^\top D\mathbf{1} = m$ and $\mathbf{1}^\top B = \mathbf{1}^\top$;

iii) $0 \preceq C \prec I$ and null$(C) =$ span$(\mathbf{1})$;

iv) B and C commute;

v) $q(D,\gamma)AB \prec (I - C)$ and $0 < \gamma \le \gamma^\star(D)$,

where $q(D,\gamma)$ and $\gamma^\star(D)$ are defined in (2.20) and (2.19), respectively.

Assumption 2.5.1 is quite mild and satisfied by a variety of algorithms. For instance, all the algorithms in Table 2.2 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 2.5.2 (Linear rate for $T_C T_f T_B$)**.** *Consider Problem* (P) *under Assumption 2.2.1, $\mu > 0$, and $G = 0$, with solution $x^\star$. Let $\{(X^k, Y^k)\}_{k \in \mathbb{N}_+}$ be the sequence generated by Algorithm* (2.23) *under Assumption 2.5.1. Then, $\left\|X^k - \mathbf{1}x^{\star\top}\right\|^2 = \mathcal{O}(\delta^k)$, with*

$$\delta \triangleq \max\Big(q(D,\gamma)\,\lambda_{\max}(AB(I-C)^{-1}), 1 - \lambda_2(C)\Big), \tag{2.24}$$

where $q(D,\gamma)$ is defined in (2.20)*.*

Proof. Since (2.23) corresponds to Algorithm (2.4) with $G = 0$, by Assumption 2.5.1 and Prop. 2.4.1, (2.23) can be equivalently rewritten in the form (2.13), with $T_g = I$; and thus the Z- and X-variables coincide. Define $X^\star = Z^\star \triangleq \mathbf{1}x^{\star\top}$. Let $\widetilde{U}^k = [(\widetilde{Z}^k)^\top, (\sqrt{C}\widetilde{Y}^k)^\top]^\top$ be the auxiliary sequence defined in (2.12) with $\widetilde{U}^\star \triangleq [\widetilde{Z}^\star, \sqrt{C}\widetilde{Y}^\star]$ the fixed point of $T = T_C T_f T_B$. Then, we have

$$\begin{aligned} &\left\|X^k - X^\star\right\|^2 = \left\|Z^k - Z^\star\right\|^2 \overset{(2.12)}{\leq} \left\|\widetilde{Z}^k - \widetilde{Z}^\star\right\|^2_{B^2} \\ &\leq \frac{\lambda_{\max}(B^2)}{\lambda_{\min}(I-C)}\left\|\widetilde{Z}^k - \widetilde{Z}^\star\right\|^2_{I-C} \leq \frac{\lambda_{\max}(B^2)}{\lambda_{\min}(I-C)}\left\|\widetilde{U}^k - \widetilde{U}^\star\right\|^2_{\Lambda_{I-C}}. \end{aligned} \tag{2.25}$$

Using (2.13) in (2.25), it is sufficient to prove that T is contractive w.r.t. the norm $\left\|\cdot\right\|_{\Lambda_{I-C}}$. To this end, consider the following chain of inequalities: $\forall\, X, Y \in \mathbb{R}^{2m \times d}$, $X_\ell, Y_\ell \in \operatorname{span}(\sqrt{C})$,

$$\begin{aligned} &\left\|T\,X - T\,Y\right\|^2_{\Lambda_{I-C}} = \left\|T_C \circ T_f \circ T_B\,(X) - T_C \circ T_f \circ T_B\,(Y)\right\|^2_{\Lambda_{I-C}} \\ &\overset{\textit{Lem. } 2.4.2}{=} \left\|T_f \circ T_B\,(X) - T_f \circ T_B\,(Y)\right\|^2_{V_{I-C}} \overset{\textit{Lem. } 2.4.3}{\leq} \left\|T_B\,(X) - T_B\,(Y)\right\|^2_{\operatorname{diag}(q(D,\gamma)\,D,\,I-C)} \\ &\overset{\textit{Lem. } 2.4.5}{=} \left\|X - Y\right\|^2_{\operatorname{diag}(q(D,\gamma)\,BDB,\,I-C)}. \end{aligned} \tag{2.26}$$

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

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

and ii) $X_\ell, Y_\ell \in \text{span}(\sqrt{C})$. The upper block term of the RHS of (2.26) can be upper bounded by $q(D,\gamma)\lambda_{\max}(AB(I-C)^{-1})\big\|X_u - Y_u\big\|^2_{\Lambda_{I-C}}$; and the lower block term of that can be upper bounded by $(1-\lambda_2(C))\big\|X_\ell - Y_\ell\big\|^2$. Together we have $\big\|T\,X - T\,Y\big\|^2_{\Lambda_{I-C}} \leq \delta\big\|X-Y\big\|^2_{\Lambda_{I-C}}$. □

Note that Theorem 2.5.2 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 2.5.1 might depend on some network parameters. This is due to the fact that $\lambda_{\max}(AB(I-C)^{-1}) \geq 1$, since $(I-C)^{-1/2}AB(I-C)^{-1/2}\mathbf{1} = \mathbf{1}$. 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 2.5.2.1 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 (2.24) is derived.

Corollary 2.5.2.1. *Consider the setting of Theorem 2.5.2, and further assume $AB \preceq I - C$. Then, $\big\|X^k - \mathbf{1}x^{\star\top}\big\|^2 = \mathcal{O}(\delta^k)$, with*

$$\delta = \max\Big(q(D,\gamma),\ 1-\lambda_2(C)\Big). \tag{2.27}$$

The stepsize that minimizes (2.27) is $\gamma = \gamma^\star(D) = \frac{2\lambda_{\min}(D)}{L+\mu\cdot\lambda_{\min}(D)}$, resulting in the contraction factor

$$\delta = \max\left(\left(\frac{\kappa - \lambda_{\min}(D)}{\kappa + \lambda_{\min}(D)}\right)^2, 1-\lambda_2(C)\right). \tag{2.28}$$

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

$$\delta^\star = \max\left\{\left(\frac{\kappa-1}{\kappa+1}\right)^2, \ 1-\lambda_2(C)\right\}. \tag{2.29}$$

Proof. Since $(I-C)^{-1/2}AB(I-C)^{-1/2}\mathbf{1} = \mathbf{1}$ and $AB \preceq I-C$, we have $\lambda_{\max}(AB(I-C)^{-1}) = 1$, which together with (2.24) yield (2.27). Eq. (2.28) 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, (2.29) is the result of the following optimization problem: $\max_{D\in\mathbb{S}^m} \lambda_{\min}(D)$, subject to $0 \prec \Sigma \preceq I$ [Assumption 2.5.1(i)] and $\mathbf{1}^\top\Sigma\mathbf{1} = m$ [Assumption 2.5.1(ii)], whose solution is $D = I$. □

2.5.2 The general case $G \neq 0$

We establish now linear convergence of Algorithm (2.4) applied to Problem (P), with $G \neq 0$. We introduce the following assumption similar to Assumption 2.5.1 for $G = 0$.

Assumption 2.5.3. *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) $\mathbf{1}^\top D\mathbf{1} = m$ and $\mathbf{1}^\top B = \mathbf{1}^\top$;

iii) $0 \preceq C \prec I$ and null$(C) =$ span$(\mathbf{1})$;

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 (2.20) and (2.19), respectively.

Condition v) in Assumption 2.5.3 is slightly stronger than its counterpart in Assumption 2.5.1 (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 2.5.4.1 below, this does not affect the smallest achievable contraction rate, which coincides with the one attainable when $G = 0$. Note that Assumption 2.5.3 is satisfied by all the algorithms in Table 2.2.

Theorem 2.5.4 (Linear rate for $T = T_C T_f T_g T_B$)**.** *Consider Problem* (P) *under Assumption 2.2.1 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* (2.4) *under Assumption 2.5.3. Then $\left\|X^k - \mathbf{1}x^{\star\top}\right\|^2 = \mathcal{O}(\delta^k)$, with*

$$\delta \triangleq \max\left(q(D,\gamma)\, \lambda_{max}(B^2(I-C)^{-1}),\ 1-\lambda_2(C)\right), \tag{2.30}$$

where $q(D,\gamma)$ is defined in (2.20)*.*

The proof of Theorem 2.5.4 is similar to that of Theorem 2.5.2 and can be found in the supplementary material.

Corollary 2.5.4.1. *Consider the setting of Theorem 2.5.4, and further assume $B^2 \preceq I - C$. Then, the same conclusions as in Corollary 2.5.2.1 hold for Algorithm* (2.4)*.*

Remark: We point out that linear convergence of Algorithm (2.4) 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 (2.30), the proved convergence rate does show a coupling between optimization and network parameters. This is consistent with existing results in the literature.

2.5.3 Discussion

- Unified convergence conditions Theorems 2.5.2 and 2.5.4 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 (2.23) and (2.4) satisfying Assumption 2.5.1 and 2.5.3, respectively–e.g., all the algorithms listed in Table 2.1. 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 [18] and [30].

- On the rate expression

We comment the expression of the rate focusing on Theorem 2.5.4 and Corollary 2.5.4.1 ($G \neq 0$); same conclusions can be drawn for Algorithm (2.23) (Theorem 2.5.2 and Corollary 2.5.2.1). Theorem 2.5.4 provides the explicit expression of the linear rate provably achievable by Algorithm (2.4), for a given choice of the weight matrices A, B and C and stepsize γ

(satisfying Assumption 2.5.3). 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 2.5.3v). **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 (2.23). For instance, consider EXTRA [4] and DIGing [11] (corresponding to Algorithm (2.23) with $B = I$, cf. Table 2.1): 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 2.5.3v) is satisfied by $B^2 \prec I - C$–a condition that is met by several algorithms in Table 2.1–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 2.5.4.1).

On the other hand, when the algorithm parameters can be freely designed, Corollary 2.5.2.1 offers the "optimal" choice, resulting in the smallest contraction factor, as in (2.29). This instance enjoys two desirable properties, namely:

(i) Network-independent stepsize: The stepsize $\gamma^\star$ in Corollary 2.5.2.1 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 [12], whose algorithm however cannot deal with the non-smooth term G and use more stringent stepsize.

(ii) Rate-separation: The rate (2.29) 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.

- 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 2.5.2.1). 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. (2.29)], 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 [31]. 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 $P_K(x) = T_K(\frac{x}{\rho_{\texttt{com}}})/T_K(\frac{1}{\rho_{\texttt{com}}})$ [31, 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 [14], [31]: $T_{k+1}(\xi) = 2\xi\, T_k(\xi) - T_{k-1}(\xi)$, $k \geq 1$, with $T_0(\xi) = 1$, $T_1(\xi) = \xi$. Also, invoking [31, Corollary 6.3], we have $\rho_C = \frac{2c^K}{1+c^{2K}}$, 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 $K = \left\lceil \log_c \left(1/\rho_{\texttt{opt}}^2 + \sqrt{1/\rho_{\texttt{opt}}^4 - 1} \right) \right\rceil$. 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 [32].

- Improvement upon existing results and tuning recommendations Theorems 2.5.2 and 2.5.4 improve upon existing convergence conditions and rate bounds (when restricted to our setting, cf. Assumptions 2.2.1 and 2.2.2). A comparison with notable distributed algorithms in the literature is presented in Table 2.1. Since all the schemes therein are special cases of Algorithm (2.23) [with the exception of [18] that is an instance of Algorithm (2.4)] (cf. Table 2.2) and satisfy Assumption 2.5.1 (or Assumption 2.5.3), one can readily apply

Theorem 2.5.2 (or Theorem 2.5.4) and determine, for each of them, a new stepsize range and achievable rate: the column "Stepsize/our result (optimal, Corollary 2.5.2.1)" reports the stepsize value $\gamma^\star(D)$ for the different algorithms (i.e., given B, C and D) while the column "Rate/δ our result" shows the resulting provably rate, as given in (2.28). 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 [4] NEXT [7]/AugDGM [5], [33] and Exact Diffusion [13], and faster linear rates for *all* the algorithms in the table.

Table 2.1 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 communication per gradient evaluation, NIDS is provably faster, achieving a linear rate of $\delta^\star \log(1/\epsilon)$, with $\delta^\star$ defined in (2.29), versus the linear rate $(\kappa/(1-\rho))\log(1/\epsilon)$ of EXTRA. In Sec. 2.7.1 we show that the ranking based on our theoretical findings in Table 2.1 is reflected by our numerical experiments–see Fig. 2.3. For the sake of fairness, we remark one more time that, the stepsize and rate expressions of some of the algorithms listed in Table 2.1 were obtained under weaker conditions on F and W than Assumptions 2.2.1 and 2.2.2.

- Generalizing existing algorithms to the case $G \neq 0$ All the algorithms listed in Table 2.1 but [7] and [18] are designed for Problem (P) with $G = 0$. Since they are special cases of our general framework and Algorithm (2.4) can deal with the case $G \neq 0$, they inherit the same feature. Their "proximal" extension is given by (2.6), with the matrices A, B and C as in original algorithm (cf. Table 2.2). Theorem 2.5.4 and Corollary 2.5.4.1 show that these new algorithms enjoy the same convergence rates of their "no-proximal" counterpart. For instance, consider AugDGM, corresponding to Algorithm (2.23) with $A = B = W^2$, $D = I$, $C = (I-W)^2$; it clearly satisfies Assumption 2.5.3 for $W \succ 0$. Its extension to the general optimization with $G \neq 0$ comes readily substituting these choices of A, B, C into (2.6) (or Algorithm 2.23), yielding

$$\begin{aligned} X^{k+1} &= \mathrm{prox}_{\gamma g}\left(Z^{k+1}\right), \\ Z^{k+2} &= (2W - W^2)Z^{k+1} + W^2(X^{k+1} - X^k) - \gamma\, W^2(\nabla f(X^{k+1}) - \nabla f(X^k)). \end{aligned} \tag{2.31}$$

As second example, consider the primal-dual scheme such as NIDS and Exact Diffusion; they correspond to Algorithm (2.23) with $A = B = \frac{I+W}{2}, C = \frac{I-W}{2}$. Similarly, we can introduce their "proximal" version as follows:

$$\begin{aligned} X^k &= \text{prox}_{\gamma g}\left(Z^k\right), \\ Z^{k+2} &= \frac{I+W}{2}\left(Z^{k+1} + X^{k+1} - X^k - \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)\right). \end{aligned} \tag{2.32}$$

2.5.4 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 $\{z_{\mathrm{j}}\}_{\mathrm{j}\in\mathcal{D}_{\mathrm{i}}}$ 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:

$$\min_{x\in\mathbb{R}^d} \sum_{\mathrm{i}\in[m]} \sum_{\mathrm{j}\in\mathcal{D}_{\mathrm{i}}} \ell(x; z_{\mathrm{j}}),$$

where $\ell(x; z_{\mathrm{j}})$ is the loss function measuring the fitness of the statistical model parameterized by x to sample z_{j}; we assume each $\ell(x; z_{\mathrm{j}})$ 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 $f_{\mathrm{i}}(x) \triangleq \sum_{\mathrm{j}\in\mathcal{D}_{\mathrm{i}}} \ell(x; z_{\mathrm{j}})$. Denote the largest and the smallest eigenvalues of $\nabla^2 f_{\mathrm{i}}(x)$ (resp. $\nabla^2 F(x)$) as L_{i} and μ_{i} (resp. $\bar{L}$ and $\bar{\mu}$). Then, each $f_{\mathrm{i}}(x)$ is $\mu \triangleq \min_{\mathrm{i}\in[m]} \mu_{\mathrm{i}}$-strongly convex and $L \triangleq \max_{\mathrm{i}\in[m]} L_{\mathrm{i}}$-smooth. Recalling $\kappa = L/\mu$, the rate in (2.29) 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 $\mathcal{O}\left(\kappa \log\left(1/\epsilon\right)\right)$ overall number of gradient evaluations. On the other hand, the complexity of the centralized gradient descent algorithm reads $\mathcal{O}\left(\frac{\bar{L}}{\bar{\mu}}\log\left(\frac{1}{\epsilon}\right)\right)$. To compare these two quantities, compute

$$
\begin{aligned}
&\left|\frac{L}{\mu}-\frac{\bar{L}}{\bar{\mu}}\right| = \frac{\left|L\bar{\mu}-\bar{L}\mu\right|}{\mu\bar{\mu}} \leq \frac{\left|L-\bar{L}\right|\bar{\mu}+\bar{L}\left|\bar{\mu}-\mu\right|}{\widetilde{\mu}^2} \\
&\leq \frac{1}{\widetilde{\mu}^2}\left(\bar{\mu}\max_{\mathrm{i}\in[m]}\left|L_{\mathrm{i}}-\bar{L}\right| + \bar{L}\max_{\mathrm{i}\in[m]}\left|\mu_{\mathrm{i}}-\bar{\mu}\right|\right) \\
&\overset{(a)}{\leq} \frac{\bar{\mu}+\bar{L}}{\widetilde{\mu}^2}\sqrt{\frac{32\widetilde{L}^2\log(dm/\delta)}{n}}, \quad \text{with probability } 1-\delta \\
&\leq 8\sqrt{2}\,\frac{\widetilde{L}^2}{\widetilde{\mu}^2}\sqrt{\frac{\log(dm/\delta)}{n}},
\end{aligned}
$$

where in (a) we used [34, Corollary 6.3.8]

$$
\max_{\mathrm{i}\in[m]}\left(\left|\mu_{\mathrm{i}}-\bar{\mu}\right|,\ \left|L_{\mathrm{i}}-\bar{L}\right|\right) \leq \left\|\nabla^2 f_{\mathrm{i}}(x)-\nabla^2 f(x)\right\|, \tag{2.33}
$$

and [35, Lemma 2]

$$
\max_{\mathrm{i}\in[m]}\left\|\nabla^2 f_{\mathrm{i}}(x)-\nabla^2 f(x)\right\| \leq \sqrt{\frac{32\widetilde{L}^2\log(dm/\delta)}{n}} \tag{2.34}
$$

with probability at least $1-\delta$. Therefore, the complexity of our algorithm becomes

$$
\mathcal{O}\left(\left(\frac{\bar{L}}{\bar{\mu}}+\widetilde{\mathcal{O}}\left(\frac{\bar{L}^2}{\bar{\mu}^2}\frac{1}{\sqrt{n}}\right)\right)\cdot\log\left(\frac{1}{\epsilon}\right)\right),
$$

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.

2.6 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) $T = T_C T_f T_B$ applied to (P) with $G = 0$; and ii) $T = T_C T_g T_f T_B$ applied to (P) with $G \neq 0$.

2.6.1 Convergence under $G = 0$

We establish sublinear convergence of Algorithm (2.23) (corresponding to $T = T_C T_f T_B$) under the following assumption.

Assumption 2.6.1. *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) $D\mathbf{1} = \mathbf{1}$ *and* $\mathbf{1}^\top B = \mathbf{1}^\top$*;*

iii) $C \succeq 0$ *and null*$(C) =$ *span*$(\mathbf{1})$*;*

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:

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

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

We begin by rewriting Algorithm (2.23) in an equivalent form given in Lemma 2.6.2, which does not have a mixing matrix multiplied to the gradient term.

Lemma 2.6.2. *Suppose Assumption 2.4.1 holds. Then, Algorithm (2.23) can be rewritten as (with $\underline{Y}^0 \triangleq 0$):*

$$X^k = B\underline{X}^k, \tag{2.35a}$$

$$\underline{X}^{k+1} = DX^k - \gamma(\nabla f(X^k) + \underline{Y}^k), \tag{2.35b}$$

$$\underline{Y}^{k+1} = \underline{Y}^k + \frac{1}{\gamma}C\underline{X}^{k+1}. \tag{2.35c}$$

Proof. since $Y^0 = 0$, we know $\text{span}(X^1), \text{span}(Y^1) \subset \text{span}(B)$. It is easy then to deduce from induction that $\text{span}(X^k)$, $\text{span}(Y^k) \subset \text{span}(B)$, $\forall k$. 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 2.6.3 and 2.6.4 below, we establish two fundamental inequalities on $\phi(X^k, Y)$ and $\phi(X, Y)$ for $X \in \text{span}(\mathbf{1})$ and $Y \in \text{span}(C)$, instrumental to prove the sublinear rate; the proofs are reported in Sec. 2.9.1 in Appendix.

Lemma 2.6.3. *Consider the setting of Theorem 2.6.5, let $\{X^k, \underline{X}^k, \underline{Y}^k\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm (2.35) under Assumption 2.6.1. Then, it holds:*

$$\begin{aligned}\phi(X^{k+1}, Y) \leq \phi(X, Y) - \frac{1}{\gamma}\left\|\underline{X}^{k+1}\right\|^2_{B-BC-AB} - \frac{1}{\gamma}\left\langle X^{k+1} - X^k, X^{k+1} - X\right\rangle_D \\ - \gamma\left\langle \underline{Y}^{k+1} - Y, \underline{Y}^{k+1} - \underline{Y}^k\right\rangle_B + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2,\end{aligned} \tag{2.36}$$

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

Lemma 2.6.4. *Under the same conditions as in Lemma 2.6.3, if $\gamma \leq \frac{\lambda_{\min}(D)}{L}$, then*

$$\phi(\widehat{X}^k, Y) - \phi(X, Y) \leq \frac{1}{2k}\left(\frac{1}{\gamma}\left\|X^0 - X\right\|^2_D + \gamma\frac{\rho(B - J)}{\lambda_2(C)}\left\|Y\right\|^2\right), \tag{2.37}$$

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

We now prove the sublinear convergence rate.

Theorem 2.6.5 (Sublinear rate for $T_C T_f T_B$)**.** *Consider Problem (P) under Assumption 2.2.1 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 (2.23) under Assumptions 2.6.1. Then, if $0 < \gamma \leq \frac{\lambda_{\min}(D)}{L}$, we have*

$$M(\widehat{X}^k) \leq \frac{1}{k}\left(\frac{1}{2\gamma}\left\|X^0 - X^\star\right\|^2_D + 2\gamma\,\frac{\rho(B - J)}{\lambda_2(C)}\left\|\nabla f(X^\star)\right\|^2\right), \tag{2.38}$$

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

Proof. Setting $X = X^\star$ in (2.36), it holds

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

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

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

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

□

Finally, we leverage Young inequality to provide the choice of γ that optimizes the rate given in Theorem 2.6.5.

Corollary 2.6.5.1. *Consider the setting of Theorem 2.6.5. The stepsize that minimizes the right hand side of (2.38) is*

$$\gamma = \min\left(\frac{\lambda_{\min}(D)}{L}, \frac{1}{2}\sqrt{\frac{\lambda_2(C)}{\rho(B-J)}}\frac{\left\| X^0 - X^\star \right\|_D}{\left\| \nabla f(X^\star) \right\|}\right), \tag{2.39}$$

leading to a sublinear rate

$$M(\widehat{X}^k) \leq \frac{1}{k}\max\left\{\frac{L\left\| X^0 - X^\star \right\|_D^2}{\lambda_{\min}(D)}, 2\sqrt{\frac{\rho(B-J)}{\lambda_2(C)}}\left\| X^0 - X^\star \right\|_D \left\| \nabla f(X^\star) \right\|\right\}. \tag{2.40}$$

Note that the stepsize in (2.39) depends on $\left\| X^0 - X^\star \right\|_D / \left\| \nabla f(X^\star) \right\|$, an information that is not generally available; we discuss this issue in Sec. 2.6.3.

2.6.2 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 (2.4), 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

$$\begin{aligned} \underline{X}^{k+1} &= DX^k - \gamma(\nabla f(X^k) + \underline{Y}^k), \\ X^{k+1} &= \operatorname{prox}_{\gamma g}\left(\underline{X}^{k+1}\right), \\ \underline{Y}^{k+1} &= \underline{Y}^k + \frac{1}{\gamma} C X^{k+1}, \end{aligned} \tag{2.41}$$

with $\underline{Y}^0 \triangleq 0$. Note that a key difference between (2.4) and the above algorithm is that the former uses $\underline{X}$ in the update of the dual variable Y, the variable before the operator $\operatorname{prox}_{\gamma g}(\cdot)$, while the latter uses the variable X, i.e., the variable after the operator $\operatorname{prox}_{\gamma g}(\cdot)$. It is not difficult to check that (2.41) subsumes many existing proximal-gradient methods, such as PG-EXTRA [19] or ID-FBBS [20] (with $D = W, C = I - W$). We present a unified result of the sublinear convergence for the algorithm (2.41), under the following assumption.

Assumption 2.6.6. *The weight matrices C, $D \in \mathbb{S}^m$ satisfy:*

i) $\mathbf{1}^\top D \mathbf{1} = m$;

ii) $C \succeq 0$ *and* $\mathit{null}(C) = \mathit{span}(\mathbf{1})$;

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

Note that the above assumption is, indeed, a customization of Assumption 2.6.1. We study convergence of Algorithm (2.41) using the following merit function measuring the progresses of the algorithms from consensus and optimality. Define

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

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

[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 (2.14).

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 2.6.5.

Theorem 2.6.7 (Sublinear rate for $T = T_C T_g T_f$)**.** *Consider Problem* (P) *under Assumption 2.2.1 with $\mu = 0$; and let $x^\star$ be an optimal solution. Let $\{(X^k, Y^k)\}_{k\geq 0}$ be the sequence generated by Algorithm* (2.41) *under Assumptions 2.6.6. Then, if $\gamma < \frac{\lambda_{\min}(D)}{L}$, we have*

$$M(\widehat{X}^k) \leq \frac{1}{k}\left(\frac{1}{2\gamma}\left\|X^0 - X^\star\right\|_D^2 + 2\gamma\,\frac{1}{\lambda_2(C)}\left\|\nabla f(X^\star)\right\|^2\right), \tag{2.42}$$

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

Corollary 2.6.7.1. *Consider the setting of Theorem 2.6.7. The stepsize that minimizes the right hand side of* (2.42) *is*

$$\gamma = \min\left(\frac{\lambda_{\min}(D)}{L}, \frac{1}{2}\sqrt{\lambda_2(C)}\frac{\left\|X^0 - X^\star\right\|_D}{\left\|\nabla f(X^\star)\right\|}\right), \tag{2.43}$$

leading to a sublinear rate

$$M(\widehat{X}^k) \leq \frac{1}{k}\max\left\{\frac{L\left\|X^0 - X^\star\right\|_D^2}{\lambda_{\min}(D)}, 2\sqrt{\frac{1}{\lambda_2(C)}}\left\|X^0 - X^\star\right\|_D\left\|\nabla f(X^\star)\right\|\right\}. \tag{2.44}$$

2.6.3 Discussion

- On rate seperation

Differently from most of the existing works, such as [4], [8], [21], the above convergence results (Corollary 2.6.5.1 and 2.6.7.1) 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 (2.40) and (2.44) 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., $\left\|\nabla f(x^\star)\right\|$). The smaller $\left\|\nabla f(x^\star)\right\|$, the more similar the objective functions agents have. For instance, when f_i's share a common minimizer, i.e., $\left\|\nabla f(x^\star)\right\| = 0$, 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 graph), $\sqrt{\frac{\rho(B-J)}{\lambda_2(C)}} \to 0$, leading to the rate of the centralized gradient algorithm [cf. (2.40)]. 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 $\left\|\nabla f(x^\star)\right\|$) or the network is well connected, the first term will dominate the second, leading to the centralized performance. The impact of the heterogeneity quantity $\left\|\nabla f(x^\star)\right\|$ on the convergence behavior is validated by our numerical results–see Section 2.

- On the choice of stepsize

The optimal stepsize, as indicated in (2.39) (resp. (2.43)), is such that the two terms in (2.38) (resp. (2.42)) are balanced. Albeit (2.39) and (2.43) generally are not implementable, due to the unknown quantity $\left\|X^0 - X^\star\right\|_D / \left\|\nabla f(X^\star)\right\|$, 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., $\left\|\nabla f(x^\star)\right\|$ 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.

2.7 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 (2.29)]: 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.

2.7.1 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

$$\min_{x\in\mathbb{R}^d}\left(\frac{1}{50}\sum_{\mathrm{i}=1}^{50}\left\|U_{\mathrm{i}}x-v_{\mathrm{i}}\right\|^2\right)+\rho\left\|x\right\|_2^2+\lambda\left\|x\right\|_1. \tag{2.45}$$

where $U_{\mathrm{i}}\in\mathbb{R}^{r\times d}$ and $v_{\mathrm{i}}\in\mathbb{R}^{r\times 1}$ 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 $M_{:,\mathrm{i}}$ (resp. $M_{\mathrm{i},:}$) 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 **agarwal2010fast**: 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 $U_{:,\mathrm{i}}=\omega U_{:,\mathrm{i}-1}+Z_{:,\mathrm{i}}$, for $\mathrm{i}=2,\ldots,d$. As a result, each row $U_{\mathrm{i},:}\in\mathbb{R}^d$ is a Gaussian random vector and its covariance matrix $\Sigma=\mathrm{cov}(U_{:,\mathrm{i}})$ 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 (2.45).

- Validating the rate separation We validate here the rate results predicted by Corollary 2.5.2.1 and 2.5.4.1. We consider Algorithm (2.4), with $A=B=\frac{I+W}{2}$ and $C=I-B$, and run two experiments. **1)** We simulated problem (2.45), with $\rho=10$ and $\omega=0.999$–this leads to an extremely large condition number, $\big((\kappa-1)/(\kappa+1)\big)^2\approx 0.9999$)–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. 2.1 plots the optimality gap $\frac{1}{\sqrt{50}}\left\|X^k-\mathbf{1}x^{\star\top}\right\|$ 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 $\big((\kappa-1)/(\kappa+1)\big)^2$ being 0.9782, 0.9845, 0.9895 and 0.9922 respectively; Fig. 2.2 plots the optimality gap (defined as in Fig. 2.1) 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. 2.1: For ill-conditioned problems–meaning $\big((\kappa-1)/(\kappa+1)\big)^2 > 1-\lambda_2(C)$–the algorithm exhibits almost identical rates, irrespectively of the specific graph instances. On the other hand, Fig. 2.2 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 $\big((\kappa-1)/(\kappa+1)\big)^2 < 1-\lambda_2(C)$.

- More on the rate separation (2.29) We simulated the following instances of Algorithm 2.4. 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 [36], 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 2.5.4.1 is satisfied). In Fig. 2.3, 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 (2.29) for the max in (2.29) to achieve the minimum value, that is, $\big\lceil 2\log(\frac{\kappa-1}{\kappa+1})/\log(\frac{1+\lambda_{m-1}(W)}{2})\big\rceil$. 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 (2.29)]. **(ii)** Less number of iterations are 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.

- Validating Table 2.1: Comparison of the "prox"-versions of existing algorithms In Fig. 2.4 we compare the "prox" version of several existing algorithms, applied to (2.45): we plot the optimality gap $\left\|X^k-\mathbf{1}x^{\star\top}\right\|$ versus the overall number of iterations (gradient

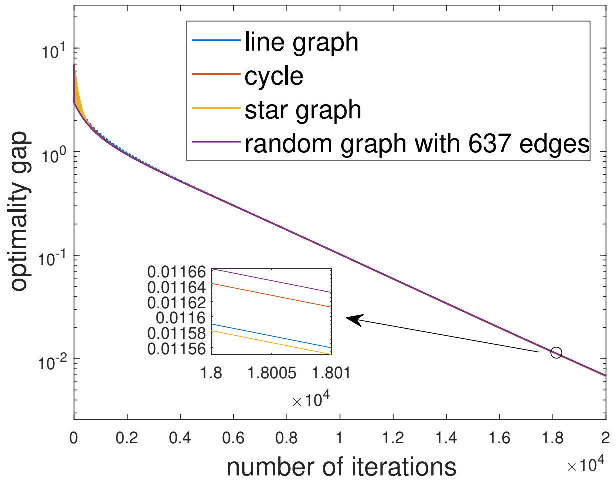


Figure 2.1. Instance of the ABC algorithm on problems of the same ill-conditioned optimization data, but over different graph topologies.

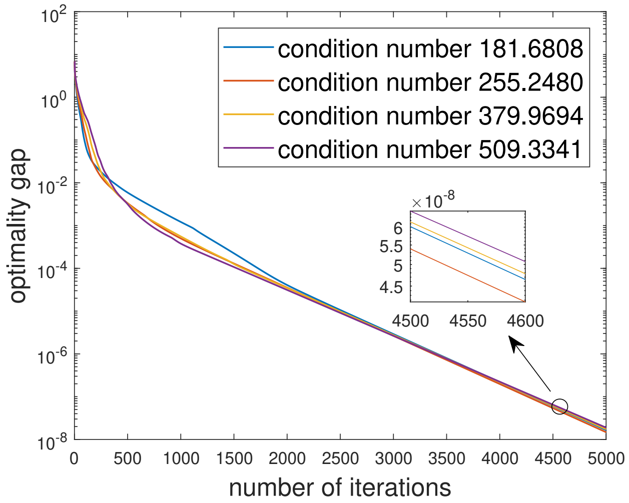


Figure 2.2. Instance of the ABC algorithm on problems over the same line graph, but with optimization data of different condition numbers.

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 (2.19). The network is the Erdos-Renyi model with connection probability of 0.25; in this setting, the max in (2.29) 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 2.1.

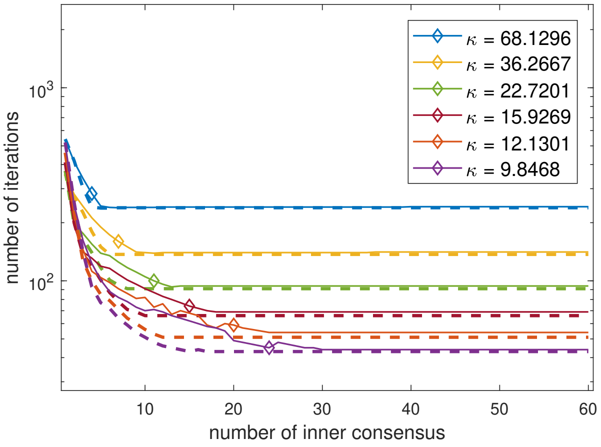


Figure 2.3. Elastic net problem: Number of iterations (gradient evaluations) needed to reach an accuracy of 10^{-8} by Algorithm 2.4 employing Chebyshev acceleration (dashed lines) and multiple rounds of consensus (solid lines).

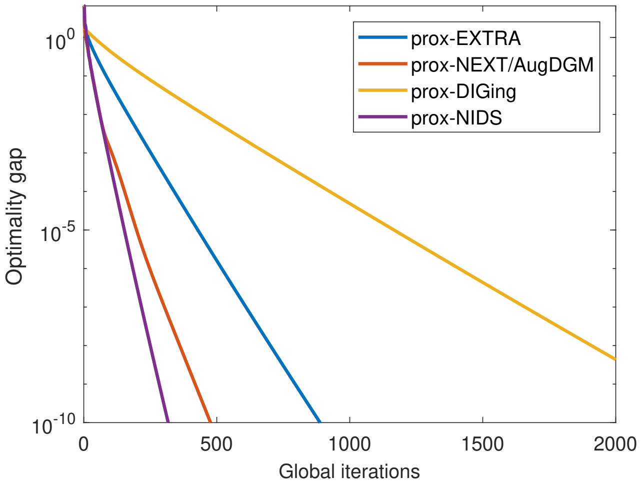


Figure 2.4. Performance comparison of the proximal extensions of some existing algorithms–these extension schemes are all new and are instances of (2.4).

2.7.2 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 [37]:

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

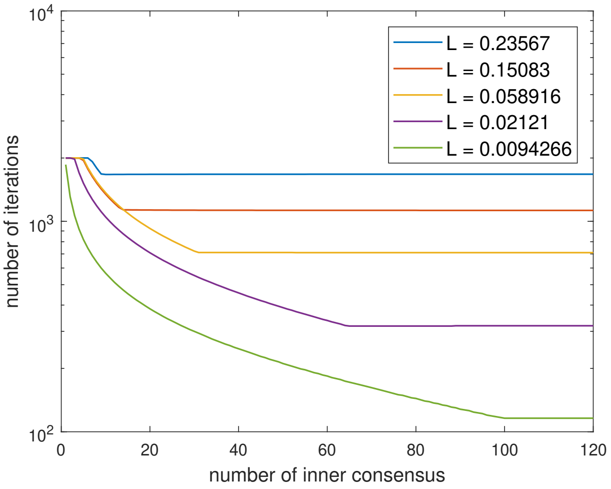


Figure 2.5. Logistic regression problem: Number of iterations (gradient evaluations) needed to reach an accuracy of 10^{-4} by Algorithm 2.23 (equivalently Algorithm 2.35) employing multiple rounds of consensus.

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 $U = [u_1, u_2, \cdots, u_{350}]^\top$ 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 $\alpha^2 L$-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 (2.39). Figure 2.5 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. 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 (2.40) where the two quantities is to be properly balanced with multiple communication steps.

- On the heterogeneity of $f_{\rm i}$'s We exemplify the role of the heterogeneity measure $\left\|\nabla f(X^\star)\right\|$ on the convergence rate, stated in Corollary 25 and Corollary 28. We consider

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

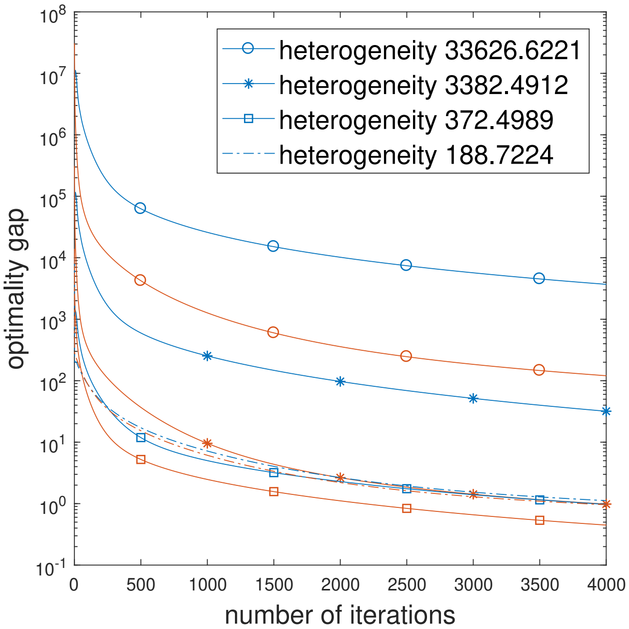


Figure 2.6. Convergence behavior of the ABC algorithm and the centralized gradient descent for problems with different level of heterogeneity (measured by $\left\|\nabla f(X^{\star})\right\|$). The blue curves are associated with the ABC algorithm while the red ones with the centralized gradient descent.

the distributed least squares problem (2.45), 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 $\left\|\nabla f(X^{\star})\right\|$ 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 $A = B = \frac{I+W}{2}$ 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. 2.6, 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.

2.7.3 Other linearly-convergent cases

To corroborate the linear convergent property of the ABC algorithm in the case of $G = 0$, we generate the same setting as Sec. 2.7.1 with the only exception that $\lambda_1 = 0$. The comparison of the performance of several instances of the ABC algorithm is reported in Fig. 2.7. On the other hand, we experiment on the setting as Sec. 2.7.1 with the exception that $\rho = 0$. In this case, although the average function $F(x)$ is strongly convex, each f_i is not strongly convex. We report the convergence behavior of several instances of the ABC algorithm in Fig. 2.8. This result indicates that the ABC algorithm still exhibits linear convergence as long as the average function $F(x)$ is strongly convex.

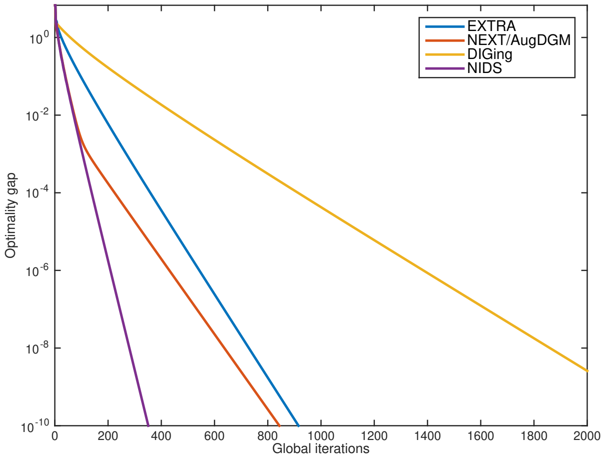


Figure 2.7. Performance comparison of some existing algorithms, which are instances of (2.4) with $G = 0$.

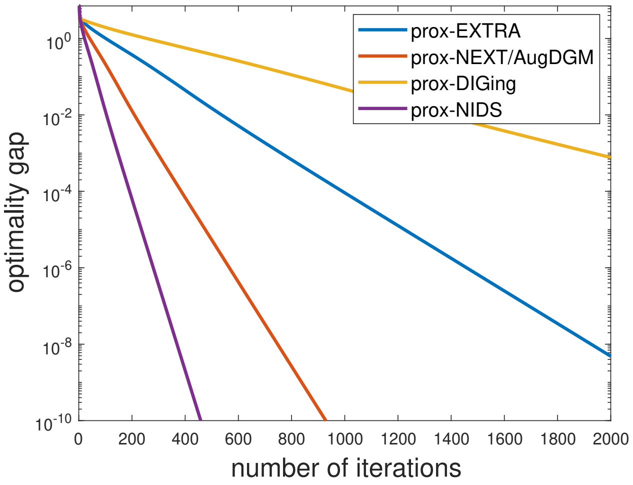


Figure 2.8. Performance comparison of several instances of the ABC algorithm on one problem, wherein the average function $F(x)$ is strongly convex while the individual ones $f_{\mathrm{i}}(x)$'s are not.

2.7.4 Weakly convex cases

We conduct simulations to support Theorem 2.6.5 and Theorem 2.6.7. In particular, for Theorem 2.6.5, we experiment the following two instances of the ABC algorithm:

$$\begin{aligned}
&\text{Algorithm-1:} \qquad A = \frac{I+W}{2}, \quad B = \frac{I+W}{2}, \quad C = \frac{I-W}{2}, \\
&\text{Algorithm-2:} \qquad A = \frac{I+W}{2}, \quad B = I, \quad C = \frac{I-W}{2}
\end{aligned}$$

on the following least squares problem

$$\min_{x\in\mathbb{R}^d} \frac{1}{50}\sum_{\mathrm{i}=1}^{50}\left\|U_{\mathrm{i}}x - v_{\mathrm{i}}\right\|^2. \tag{2.46}$$

where $U_{\mathrm{i}} \in \mathbb{R}^{10\times 200}$ and $v_{\mathrm{i}} \in \mathbb{R}^{10\times 1}$ are the feature vector and labels accessible by node i. We generate the matrix $U = [U_1; U_2; \cdots; U_{50}] \in \mathbb{R}^{500\times 200}$ according to the procedure described in section VII-A in our manuscript with $\omega = 0.8$. Then we set $v = U\hat{x} + \xi$, where each component of $\hat{x}$ and ξ is drawn i.i.d. from the standard normal. The performance of the two algorithmic instances on this problem is shown in Fig 2.9.

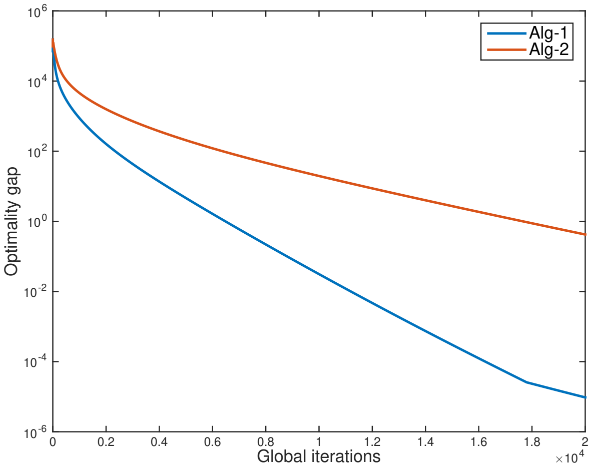


Figure 2.9. Performance comparison of two instances of the ABC algorithm (cf. (34) in the revised manuscript), in solving a weakly convex problem with $G = 0$.

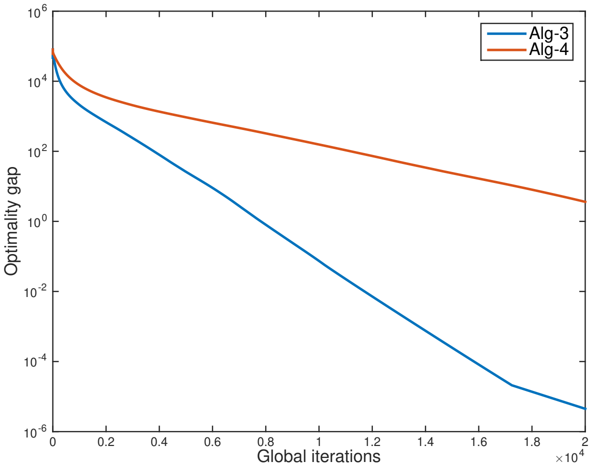


Figure 2.10. Performance comparison of two instances of the ABC algorithm (cf. (40) in the revised manuscript), in solving a weakly convex problem with $G \neq 0$.

For Theorem 2.6.7, we experiment the following two instances of the algorithm given in Section VI-B:

$$\text{Algorithm-3:} \qquad B = I, \quad C = \frac{I - W}{2}, \quad D = \frac{I + W}{2}$$

$$\text{Algorithm-4:} \qquad B = I, \quad C = \frac{I - W}{2}, \quad D = \left(\frac{I + W}{2}\right)^2$$

on the LASSO problem:

$$\min_{x\in\mathbb{R}^d}\left(\frac{1}{50}\sum_{\mathrm{i}=1}^{50}\left\|U_{\mathrm{i}}x-v_{\mathrm{i}}\right\|^2\right)+\left\|x\right\|_1. \tag{2.47}$$

The matrix U is generated in the same way as problem (2.46). Then we generate $\hat{x}$ with sparsity level 0.3 with each nonzero entry drawn i.i.d. from the standard normal. We then set $v = U\hat{x} + \xi$, with each component of ξ drawn i.i.d. from the standard normal. The performance of the two algorithmic instances on this problem is shown in Fig 2.10.

2.8 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.

2.9 Appendix: Proofs of Theorems

2.9.1 Proof of Lemma 2.6.3

Since f is L-smooth, we have

$$\begin{aligned}
&f(X^{k+1}) \\
&\leq f(X^k) + \left\langle \nabla f(X^k), X^{k+1} - X^k \right\rangle + \frac{L}{2}\left\| X^{k+1} - X^k \right\|^2 \\
&\overset{(a)}{\leq} f(X) + \left\langle \nabla f(X^k), X^k - X \right\rangle + \left\langle \nabla f(X^k), X^{k+1} - X^k \right\rangle + \frac{L}{2}\left\| X^{k+1} - X^k \right\|^2 \\
&= f(X) + \left\langle \nabla f(X^k), X^{k+1} - X \right\rangle + \frac{L}{2}\left\| X^{k+1} - X^k \right\|^2.
\end{aligned} \tag{2.48}$$

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 (2.35b) as follows

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

where we have used (2.35c) to obtain the last relation. Now, substituting the above relation into (2.48), we further have

$$\begin{aligned}
f(X^{k+1}) \leq\ & f(X) - \frac{1}{\gamma}\left\langle (I - C)\underline{X}^{k+1}, X^{k+1} - X \right\rangle \\
&+ \frac{1}{\gamma}\left\langle DX^k, X^{k+1} - X \right\rangle - \left\langle \underline{Y}^{k+1}, X^{k+1} - X \right\rangle + \frac{L}{2}\left\| X^{k+1} - X^k \right\|^2
\end{aligned} \tag{2.49}$$

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

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

where we have used (2.35c) to obtain the last relation. Knowing that $X = B\underline{X}$ from (2.35a), we complete the proof.

2.9.2 Proof of Lemma 2.6.4

Invoking Lemma 2.6.3 and using the identity

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

we have that

$$\begin{aligned}
&\phi(X^{k+1},Y)\\
&\leq \phi(X,Y)-\frac{1}{2\gamma}\left(\left\|X^{k+1}-X\right\|_D^2-\left\|X^k-X\right\|_D^2\right)-\frac{1}{\gamma}\left\|\underline{X}^{k+1}\right\|_{B-BC-AB}^2-\left\|X^{k+1}-X^k\right\|_{\frac{1}{2\gamma}D-\frac{L}{2}I}^2\\
&\quad-\frac{\gamma}{2}(\left\|\underline{Y}^{k+1}-Y\right\|_B^2-\left\|\underline{Y}^k-Y\right\|_B^2+\left\|\underline{Y}^{k+1}-\underline{Y}^k\right\|_B^2)\\
&\overset{(a)}{=}\phi(X,Y)-\frac{1}{2\gamma}\left(\left\|X^{k+1}-X\right\|_D^2-\left\|X^k-X\right\|_D^2\right)-\frac{1}{\gamma}\left\|\underline{X}^{k+1}\right\|_{B-\frac{1}{2}BC-AB}^2-\left\|X^{k+1}-X^k\right\|_{\frac{1}{2\gamma}D-\frac{L}{2}I}^2\\
&\quad-\frac{\gamma}{2}\left(\left\|\underline{Y}^{k+1}-Y\right\|_B^2-\left\|\underline{Y}^k-Y\right\|_B^2\right)\\
&\overset{(b)}{\leq}\phi(X,Y)-\frac{1}{2\gamma}\left(\left\|X^{k+1}-X\right\|_D^2-\left\|X^k-X\right\|_D^2\right)-\frac{\gamma}{2}\left(\left\|\underline{Y}^{k+1}-Y\right\|_B^2-\left\|\underline{Y}^k-Y\right\|_B^2\right) \quad (2.50)
\end{aligned}$$

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

Then, averaging (2.50) over k from 0 to $t-1$, we have

$$\begin{aligned}
&\frac{1}{t}\sum_{k=0}^{t-1}\left(\phi(X^{k+1},Y)-\phi(X,Y)\right)\\
&\leq-\frac{1}{2\gamma t}\left(\left\|X^t-X\right\|_D^2-\left\|X^0-X\right\|_D^2\right)-\frac{\gamma}{2t}\left(\left\|\underline{Y}^t-Y\right\|_B^2-\left\|\underline{Y}^0-Y\right\|_B^2\right)\\
&\overset{(a)}{\leq}\frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0-X\right\|_D^2+\gamma\frac{1}{\lambda_2(C)}\left\|Y\right\|_B^2\right) \qquad (2.51)\\
&\overset{(b)}{=}\frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0-X\right\|_D^2+\gamma\frac{1}{\lambda_2(C)}\left\|Y\right\|_{B-J}^2\right)\\
&\leq\frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0-X\right\|_D^2+\gamma\frac{\rho(B-J)}{\lambda_2(C)}\left\|Y\right\|^2\right)
\end{aligned}$$

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

2.9.3 Proof of Theorem 2.5.4

This proof is similar to that of Theorem 2.5.2, except that in the following chain of inequalities, we need to tackle the additional operator T_g. For $\forall\, X, Y \in \mathbb{R}^{2m\times d}$, $X_\ell, Y_\ell \in \operatorname{span}(\sqrt{C})$,

$$
\begin{aligned}
&\left\| T\,X - T\,Y \right\|^2_{\Lambda_{I-C}} \\
&= \left\| T_c \circ T_f \circ T_g \circ T_B\left(X\right) - T_c \circ T_f \circ T_g \circ T_B\left(Y\right) \right\|^2_{\Lambda_{I-C}} \\
&\overset{Lm.\ 2.4.2}{=} \left\| T_f \circ T_g \circ T_B\left(X\right) - T_f \circ T_g \circ T_B\left(Y\right) \right\|^2_{V_{I-C}} \\
&\overset{Lm.\ 2.4.3}{\leq} \left\| T_g \circ T_B\left(X\right) - T_g \circ T_B\left(Y\right) \right\|^2_{\operatorname{diag}(q(D,\gamma)\, D,\, I-C)} \\
&\quad\leq \quad \left\| T_g \circ T_B\left(X\right) - T_g \circ T_B\left(Y\right) \right\|^2_{\operatorname{diag}(q(D,\gamma)\, I,\, I-C)} \\
&\overset{Lm.\ 2.4.4}{\leq} \left\| T_B\left(X\right) - T_B\left(Y\right) \right\|^2_{\operatorname{diag}(q(D,\gamma)\, I,\, I-C)} \\
&\overset{Lm.\ 2.4.5}{=} \left\| X - Y \right\|^2_{\operatorname{diag}(q(D,\gamma)\, B^2,\, I-C)}.
\end{aligned}
$$

To obtain the final result, it remains to notice that: i) for all $(Z)_u \in \mathbb{R}^{m\times d}$,

$$
\begin{aligned}
&\|(Z)_u\|^2_{B^2} = \|(I-C)^{\frac{1}{2}}(Z)_u\|^2_{B^2(I-C)^{-1}} \leq \lambda_{\max}(B^2(I-C)^{-1})\|(I-C)^{\frac{1}{2}}(Z)_u\|^2 \\
&= \lambda_{\max}(B^2(I-C)^{-1})\left\|(Z)_u\right\|^2_{I-C};
\end{aligned}
$$

and ii) $X_\ell, Y_\ell \in \operatorname{span}(\sqrt{C})$.

2.9.4 Proof of Theorem 2.6.7

Algorithm (2.41) reads

$$
\begin{aligned}
\underline{X}^{k+1} &= DX^k - \gamma(\nabla f(X^k) + \underline{Y}^k), \\
X^{k+1} &= \operatorname{prox}_{\gamma g}\left(\underline{X}^{k+1}\right), \\
\underline{Y}^{k+1} &= \underline{Y}^k + \frac{1}{\gamma} C X^{k+1}.
\end{aligned}
\tag{2.52}
$$

The structure of this proof is similar to the proof of Theorem 2.6.5. We first establish two fundamental inequalities that are valid for any pair (X, Y) such that $X \in \operatorname{span}(\mathbf{1})$ and

$Y \in \mathrm{span}(C)$ (cf. Lemma 2.9.1 and Lemma 2.9.2); and then apply these results with $X = X^\star$ and two choices of Y to get the result of the sublinear convergence and rate separation.

Lemma 2.9.1. *Consider the setting of Theorem 2.6.7, let $\{X^k, \underline{X}^k, \underline{Y}^k\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm (2.52) under Assumption 2.6.6. Then for all $X \in span(\mathbf{1})$ and $Y \in span(C)$ it holds*

$$\begin{aligned}\phi(X^{k+1}, Y) \leq \phi(X, Y) + \frac{1}{\gamma}\left\langle X^k - X^{k+1}, X^{k+1} - X\right\rangle_D \\ - \left\langle \underline{Y}^{k+1} - Y, X^{k+1} - X\right\rangle + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2 - \frac{1}{\gamma}\left\|X^{k+1}\right\|^2_{I-C-D}.\end{aligned}$$

Lemma 2.9.2. *Under the same conditions as Lemma 2.9.1, if $\gamma \leq \frac{\lambda_{\min}(D)}{L}$, then for all $X \in span(\mathbf{1})$ and $Y \in span(C)$ it holds*

$$\phi(\widehat{X}^t, Y) - \phi(X, Y) \leq \frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0 - X\right\|^2_D + \gamma\frac{1}{\lambda_2(C)}\left\|Y\right\|^2\right). \tag{2.53}$$

2.9.5 Proof of Lemma 2.9.1

The proof is similar to that of Lemma 2.6.3.

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

According to $X^{k+1} = \operatorname{prox}_{\gamma g}\left(\underline{X}^{k+1}\right)$, we have $g(X^{k+1}) - g(X) \leq \frac{1}{\gamma}\left\langle \underline{X}^{k+1} - X^{k+1}, X^{k+1} - X \right\rangle$. We define $\phi(X, Y) = f(X) + g(X) + \langle X, Y \rangle$. Then we have for $X \in \operatorname{span}(\mathbf{1})$ and $Y \in \operatorname{span}(C)$,

$$\begin{aligned}
&\phi(X^{k+1}, Y) \leq \phi(X, Y) + \frac{1}{\gamma}\left\langle D(X^k - X^{k+1}), X^{k+1} - X \right\rangle \\
&\quad - \left\langle \underline{Y}^{k+1} - Y, X^{k+1} - X \right\rangle + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2 - \frac{1}{\gamma}\left\langle (I - C - D)X^{k+1}, X^{k+1} - X \right\rangle \\
&= \phi(X, Y) + \frac{1}{\gamma}\left\langle X^k - X^{k+1}, X^{k+1} - X \right\rangle_D - \left\langle \underline{Y}^{k+1} - Y, X^{k+1} - X \right\rangle + \frac{L}{2}\left\|X^{k+1} - X^k\right\|^2 \\
&\quad - \frac{1}{\gamma}\left\|X^{k+1}\right\|_{I-C-D}^2
\end{aligned} \tag{2.54}$$

2.9.6 Proof of Lemma 2.9.2

Continuing from (2.54), we have

$$\begin{aligned}
&\phi(X^{k+1}, Y) \leq \phi(X, Y) - \frac{1}{2\gamma}\left(\left\|X^{k+1} - X\right\|_D^2 - \left\|X^k - X\right\|_D^2\right) \\
&\quad - \left\langle \underline{Y}^{k+1} - Y, CX^{k+1} \right\rangle_{(J+C)^{-1}} - \left\|X^{k+1} - X^k\right\|_{\frac{1}{2\gamma}D - \frac{L}{2}}^2 - \frac{1}{\gamma}\left\|X^{k+1}\right\|_{I-C-D} \\
&= \phi(X, Y) - \frac{1}{2\gamma}\left(\left\|X^{k+1} - X\right\|_D^2 - \left\|X^k - X\right\|_D^2\right) \\
&\quad - \frac{\gamma}{2}\left(\left\|\underline{Y}^{k+1} - Y\right\|_{(J+C)^{-1}}^2 - \left\|\underline{Y}^k - Y\right\|_{(J+C)^{-1}}^2\right) - \frac{1}{\gamma}\left\|X^{k+1}\right\|_{I-\frac{C}{2}-D} - \left\|X^{k+1} - X^k\right\|_{\frac{1}{2\gamma}D - \frac{L}{2}}^2 \\
&\leq \phi(X, Y) - \frac{1}{2\gamma}\left(\left\|X^{k+1} - X\right\|_D^2 - \left\|X^k - X\right\|_D^2\right) - \frac{\gamma}{2}\left(\left\|\underline{Y}^{k+1} - Y\right\|_{(J+C)^{-1}}^2 - \left\|\underline{Y}^k - Y\right\|_{(J+C)^{-1}}^2\right),
\end{aligned}$$

where the last step is due to that $I - \frac{C}{2} - D \succeq 0$ and $\gamma \leq \frac{\lambda_{\min}(D)}{L}$. Then, averaging the above over k from 0 to $t-1$, we have

$$\begin{aligned}
&\frac{1}{t}\sum_{k=0}^{t-1}\left(\phi(X^{k+1}, Y) - \phi(X, Y)\right) \\
&\leq -\frac{1}{2\gamma t}\left(\left\|X^t - X\right\|_D^2 - \left\|X^0 - X\right\|_D^2\right) - \frac{\gamma}{2t}\left(\left\|\underline{Y}^t - Y\right\|_{(J+C)^{-1}}^2 - \left\|\underline{Y}^0 - Y\right\|_{(J+C)^{-1}}^2\right) \\
&\leq \frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0 - X\right\|_D^2 + \gamma\frac{1}{\lambda_2(C)}\left\|Y\right\|^2\right).
\end{aligned}$$

Using the convexity of ϕ completes the proof.

For notational simplicity, we set $r(X) = f(X) + g(X)$. From (2.53), we have

$$\begin{aligned}\phi(\widehat{X}^t, Y) - \phi(X^\star, Y) &= r(\widehat{X}^t) - r(X^\star) - \left\langle \widehat{X}^t - X^\star, Y \right\rangle \\ &= r(\widehat{X}^t) - r(X^\star) - \left\langle \widehat{X}^t, Y \right\rangle \leq h(\left\| Y \right\|),\end{aligned} \tag{2.55}$$

where $h(\cdot) = \frac{1}{2t}\left(\frac{1}{\gamma}\left\|X^0 - X^\star\right\|_D^2 + \gamma\frac{1}{\lambda_2(C)}(\cdot)^2\right)$. Now setting $Y = -2\frac{(I-J)\widehat{X}^t}{\|(I-J)\widehat{X}^t\|}\left\|Y^\star\right\|$. The rest of the proof is similar to that in Theorem 2.6.5.

2.9.7 Proof of linear rate under strong convexity of F

Theorem 2.9.3. *Suppose $B^2 \leq I - C$. Let $\kappa_g = \frac{L}{\mu_g}$ with μ_g being the strong convexity of $\frac{1}{m}F$. Then, if $\gamma \leq \min\left\{\frac{2\mu_g(1-\lambda_{m-1}(D))}{\mu_g^2+4L^2},\ \frac{\lambda_{\min}(D)}{L}\right\}$, we have $\left\|T\widetilde{U} - T\widetilde{U}^\star\right\|_{\Lambda_{I-C}}^2 \leq \lambda\left\|\widetilde{U} - \widetilde{U}^\star\right\|_{\Lambda_{I-C}}^2$, with $\lambda = \max\{1 - \gamma\frac{\mu_g}{2},\ I - \lambda_2(C)\}$*

Proof. We prove

$$\left\|DX - \gamma\nabla f(X) - DX^\star + \gamma\nabla f(X^\star)\right\|^2 \leq \left(1 - \gamma\frac{\mu_g}{2}\right)\left\|X - X^\star\right\|^2.$$

The rest of the proof follows the same steps as Theorem 2.5.2. Denote $\bar{X} \triangleq JX$, the following holds:

$$\begin{aligned}&\langle X - X^\star, \nabla f(X) - \nabla f(X^\star)\rangle \\ &= \left\langle \bar{X} - X^\star, \nabla f(X) - \nabla f(\bar{X})\right\rangle + \left\langle \bar{X} - X^\star, \nabla f(\bar{X}) - \nabla f(X^\star)\right\rangle + \left\langle X - \bar{X}, \nabla f(\bar{X}) - \nabla f(X^\star)\right\rangle \\ &\qquad + \left\langle X - \bar{X}, \nabla f(X) - \nabla f(\bar{X})\right\rangle \\ &\geq \mu_g\left\|\bar{X} - X^\star\right\|^2 - 2L\left\|\bar{X} - X^\star\right\|\left\|(I-J)X\right\| \geq (\mu_g - \beta)\left\|\bar{X} - X^\star\right\|^2 - \frac{L^2}{\beta}\left\|(I-J)X\right\|^2,\end{aligned}$$

where we used that the overall function $\bar{f}$ is μ_g-strongly convex to obtain the first inequality and Young's inequality for the second inequality. Setting $\beta = \frac{\mu_g}{2}$ leads to

$$\langle X - X^\star, \nabla f(X) - \nabla f(X^\star)\rangle \geq \frac{\mu_g}{2}\left\|\bar{X} - X^\star\right\|^2 - \frac{2L^2}{\mu_g}\left\|(I-J)X\right\|^2. \tag{2.56}$$

Using the similar steps as in Lemma 2.4.3 and assuming that $\gamma \leq \frac{\lambda_{\min}(D)}{L}$, we have

$$\begin{aligned}
&\left\|DX - \gamma\nabla f(X) - DX^\star + \gamma\nabla f(X^\star)\right\|^2 \\
&\leq \left\|DX - \gamma\nabla f(X) - DX^\star + \gamma\nabla f(X^\star)\right\|^2_{D^{-1}} \\
&\overset{(a)}{\leq} \left\|X - X^\star\right\|^2_D - \gamma\,(2 - \frac{\gamma L}{\lambda_{\min}(D)})\,\langle X - X^\star, \nabla f(X) - \nabla f(X^\star)\rangle \\
&\leq \left\|X - X^\star\right\|^2_D - \gamma\,\langle X - X^\star, \nabla f(X) - \nabla f(X^\star)\rangle \\
&\overset{(2.56)}{\leq} \left\|X - X^\star\right\|^2_D - \gamma\,(\frac{\mu_g}{2}\left\|X - X^\star\right\|^2_J - \frac{2L^2}{\mu_g}\left\|X - X^\star\right\|^2_{I-J}) \\
&= \left(1 - \gamma\,\frac{\mu_g}{2}\right)\left\|X - X^\star\right\|^2 + \gamma\,(\frac{\mu_g}{2} + \frac{2L^2}{\mu_g})\left\|X - X^\star\right\|^2_{I-J} - \left\|X - X^\star\right\|^2_{I-D} \\
&\overset{(b)}{\leq} \left(1 - \gamma\,\frac{\mu_g}{2}\right)\left\|X - X^\star\right\|^2 + \left(\gamma\,(\frac{\mu_g}{2} + \frac{2L^2}{\mu_g}) - 1 + \lambda_{m-1}(D)\right)\left\|X - X^\star\right\|^2_{I-J}
\end{aligned} \tag{2.57}$$

where (a) is due to

$$\begin{aligned}
\left\|\nabla f(X) - \nabla f(X^\star)\right\|^2_{D^{-1}} &\leq \frac{1}{\lambda_{\min}(D)}\left\|\nabla f(X) - \nabla f(X^\star)\right\|^2 \\
&\leq \frac{L}{\lambda_{\min}(D)}\,\langle X - X^\star, \nabla f(X) - \nabla f(X^\star)\rangle\,;
\end{aligned}$$

and (b) is due to $I - D \succeq (1 - \lambda_{m-1}(D))(I - J)$. Setting $\gamma \leq \min\left\{\frac{2\mu_g(1-\lambda_{m-1}(D))}{\mu_g^2+4L^2},\ \frac{\lambda_{\min}(D)}{L}\right\}$ gives the desired result. □

3. AN OPTIMAL DECENTRALIZED ALGORITHM: OPTRA

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

The novel results of this chapter have been published in

- Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari. ”Accelerated primal-dual algorithms for distributed smooth convex optimization over networks.” In International Conference on Artificial Intelligence and Statistics, pp. 2381-2391. PMLR, 2020.

3.1 Introduction

We study distributed smooth convex optimization over a fixed undirected graph:

$$\min_{x\in\mathbb{R}^d} F(x) \triangleq \sum_{\mathrm{i}=1}^{m} f_{\mathrm{i}}(x), \tag{3.1}$$

which is a special instance of Problem (P) with $G = 0$. We assume each $f_{\mathrm{i}} : \mathbb{R}^d \to \mathbb{R}$ to be smooth and convex. The focus of this chapter is on *optimal rate* decentralized algorithms for Problem (3.1) that use only gradient information and gossip communications. By *optimal* we mean that these algorithms provably achieve lower complexity bounds for such a class of problems and oracle decentralized algorithms. We start from the literature review.

3.1.1 Literature Review

Primal [5], [7], [8], [10], [11], [38]–[41] and primal-dual distributed methods [4], [17], [42]–[44] applicable to Problem (3.1) have been extensively studied in the literature, enjoying different convergence rates. In general, these rates are not optimal for several reasons: i) the schemes do not employ any acceleration on the local optimization step and/or communications; or ii) they do not balance optimally the number of optimization and communication steps. Optimal rates of first-order distributed algorithms have been recently studied in [14], [45]–[50] for different classes of optimization problems and network topologies; they however are not optimal or applicable to the formulation considered in this chapter.

Optimal lower complexity bounds and matching distributed algorithms have been recently investigated in [14] for smooth strongly convex functions, in [45] for nonsmooth convex functions, and in [46] for smooth nonconvex functions. Fully connected networks have been considered in [49], [50]. However, to our knowledge, no first-order gossip algorithm is known that achieves *both* computation *and* communication lower complexity bound for the minimization of *smooth convex* functions over graphs. Attempts of designing accelerated distributed algorithms for Problem (3.1) can be found in [47], [51], [52] and briefly discussed next. The scheme in [52] combines the technique of gradient tracking [5], [7], [11] with Nesterov acceleration of local computations and achieves an $\epsilon > 0$ solution in $O\left(1/\epsilon^{5/7}\right)$ gradient and communication steps, under the assumption that the solution set of the optimization problem (3.1) is compact. Algorithm 7 in [47] is designed for general smooth convex objectives; it reaches an ϵ solution in $O\left(\sqrt{L_f/(\eta\,\epsilon)}\ \log 1/\epsilon\right)$ outer loops of communications and $O\left(\sqrt{L_f/\epsilon}\log 1/\epsilon\right)$ inner loops of computations (per communication), resulting in an overall gradient evaluations of $O\left(L_f/(\epsilon\sqrt{\eta})\ \log^2 1/\epsilon\right)$, which do not match existing lower bounds. The subsequent work [51] proposes an accelerated penalty-based method with increasing penalty values; the algorithm achieves the lower bound of $O\left(\sqrt{L_f/\epsilon}\right)$ gradient evaluations but at the cost of an *increasing* number of communications per gradient evaluation (iteration)–namely: $O\left(\sqrt{L_f/\left(\eta\epsilon\right)}\log 1/\epsilon\right)$, which makes it not optimal in terms of communication steps.

3.1.2 Summary of Contributions

We propose a novel family of primal-dual-based distributed algorithms for Problem (3.1) that use *only gradient* information and gossip communications. The algorithms can also employ acceleration on the computation and communications. We provide a unified analysis of their convergence rate, measured in terms of the Bregman distance associated to the saddle point reformation of (3.1). When acceleration on both computation and communications is properly designed, the proposed algorithms are shown to be optimal, in the sense that they match existing complexity lower bounds [51], rewritten in terms of the Bregman distance metric. Furthermore, differently from [14], [47], our algorithms do not require any information on the Fenchel conjugate of the agents' functions, which significantly enlarge the class of functions to which provably optimal rate algorithms can be applied to. Hence, we termed our algorithms OPTRA (*optimal conjugate-free distributed primal-dual methods*) (OPTRA). Our preliminary numerical results show that OPTRA compare favorably with existing distributed accelerated methods [47], [51], [52] proposed for Problem (3.1), which supports our theoretical findings.

Technical novelties. While the genesis of OPTRA finds roots in the primal-dual algorithm [53] and employs Nesterov acceleration similar to [54] (which also builds on [53]), there are some substantial differences between the proposed distributed algorithms and the aforementioned schemes [53], [54], which are briefly discussed next. The scheme in [53] is meant for abstract saddle-point problems and so [54] does; the focus therein is not on distributed optimization. Hence, communications over networks are not explicitly accounted for. Specifically, both [53] and [54] only accelerate the computation but not the communication (networking) component (cf., [53, Alg. 2] and [54, Alg. 2]). On the other hand, OPTRA adopts Nesterov *and* Chebyshev acceleration to balance computation and communication, so that lower complexity bounds on both are achieved (in terms of Bregman distance). This is a major novelty with respect to [53], [54]. Because of these differences, the convergence analysis of OPTRA can not be deduced or easily adapted from that of [53], [54]; a new convergence proof is therefore provided, which shows an explicit dependence of the rate on key network parameters.

3.2 Problem formulation

3.2.1 Distributed optimization over networks

We study Problem (3.1) under the following assumptions.

Assumption 3.2.1. *(i) Each cost function* $f_{\mathrm{i}} : \mathbb{R}^d \to \mathbb{R}$ *is convex and* $L_{f_{\mathrm{i}}}$*-smooth; define* $L_f \triangleq \max_{\mathrm{i}=1}^{m} L_{f_{\mathrm{i}}}$. *(ii) Problem* (3.1) *has a solution.*

Network model Agents are embedded in a communication network, modeled as an undirected graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{V}$ is the set of vertices–the agents–and $\mathcal{E}$ is the set of edges; $\{\mathrm{i}, \mathrm{j}\} \in \mathcal{E}$ if there is a communication link between agent i and agent j. We assume that the graph has no self-loops, that is, $\{\mathrm{i}, \mathrm{i}\} \notin \mathcal{E}$. We use $\mathcal{N}_{\mathrm{i}} \triangleq \{\mathrm{j} | \{\mathrm{i}, \mathrm{j}\} \in \mathcal{E}\}$ to denote the set of neighbors of agent i.

Definition 3.2.1 (Graph Induced Matrix)**.** *The symmetric matrix* $\mathbf{S} = [s_{\mathrm{ij}}] \in \mathbb{R}^{m \times m}$ *is said to be induced by the graph* $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ *if* $s_{\mathrm{ij}} \neq 0$ *only if* $\mathrm{i} = \mathrm{j}$ *or* $\{\mathrm{i}, \mathrm{j}\} \in \mathcal{E}$. *The set of such matrices is denoted by* $\mathcal{W}_{\mathcal{G}}$.

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

Assumption 3.2.2. *Given the graph $\mathcal{G}$, the gossip matrix $L \in \mathbb{R}^{m\times m}$ satisfies:*

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

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

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

where $\{\lambda_{\mathrm{i}}\}_{\mathrm{i}=1}^m$ *are the eigenvalues of* L*.*

It is not difficult to check a gossip matrix satisfying Assumption 3.2.2 always exists if the associated graph is connected; see, e.g., [55]. Several gossip matrices have been considered in the literature; we refer the reader to [36], [56] and references therein for specific examples.

3.2.2 Saddle-point reformulation

A standard approach for solving (3.1) consists in rewriting the optimization problem in the so-called consensus optimization form, that is

$$\min_{X\in\mathbb{R}^{m\times d}} f(X) + \iota_{\mathcal{C}}(X), \tag{3.2}$$

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

To solve Problem (3.2), we consider the following closely related saddle point formulation

$$\max_{Y\in\mathbb{R}^{m\times d}} \min_{X\in\mathbb{R}^{m\times d}} \Phi(X,Y) \triangleq f(X) + \langle Y, X\rangle - \iota_{\mathcal{C}^\perp}(Y), \tag{3.3}$$

where $\mathcal{C}^\perp$ is the space orthogonal to $\mathcal{C}$ and $\Phi(X,Y)$ is the Lagrangian associated to problem (3.2). By Assumption 3.2.1, strong duality holds for (3.3); hence, (3.3) admits a primal-dual optimal solution pair $(X^\star, Y^\star) \in \mathcal{D} \triangleq \mathbb{R}^{m\times d} \times \mathcal{C}^\perp$ that satisfies the following KKT conditions

$$\text{(Lagrangian Optimality)} \qquad Y^\star = -\nabla f(X^\star), \tag{3.4a}$$

$$\text{(Primal Feasibility)} \qquad X^\star \in \mathcal{C}, \tag{3.4b}$$

and the saddle-point property $\Phi(X^\star, Y) \leq \Phi(X^\star, Y^\star) \leq \Phi(X, Y^\star)$, for all $(X, Y) \in \mathcal{D}$. Note that $X^\star$ solves Problem (3.2) and thus it is also a solution of the original formulation (3.1).

Using (3.3) and (3.4), one can write

$$\begin{aligned} \Phi(X, Y^\star) - \Phi(X^\star, Y) &= f(X) + \langle Y^\star, X\rangle - f(X^\star) - \langle Y, X^\star\rangle \\ &\overset{(3.4)}{=} f(X) - f(X^\star) - \langle \nabla f(X^\star), X - X^\star\rangle \triangleq G(X, X^\star) \geq 0, \end{aligned} \tag{3.5}$$

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

Proposition 3.2.1. *Let $X^\star$ be any optimal solution of (3.2); the following hold for G defined in (3.5):*

(a) $\bar{X}$ is an optimal solution of (3.2) if and only if $\bar{X} \in \mathcal{C}$ and $G(\bar{X}, X^\star) = 0$;

(b) $G(X, \bullet)$ is constant over the solution set of (3.2).

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

Remark 3.2.3. *We will use G as metric to assess the (worst-case) convergence rate of the proposed algorithms as well as to state lower complexity bounds. Note that, since f is not assumed to be strictly convex, $G(X) = 0$ does not imply $X = X^\star$, but it is only a necessary condition for X to be optimal (cf. Proposition 3.2.1(a)). Still, G is a valid merit function for both purposes above, as explained next. First, $G(X) > \epsilon$ implies that X is ϵ "far" away (in the G-measure) from any optimal solution of (3.2); hence, a lower bound in terms of G is an informative measure. Furthermore, when it comes to the convergence rate analysis of distributed algorithms, Proposition 3.2.1-(a) legitimates the use of (the decay rate of) G along the agents' iterates $\{X^k\}_{k=0}^\infty$, as the distance of X^k from $\mathcal{C}$ is proved to be vanishing–see Sec. 3.4.*

3.3 Preliminaries: Lower Complexity Bounds

To benchmark the distributed algorithms to be introduced, we recall here existing lower complexity bounds for decentralized first-oder schemes belonging to the same oracle class of the proposed algorithms. The difference from the literature is that we will write such bounds in terms of the Bregman distance G. We begin introducing the distributed oracle model (cf. Sec. 3.3.1), followed by the lower complexity bound (cf. Sec. 3.3.2).

3.3.1 Decentralized first-order oracle

Given Problem (3.1) over the graph $\mathcal{G}$, we consider distributed algorithms wherein each agent i controls a local variable $x_{\mathrm{i}} \in \mathbb{R}^d$, which is an estimate of the shared optimization variable x in (3.1). The value of x_{i} at (continuous) time $t \in \mathbb{R}_+$ is denoted by $x_{\mathrm{i}}^{(t)}$. To update its own variable, each agent i: 1) has access to the gradient of its own function–we assume that the time to inquire such a gradient is normalized to one; and 2) can communicate values (vectors in $\mathbb{R}^d$) to (some of) its neighbors $\mathrm{j} \in \mathcal{T}_{\mathrm{i}}$–this communication requires a time $\tau_c \in \mathbb{R}_+$ (which may be smaller or greater than one). Each update $x_{\mathrm{i}}^{(t)}$ is generated according to the following general *black-box procedure.*

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

$$x_{\mathrm{i}}^{(t)} \in \underbrace{\mathrm{span}(x_{\mathrm{j}}^{(s)} \,|\, \mathrm{j} \in \mathcal{N}_{\mathrm{i}} \ \text{ and } \ 0 \leq s < t - \tau_c)}_{\text{local communication}} + \underbrace{\mathrm{span}(x_{\mathrm{i}}^{(s)}, \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{(s)}) \,|\, 0 \leq s < t-1)}_{\text{local computation}}, \tag{3.6}$$

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

The oracle (3.6) allows each agent to use all the historical values of its local gradients (local computations) as well as the historical values of the decision variables received from its neighbors (local communications). Furthermore, (3.6) also captures algorithms employing multiple rounds of communications (resp. gradient computations) per gradient evaluation (resp. communication). In the supporting material (Appendix 3.4.2), we show that, in fact, the above oracle accounts for most existing distributed algorithms, such as primal-dual methods [4] as well as gradient tracking methods [5], [7], [8], [11].

A similar black-box procedure has been introduced in [14] for strongly convex instances of (3.1). The difference with [14] is that the oracle in (3.6) cannot return the gradient of the conjugate of the f_{i}'s. The reason of considering such "less powerful" methods is that, in practice, it is hard to compute the gradient of conjugate functions. This means that the gossip (dual-based) methods in [14] do not belong to the oracle considered in this chapter.

3.3.2 Lower complexity bounds

We state now lower complexity bounds in the G-metric for the class of algorithms $\mathcal{A}$ applied to Problem (3.2) [and thus (3.1)] over a connected graph $\mathcal{G}$. In Section 3.4 we will introduce a primal-dual distributed algorithm that indeed converges to an optimal solution of (3.2) driving G to zero at a rate that matches the lower complexity bound. Proofs of the results are available as supporting material.

Theorem 3.3.1. *Consider Problem (3.1) under Assumption 3.2.1 and let $\mathcal{G}$ be a connected graph. For any given $\eta \in (0, 1]$ and $L_f > 0$, there exists a gossip matrix $L \in \mathcal{W}_{\mathcal{G}}$ with eigengap $\eta \triangleq \frac{\lambda_2(L)}{\lambda_m(L)}$, and a set of local cost functions $\{f_{\mathrm{i}}\}_{\mathrm{i}=0}^{m}$, $f_{\mathrm{i}} : \mathbb{R}^d \to \mathbb{R}$, with $f(X) = \sum_{\mathrm{i}} f_{\mathrm{i}}(x_{\mathrm{i}})$ being L_f-smooth such that, for any first-order gossip algorithm in $\mathcal{A}$ using L, we have*

$$G\Big(X^{(t)}\Big) = \Omega\left(\frac{L_f R^2}{\Big(\frac{t}{1+\left\lceil\frac{1}{5\sqrt{\eta}}\right\rceil\tau_c}+2\Big)^2} + \frac{R\Big\|\nabla f(X^\star)\Big\|}{\frac{t}{1+\left\lceil\frac{1}{5\sqrt{\eta}}\right\rceil\tau_c}+2}\right), \tag{3.7}$$

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

$$\frac{L_f R^2}{t \,\Big/ \Big(1+\left\lceil\frac{1}{5\sqrt{\eta}}\right\rceil\tau_c\Big)} = \Theta\left(R\Big\|\nabla f(X^\star)\Big\|\right). \tag{3.8}$$

Corollary 3.3.1.1. *In the setting of Theorem 3.3.1, the overall time needed by any first-order algorithm in $\mathcal{A}$ using the gossip matrix L to drive G below $\epsilon > 0$, with f given in Theorem 3.3.1, is*

$$\Omega\left(\left(1+\frac{1}{\sqrt{\eta}}\tau_c\right)\left(\sqrt{\frac{L_f R^2}{\epsilon}} + \frac{R\Big\|\nabla f(X^\star)\Big\|}{\epsilon}\right)\right). \tag{3.9}$$

Notice that, because of (3.8), the lower bound (3.9) can be equivalently stated as

$$\Omega\left(\left(1+\frac{1}{\sqrt{\eta}}\tau_c\right)\sqrt{\frac{L_f R^2}{\epsilon}}\right). \tag{3.10}$$

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

$$\max_{\mathrm{i}\in\mathcal{V}}(F(x_{\mathrm{i}}) - \min_{x\in\mathbb{R}^d} F(x)) \tag{3.11}$$

has the same expression as (3.7) [and thus (3.9) and (3.10)] up to some constants. This observation is also reported in [51] without proof, and stated formally below for completeness (the proof can be found in the supporting material).

Theorem 3.3.2 (Lower bound on the objective-error)**.** *In the setting of Theorem 3.3.1, the overall time needed by any first-order algorithm in $\mathcal{A}$ using the gossip matrix L to drive the objective-error-based metric, $\max_{\mathrm{i}\in\mathcal{V}}(F(x_{\mathrm{i}}) - \min_{x\in\mathbb{R}^d} F(x))$, below $\epsilon > 0$, with f given in Theorem 3.3.1, is given by (3.9) [or, equivalently, by (3.10)].*

Remark 3.3.3 (Balancing computations & communications)**.** *The above lower bounds tell us that one cannot reach an ϵ-solution of* (3.2) *(measured either in terms of the G or FEM-metrics) in less than $O\left(\sqrt{L_f R^2/\epsilon} + R\|\nabla f(X^\star)\|/\epsilon\right)$ computing time and $O(\tau_c/\sqrt{\eta}\cdot(\sqrt{L_f R^2/\epsilon} + R\|\nabla f(X^\star)\|/\epsilon))$ communication time. Since the time for a single gradient evaluation has been normalized to one, the former lower bound corresponds also to the overall number of gradient evaluations while the overall communication steps read*

$$\Omega\left(1/\sqrt{\eta}\cdot\left(\sqrt{L_f R^2/\epsilon} + R\|\nabla f(X^\star)\|/\epsilon\right)\right).$$

This sheds light also on the optimal balance between computation and communication: the optimal number of communication steps per gradient evaluations is $\lceil 1/\sqrt{\eta}\rceil$. In the next section, we introduce a distributed, gossip-based algorithm that achieves lower complexity bounds in the G-metric.

3.4 Distributed primal-dual algorithms

3.4.1 A general primal-dual scheme

A gamut of primal-dual algorithms has been proposed in the literature to solve Problem (3.2) in a centralized setting; see, e.g., [53], [57] and references therein for details. Building on [53], [57], here, we propose a general primal-dual algorithm to solve the saddle point problem (3.3) in a *distributed* manner. The algorithm reads: given X^k and Y^k at iteration $k \in \mathbb{T}_+$,

$$X^{k+1} = A(X^k - \gamma(\nabla f(X^k) + \hat{Y}^k)), \tag{3.12a}$$

$$Y^{k+1} = Y^k + \tau B X^{k+1}, \tag{3.12b}$$

$$\hat{Y}^{k+1} = Y^{k+1} + \beta(Y^{k+1} - Y^k), \tag{3.12c}$$

where Y^k is the dual vector variable; γ and τ are the primal and dual step-sizes common to all the agents; $\beta \in [-1\ 1]$ is a free parameter to be determined; and $A, B \in \mathbb{R}^{m\times m}$ satisfy the following assumption.

Assumption 3.4.1. *The weight matrices A, B in (3.12) are such that*

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

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

Remark 3.4.2. *Several choices for A and B satisfying Assumption 3.4.1 are possible, resulting in a gamut of specific algorithms, obtained as instances of (3.12). Note that, when A and B satisfy also Assumption 3.2.2, all these algorithms are implementable over the graph $\mathcal{G}$. Several examples of such distributed algorithms are discussed in details in Appendix 3.4.2. Here, we only mention that the gradient tracking methods [5], [7], [8], [11] and primal-dual methods, such as EXTRA [4], are all special cases of (3.12); the former schemes are obtained setting $A = W^2$ and $B = (I - W)^2$, where $W \in \mathcal{W}_\mathcal{G}$ is the weight matrix used by the agents to employ the (perturbed) consensus step; and EXTRA is obtained setting $A = W$*

and $B = I - W$. We begin studying convergence of the general primal-dual algorithm (3.12), under the following tuning of the free parameters:

$$\gamma = \frac{\nu}{\nu L_f + 1}, \quad \tau = \frac{1}{\nu \lambda_m(B)}, \quad (1 - \gamma L_f)I - \gamma\tau B \succeq 0, \tag{3.13}$$

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

Theorem 3.4.3. *Consider Problem (3.1) under Assumption 3.2.1. Given (X^0, Y^0), let $\{(X^k, Y^k)\}_{k=1}^{\infty}$ be the sequence generated by the algorithm in (3.12), under Assumption 3.4.1 and the setting in (3.13). Define $\bar{X}^k \triangleq \frac{1}{k-1}\sum_{t=2}^{k} X_t$ and $R \triangleq \|X^1 - X^\star\|$, Then, the following hold: (i) $\{X^k\}_{k=0}^{\infty}$ converges to an optimal solution $X^\star$ of (3.2) [thus $X^\star = \mathbf{1}x^\star$, with $x^\star$ being optimal for (3.1)]; therefore $\lim_{k\to\infty} G(X^k) = 0$; and (ii) the number of iterations needed for $G(\bar{X}^k)$ to go below $\epsilon > 0$ is*

$$O\left(\frac{L_f R^2}{\epsilon} + \frac{1}{\sqrt{\eta(B)}} \frac{R\|\nabla f(X^\star)\|}{\epsilon}\right). \tag{3.14}$$

The proof of the theorem can be found in the supporting material. Note that the convergence rate (3.14) does not match the lower bound given in Theorem 3.3.1. For instance, consider as concrete example the choice $A = I - L$ and $B = L$; and let $\tau_c \in \mathbb{R}_+$ (resp. 1) be the time for each agent to perform a single communication to its neighbors (resp. gradient evaluation). The time complexity of the primal-dual algorithm (3.12) becomes

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

To match the lower lower bound given in Theorem 3.3.1, our next step is accelerating the algorithm, both the computational part and the communication step; we leverage Nesterov acceleration [29] for the optimization step while employ Chebyshev polynomials [31] to accelerate communications. To provide some insight of our construction, we begin with the former acceleration; the latter is added in Section 3.4.4.

3.4.2 Review of existing distributed algorithms and their connections

This section shows the generality of the first-order oracle $\mathcal{A}$ in (3.6) and the proposed distributed primal-dual algorithmic framework (3.12) by casting several existing distributed algorithms in the oracle form (3.6) and algorithmic form (3.12).

- Some distributed optimization methods

Distributed gradient methods One of the first distributed algorithms for Problem (3.1) was proposed in the seminal work [58] and called Distributed Gradient Algorithm (DGD). DGD employing constant step-size can be written in compact form as:

$$X^{k+1} = WX^k - \gamma \nabla f(X^k), \tag{3.15}$$

where $W \in \mathcal{W}_{\mathcal{G}}$. Defining $X^{(t_k)} = X^k$, DGD can be rewritten in a piece-wise continuous form as

$$\begin{aligned} X^{(t_{k+1})} &= WX^{(t_k)} - \gamma \nabla f(X^{(t_k)}), \\ X^{(t)} &= X^{(t_k)},\ t_k \leq t < t_{k+1}, \end{aligned} \tag{3.16}$$

which is an instance of the oracle $\mathcal{A}$.

Distributed gradient tracking methods The distributed gradient tracking algorithm, first proposed in [5], [7] and further analyzed in [8], [11], reads

$$X^{k+1} = WX^k - \gamma Y^k \tag{3.17a}$$

$$Y^{k+1} = WY^k + \nabla f(X^{k+1}) - \nabla f(X^k) \tag{3.17b}$$

where Y_k is an auxiliary variable aiming at tracking the gradient of the sum-cost function. The above algorithm is proved to converge at linear rate to a solution of Problem (3.2), under proper conditions on the stepsize γ. To show its relationship to the oracle, we first rewrite (3.17) absorbing the tracking variable Y, which yields

$$X^{k+2} = 2WX^{k+1} - W^2X^k - \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)),$$

with $X^1 = WX^0 - \gamma\nabla f(X^0)$. It is clear that the gradient tracking algorithm belongs to the oracle $\mathcal{S}$, as each iteration k only involves the historical neighboring information and local gradients at $k-1$ and $k-2$.

Distributed primal-dual methods Distributed primal-dual algorithms can be generally written in the following form [4]

$$X^{k+1} = WX^k - \gamma\nabla f(X^k) - Y^k \tag{3.18a}$$

$$Y^{k+1} = Y^k + (I - W)X^{k+1} \tag{3.18b}$$

where Y_k is the dual variable. When $Y^0 = 0$, the algorithm 3.18 can solve problem (3.2). Evaluating (3.18a) at $k+1$ and substituting it into (3.18b) yields

$$X^{k+2} = 2WX^{k+1} - WX^k - \gamma(\nabla f(X^{k+1}) - \nabla f(X^k)), \tag{3.19}$$

with $X^1 = WX^0 - \gamma W\nabla f(X^0)$. It is easy to check that (3.19) belongs to the oracle $\mathcal{A}$.

Remark 3.4.4. *There are some other distributed algorithms that do not belong to the categories above such as [59]. However, using similar arguments as above, one can show that they are instances of the oracle $\mathcal{A}$.*

- Connections between gradient tracking and primal-dual methods

We reveal here an unknown interesting connection between primal-dual methods and gradient tracking based methods. More specifically, setting in (3.12a) $A = W^2$ and $B = (\mathbf{I} - \mathbf{W})^2$, one can easily recover gradient tracking methods from the primal-dual ones. To simplify the presentation, we consider a slightly different form of (3.12a), that is

$$X^{k+1} = W^2(X^k - \gamma(\nabla f(X^k)) + (I - W)Y^k, \tag{3.20a}$$

$$Y^{k+1} = Y^k + (I - W)X^{k+1}. \tag{3.20b}$$

Then, from (3.20a), we have at iteration $k+1$

$$X^{k+2} = W^2X^{k+1} - \gamma W^2\nabla f(X^{k+1}) - (I - W)Y^{k+1}$$

Subtracting (3.20a) from the above equation we have

$$\begin{aligned} X^{k+2}-X^{k+1} &= W^2X^{k+1}-W^2X^k-\gamma W^2(\nabla f(X^{k+1})-\nabla f(X^k))-(I-W)(Y^{k+1}-Y^k) \\ &= W^2X^{k+1}-W^2X^k-\gamma W^2(\nabla f(X^{k+1})-\nabla f(X^k))-(I-W)^2X^{k+1} \\ &= 2WX^{k+1}-X^{k+1}-W^2X^k-\gamma W^2(\nabla f(X^{k+1})-\nabla f(X^k)). \end{aligned} \tag{3.21}$$

Rearranging terms leads to

$$X^{k+2}-WX^{k+1}=W(X^{k+1}-WX^k)-\gamma W^2(\nabla f(X^{k+1})-\nabla f(X^k)).$$

Let $-\gamma WY^k=X^{k+1}-WX^k$ and suppose W is invertible. Then, we have

$$X^{k+1}=W(X^k-\gamma Y^k) \tag{3.22a}$$

$$Y^{k+1}=W(Y^k+\nabla f(X^{k+1})-\nabla f(X^k)) \tag{3.22b}$$

which is exactly the standard gradient tracking method in the ATC form [5], [7].

3.4.3 Nesterov-based accelerated primal-dual algorithms

We accelerate the primal-dual algorithm (3.12) as follows:

$$u^{k+1}=A(X^k-\gamma(\nabla f(X^k)+\hat{Y}^k)), \tag{3.23a}$$

$$X^{k+1}=u^{k+1}+\alpha_k(u^{k+1}-u^k), \tag{3.23b}$$

$$\hat{X}^{k+1}=\sigma_kX^{k+1}+(1-\sigma_k)u^{k+1} \tag{3.23c}$$

$$Y^{k+1}=Y^k+\tau_kB\hat{X}^{k+1}, \tag{3.23d}$$

$$\hat{Y}^{k+1}=Y^{k+1}+\beta_k(Y^{k+1}-Y^k), \tag{3.23e}$$

where $u^k,\hat{X}^k,\hat{Y}^k$ are auxiliary variables and $\alpha_k,\sigma_k,\tau_k,\beta_k$ are parameters to be properly chosen. Roughly speaking, (3.23a), (3.23d) and (3.23e) are the standard primal-dual steps while (3.23b) and (3.23c) are the extra steps meant for the acceleration, with (3.23b) being the standard Nesterov momentum step and (3.23c) being a correction step. Note that setting

$\alpha_k \equiv 0, \sigma_k \equiv 1, \tau_k \equiv \tau, \beta_k \equiv 1$, the algorithm reduces to the primal-dual method (3.12). We provide next an instance of (3.23) that is suitable for a distributed implementation.

Choose the free parameters in (3.23) as follows: denoting by $T \in \mathbb{T}_+$ the total number of iterations k performed by the algorithm, set

$$\begin{aligned} &A = I - L/\lambda_m(L),\ B = L/\lambda_m(L),\ \gamma = \frac{\nu}{\nu L_f + T}, \\ &\tau = \frac{1}{\nu T\, \lambda_m(B)}\ , \frac{1}{\theta_k} = \frac{1+\sqrt{1+4(\frac{1}{\theta_{k-1}})^2}}{2} \text{ with } \theta_1 = 1, \\ &\sigma_k = \frac{1}{\theta_{k+1}},\ \alpha_k = \frac{\theta_{k+1}}{\theta_k} - \theta_{k+1},\ \beta_k = \frac{\tau_{k+1}}{\tau_k},\ \tau_k = \frac{\tau}{\theta_k}. \end{aligned} \tag{3.24}$$

The resulting scheme is summarized in Algorithm 1, and its convergence properties are stated in Theorem 3.4.5. We point out that Theorem 3.4.5, although stated for Algorithm 1, can be readily extended to the more general accelerated primal-dual scheme (3.23), with other choices of A and B just satisfying Assumption 3.4.1.

Algorithm 1 OPTRA-N

```
Input: number of iterations $T$, Laplacian matrix $L$, parameter $\nu = \sqrt{\eta(B)}$
Output: $(u^T, Y^T)$
Initialization: $y_{\mathrm{i}}^1 = 0, \forall \mathrm{i} \in \mathcal{V}$ and $\theta_1 = 1$
$\hat{Y}^1 = \tau_1 B X^1$, $u^1 = X^1$
for $k = 1, 2, ..., T$ do
    compute $\theta_k$ according to (3.24),
    for $\forall \mathrm{i} \in \mathcal{V}$ do in parallel
        compute the next iterate according to (3.23), using the tuning as in (3.24),
Return $(u^T, Y^T)$
```

Theorem 3.4.5. *Consider Problem* (3.1) *under Assumption 3.2.1; let $u^{(t)}$ be the value of the u-vector generated by Algorithm 1 at time $t \in \mathbb{R}_+$, under Assumptions 3.2.2 and 3.4.1, and the parameter setting in* (3.24)*. Then, we have*

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

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

$$G(u^{(t)}) = O\left(\frac{L_f R^2}{\left(\frac{t}{1+\tau_c}\right)^2} + \frac{R\left\|\nabla f(X^\star)\right\|}{\sqrt{\eta}\frac{t}{1+\tau_c}}\right). \tag{3.25}$$

Furthermore, the consensus error decays at

$$\left\|\left(I - \frac{\mathbf{1}\mathbf{1}^T}{m}\right)u^{(t)}\right\| = O\left(\frac{L_f R^2}{\left\|\nabla f(X^\star)\right\|\left(\frac{t}{1+\tau_c}\right)^2} + \frac{R^2 + \left\|\nabla f(X^\star)\right\|^2}{\left\|\nabla f(X^\star)\right\|\sqrt{\eta}\frac{t}{1+\tau_c}}\right). \tag{3.26}$$

While the convergence time of Algorithm 1 benefits from the Nesterov acceleration of the computation step, it is not optimal in terms of communications (optimal dependence on η). In fact, when the network is poorly connected, the second term on the RHS of (3.25) becomes dominant with respect to the first one, and (3.25) overall will be larger than (3.7). This is due to the fact that Algorithm 1 performs a one-consensus-one-gradient update while the lower bound shows an optimal ratio of $\lceil 1/\sqrt{\eta}\rceil$ (cf. Remark 3.3.3). This optimal ratio can be achieved accelerating also the communication step, as described in the next section.

3.4.4 Optimal primal-dual algorithms with Chebyshev acceleration

We employ the acceleration of the communication step in Algorithm 1 by replacing the gossip matrix L by $P_K(L)$, where $P_K(\cdot)$ is a polynomial of degree at most K that maximizes the eigengap of $P_K(L)$, for a fixed K. This leads to a widely used acceleration scheme known as Chebyshev acceleration and the choice $P_K(x) = 1 - T_K(c_1(1-x))/T_K(c_1)$, with $c_1 = (1+\eta(L))/(1-\eta(L))$ and $T_K(\cdot)$, are the Chebyshev polynomials [31]. It is not difficult to check that such a $P_K(L)$ is still a gossip matrix. Using in (3.23) the following setting:

$$\begin{gathered} A = I - c_2 P_K(L),\ B = P_K(L),\ K = \left\lceil 1/\sqrt{\eta(L)}\right\rceil, \text{ with} \\ c_2 = \left(1 + 2\frac{c_0^K}{(1+c_0^{2K})}\right)^{-1},\ c_0 = \frac{1-\sqrt{\eta(L)}}{1+\sqrt{\eta(L)}}, \end{gathered} \tag{3.27}$$

Algorithm 2 OPTRA

```
Input: number of iterations $T$, Laplacian matrix $\widetilde{L}$, number of inner consensus $K = \left\lceil \frac{1}{\sqrt{\eta(L)}} \right\rceil$,
$c_0 = \frac{1-\sqrt{\eta(\widetilde{L})}}{1+\sqrt{\eta(\widetilde{L})}}$, $c_1 = \frac{1+\eta(\widetilde{L})}{1-\eta(\widetilde{L})}$, $c_2 = 1/\left(1+2\frac{c_0^K}{1+c_0^{2K}}\right)$, $\tau = \frac{c_2}{\nu T}$, $\gamma = \frac{\nu}{\nu L_f+T}$, $\nu = 1$.
Initialization: $Y^0 = 0$;        Preprocessing: $L = \frac{2}{\lambda_2(\widetilde{L})+\lambda_n(\widetilde{L})}\widetilde{L}$.
Output: $(u^T, Y^T)$
$\hat{Y}^1 = \tau_1 \cdot \textsc{AcceleratedGossip}(X^1, L, K)$, $u^1 = X^1$
for $k = 1, 2, ..., T$ do
    $u^{k+\frac{1}{2}} = X^k - \gamma\left(\nabla f(X^k) + \hat{Y}^k\right)$,
    $u^{k+1} = u^{k+\frac{1}{2}} - c_2 \cdot \textsc{AcceleratedGossip}(u^{k+\frac{1}{2}}, L, K)$,
    $X^{k+1} = u^{k+1} + \left(\frac{\theta_{k+1}}{\theta_k} - \theta_{k+1}\right)(u^{k+1} - u^k)$,
    $\hat{X}^{k+1} = \frac{1}{\theta_{k+1}}X^{k+1} + \left(1 - \frac{1}{\theta_{k+1}}\right)u^{k+1}$,
    $Y^{k+1} = Y^k + \frac{\tau}{\theta_k}\textsc{AcceleratedGossip}(\hat{X}^{k+1}, L, K)$,
    $\hat{Y}^{k+1} = Y^{k+1} + \frac{\theta_k}{\theta_{k+1}}(Y^{k+1} - Y^k)$,
Return $(u^T, Y^T)$.

procedure $\textsc{AcceleratedGossip}(X, L, K)$
    $a_0 = 1, a_1 = c_1$
    $Z_0 = X, Z_1 = c_1(I - L)X$
    for $k = 1$ to $K - 1$ do
        $a_{k+1} = 2c_1 a_k - a_{k-1}$
        $Z_{k+1} = 2c_1(I - L)Z_k - Z_{k-1}$
    return $Z_0 - \frac{Z_K}{a_K}$
```

leads to the distributed scheme described in Algorithm 2, whose convergence rate achieves the lower bound (3.9), as proved in Theorem 3.4.6 below. Note that, although the idea of using Chebyshev polynomial has been used in some (centralized and distributed) algorithms in the literature [14], [31], Algorithm 2 substantially differs from existing schemes. Furtheremore, [14], [31] are not rate optimal in the distributed setting considered in this work.

Theorem 3.4.6. *Consider Problem* (3.1) *under Assumption 3.2.1; let $u^{(t)}$ be the value of the u-vector generated by algorithm 2 at time $t \in \mathbb{R}_+$, under Assumptions 3.2.2 and 3.4.1,* *the parameter setting in* (3.24), *and employing the Chebyshev acceleration* (3.27). *Then, the following hold:*

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

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

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

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

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

According to Theorem 3.4.6, given $\epsilon > 0$, the time needed by the algorithm to drive G below $\epsilon > 0$ is

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

which matches the lower complexity bound given in (3.9). Note that the optimality is stated in terms of the G-metric and does not imply that the algorithm is rate optimal also in the FEM-metric (3.11), which to date remains an open question. In our experiments (cf. Sec. 5.4) we observed i) the similar behavior of these two errors measured in different metrics as a function of the total number of computations and communications; and ii) that Algorithm 2 in fact outperforms existing distributed schemes.

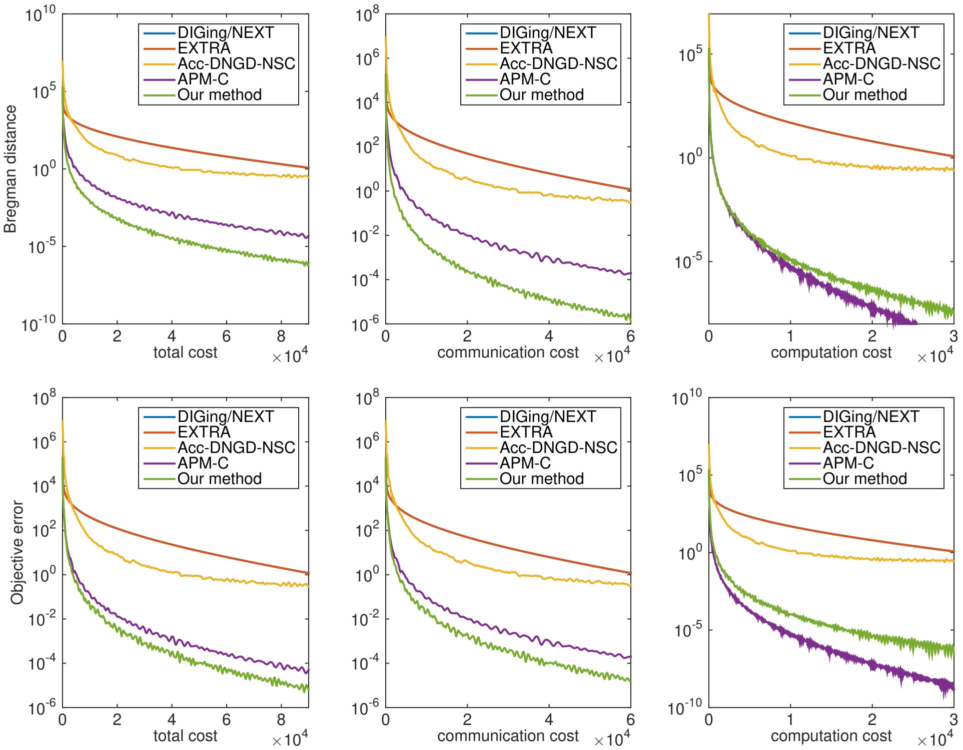


Figure 3.1. Comparison of distributed first-order gradient algorithms. The first row of panels shows the objective error versus the total cost (left panel), the communication cost (middle panel), and the gradient computation cost (right panel). The second row plots the Bregman distance versus the same quantities as in the first row. Note that the curves of DIGing/NEXT overlap with that of EXTRA.

3.5 Numerical Results

We present here some numerical results validating our theoretical findings. We compare the proposed optimal rate algorithm–OPTRA–with existing accelerated algorithms designed for convex smooth problems, namely: Acc-DNGD-NSC [52] and APM-C [51]. We also incuded the gradient tracking method (DIGing/NEXT) [7] and the primal-dual method EXTRA [4]; they are non accelerated schemes but generally perform quite well in practice, achieving linear rate for smooth and strongly convex optimization problems [10], [11].

3.5.1 Decentralized linear regression

We tested the above algorithms on a distributed least squares regression problem, in the form $\min_{x\in\mathbb{R}^d} \left\|Ax - b\right\|^2$, where $A = [A_1; A_2; \cdots; A_m] \in \mathbb{R}^{mr\times d}$ and $b = [b_1; b_2; \cdots; b_m] \in \mathbb{R}^{mr\times 1}$, with $A_\mathrm{i} \in \mathbb{R}^{r\times d}$ and $b_\mathrm{i} \in \mathbb{R}^{r\times 1}$, $r = 10$, $d = 500$, and $m = 20$. Note that each agent i can only access the data $(A_\mathrm{i}, b_\mathrm{i})$. We generated the matrix A of the feature vectors according to the following procedure, proposed in [60]. We first generate a random matrix Z with each entry i.i.d. drawn from $\mathcal{T}(0, 1)$. Using a control parameter $\omega \in [0, 1)$, we generate columns of A ($M_{:,\mathrm{i}}$ and $M_{\mathrm{i},:}$ denote the i-th column and i-th row of a matrix M, respectively) so that the first column is $A_{:,1} = Z_{:,1}/\sqrt{1-\omega^2}$ and the rest are recursively set as $A_{:,\mathrm{i}} = \omega A_{:,\mathrm{i}-1} + Z_{:,\mathrm{i}}$, for $\mathrm{i} = 2, \ldots, d$. As result, each row $A_{\mathrm{i},:} \in \mathbb{R}^d$ is a Gaussian random vector and its covariance matrix $\Sigma = \mathrm{cov}(A_{:,\mathrm{i}})$ is the identity matrix if $\omega = 0$ and becomes extremely ill-conditioned as $\omega \to 1$; we set $\omega = 0.95$. Finally we generate $x_0 \in \mathbb{R}^d$ with each entry i.i.d. drawn from $\mathcal{T}(0, 1)$, and set $b = Ax_0 + \boldsymbol{\xi}$, where each component of the noise $\boldsymbol{\xi}$ is i.i.d. drawn from $\mathcal{T}(0, 0.25)$. We simulated a network of $m = 20$ agents, connected throughout a communication graph, generated using the Erdös-RéTyi model; the probability of having an edge between any two nodes is set to 0.1. We calculated L_f from the generated data and used the exact value whenever this parameter is needed. We tuned the free parameters of the simulated algorithms manually to achieve the best practical performance for each algorithm. This leads to the following choices: **i)** the step size of DIGing/NEXT and EXTRA is set to 10^{-5}; **ii)** for Acc-DNGD-NSC, we used the fixed step-size rule, with $\eta = 0.005/L_f$ (the one provided in [52, Th. 5] is too conservative, resulting in poor practical performance); **iii)** for APM-C, we set (see notation therein) $T_k = \left\lceil c \cdot (\log k/\sqrt{1-\sigma_2(W)})\right\rceil$, with $c = 0.2$ and $\beta_0 = 10^4$; and for **iv)** for our algorithm, we set $\nu = 100$ and $K = 2$.

Our experiments are reported in Figure 3.1, where we plot the Bregman distance (first row of panels) and FEM-metric (3.11) (second row of panels) versus the overall number of communications and computations performed by each agent (left plot), the number of communications (middle plot), and the number of computations (right plot). The following comments are in order. The accelerated schemes converge faster than the non-accelerated schemes NEXT/DIGing and EXTRA (whose curves are coincident in all panels). In our experiments (not all reported), we observed that this gap is quite evident when problems are ill-conditioned. From the right panel, one can see that APM-C performs better than OPTRA and Acc-DNGD-NSC in terms of overall number of gradient evaluations, which is expected since APM-C employs an increasing number of communication steps per gradient evaluation. On the other hand, APM-C suffers from high communication cost (which is evident from the middle panel), making it not competitive with respect to the proposed OPTRA in terms of communications. When both communication and computation costs are considered (left panel), OPTRA outperforms all the other simulated schemes, which support our theoretical findings.

3.5.2 Decentralized logistic regression

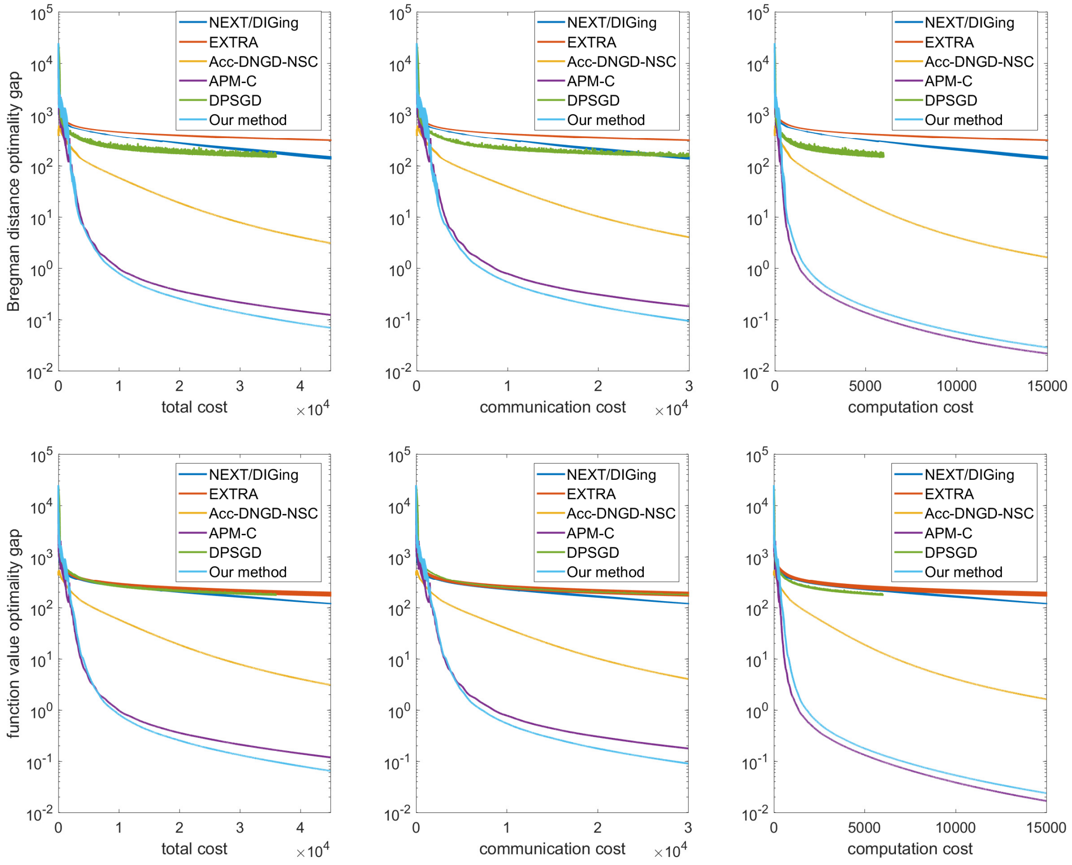


Figure 3.2. Comparison of distributed first-order gradient algorithms for solving the decentralized logistic regression problem in terms of both the Bregman distance and the traditional FEM-metric.

To further verify the effectiveness of our proposed scheme, we also include a decentralized logistic regression task on the Parkinson's Disease Classification Data Set[1]. We preprocess the data by deleting the first column-id number, rescaling feature values to the range $(0, 1)$, and changing the label notation from $\{1, 0\}$ to $\{1, -1\}$. We denote the processed data set as $\{(u_\mathrm{i},\, y_\mathrm{i})\}_{\mathrm{i}\in\mathcal{D}}$, where $u_\mathrm{i} \in \mathbb{R}^d$ is the feature vector and $y_\mathrm{i} \in \{1, -1\}$ is the label of the i-th observation. We simulated a network of 60 agents, generated by the Erdös-RéTyi model with the parameter of connection probability as 0.1. Then, we distributed the data set to all agents evenly, corresponding to a partition of the index set $\mathcal{D}$ across agents as $\mathcal{D} = \cup_{\mathrm{i}=1}^{60}\mathcal{D}_\mathrm{i}$. The decentralized logistic regression problem reads

$$\min_{x\in\mathbb{R}} \sum_{\mathrm{i}=1}^{m} \sum_{\mathrm{j}\in\mathcal{D}_\mathrm{i}} \log\left(1 + \exp(-y_\mathrm{j} u_\mathrm{j}^\top x)\right).$$

We estimated L_f for the problem as $L_f = 24$ and tuned the free parameters of the simulated algorithms manually to achieve the best practical performance for each algorithm. This leads to the following choices: **i)** the step size of NEXT/DIGing is set to 0.01; **ii)** the step size of EXTRA is set to 0.005; **iii)** for Acc-DNGD-NSC, we used the fixed step-size rule, with $\eta = 0.01/L_f$; **iv)** for APM-C, we set (see notation therein) $T_k = \left\lceil c \cdot (\log k/\sqrt{1 - \sigma_2(W)})\right\rceil$, with $c = 0.2$ and $\beta_0 = 10^4$; **v)** for DPSGD, we set its step size as 0.001 and the portion of batch size to the full local data set as 20% and for **vi)** for our algorithm, we set $\nu = 1500$ and $K = 2$.

The experiment result is reported in Figure 3.2. The first row of panels shows the Bregman distance versus the total cost (left panel), the communication cost (middle panel), and the gradient computation cost (right panel). The second row plots the FEM-metric versus the same quantities as in the first row. Both the communication time unit and the computation time unit for a full epoch of local data is set as 1. For DPSGD, the computation time unit is scaled in proportion to the local batch size. The only existing algorithm that has a comparable performance with the proposed OPTRA is APM-C. As discussed in the task of decentralized linear regression, APM-C performs better than OPTRA in terms of the

[1] ↑The data set is available at https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification

number of gradient computations, while suffers from high communication cost. In terms of the overall number of communications and computations, OPTRA outperforms all the other simulated schemes under the above setting.

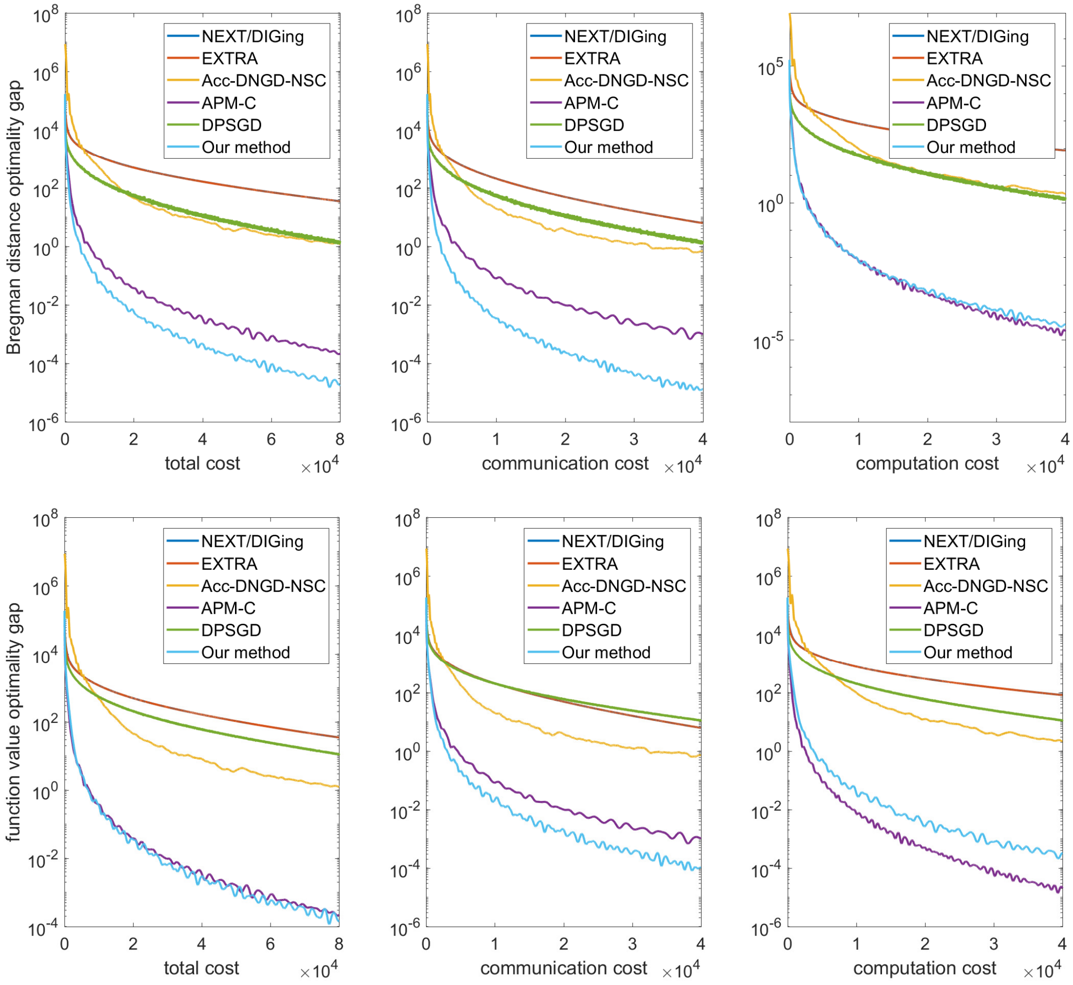


Figure 3.3. Comparison for distributed algorithms for solving the decentralized linear regression problem with the communication time unit being "1" and the computation time unit "5" for a full epoch of local data.

3.5.3 Different ratio of communication time versus computation time

In all the previous experiments, we set both the communication time unit and the computation time unit for a full epoch of local data as 1. To incorporate scenarios where a full epoch computation of local gradient is much more expensive than one communication process, we re-conducted the previous experiments in the setting where the communication

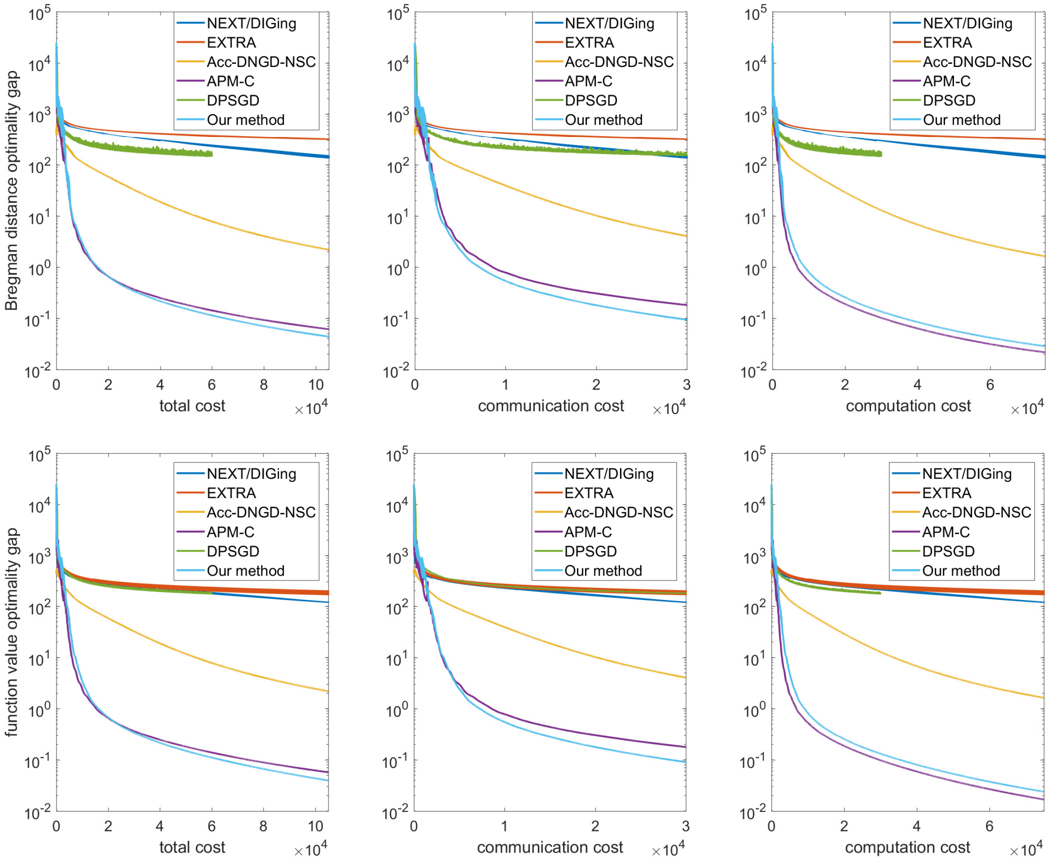


Figure 3.4. Comparison for distributed algorithms for solving the decentralized logistic regression problem, in the setting where the communication time unit is 1 while the computation time unit for a full epoch of local data is 5.

time unit is 1 while the computation time unit for a full epoch of local data is 5. Note that all the process of data generation and parameters tunings are the same as in the Sec. 6.2. The results are reported in Figure 3.3 and Figure 3.4 respectively for decentralized linear regression problem and the decentralized logistic regression problem. It can be seen that OPTRA outperforms all the other simulated schemes in terms of the overall number of communications and computations, especially when the communication cost is not negligible.

3.6 Conclusion

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

3.7 Appendix: Proofs of Theorems

3.7.1 Proof of Proposition 3.2.1

Statement (a) is a direct result of [61, Prop. 6.1.1]. We prove next statement (b). Suppose that there are two optimal solutions $X^\star$ and $\widetilde{X}^\star$ such that

$$\nabla f(X^\star),\ \nabla f(\widetilde{X}^\star) \in \mathcal{C}^\perp, \qquad X^\star,\ \widetilde{X}^\star \in \mathcal{C} \qquad \text{and} \qquad f(X^\star) = f(\widetilde{X}^\star).$$

Since $G(X, X^\star) = f(X) - f(X^\star) - \langle \nabla f(X^\star), X - X^\star \rangle \geq 0$ for all $X \in \mathbb{R}^{m\times d}$, and $G(\widetilde{X}^\star, X^\star) = 0$, $\widetilde{X}^\star$ is the global minimizer of G. Hence, it must be $\nabla f(X^\star) = \nabla f(\widetilde{X}^\star)$, implying

$$\begin{aligned} G(X, X^\star) &= f(X) - f(X^\star) - \langle \nabla f(X^\star), X - X^\star \rangle \\ &= f(X) - f(\widetilde{X}^\star) - \left\langle \nabla f(\widetilde{X}^\star), X - \widetilde{X}^\star \right\rangle = G(X, \widetilde{X}^\star), \qquad \forall X \in \mathbb{R}^{m\times d}, \end{aligned} \tag{3.28}$$

where we have used the fact that $\langle \nabla f(Z), Z \rangle = 0$ for any optimal solution Z.

3.7.2 Proof of Theorem 3.3.1

As elaborated in Section 3.3.1, to study the lower complexity bound of the first order distributed oracle $\mathcal{A}$ solving Problem (3.2) [and thus (3.1)], one can consider ϵ-solutions (i.e., $\bar{X} \in \mathbb{R}^{m\times d}$ such that $G(\bar{X}) \leq \epsilon$) of the following convex optimization problem:

$$\min_{X \in \mathbb{R}^{m\times d}} G(X) = f(X) - \langle \nabla f(X^\star), X - X^\star \rangle - f(X^\star). \tag{3.29}$$

The proof is based on building a worst-case objective function in (3.29) and network graph for which the lower bound is achieved by the best available gossip, distributed algorithm in the oracle $\mathcal{A}$. To do so we build on the cost function first introduced in [50] for a fully connected network and later used for a peer-to-peer network in [14], both for smooth strongly convex problems. Since we use a different metric (the Bregman distance) to define the lower bound and consider smooth convex problems (not necessarily strongly-convex), the analysis in [14] cannot be readily applied to our setting and an ad-hoc proof of the theorem is needed.

The path of our proof is the following: i) We start with a simple network consisting of two agents such that the diameter of the network will not come into play–see Sec. 3.7.2; and ii) then we extend our results to a general network composed by an arbitrary number of agents–see Sec. 3.7.3.

- A simple two-agent network We state the result on the simple two-agent network as the following.

Theorem 3.7.1. *Consider a two-agent network with cost functions given in* (3.30). *Let* $\{X^k\}_{k=0}^{\infty}$ *be the sequence generated by any first-order algorithm* $\mathcal{A}$. *Suppose* $0 \leq k \leq \frac{d-1}{2}$. *Then, we have*

$$G(X^k) = \Omega\left(\frac{L_f\left\|X^0 - X^\star\right\|^2}{(k+1)^2} + \frac{\left\|X^0 - X^\star\right\|\left\|\nabla f(X^\star)\right\|}{k+1}\right).$$

We prove the above result in three steps: i) we construct the hard function in Sec. 3.7.2, which is the worst-case function for all methods belonging to the oracle $\mathcal{A}$; ii) we introduce some intermediate result in Sec. 3.7.2, which is related to our specific metric–the Bregman distance G, and iii) building on step i-ii, we derive the lower bound in Sec. 3.7.2.

- Construction of the hard function Consider a network composed of two agents. The idea of the proof of the lower complexity bound relies on splitting the "hard" function used by Nesterov to prove the iteration complexity of first-order gradient methods for (centralized) smooth convex problems across the agents[29, Chapter 2]. More specifically, consider the following cost functions for the two agents:

$$\begin{cases} f_{1,[k]}(x_1) = \frac{L_f}{8} x_1^\top A_{1,[k]} x_1 - \frac{L_f}{4} \mathrm{e}_1^\top x_1, \\ f_{2,[k]}(x_2) = \frac{L_f}{8} x_2^\top A_{2,[k]} x_2, \end{cases} \tag{3.30}$$

where

$$A_{1,[k]} \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & -1 & \cdots \\ 0 & 0 & 0 & -1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad A_{2,[k]} \triangleq \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & \cdots \\ -1 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & -1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.31}$$

are two $d \times d$ matrices with their leading principal minors of order $k \in [1, d]$ having non-zero block diagonals while the rest being zero.

The key idea of Nesterov proof for the lower complexity bound of centralized first-order gradient methods consists in designing the "hardest" function to be minimized by any method belonging to the oracle. This function was shown to be such that, at iteration k, all these methods produce a new iterate whereby only the kth component is updated. The choice of the two agents' cost functions in (3.30) follows the same rationale: the structure of $A_{1,[k]}$ and $A_{2,[k]}$ is such that none of the two agents is able to make progresses towards optimality, i.e., updating the next component in their local optimization vector (with odd index for agent 1 and even index for agent 2) just performing local gradient updates and without communication with each other. This means that at certain stages a communication between the two agents is necessary for the algorithm to make progresses towards optimality. Building on the above idea, we begin establishing the lower complexity bound for the two-agent network problem in terms of gradient evaluations.

- Intermediate results

Now substituting $f(X) = f_{1,[k]}(x_1) + f_{2,[k]}(x_2)$ in (3.29) and ignoring constants, we obtain

$$\min_{X \in \mathbb{R}^{2\times d}} f_{[k]}(X) \triangleq f_{1,[k]}(x_1) + f_{2,[k]}(x_2) - \left\langle [\nabla f_{1,[k]}(x_1^\star), \nabla f_{2,[k]}(x_2^\star)]^\top, X \right\rangle . \tag{3.32}$$

We denote the optimal function value of the above problem as $f_{[k]}^\star$. It is obvious that, when agents reach consensus, i.e., $x_1 = x_2$, the function $f_{[k]}(X)$ will reduce to the Nesterov's "hard" function [29, Section 2.1.2], for which we have the optimal solution

$$x_1^\star = x_2^\star = [\underbrace{\frac{k}{k+1}, \frac{k-1}{k+1}, ..., \frac{1}{k+1}}_{\text{the first } k \text{ components}}, 0, \dots, 0]^\top \in \operatorname{span}(\mathrm{e}_1, \mathrm{e}_2, ..., \mathrm{e}_k),$$

and it yields

$$\left\|X^\star\right\|^2 = \left\|x_1^\star\right\|^2 + \left\|x_2^\star\right\|^2 \le \frac{2}{3}(k+1) \tag{3.33}$$

and $f_{[k]}^\star = \frac{L_f}{8}(-1 + \frac{1}{k+1})$. Also, we have

$$\begin{cases} \nabla f_{1,[k]}(x_1^\star) = \frac{L_f}{4}(A_{1,[k]}x_1^\star - \mathrm{e}_1) = -\frac{L_f}{4}\frac{1}{k+1}a_{[k]} \\ \nabla f_{2,[k]}(x_2^\star) = \frac{L_f}{4}A_{2,[k]}x_2^\star = \frac{L_f}{4}\frac{1}{k+1}a_{[k]}, \end{cases} \tag{3.34}$$

where

$$a_{[k]} = [\underbrace{1, -1, 1, -1, 1, -1, ...}_{1/-1 \text{ alternates } k \text{ times}}, 0, \dots, 0]^\top.$$

Thus, we further have

$$\left\|\nabla f(X^\star)\right\| = \sqrt{\left\|\nabla f_{1,[k]}(x_1^\star)\right\|^2 + \left\|\nabla f_{2,[k]}(x_2^\star)\right\|^2} = \sqrt{\frac{2L_f^2 a_{[k]}^\top a_{[k]}}{16(k+1)^2}} = \frac{\sqrt{2k}L_f}{4(k+1)}. \tag{3.35}$$

Note that quantities (3.33) and (3.35) will be useful later to relate the complexities with $\left\|X^0 - X^\star\right\|$ and $\left\|\nabla f(X^\star)\right\|$. According to (3.34), Problem (3.32) further becomes

$$\min_{X \in \mathbb{R}^{2\times d}} f_{[k]}(X) = f_{1,[k]}(x_1) + f_{2,[k]}(x_2) + \frac{L_f}{4(k+1)} \left\langle a_{[k]}, x_1 - x_2 \right\rangle . \tag{3.36}$$

In the following, we study the above problem when the local variables x_1 and x_2 are restricted to the truncating subspace of $\mathbb{R}^d$, as a stepping stone to prove Theorem 3.7.1.

Let $\mathbb{R}^{k,d} \triangleq \mathrm{span}(\mathrm{e_i} \in \mathbb{R}^d \,|\, 1 \leq \mathrm{i} \leq k)$ denote the subspace composed of vectors whose only first k components are possibly non-zeros and $\mathcal{L}^k \triangleq \mathrm{span}(\nabla f_\mathrm{i}(x_\mathrm{i}^l) \,|\, 0 \leq l \leq k-1, \mathrm{i} \in \mathcal{V})$. It should be noted that the local cost functions constructed in (3.30) are dependent on k, but hereafter subscripts indicating this dependence are omitted for simplicity.

Lemma 3.7.2 (Linear Span)**.** *Let $\{X^k\}_{k=0}^\infty$ be the sequence generated by any distributed first-order algorithm $\mathcal{A}$ with $X^0 = 0$. Then, $x_\mathrm{i}^k \in \mathcal{L}^k$ for all $k \geq 0$ and all $\mathrm{i} \in \mathcal{V}$.*

The proof of the above lemma is straightforward, since local communication steps do not change the space spanned by the historical gradient vectors generated over the network.

Lemma 3.7.3. *Let $X^0 = 0$. For the two-agent problem* (3.30)*, we have $\mathcal{L}^k \subseteq \mathbb{R}^{k,d}$.*

Proof. Since $X^0 = 0$, we have $\nabla f_1(x_1^0) = -\frac{L_f}{4}\mathrm{e}_1 \in \mathbb{R}^{1,d}, \nabla f_2(x_2^0) = 0 \in \mathbb{R}^{1,d}$ and thus $\mathcal{L}^1 = \mathrm{span}(\nabla f_1(x_1^0), \nabla f_2(x_2^0)) \subseteq \mathbb{R}^{1,d}$. Now, let $x_\mathrm{i}^\mathrm{j} \in \mathcal{L}^\mathrm{j} \subseteq \mathbb{R}^{\mathrm{j},d}$. Without loss of generality, let us assume j is odd. Then, according to the structure of ∇f_1, we have $\nabla f_1(x_1^\mathrm{j}) = \frac{L_f}{4}(A_{1,[k]}x_1^\mathrm{j} - \mathrm{e}_1) \in \mathbb{R}^{\mathrm{j},d}$, but multiplying $A_{1,[k]}$ from the left of $x_1^\mathrm{j} \in \mathbb{R}^{\mathrm{j},d}$ will not increase the number of nonzeros to $\mathrm{j}+1$. By contrast, for ∇f_2, we have $\nabla f_2(x_2^\mathrm{j}) = \frac{L_f}{4}A_{2,[k]}x_2^\mathrm{j} \in \mathbb{R}^{\mathrm{j}+1,d}$ and $A_{2,[k]}$ is now able to increase the number of non-zeros. Therefore, we have $\mathcal{L}^{\mathrm{j}+1} = \mathcal{L}^\mathrm{j} + \mathrm{span}(\nabla f_1(x_1^\mathrm{j}), \nabla f_2(x_2^\mathrm{j})) \subseteq \mathbb{R}^{\mathrm{j}+1,d}$ and we can complete the proof by induction. □

Lemma 3.7.4. *Consider Problem* (3.36)*. Let $f_{[k,\mathrm{j}]}^\star \triangleq \min_{X_\mathrm{i} \in \mathbb{R}^{\mathrm{j},d}, \forall \mathrm{i} \in \mathcal{V}} f_{[k]}(X)$; we have*

$$f_{[k,\mathrm{j}]}^\star = -\frac{L_f}{8}\left(\frac{k^2}{(k+1)^2} + \frac{\mathrm{j}}{(k+1)^2}\right).$$

Proof. Let $x_{\mathrm{i}} \in \mathbb{R}^{1,d}$, $\mathrm{i} \in \mathcal{V}$. Then, the cost function in (3.36) becomes

$$f_{[k,1]}(X) \triangleq \frac{L_f}{4}[0.5x_{11}^2 - x_{11} + \frac{1}{(k+1)}(x_{11} - x_{21}) + 0.5x_{21}^2]$$

which attains the optimum $f^{\star}_{[k,1]} = \frac{L_f}{8}(-\frac{k^2}{(k+1)^2} - \frac{1}{(k+1)^2})$.

Likewise, letting $x_{\mathrm{i}} \in \mathbb{R}^{2,d}$, $\mathrm{i} \in \mathcal{V}$, we have

$$f_{[k,2]}(X) \triangleq \frac{L_f}{4}[0.5x_{11}^2 + 0.5x_{12}^2 - x_{11} - \frac{1}{k+1}(x_{21} - x_{11}) + \frac{1}{k+1}(x_{22} - x_{12}) + 0.5(x_{21} - x_{22})^2]$$

which yields $f^{\star}_{[k,2]} = \frac{L_f}{8}(-\frac{k^2}{(k+1)^2} - \frac{2}{(k+1)^2})$. Also, for $X_{\mathrm{i}} \in \mathbb{R}^{3,d}$, $\mathrm{i} \in \mathcal{V}$, we have

$$\begin{aligned} f_{[k,3]}(X) \triangleq \frac{L_f}{4}[&0.5x_{11}^2 + 0.5(x_{12} - x_{13})^2 - x_{11} - \frac{1}{k+1}(x_{21} - x_{11}) + \frac{1}{k+1}(x_{22} - x_{12}) \\ &- \frac{1}{k+1}(x_{23} - x_{13}) + 0.5(x_{21} - x_{22})^2 + 0.5x_{23}^2], \end{aligned} \tag{3.37}$$

which gives $f^{\star}_{[k,3]} = \frac{L_f}{8}(-\frac{k^2}{(k+1)^2} - \frac{3}{(k+1)^2})$.

In fact, by induction, it is not difficult to show that, when j is odd, for $X_{\mathrm{i}} \in \mathbb{R}^{\mathrm{j},d}$, $\mathrm{i} \in \mathcal{V}$, we have

$$\begin{aligned} f_{[k,\mathrm{j}]}(X) \triangleq \frac{L_f}{4}\Bigg(&0.5x_{11}^2 - \frac{k}{k+1}x_{11} + \sum_{\mathrm{i}=1}^{\frac{\mathrm{j}-1}{2}}\Big(0.5(x_{2(2\mathrm{i})} - x_{2(2\mathrm{i}-1)})^2 - \frac{1}{k+1}(x_{2(2\mathrm{i})} - x_{2(2\mathrm{i}-1)}\Big) \\ &+0.5x_{2\mathrm{j}}^2 - \frac{1}{k+1}x_{2\mathrm{j}} + \sum_{\mathrm{i}=1}^{\frac{\mathrm{j}-1}{2}}\Big(0.5(x_{1(2\mathrm{i})} - x_{1(2\mathrm{i}+1)})^2 - \frac{1}{k+1}(x_{2\mathrm{i}} - x_{1(2\mathrm{i}+1)}\Big)\Bigg), \end{aligned} \tag{3.38}$$

which yields

$$f^{\star}_{[k,\mathrm{j}]} = -\frac{L_f}{8}\left(\frac{k^2}{(k+1)^2} + \frac{\mathrm{j}}{(k+1)^2}\right).$$

When j is even, for $X_{\mathrm{i}} \in \mathbb{R}^{\mathrm{j},d}$, $\mathrm{i} \in \mathcal{V}$, we have

$$f_{[k,\mathrm{j}]}(X) = \frac{L_f}{4}\left(0.5x_{11}^2 - \frac{k}{k+1}x_{11} + \sum_{\mathrm{i}=1}^{\frac{\mathrm{j}}{2}}\left(0.5(x_{2(2\mathrm{i})} - x_{2(2\mathrm{i}-1)})^2 - \frac{1}{k+1}(x_{2(2\mathrm{i})} - x_{2(2\mathrm{i}-1)}\right)\right. \\ \left. +0.5x_{1\mathrm{j}}^2 - \frac{1}{k+1}x_{1\mathrm{j}} + \sum_{\mathrm{i}=1}^{\frac{\mathrm{j}}{2}-1}\left(0.5(x_{1(2\mathrm{i})} - x_{1(2\mathrm{i}+1)})^2 - \frac{1}{k+1}(x_{2\mathrm{i}} - x_{1(2\mathrm{i}+1)}\right)\right) \tag{3.39}$$

which also yields

$$f^\star_{[k,\mathrm{j}]} = -\frac{L_f}{8}\left(\frac{k^2}{(k+1)^2} + \frac{\mathrm{j}}{(k+1)^2}\right).$$

The proof is completed by combining the two cases above. □

- Proof of Theorem 3.7.1

We can now prove the theorem. Let us fix k and apply the first-order gossip algorithm $\mathcal{A}$ to minimize $f_{[2k+1]}$. Since $X^0 = 0$, invoking Lemma 3.7.4, we have

$$\begin{aligned} G(X^k) &= f_{[2k+1]}(X^k) - f^\star_{[2k+1]} \geq \min_{X \in \mathbb{R}^{k,d}} f_{[2k+1]}(X) - f^\star_{[2k+1]} = f^\star_{[2k+1,k]} - f^\star_{[2k+1]} \\ &\geq \frac{L_f}{8}(1 - \frac{1}{2(k+1)} - \frac{(2k+1)^2}{4(k+1)^2} - \frac{k}{4(k+1)^2}) \\ &= \frac{L_f}{32(k+1)} = \Theta\left(\frac{L_f\left\|X^0 - X^\star\right\|^2}{(k+1)^2} + \frac{\left\|X^0 - X^\star\right\|\left\|\nabla f(X^\star)\right\|}{(k+1)}\right), \end{aligned} \tag{3.40}$$

where the last inequality comes from the previously developed facts $\left\|X^\star\right\|^2 = \Theta\left(k+1\right)$, $\left\|\nabla f(X^\star)\right\| = \Theta\left(\frac{L_f}{\sqrt{k+1}}\right)$ and thus $\left\|X^0 - X^\star\right\|^2 = \Theta\left(\frac{k+1}{L_f}\left\|X^0 - X^\star\right\|\left\|\nabla f(X^\star)\right\|\right)$. This completes the proof for the two-agent network. □

Remark 3.7.5. *The lower bound we develop in Theorem 3.7.1 for distributed scenarios has similar structure of that of the recent paper [62], where the lower bound is derived for general equality-constrained problems in centralized scenarios (i.e., $Ax = b$). Notice that the results and techniques therein can not apply to our distributed setting, as we require $b = 0$ and $A \in \mathcal{W}_{\mathcal{G}}$ while the lower bound in [62] is determined by a choice of b and A that does not meet our requirement.*

3.7.3 Proof of Theorem 3.3.1

Following the same path of [14], we now extend the above analysis to the general network setting (arbitrary number of agents) by employing a line graph and constructing certain number of pairwise two-agent networks as in (3.30) from the left and the right of the line graph, respectively, yielding two subgroups. Between these two subgroups, we place a number (proportional to the diameter of the network) of agents with zero cost functions to ensure the necessity of communications between the agents in the two subgroups. To prove the time complexity lower bound, we then leverage the effect of the network by establishing the connection between the diameter of the network and the eigengap of the gossip matrix.

Let $\eta_n = \frac{1-\cos\left(\frac{\pi}{T}\right)}{1+\cos\left(\frac{\pi}{T}\right)}$. For a given $\eta \in (0, 1]$, there exists $n \geq 2$ such that $\eta_n \geq \eta > \eta_{n+1}$. We treat the cases $n = 2$ and $n \geq 3$ separately. Let us first consider the case $n \geq 3$. There exists a line graph of $m = n$ agents and associated Laplacian weight matrix with eigengap η. Now, let us define two subsets of agents as $\mathcal{A}_l = \left\{\mathrm{i}\middle|1 \leq \mathrm{i} \leq \lceil \zeta m \rceil\right\}$ and $\mathcal{A}_r = \left\{\mathrm{i}\middle|\lfloor (1-\zeta)m \rfloor + 1 \leq \mathrm{i} \leq m\right\}$, which lie on the left and the right of the line graph, respectively; the parameter $\zeta \in (0, \frac{1}{2})$ is to be determined. The distance between the two subsets is thus $d_c \triangleq \lfloor (1-\zeta)m \rfloor + 1 - \lceil \zeta m \rceil$. The class of local functions is defined as follows

$$f_{\mathrm{i}} = \begin{cases} \frac{L_f}{8} x_{\mathrm{i}}^\top A_{1,[k]} x_{\mathrm{i}} - \frac{L_f}{4} \mathrm{e}_1^\top x_{\mathrm{i}} & \forall \mathrm{i} \in \mathcal{A}_l \\ \frac{L_f}{8} x_{\mathrm{i}}^\top A_{2,[k]} x_{\mathrm{i}} & \forall \mathrm{i} \in \mathcal{A}_r \\ 0 & \text{otherwise} \end{cases} \tag{3.41}$$

where $A_{1,[k]}, A_{2,[k]}$ are the two matrices defined in (3.31). Similarly to the two-agent network case (cf. Sec. 3.7.2), we have

$$\left\|X^\star\right\|^2 \leq \frac{m}{3}(k+1), \ \left\|\nabla f(X^\star)\right\| \leq \sqrt{2(\zeta m + 1)} \frac{\sqrt{k} L_f}{4(k+1)},$$

and Problem (3.29) becomes

$$\min_{X \in \mathbb{R}^{m \times d}} f_{[k]}(X) = \sum_{\mathrm{i}=1}^{\lceil \zeta m \rceil} f_{\mathrm{i}}(x_{\mathrm{i}}) + f_{m+1-\mathrm{i}}(x_{m+1-\mathrm{i}}) + \frac{L_f}{4(k+1)} \left\langle a_{[k]}, x_{\mathrm{i}} - x_{m+1-\mathrm{i}} \right\rangle \tag{3.42}$$

which further yields

$$f^\star_{[k]} = \lceil \zeta m \rceil \frac{L_f}{8}(-1 + \frac{1}{k+1}) \quad \text{and} \quad f^\star_{[k,\mathrm{i}]} = -\lceil \zeta m \rceil \frac{L_f}{8} \left(\frac{k^2}{(k+1)^2} + \frac{\mathrm{i}}{(k+1)^2} \right).$$

Let each row of X^k belongs to $\mathbb{R}^{k,d}$. Then, since $X^0 = 0$, we have

$$\begin{aligned} G(X^k) &= f_{[2k+1]}(X^k) - f^\star_{[2k+1]} \geq \min_{x_\mathrm{i} \in \mathbb{R}^{k,d}} f_{[2k+1]}(X) - f^\star_{[2k+1]} = f^\star_{[2k+1,k]} - f^\star_{[2k+1]} \\ &\geq \frac{\zeta m L_f}{8}(1 - \frac{1}{2(k+1)} - \frac{(2k+1)^2}{4(k+1)^2} - \frac{k}{4(k+1)^2}) \\ &= \frac{\zeta m L_f}{32(k+1)} = \Theta \left(\frac{L_f \left\|X^0 - X^\star\right\|^2}{(k+1)^2} + \frac{\left\|X^0 - X^\star\right\| \left\|\nabla f(X^\star)\right\|}{k+1} \right). \end{aligned} \tag{3.43}$$

Similarly as the two-agent case, one can verify that $\left\|X^0 - X^\star\right\|^2 = \Theta\Big(\frac{k+1}{L_f}\left\|X^0 - X^\star\right\|\left\|\nabla f(X^\star)\right\|\Big)$.

To have at least one non-zero element at the kth component among the local copies of agents in both of the above two subsets, one must perform at least k local computation steps and $(k-1)d_c$ communication steps. Thus, we have

$$k \leq \lfloor * \rfloor \frac{t-1}{1 + d_c \tau_c} + 1 \leq \frac{t}{1 + d_c \tau_c} + 1. \tag{3.44}$$

Choosing $\zeta = \frac{1}{32}$, we have

$$\begin{aligned} d_c &= \lfloor * \rfloor (1-\zeta)m + 1 - \lceil \zeta m \rceil \geq (1 - 2\zeta)m - 1 \\ &= \frac{15}{16} m - 1 \overset{(a)}{\geq} \frac{15}{16}\left(\sqrt{\frac{2}{\eta}} - 1 \right) - 1 \overset{(b)}{\geq} \frac{1}{5\sqrt{\eta}}, \end{aligned}$$

where (a) is due to $\eta > \eta_{m+1} > \frac{2}{(m+1)^2}$ and (b) is due to $\eta \leq \eta_3 = \frac{1}{3}$. Further, since d_c is an integer, we have $d_c \geq \left\lceil \frac{1}{5\sqrt{\eta}} \right\rceil$. Combining (3.43) and (3.44) leads to

$$G(X^{(t)}) \geq \Omega \left(\frac{L_f \left\|X^0 - X^\star\right\|^2}{(\frac{t}{1 + \lceil \frac{1}{5\sqrt{\eta}} \rceil \tau_c} + 2)^2} + \frac{\left\|X^0 - X^\star\right\| \left\|\nabla f(X^\star)\right\|}{\frac{t}{1 + \lceil \frac{1}{5\sqrt{\eta}} \rceil \tau_c} + 2} \right). \tag{3.45}$$

We focus now on the case $n = 2$. Consider a complete graph of 3 agents with associated Laplacian matrix having eigengap equal to η. The agents' cost functions are

$$f_{\mathrm{i}} = \begin{cases} \frac{L_f}{8} x_{\mathrm{i}}^\top A_{1,[k]} x_{\mathrm{i}} - \frac{L_f}{4} \mathrm{e}_1^\top X_{\mathrm{i}} & \mathrm{i} = 1 \\ \frac{L_f}{8} x_{\mathrm{i}}^\top A_{2,[k]} x_{\mathrm{i}} & \mathrm{i} = 2 \\ 0 & \mathrm{i} = 3 \end{cases}$$

Following similar steps as above, one can show that

$$G(X^k) \geq \Omega \left(\frac{L_f \left\| X^0 - X^\star \right\|^2}{(k+1)^2} + \frac{\left\| X^0 - X^\star \right\| \left\| \nabla f(X^\star) \right\|}{(k+1)} \right) \text{ with } k \leq \frac{t}{1+\tau_c} + 1 \text{ and } 1 \geq \left\lceil \frac{1}{5\sqrt{\eta}} \right\rceil,$$

which leads to the same expression of the lower bound as in (3.45). This concludes the proofs.

3.7.4 Proof of Theorem 3.3.2

The proof follows the similar line of [29, Section 2.1.2]. We consider the same set of local cost functions as depicted in (3.41), with the subscript $[k]$ of the A matrices replaced by $[2k+1]$. Then, it is not difficult to see that $\min_{x \in \mathbb{R}^d} F(x) = f^\star_{[2k+1]}$ and, for any $x \in \mathbb{R}^{k,d}$, we have

$$\begin{aligned} F(x) - f^\star_{[2k+1]} &= \min_{y \in \mathbb{R}^{k,d}} F(y) - f^\star_{[2k+1]} = f^\star_{[k]} - f^\star_{[2k+1]} \\ &= \lceil \zeta m \rceil \frac{L_f}{8} \left(-1 + \frac{1}{k+1} + 1 - \frac{1}{2k+1+1} \right) = \lceil \zeta m \rceil \frac{L_f}{16} \frac{1}{k+1} = \Theta\left(\frac{L_f \, m}{k+1} \right). \end{aligned} \tag{3.46}$$

For the cost functions as mentioned above, one can also verify that (cf. Appendix 3.7.2)

$$\left\| X^\star - X^0 \right\|^2 = \Theta\left(m(k+1) \right), \quad \left\| \nabla f(X^\star) \right\| = \Theta\left(\frac{\sqrt{m} L_f}{\sqrt{k+1}} \right),$$

and thus $\frac{L_f\|X^\star - X^0\|^2}{k+1} = \Theta\left(\left\|X^\star - X^0\right\|\left\|\nabla f(X^\star)\right\|\right)$. As a result, the RHS of (3.46) can be rewritten as:

$$\Theta\left(\frac{L_f\left\|X^\star - X^0\right\|^2}{(k+1)^2} + \frac{\left\|X^\star - X^0\right\|\left\|\nabla f(X^\star)\right\|}{k+1}\right), \text{ or equivalently, } \quad \Theta\left(\frac{L_f\left\|X^\star - X^0\right\|^2}{(k+1)^2}\right),$$

which translate to the following lower bounds in terms of number of iterations, respectively:

$$\Omega\left(\sqrt{\frac{L_f\left\|X^0 - X^\star\right\|^2}{\epsilon}} + \frac{\left\|X^0 - X^\star\right\|\left\|\nabla f(X^\star)\right\|}{\epsilon}\right) \quad \text{and} \quad \Omega\left(\sqrt{\frac{L_f\left\|X^0 - X^\star\right\|^2}{\epsilon}}\right).$$

The rest of proof follows by the same argument as in Section 3.7.3 to relate k to the absolute time t as well as the eigengap η of the network.

3.7.5 Intermediate results

Lemma 3.7.6 (Fundamental Inequality I)**.** *Consider Algorithm (3.23). We define* $\tau = \frac{1}{\nu T \lambda_m(B)}$. *Then we have*

$$\sigma_k = \frac{1}{\theta_{k+1}}, \quad \alpha_k = \frac{\theta_{k+1}}{\theta_k} - \theta_{k+1}, \quad \beta_k = \frac{\tau_{k+1}}{\tau_k}, \quad \tau_k = \frac{\tau}{\theta_k}.$$

Suppose Assumptions 3.2.1 and 3.4.1 hold. Then, for any $X \in \mathbb{R}^{m\times d}$ *and* $Y \in \mathcal{C}^\perp$, *we have*

$$\begin{aligned}&\Phi(u^{k+1}, Y) - \Phi(AX, Y) + h(u^{k+\frac{1}{2}}) - h(X)\\ &\le -\left\langle Y^{k+1} - Y, u^{k+1} - X\right\rangle - \frac{1}{\gamma}\left\langle \theta_k(I - \frac{\gamma\tau}{\theta_k^2}B)(\hat{X}^{k+1} - \hat{X}^k), u^{k+1} - AX\right\rangle + \frac{L_f}{2}\left\|u^{k+1} - X^k\right\|^2,\end{aligned} \tag{3.47}$$

where $h(\cdot) = \frac{1}{2\gamma}\left\|\cdot\right\|^2_{A-A^2}$, $u^{k+\frac{1}{2}} = X^k - \gamma(\nabla f(X^k) + \hat{Y}^k)$ *and* $L_f = \max_i\{L_{f_i}\}$.

Proof. Since f is L_f-smooth by Assumption 3.2.1, we have

$$f(u^{k+1}) \le f(X^k) + \left\langle \nabla f(X^k), u^{k+1} - X^k\right\rangle + \frac{L_f}{2}\left\|u^{k+1} - X^k\right\|^2,$$

and using $f(AX) \geq f(X^k) + \left\langle \nabla f(X^k), AX - X^k \right\rangle$, further gives

$$f(u^{k+1}) \leq f(AX) + \left\langle \nabla f(X^k), u^{k+1} - AX \right\rangle + \frac{L_f}{2} \left\| u^{k+1} - X^k \right\|^2. \tag{3.48}$$

Also, subtracting Au^{k+1} from both sides of (3.23a), multiplying (3.23d) by γA, and adding the obtained two equations while using (3.23e) lead to

$$\begin{aligned}(I - A)u^{k+1} &= -A \left(u^{k+1} - X^k + \gamma \left(\nabla f(X^k) + Y^{k+1} \right) \right) - \gamma AB(\tau_{k-1}\beta_{k-1}\hat{X}^k - \tau_k \hat{X}^{k+1}) \\ &\overset{(*)}{=} -A \left(u^{k+1} - X^k + \gamma \left(\nabla f(X^k) + Y^{k+1} \right) \right) - \frac{\gamma\tau}{\theta_k} AB(\hat{X}^k - \hat{X}^{k+1}),\end{aligned}$$

where in $(*)$ we used $\beta_{k-1} = \frac{\tau_k}{\tau_{k-1}}$, $\tau_k = \frac{\tau}{\theta_k}$. Notice that, for the above derivation, we implicitly assume that $k \geq 2$. However, with the definition of $\hat{X}^1 \triangleq X^1$ and the fact that $\hat{Y}^1 = \tau_1 BX^1$, we still have $(I - A)u^2 = -A\left(u^2 - X^1 + \gamma\left(\nabla f(X^1) + Y^2\right)\right) - \frac{\gamma\tau}{\theta_1} AB(\hat{X}^1 - \hat{X}^2)$.

Multiplying $u^{k+\frac{1}{2}} - X$ from both sides of the above equation and using the convexity of $h(\cdot)$ and the fact that $u^{k+1} = Au^{k+\frac{1}{2}}$ we obtain

$$h(u^{k+\frac{1}{2}}) \leq h(X) - \frac{1}{\gamma} \left\langle u^{k+1} - X^k + \gamma \left(\nabla f(X^k) + Y^{k+1} \right) - \frac{\gamma\tau}{\theta_k} B(\hat{X}^{k+1} - \hat{X}^k), u^{k+1} - AX \right\rangle. \tag{3.49}$$

Since $\sigma_k = \frac{1}{\theta_{k+1}}$ and $\alpha_k = \frac{\theta_{k+1}}{\theta_k} - \theta_{k+1}$, using (3.23b) and (3.23c) leads to

$$\theta_k \hat{X}^{k+1} = u^{k+1} - (1 - \theta_k)u^k \tag{3.50}$$

and

$$\begin{aligned}\hat{X}^{k+1} - \hat{X}^k &= \frac{1}{\theta_k}(u^{k+1} - (1 - \theta_k)u^k) - \frac{1}{\theta_{k-1}}(u^k - (1 - \theta_{k-1})u^{k-1}) \\ &= \frac{1}{\theta_k} u^{k+1} - \frac{1}{\theta_k} \left((1 - \theta_k)u^k + \frac{\theta_k}{\theta_{k-1}} u^k - \frac{\theta_k}{\theta_{k-1}}(1 - \theta_{k-1})u^{k-1} \right) \\ &= \frac{1}{\theta_k} u^{k+1} - \frac{1}{\theta_k} \left(u^k + (\frac{\theta_k}{\theta_{k-1}} - \theta_k)(u^k - u^{k-1}) \right) \\ &\overset{(3.23b)}{=} \frac{1}{\theta_k}(u^{k+1} - X^k).\end{aligned} \tag{3.51}$$

We implicitly assumed $k \geq 2$; still we have $\hat{X}^2 - \hat{X}^1 = \frac{1}{\theta_k}(u^2 - X^1)$, recalling that $\hat{X}^1 = X^1$. Thus, (3.49) becomes

$$h(u^{k+\frac{1}{2}}) \leq h(X) - \frac{1}{\gamma}\left\langle \theta_k(I - \frac{\gamma\tau}{\theta_k^2}B)(\hat{X}^{k+1} - \hat{X}^k) + \gamma\left(\nabla f(X^k) + Y^{k+1}\right), u^{k+1} - AX \right\rangle. \tag{3.52}$$

Combining (3.48) and (3.52) yields: for any $X \in \mathbb{R}^{m\times d}$ and $Y \in \mathcal{C}^\perp$,

$$\begin{aligned} &f(u^{k+1}) + h(u^{k+\frac{1}{2}}) + \left\langle Y, u^{k+1} - AX\right\rangle - f(AX) - h(X) \\ &\leq \left\langle -(Y^{k+1} - Y) - \frac{1}{\gamma}\theta_k(I - \frac{\gamma\tau}{\theta_k^2}B)(\hat{X}^{k+1} - \hat{X}^k), u^{k+1} - AX \right\rangle + \frac{L_f}{2}\left\|u^{k+1} - X^k\right\|^2, \end{aligned} \tag{3.53}$$

which, recalling that $\Phi(X,Y) = f(X) + \langle Y, X\rangle$, completes the proof. □

Lemma 3.7.7 (Fundamental Inequality II)**.** *In the setting of Lemma 3.7.6, let* $\frac{1}{\theta_{k-1}^2} - \frac{1-\theta_k}{\theta_k^2} = 0$*, that is,* $\frac{1}{\theta_k} = \frac{1+\sqrt{1+4(\frac{1}{\theta_{k-1}})^2}}{2}$*, with* $\theta_1 = 1$ *and* $(1-\gamma L_f)I - \frac{\gamma\tau}{\theta_k^2}B \succeq 0$*, for all* $1 \leq k \leq T-1$*. Suppose Assumptions 3.2.1 and 3.4.1 hold. Then, for any* $X \in \mathcal{C}, Y \in \mathcal{C}^\perp$*, we have*

$$\Phi(u^T, Y) - \Phi(X,Y) \leq \frac{1}{T^2}\left(\frac{2}{\gamma}\left\|u^1 - X\right\|^2 + \frac{2}{\tau\lambda_2(B)}\left\|Y\right\|^2\right). \tag{3.54}$$

where N *is the overall number of iterations.*

Proof. Applying Lemma 3.7.6 with $X \in \mathcal{C}$, we have (note that $AX = X$ by Assumption 3.4.1)

$$\begin{aligned} &\Phi(u^{k+1}, Y) - \Phi(X,Y) + h(u^{k+\frac{1}{2}}) - h(X) \\ &\leq -\left\langle Y^{k+1} - Y, u^{k+1} - X\right\rangle - \left\langle \frac{1}{\gamma}\theta_k(I - \frac{\gamma\tau}{\theta_k^2}B)(\hat{X}^{k+1} - \hat{X}^k), u^{k+1} - X \right\rangle + \frac{L_f}{2}\left\|u^{k+1} - X^k\right\|^2. \end{aligned} \tag{3.55}$$

Likewise, with $X = u^{k-\frac{1}{2}}$ we have

$$\begin{aligned} &\Phi(u^{k+1}, Y) - \Phi(u^k,Y) + h(u^{k+\frac{1}{2}}) - h(u^{k-\frac{1}{2}}) \\ &\leq -\left\langle Y^{k+1} - Y, u^{k+1} - u^k\right\rangle - \left\langle \frac{1}{\gamma}\theta_k(I - \frac{\gamma\tau}{\theta_k^2}B)(\hat{X}^{k+1} - \hat{X}^k), u^{k+1} - u^k \right\rangle + \frac{L_f}{2}\left\|u^{k+1} - X^k\right\|^2. \end{aligned} \tag{3.56}$$

Let $V_k = \Phi(u^k, Y) - \Phi(X, Y) + h(u^{k+\frac{1}{2}}) - h(X)$. Then, multiplying (3.56) by $1-\theta_k$ and (3.55) by θ_k, and combing the obtained equations yield

$$
\begin{aligned}
& V_{k+1} - (1-\theta_k)V_k \\
& \leq -\left\langle Y^{k+1} - Y, u^{k+1} - \theta_k X - (1-\theta_k)u^k \right\rangle \\
& - \frac{1}{\gamma}\left\langle \theta_k (I - \frac{\gamma\tau}{\theta_k^2}B)(\hat{X}^{k+1} - \hat{X}^k), u^{k+1} - \theta_k X - (1-\theta_k)u^k \right\rangle + \frac{L_f}{2}\left\| u^{k+1} - X^k \right\|^2 \\
& \overset{(3.50)}{=} -\theta_k \left\langle Y^{k+1} - Y, \hat{X}^{k+1} - X \right\rangle - \frac{\theta_k^2}{\gamma}\left\langle \hat{X}^{k+1} - \hat{X}^k, \hat{X}^{k+1} - X \right\rangle_{I - \frac{\gamma\tau}{\theta_k^2}B} + \frac{\theta_k^2 L_f}{2}\left\| \hat{X}^{k+1} - \hat{X}^k \right\|^2 \\
& = -\frac{\theta_k^2}{\tau}\left\langle Y^{k+1} - Y, Y^{k+1} - Y^k \right\rangle_{(B+J)^{-1}} - \frac{\theta_k^2}{\gamma}\left\langle \hat{X}^{k+1} - \hat{X}^k, \hat{X}^{k+1} - X \right\rangle_{I - \frac{\gamma\tau}{\theta_k^2}B} + \frac{\theta_k^2 L_f}{2}\left\| \hat{X}^{k+1} - \hat{X}^k \right\|^2,
\end{aligned}
\tag{3.57}
$$

where in the last equality we used $\mathbf{1}^\top Y^k = 0, \forall k \geq 1$ and the following result (recall $BJ = JB = 0$ and $Y \in \mathcal{C}^\perp$):

$$
\begin{aligned}
& \left\langle Y^{k+1} - Y, \hat{X}^{k+1} - X \right\rangle \\
& \quad = \left\langle (B+J)^{-1}(B+J)(Y^{k+1} - Y), \hat{X}^{k+1} - X \right\rangle \\
& \quad = \left\langle B(B+J)^{-1}(Y^{k+1} - Y), \hat{X}^{k+1} - X \right\rangle \\
& \quad = \left\langle Y^{k+1} - Y, B(\hat{X}^{k+1} - X) \right\rangle_{(B+J)^{-1}} \\
& \quad \overset{(3.23d)}{=} \frac{\theta_k}{\tau}\left\langle Y^{k+1} - Y, Y^{k+1} - Y^k \right\rangle_{(B+J)^{-1}}.
\end{aligned}
\tag{3.58}
$$

Dividing θ_k^2 from both sides of (3.57) leads to

$$
\begin{aligned}
& \frac{V_{k+1}}{\theta_k^2} - \frac{1-\theta_k}{\theta_k^2}V_k \\
& \leq -\frac{1}{\gamma}\left\langle \hat{X}^{k+1} - \hat{X}^k, \hat{X}^{k+1} - X \right\rangle_{I - \frac{\gamma\tau}{\theta_k^2}B} - \frac{1}{\tau}\left\langle Y^{k+1} - Y, Y^{k+1} - Y^k \right\rangle_{(B+J)^{-1}} + \frac{L_f}{2}\left\| \hat{X}^{k+1} - \hat{X}^k \right\|^2 \\
& = -\frac{1}{2\gamma}\left(\left\| \hat{X}^{k+1} - \hat{X}^k \right\|^2_{(1-\gamma L_f)I - \frac{\gamma\tau}{\theta_k^2}B} + \left\| \hat{X}^{k+1} - X \right\|^2_{I - \frac{\gamma\tau}{\theta_k^2}B} - \left\| \hat{X}^k - X \right\|^2_{I - \frac{\gamma\tau}{\theta_k^2}B} \right) \\
& \quad - \frac{1}{2\tau}\left(\left\| Y^{k+1} - Y^k \right\|^2_{(B+J)^{-1}} + \left\| Y^{k+1} - Y \right\|^2_{(B+J)^{-1}} - \left\| Y^k - Y \right\|^2_{(B+J)^{-1}} \right),
\end{aligned}
\tag{3.59}
$$

where in the last equality we used

$$2\left\langle a-c,b-c\right\rangle_G=\left\|a-c\right\|_G^2+\left\|b-c\right\|_G^2-\left\|a-b\right\|_G^2,\quad \forall a,\,b\in\mathbb{R}^{m\times d}.$$

Summing (3.59) over k from 1 to $T-1$ yields

$$\begin{aligned}&\frac{V_T}{\theta_{T-1}^2}-\frac{1-\theta_1}{\theta_1^2}V_1+\sum_{k=2}^{T-1}\left(\frac{1}{\theta_{k-1}^2}-\frac{1-\theta_k}{\theta_k^2}\right)V_k\\&\le-\frac{1}{2\gamma}\sum_{k=1}^{T-1}\left\|\hat{X}^{k+1}-\hat{X}^k\right\|_{(1-\gamma L_f)I-\frac{\gamma\tau}{\theta_k^2}B}^2-\frac{1}{2\gamma}\sum_{k=2}^{T-1}\gamma\tau(\frac{1}{\theta_k^2}-\frac{1}{\theta_{k-1}^2})\left\|\hat{X}^k-X\right\|_B^2\\&\quad-\frac{1}{2\gamma}\left(\left\|\hat{X}^T-X\right\|_{I-\frac{\gamma\tau}{\theta_{T-1}^2}B}^2-\left\|\hat{X}^1-X\right\|_{I-\frac{\gamma\tau}{\theta_1^2}B}^2\right)\\&\quad-\frac{1}{2\tau}\left(\sum_{k=1}^{T-1}\left\|Y^{k+1}-Y^k\right\|_{(B+J)^{-1}}^2+\left\|Y^T-Y\right\|_{(B+J)^{-1}}^2-\left\|Y^1-Y\right\|_{(B+J)^{-1}}^2\right).\end{aligned}\tag{3.60}$$

Recalling that $\frac{1}{\theta_{k-1}^2}-\frac{1-\theta_k}{\theta_k^2}=0$ and $\theta_1=1$, by induction it is easy to see that $k+1>\frac{1}{\theta_k}\ge\frac{k+1}{2}$ and thus $\frac{1}{\theta_k^2}-\frac{1}{\theta_{k-1}^2}=\frac{1}{\theta_k}>0$. Then, with $\hat{X}^1=X^1\triangleq u^1,Y^1\triangleq 0$, (3.60) can be simplified as

$$\begin{aligned}\frac{T^2}{4}V_T+\sum_{\mathrm{i}=1}^{T}\frac{1}{2\tau}\left\|Y^{k+1}-Y^k\right\|_{(B+J)^{-1}}^2+\frac{1}{2\gamma}\sum_{k=1}^{T-1}\left\|\hat{X}^{k+1}-\hat{X}^k\right\|_{(1-\gamma L_f)I-\frac{\gamma\tau}{\theta_k^2}B}^2&\\\le\frac{1}{2\gamma}\left\|u^1-X\right\|_{I-\frac{\gamma\tau}{\theta_1^2}B}^2+\frac{1}{2\tau}\left\|Y\right\|_{(B+J)^{-1}}^2.&\end{aligned}\tag{3.61}$$

Since $\rho\left((B+J)^{-1}\right)=\frac{1}{\lambda_{\min}(B+J)}=\frac{1}{\lambda_2(B)}$, $B\succeq 0$ and $(1-\gamma L_f)I-\frac{\gamma\tau}{\theta_k^2}B\succeq 0$, we further have

$$\frac{T^2}{4}V_T\le\frac{1}{2\gamma}\left\|u^1-X\right\|^2+\frac{1}{2\tau}\frac{\left\|Y\right\|^2}{\lambda_2(B)},\tag{3.62}$$

which, together with the fact that $V_k\ge\Phi(u^k,Y)-\Phi(X,Y)$, completes the proof. □

3.7.6 Proof of Theorem 3.4.3

Note that the primal-dual method (3.12) is a special case of the update (3.23) with the setting $\theta_k\equiv 1,\alpha_k\equiv 0,\sigma_k\equiv 1,\tau_k\equiv\tau,\beta_k\equiv 1$. Furthermore, γ and τ defined in (3.13) satisfy

$(1-\gamma L_f)I - \gamma\tau B \geq 0$ and $X^k \equiv u^k$. Invoking (3.60) with these parameter settings and $X \triangleq X^\star$, $Y \triangleq Y^\star = -\nabla f(X^\star)$, $\hat{X}^1 \triangleq X^1, Y^1 \triangleq 0$, we have

$$\sum_{k=2}^{T}(\Phi(X^k,Y^\star) - \Phi(X^\star,Y^\star)) \leq \frac{1}{2\gamma}\left\|X^1 - X^\star\right\|^2 + \frac{1}{2\tau}\frac{\left\|Y^\star\right\|^2}{\lambda_2(B)}, \tag{3.63}$$

Let $\bar{X}^T \triangleq \frac{1}{T-1}\sum_{k=2}^{T} X_k$. Using the convexity of Φ, we furhter have

$$\begin{aligned}\Phi(\bar{X}^T,Y^\star) - \Phi(X^\star,Y^\star)) &\leq \frac{1}{T-1}\left(\frac{1}{2\gamma}\left\|X^1 - X^\star\right\|^2 + \frac{1}{2\tau}\frac{\left\|Y^\star\right\|^2}{\lambda_2(B)}\right)\\ &\leq \frac{1}{T-1}\left(\frac{L_f}{2}\left\|X^1 - X^\star\right\|^2 + \frac{1}{2\nu}\left\|X^1 - X^\star\right\|^2 + \frac{\nu}{2\eta(B)}\left\|Y^\star\right\|^2\right).\end{aligned} \tag{3.64}$$

Setting $\nu = \frac{\sqrt{\eta(B)}\|X^1 - X^\star\|}{\nabla f(X^\star)}$ yields (3.14). Finally, it follows from (3.60) that $\hat{X}^T = X^T$ is bounded, for every $T \in \mathbb{N}_+$. The rest of proof is to show that $X^T \to X^\star$, which follows the standard cluster point analysis, as in the proof of [53, Th. 1] (refer also to [63, Remark 3]).

3.7.7 Proof of Theorem 3.4.5

Since $\gamma = \frac{\nu}{\nu L_f + T}, \tau = \frac{1}{\nu T \lambda_m(B)}$ and $\frac{1}{k+1} < \theta_k < \frac{2}{k+1}$, we have

$$(1-\gamma L_f)I - \frac{\gamma\tau}{\theta_k^2}B \succeq 0, \forall 1 \leq k \leq T-1. \tag{3.65}$$

Then, invoking Lemma 3.7.7 with $X = X^\star, Y = Y^\star = -\nabla f(X^\star)$ and knowing that $\Phi(u^k, Y^\star) - \Phi(X^\star, Y^\star) = G(u^k) \geq 0$ (cf., the relation (3.5) in the main text), we obtain

$$\begin{aligned}G(u^T) &\leq \frac{\frac{2}{\gamma}R_x + \frac{2}{\tau}\frac{R_y}{\lambda_2(B)}}{T^2} = \frac{2(L_f + T/\nu)R_x + 2\nu T\lambda_m(B)\frac{R_y}{\lambda_2(B)}}{T^2}\\ &= \frac{2L_f R_x}{T^2} + \frac{\frac{2}{\nu}R_x + 2\nu\frac{R_y}{\eta(B)}}{T},\end{aligned} \tag{3.66}$$

where $R_x = \left\|u^1 - X^\star\right\|^2, R_y = \left\|\nabla f(X^\star)\right\|^2$. Setting $\nu = \sqrt{\eta(B)}$ we have

$$G(u^T) \leq \frac{2L_f}{T^2}R_x + \frac{2}{\sqrt{\eta(B)}T}(R_x + R_y),$$

which, together with the time $(1+t_c)$ needed at each iteration, gives the overall time complexity.

If we set[2] $\nu = \sqrt{\frac{\eta(B)R_x}{R_y}}$, we have

$$G(u^T, X^\star) \leq \frac{2L_f R_x}{T^2} + \frac{4\sqrt{R_x R_y}}{\sqrt{\eta(B)}T},$$

which matches the lower bound also with respect to R_x, R_y.

In the following, we show that both consensus error and the absolute value of the objective error will converge at the same rate as the Bregman distance. Invoking Lemma 3.7.7 with $X = X^\star$, $\gamma = \frac{\nu}{\nu L_f + T}$, $\tau = \frac{1}{\nu T \lambda_m(B)}$ and $\nu = \sqrt{\eta(B)}$, we have

$$f(u^T) - f(X^\star) + \left\langle u^T, Y \right\rangle = \Phi(u^T, Y) - \Phi(X^\star, Y) \leq \phi(\left\|Y\right\|) \tag{3.67}$$

where $\phi(\cdot) \triangleq \frac{2L_f R_x}{T^2} + \frac{\frac{2}{\sqrt{\eta}}(R_x + (\cdot)^2)}{T}$.

Now, setting $Y = 2\frac{\tilde{u}^T}{\|\tilde{u}^T\|}\left\|Y^\star\right\|$ where $\tilde{u}^T = (I - \frac{\mathbf{1}\mathbf{1}^T}{m})u^T$, we have

$$f(u^T) - f(X^\star) + 2\left\|Y^\star\right\|\left\|\tilde{u}^T\right\| \leq \phi(2\left\|Y^\star\right\|)$$

Also, since $f(u^T) - f(X^\star) + \left\langle u^T, Y^\star \right\rangle \geq 0$, we have $f(u^T) - f(X^\star) \geq -\left\|Y^\star\right\|\left\|\tilde{u}^T\right\|$. Thus, combining the above two inequalities yields

$$\left\|\tilde{u}^T\right\| \leq \frac{\phi(2\left\|Y^\star\right\|)}{\left\|Y^\star\right\|} \text{ and } \left|f(u^T) - f(X^\star)\right| \leq \phi(2\left\|Y^\star\right\|).$$

□

[2] ↑Note that this requires accurate estimates on the ratio of R_x/R_y, which, indeed, plays a key role of trade-off parameter balancing gradient computation steps and communication steps.

3.7.8 Proof of Theorem 3.4.6

Following the similar lines in [14, Theorem 4], we first consider the normalized Laplacian L has a spectrum in $[1-c_1^{-1}, 1+c_1^{-1}]$. According to [14], [31], the Chebyshev polynomail $P_K(x) = 1 - \frac{T_K(c_1(1-x))}{T_K(c_1)}$ is the solution of the following problem

$$\min_{p\in\mathbb{P}_K, p(0)=0} \max_{x\in[1-c_1^{-1}, 1+c_1^{-1}]} |p(x)-1| .$$

As a result, we have

$$\max_{x\in[1-c_1^{-1}, 1+c_1^{-1}]} |P_K(x)-1| \leq 2\frac{c_0^K}{1+c_0^{2K}}. \tag{3.68}$$

Define $\delta = 2\frac{c_0^K}{1+c_0^{2K}}$. Since Algorithm 2 amounts to an instance of Procedure (3.23) with $A = I - c_2 \cdot P_K(L)$ and $B = P_K(L)$, its convergence proof follows the same lines as that of Theorem 3.4.5 with the following properties of $P_K(L)$: i) $P_K(L)$ is symmetric; ii) according to (3.68), $0 \preceq I - c_2 \cdot P_K(L) \preceq I$ and $P_K(L) \succeq 0$, and $\mathrm{null}(P_K(L)) = \mathcal{C}$; iii) The values given for γ and τ in Algorithm 2 ensures that $(1-\gamma L_f)I - \frac{\gamma\tau}{\theta_k^2} P_K(L) \succeq 0$, analogus to (3.65). Therefore we have

$$\begin{aligned} G(u^T) &\leq \frac{\frac{2}{\gamma}R_x + \frac{2}{\tau}\frac{R_y}{\lambda_{\min}(P_K(L)+J)}}{T^2} \leq \frac{2(L_f + T/\nu)R_x + 2\nu T(1+\delta)\frac{R_y}{1-\delta}}{T^2} \\ &= \frac{2L_f R_x}{T^2} + \frac{\frac{2}{\nu}R_x + 2\,\nu\,R_y\frac{1+\delta}{1-\delta}}{N} \overset{(*)}{=} \frac{2L_f R_x}{T^2} + \frac{4\sqrt{R_x R_y}}{N}\sqrt{\frac{1+\delta}{1-\delta}}, \end{aligned} \tag{3.69}$$

where (*) requires a specified ν. Finally, we have

$$\sqrt{\frac{1+\delta}{1-\delta}} = \left(1 + \left(\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\right)^K\right) \Big/ \left(1 - \left(\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\right)^K\right).$$

Taking $K = \left\lceil \frac{1}{\sqrt{\eta}} \right\rceil$, we have

$$\begin{aligned}
&\left(\frac{1-\sqrt{\eta}}{1+\sqrt{\eta}}\right)^{\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil} = \left(1 - \frac{2\sqrt{\eta}}{1+\sqrt{\eta}}\right)^{\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil} \leq \left(1 - \frac{2}{1+\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil}\right)^{\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil} \\
&\overset{(*)}{\leq} \left(1 - \frac{1}{\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil}\right)^{\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil} < \mathrm{e}^{-1},
\end{aligned}$$

where (*) is due to the fact that $\left\lceil \frac{1}{\sqrt{\eta}} \right\rceil \geq 1$. Thus, we have $\sqrt{\frac{1+\delta}{1-\delta}} \leq \frac{1+\mathrm{e}^{-1}}{1-\mathrm{e}^{-1}} \leq 2.5$, which, together with the time $(1 + K\tau_c)$ needed at each iteration, gives the time complexity as announced. □

4. ASYNCHRONOUS DECENTRALIZED ALGORITHM - PART I: P-ASY-PUSH-SUM

In the remaining of the dissertation, we study *asynchronous* multi-agent distributed/decentralized optimization (P) over static digraphs. We commit to a general *asynchronous* decentralized setting, whereby i) agents can update their local variables as well as communicate with their neighbors at any time, without any form of coordination; and ii) they can perform their local computations using (possibly) delayed, out-of-sync information from the other agents. Delays need not be known to the agent or obey any specific profile, and can also be time-varying (but bounded). As the gradient tracking mechanism is a key enabler for synchronous distributed optimization algorithms to match the rate of centralized algorithms and our ultimate goal is to design asynchronous distributed optimization algorithms, we are motivated to design firstly a gradient tracking mechanism which is robust against asynchrony. Thus in this chapter, we propose a general asynchronous signal tracking algorithm. Later with the asynchronous tracking algorithm estimating locally the average of agents' gradients, we propose an asynchronous distributed algorithm ASY-SONATA, for unconstrained smooth, convex and nonconvex optimization in Chapter 5. We further extend the algorithm ASY-SONATA to ASY-DSCA to deal with constrained nonsmooth, convex and nonconvex optimization problems in Chapter 6.

The novel results of this chapter have been published in

- Ye Tian, Ying Sun, and Gesualdo Scutari. "ASY-SONATA: Achieving linear convergence in distributed asynchronous multiagent optimization." In 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 543-551. IEEE, 2018.

- Ye Tian, Ying Sun, and Gesualdo Scutari. "Achieving Linear Convergence in Distributed Asynchronous Multiagent Optimization." IEEE Transactions on Automatic Control 65, no. 12 (2020): 5264-5279.

4.1 Introduction

In the decentralized/distributed setting, due to the lack of global knowledge on Problem (P) and the networked setting, computation has to be performed in a collaborative manner: agents can only receive/send information from/to its immediate neighbors.

As the problem and network size scale, synchronizing the entire multiagent system becomes inefficient or infeasible. Synchronous schedules require a global clock, which is against the gist of removing the central controller as in decentralized optimization. This calls for the development of *asynchronous* decentralized learning algorithms. In addition, asynchronous modus operandi brings also benefits such as mitigating communication and/or memory-access congestion, saving resources (e.g., energy, computation, bandwidth), and making algorithms more fault-tolerant. Therefore, asynchronous decentralized algorithms have the potential to prevail in large scale learning problems. In the remaining of the thesis, we consider the following very general, abstract, asynchronous model [64]:

(i) Agents can perform their local computations as well as communicate (possibly in parallel) with their immediate neighbors at any time, without any form of coordination or centralized scheduling; and

(ii) when solving their local subproblems, agents can use outdated information from their neighbors.

In (ii) no constraint is imposed on the delay profiles: delays can be arbitrary (but bounded), time-varying, and (possibly) dependent on the specific activation rules adopted to wakeup the agents in the network. This model captures in a unified fashion several forms of asynchrony: some agents execute more iterations than others; some agents communicate more frequently than others; and inter-agent communications can be unreliable and/or subject to unpredictable, time-varying delays.

In this chapter, we aim to solve the asynchronous signal tracking problem. Each agent i owns a possibly time-varying signal $\{u_{\mathrm{i}}^k\}_{k\in\mathbb{N}_0}$; the goal of the system is to asymptotically track the average signal $\bar{u}^k \triangleq (1/m)\cdot\sum_{\mathrm{i}=1}^m u_{\mathrm{i}}^k$, that is,

$$\lim_{k\to\infty}\left\|y_{\mathrm{i}}^{k+1}-\bar{u}^{k+1}\right\|=0,\quad \forall \mathrm{i}\in\mathcal{V}. \tag{4.1}$$

Note that when the signal is time-invariant, i.e., $u_{\mathrm{i}}^k\equiv u_{\mathrm{i}},\ \forall \mathrm{i}\in\mathcal{V}$, the above problem reduces to the asynchronous average consensus problem.

4.1.1 Literature Review

Distributed average consensus and signal tracking problem has been studied extensively in the community of control. Continuous-time average consensus has been studied in [65], [66]; the counterpart in discrete time has been studied in [36], [67]–[69], with [67] proposing the renowned scheme push-sum on general directed graphs and [36] focusing on the fastest average consensus schemes. As for the signal tracking (dynamic tracking) problem, numerous schemes have been proposed in the continuous-time domain [70], [71] and also the discrete-time domain [72].

However, all the schemes mentioned above assume perfect communication and synchronous update. In this dissertation, we are instead interested in schemes that are robust against unreliable links and asynchrony. In [73], the authors designed a synchronous average consensus algorithm robust to packet losses; the scheme was further extended in [74] to deal with uncoordinated (deterministic) agents' activations. However, none of them can deal with arbitrary bounded delays but packet losses; [73] is *synchronous*; and [74] is not parallel scheme, as at each iteration only one agent is allowed to wake up and transmit information to its neighbors. In particular, [74] cannot model synchronous parallel (Jacobi) updates.

4.1.2 Summary of Contributions

The review of the literature clearly showed that there exits no consensus/tracking scheme that is robust against imperfect communications and asynchrony. This chapter proposes a general asynchronous signal tracking algorithm for problem (4.1), over *directed* graphs. The proposed algorithm has the following attractive features: (a) it is *parallel and asynchronous [in the sense (i) and (ii)]*–multiple agents can be activated at the same time (with no coordination) and/or outdated information can be used in the agents' updates; (b) it is implementable over *digraph*; (c) it converges in a geometric rate when solving the asynchronous average consensus problem; and (d) it does not have any tuning parameter or step size.

On the technical side, the asynchronous agent system is reduced to a synchronous "augmented" one with no delays by adding virtual agents to the graph. While this idea was first explored in [73], [75],[76], the proposed enlarged system and algorithm differ from those used therein, which cannot deal with the general asynchronous model considered here–see Remark 4.4.1 in Sec.4.4.

4.2 Problem Setup and Preliminaries

4.2.1 Problem Setup

We study Problem (4.1) under the following assumptions.

On the communication network: The communication network of the agents is modeled as a fixed, directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, m\}$ is the set of nodes (agents), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (communication links). If $(\mathrm{i}, \mathrm{j}) \in \mathcal{E}$, it means that agent i can send information to agent j. We assume that the digraph does not have self-loops. We denote by $\mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$ the set of *in-neighbors* of node i, i.e., $\mathcal{N}_{\mathrm{i}}^{\mathrm{in}} \triangleq \{\mathrm{j} \in \mathcal{V} \mid (\mathrm{j}, \mathrm{i}) \in \mathcal{E}\}$ while $\mathcal{N}_{\mathrm{i}}^{\mathrm{out}} \triangleq \{\mathrm{j} \in \mathcal{V} \mid (\mathrm{i}, \mathrm{j}) \in \mathcal{E}\}$ is the set of *out-neighbors* of agent i. We make the following standard assumption on the graph connectivity.

Assumption 4.2.1. *The graph $\mathcal{G}$ is strongly connected.* □

4.2.2 Preliminaries: the sum-push algorithm

We first consider the average consensus problem in the multi-agent setting. This problem can be solved using the push-sum algorithm [67]. In view of our asynchronous implementation later, it is convenient to rewrite the push-sum algorithm breaking the "push" and "sum" steps in two *separate* actions and switch their order. While there is no advantage in doing that in a synchronous setting, this will simplify the presentation of its asynchronous counterpart as well as lead to a more flexible asynchronous implementation.

The sum-push reads: given z_{i}^k, ϕ_{i}^k, ρ_{ij}^k, and σ_{ij}^k, at iteration $k \in \mathbb{N}_+$, each agent $\mathrm{i} \in \mathcal{V}$ performs

$$\text{Sum:} \qquad \begin{cases} z_{\mathrm{i}}^{k+\frac{1}{2}} = z_{\mathrm{i}}^k + \sum\limits_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}^{\mathrm{in}}} \rho_{\mathrm{ij}}^k, \\ \phi_{\mathrm{i}}^{k+\frac{1}{2}} = \phi_{\mathrm{i}}^k + \sum\limits_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}^{\mathrm{in}}} \sigma_{\mathrm{ij}}^k; \end{cases} \tag{4.2}$$

$$\text{Push:} \qquad \begin{cases} z_{\mathrm{i}}^{k+1} = a_{\mathrm{ii}}\, z_{\mathrm{i}}^{k+\frac{1}{2}}, \\ \phi_{\mathrm{i}}^{k+1} = a_{\mathrm{ii}}\, \phi_{\mathrm{i}}^{k+\frac{1}{2}}, \\ \rho_{\mathrm{ji}}^{k+1} = a_{\mathrm{ji}}\, z_{\mathrm{i}}^{k+\frac{1}{2}}, \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}, \\ \sigma_{\mathrm{ji}}^{k+1} = a_{\mathrm{ji}}\, \phi_{\mathrm{i}}^{k+\frac{1}{2}}, \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}; \end{cases} \tag{4.3}$$

$$y_{\mathrm{i}}^{k+1} = \frac{z_{\mathrm{i}}^{k+1}}{\phi_{\mathrm{i}}^{k+1}}; \tag{4.4}$$

where $\phi_{\mathrm{i}}^0 = 1$, $\rho_{\mathrm{ij}}^k = 0$, and $\sigma_{\mathrm{ij}}^0 = 0$, for all $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$, and the weight-matrix $A \triangleq (a_{\mathrm{ij}})_{\mathrm{i,j}=1}^m$ satisfies the following assumption:

Assumption 4.2.2. *The weight matrix* $A \triangleq (a_{\mathrm{ij}})_{\mathrm{i,j}=1}^m$ *satisfy:*

- **(i)** $\exists\, \bar{m} > 0$ *such that:* $a_{\mathrm{ii}} \geq \bar{m}$, $\forall \mathrm{i} \in \mathcal{V}$; $a_{\mathrm{ij}} \geq \bar{m}$, *for all* $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$; *and* $a_{\mathrm{ij}} = 0$, *otherwise;*
- **(ii)** A *is column-stochastic, that is,* $A^\top \mathbf{1} = \mathbf{1}$.

In words, at iteration k, every agent i first performs the "sum" step (4.2) and builds the new mass $z_{\mathrm{i}}^{k+1/2}$: it sums its current information z_{i}^k with the one broadcasted by its in-neighbors–ρ_{ij}^k is the information sent to i by agent $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$. Then, the "push" step (4.3) follows: $z_{\mathrm{i}}^{k+1/2}$ is "pushed back" (sent) to the out-neighbors $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}$ and agent i itself; out of the total mass $z_{\mathrm{i}}^{k+1/2}$, each $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}$ receives the fraction $\rho_{\mathrm{ji}}^{k+1} = a_{\mathrm{ji}}\, z_{\mathrm{i}}^{k+1/2}$, with agent i getting $a_{\mathrm{ii}}\, z_{\mathrm{i}}^{k+1/2}$, which determines the update $z_{\mathrm{i}}^{k} \to z_{\mathrm{i}}^{k+1}$. It is not difficult to check that the overall mass in the system does not change over the time and equals the initial mass:

$$\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{k+1} + \sum_{(\mathrm{j},\mathrm{i})\in\mathcal{E}} \rho_{\mathrm{ij}}^{k+1} = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{k} + \sum_{(\mathrm{j},\mathrm{i})\in\mathcal{E}} \rho_{\mathrm{ij}}^{k} = \cdots = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{0}.$$

The ϕ-variables follow the same evolution of the z-variables, thus satisfying a similar property. Finally, consistently with the push-sum, the y-variables in (4.4), can be regarded as agent i's estimate of the average. In fact, it is not difficult to check that, if a consensus is achieved on the y_{i}'s, i.e., $\lim_{k\to\infty} z_{\mathrm{i}}^{k+1}/\phi_{\mathrm{i}}^{k+1} = c^{\infty}$ for all $\mathrm{i} \in \mathcal{V}$, then it must be $c^{\infty} = (1/m) \cdot \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{0}$.

In the following, we break the synchronism in the sum-push scheme.

4.3 Perturbed Asynchronous Sum-Push

Consider the following asynchronous setting: i) multiple agents compute and communicate independently without coordination; ii) communication latency and uncoodinated computations result in (possibly time-varying) delays. This means that some agents can execute more iterations than others and, in general they no longer use the most recent information from its neighbors; also, some information can get lost. As a consequence, the key property of the synchronous sum-push–the preservation of the overall mass–would not be guaranteed.

We robustify the sum-push building on the idea first introduced in [73] and further developed in [74], [77]: each ρ_{ji} (resp. σ_{ji}) no longer represents the *current* mass-fraction $a_{\mathrm{ji}}\, z_{\mathrm{i}}$ (resp. $a_{\mathrm{ji}}\, \phi_{\mathrm{i}}$), meant for node $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}$, but it is instead the *running-sum* of the mass $a_{\mathrm{ji}}\, z_{\mathrm{i}}$ (resp. $a_{\mathrm{ji}}\, \phi_{\mathrm{i}}$) that has been generated for j *up to the current activation* of agent i. In addition, every agent i maintains, for every $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$, a local buffer $\tilde{\rho}_{\mathrm{ij}}$ (resp. $\tilde{\sigma}_{\mathrm{ij}}$) storing the value of ρ_{ij} (resp. σ_{ij}) that it has used in its *last* (past) update. With this construction, we build next one iteration of the asynchronous sum-push algorithm. We discuss the updates of the z, ρ, and $\tilde{\rho}$-variables only; the one of the ϕ, σ, and $\tilde{\sigma}$ follows the same argument.

Suppose agent i^k wakes up at iteration k. The state of agent i^k is described by the variables z_{i^k}, ϕ_{i^k}, $\rho_{\mathrm{j}\mathrm{i}^k}$, $\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}$, $\sigma_{\mathrm{j}\mathrm{i}^k}$, and $\tilde{\sigma}_{\mathrm{i}^k\mathrm{j}}$. However, the ρ-variables may no longer contain the current information from its in-neighbors. More specifically, agent i^k does not have access to the current vector $\rho^k_{\mathrm{i}^k\mathrm{j}}$ from $\mathrm{j} \in \mathcal{N}^{\mathrm{in}}_{\mathrm{i}^k}$, but it will use instead the delayed estimate $\rho^{k-d^k_{\mathrm{j}}}_{\mathrm{i}^k\mathrm{j}}$, where $0 \leq d^k_{\mathrm{ij}} \leq D$ is the delay (assumed to be bounded). By definition, the local buffer $\tilde{\rho}^k_{\mathrm{i}^k\mathrm{j}}$ stores the value of $\rho_{\mathrm{i}^k\mathrm{j}}$ that agent i^k used in its previous update. If the information in $\rho^{k-d^k_{\mathrm{j}}}_{\mathrm{i}^k\mathrm{j}}$ is not older than the one in $\tilde{\rho}^k_{\mathrm{i}^k\mathrm{j}}$, the difference $\rho^{k-d^k_{\mathrm{j}}}_{\mathrm{i}^k\mathrm{j}} - \tilde{\rho}^k_{\mathrm{i}^k\mathrm{j}}$ will capture the sum of the $a_{\mathrm{i}^k\mathrm{j}}z_{\mathrm{j}}$'s that have been generated by $\mathrm{j} \in \mathcal{N}^{\mathrm{in}}_{\mathrm{i}^k}$ for i^k up until $k - d^k_{\mathrm{j}}$ and not used by agent i^k yet; otherwise $\rho^{k-d^k_{\mathrm{j}}}_{\mathrm{i}^k\mathrm{j}}$ will be discarded, as no new information has been acquired. For instance, in a synchronous setting, one would have $\rho^k_{\mathrm{ij}} - \tilde{\rho}^k_{\mathrm{ij}} = a_{\mathrm{ij}}z^k_{\mathrm{j}}$. This naturally suggests the following modification of the steps (4.3)-(4.4) to preserve the total mass of the system at every iteration:

$$\text{Sum:} \qquad z^{k+\frac{1}{2}}_{\mathrm{i}^k} = z^k_{\mathrm{i}^k} + \sum_{\mathrm{j}\in\mathcal{N}^{\mathrm{in}}_{\mathrm{i}^k}} \left(\rho^{k-d^k_{\mathrm{j}}}_{\mathrm{i}^k\mathrm{j}} - \tilde{\rho}^k_{\mathrm{i}^k\mathrm{j}} \right), \tag{4.5}$$

$$\text{Push:} \qquad \begin{cases} z^{k+1}_{\mathrm{i}^k} = a_{\mathrm{i}^k\mathrm{i}^k}\, z^{k+\frac{1}{2}}_{\mathrm{i}^k}, \\ \rho^{k+1}_{\mathrm{j}\mathrm{i}^k} = \rho^k_{\mathrm{j}\mathrm{i}^k} + a_{\mathrm{j}\mathrm{i}^k}\, z^{k+\frac{1}{2}}_{\mathrm{i}^k}, \quad \forall \mathrm{j} \in \mathcal{N}^{\mathrm{out}}_{\mathrm{i}^k}; \end{cases} \tag{4.6}$$

$$\text{Mass-buffer:} \qquad \tilde{\rho}^{k+1}_{\mathrm{i}^k\mathrm{j}} = \rho^{k-d^k_{\mathrm{j}}}_{\mathrm{i}^k\mathrm{j}}, \qquad \forall \mathrm{j} \in \mathcal{N}^{\mathrm{in}}_{\mathrm{i}^k} \tag{4.7}$$

while $y^{k+1}_{\mathrm{i}^k} = z^{k+1}_{\mathrm{i}^k}/\phi^{k+1}_{\mathrm{i}^k}$ [cf. (4.4)]; where $\phi^0_{\mathrm{i}} = 1$, for all $\mathrm{i} \in \mathcal{V}$, and $\rho^k_{\mathrm{ij}} = \tilde{\rho}^k_{\mathrm{ij}} = 0$, for all $k = -D, \ldots, 0$, and $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$. Note that, differently from the synchronous case [cf. (4.3)], in (4.6), ρ^k_{ji} is now updated *recursively*, to build the running-sum of the mass $a_{\mathrm{ji}}z_{\mathrm{i}}$. After the push-step, in (5.7), the buffer is updated to account for the use of new information from $\mathrm{j} \in \mathcal{N}^{\mathrm{in}}_{\mathrm{i}^k}$.

With the above modifications, the total mass in the systems is preserved at each iteration, as shown next. Consider only the z-variables; similar argument applies to the ϕ-variables. The total mass associated with the z-variables at iteration k is defined as

$$\mathfrak{m}^k_z \triangleq \sum_{\mathrm{i}=1}^{m} z^k_{\mathrm{i}} + \sum_{(\mathrm{j},\mathrm{i})\in\mathcal{E}} (\rho^k_{\mathrm{ij}} - \tilde{\rho}^k_{\mathrm{ij}}). \tag{4.8}$$

We show next that $\mathfrak{m}_z^{k+1} = \mathfrak{m}_z^{k} = \cdots = \mathfrak{m}_z^{0} = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{0}$. Since agent i^k triggers $k \to k+1$, it is sufficient to show

$$\begin{aligned} & z_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k+1} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k+1}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k+1} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k+1}) \\ & = z_{\mathrm{i}^k}^{k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k}). \end{aligned} \tag{4.9}$$

Using (4.5)-(5.7), we can write

$$\begin{aligned} & z_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k+1} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k+1}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k+1} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k+1}) \\ & \overset{(a)}{=} a_{\mathrm{i}^k\mathrm{i}^k} z_{\mathrm{i}^k}^{k+\frac{1}{2}} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k} + a_{\mathrm{j}\mathrm{i}^k} z_{\mathrm{i}^k}^{k+\frac{1}{2}} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k}) \\ & \overset{(b)}{=} z_{\mathrm{i}^k}^{k+\frac{1}{2}} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k}) \\ & \overset{(4.5)}{=} z_{\mathrm{i}^k}^{k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k}) \\ & = z_{\mathrm{i}^k}^{k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{j}\mathrm{i}^k}^{k} - \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k}) \end{aligned} \tag{4.10}$$

where in (a) we used i) (4.6)-(5.7), ii) $\rho_{\mathrm{i}^k\mathrm{j}}^{k+1} = \rho_{\mathrm{i}^k\mathrm{j}}^{k}$, for all $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$, and iii) $\tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k+1} = \tilde{\rho}_{\mathrm{j}\mathrm{i}^k}^{k}$, for all $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}$; and in (b), we used $a_{\mathrm{i}^k\mathrm{i}^k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} a_{\mathrm{j}\mathrm{i}^k} = 1$. The mass preservation property above ensures that, if a consensus is reached, i.e., $\lim_{k\to\infty} z_{\mathrm{i}}^k/\phi_{\mathrm{i}}^k = c^\infty$ for all $\mathrm{i} \in \mathcal{V}$, then it must be $c^\infty = (1/m) \cdot \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{0}$.

4.3.1 P-ASY-PUSH-SUM

We are now ready to introduce P-ASY-SUM-PUSH. Consider an asynchronous setting wherein agents compute and communicate independently without coordination. Every agent i maintains state variables z_{i}, ϕ_{i}, y_{i}, along with the following auxiliary variables that are instrumental to deal with uncoordinated activations and delayed information: i) the cumulative-mass variables ρ_{ji} and σ_{ji}, with $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}$, which capture the cumulative (sum) information generated by agent i up to the current time and to be sent to agent $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}}$; consequently, ρ_{ij} and σ_{ij} are received by i from its in-neighbors $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$; and ii) the buffer variables $\tilde{\rho}_{\mathrm{ij}}$ and $\tilde{\sigma}_{\mathrm{ij}}$, with $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$, which store the information sent from $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$ to i and used by i in its last update. Values of these variables at iteration $k \in \mathbb{N}_0$ are denoted by the same symbols with the superscript "k". Note that, because of the asynchrony, each agent i might have outdated ρ_{ij} and σ_{ij}; $\rho_{\mathrm{ij}}^{k-d_{\mathrm{j}}^k}$ (resp. $\sigma_{\mathrm{ij}}^{k-d_{\mathrm{j}}^k}$) is a delayed version of the current ρ_{ij}^k (resp. σ_{ij}^k) owned by j at time k, where $0 \leq d_{\mathrm{j}}^k \leq D < \infty$ is the delay. Similarly, $\tilde{\rho}_{\mathrm{ij}}$ and $\tilde{\sigma}_{\mathrm{ij}}$ might differ from the last information generated by j for i, because agent i might not have received that information yet (due to delays) or never will (due to packet losses).

The proposed asynchronous algorithm, P-ASY-SUM-PUSH, is summarized in Algorithm 3. A global iteration clock (not known to the agents) is introduced: $k \to k+1$ is triggered based upon the completion from one agent, say i^k, of the following actions. **(S.2):** agent i^k maintains a local variable $\tau_{\mathrm{i}^k\mathrm{j}}$, for each $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$, which keeps track of the "age" (generated time) of the (ρ, σ)-variables that it has received from its in-neighbors and *already* used. If $k - d_{\mathrm{j}}^k$ is larger than the current counter $\tau_{\mathrm{i}^k\mathrm{j}}^{k-1}$, indicating that the received (ρ, σ)-variables are newer than those currently stored, agent i^k accepts $\rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}$ and $\sigma_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}$, and updates $\tau_{\mathrm{i}^k\mathrm{j}}$ as $k - d_{\mathrm{j}}^k$; otherwise, the variables will be discarded and $\tau_{\mathrm{i}^k\mathrm{j}}$ remains unchanged. Note that (4.11) can be performed without any coordination. It is sufficient that each agent attaches a time-stamp to its produced information reflecting it local timing counter. We describe next the other steps, assuming that new information has come in to agent i^k, that is, $\tau_{\mathrm{i}^k\mathrm{j}} = k - d_{\mathrm{j}}^k$. **(S.3.1):** In (4.12), agent i^k builds the intermediate "mass" $z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ based upon its current information $z_{\mathrm{i}^k}^k$ and $\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k$, and the (possibly) delayed one from its in-neighbors, $\rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}$; and $\epsilon^k \in \mathbb{R}^d$ is an exogenous perturbation (later this perturbation will be properly chosen to accomplish specific goals, see Sec. 5.3). Note that the way agent i^k forms its own estimates $\rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}$ is *immaterial* to the description of the algorithm. The local buffer $\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k$ stores the value of $\rho_{\mathrm{i}^k\mathrm{j}}$ that agent i^k used in its last update. Therefore, if the information in $\rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k}$ is not older than the one in $\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k$, the difference $\rho_{\mathrm{i}^k\mathrm{j}}^{k-d_{\mathrm{j}}^k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k$ in (4.12) will capture the sum of the $a_{\mathrm{i}^k\mathrm{j}} z_{\mathrm{j}}$'s that have been generated by $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$ for i^k up until $k - d_{\mathrm{j}}^k$ and not used by agent i^k yet. For instance, in a synchronous setting, one would have $\rho_{\mathrm{i}^k\mathrm{j}}^k - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k = a_{\mathrm{i}^k\mathrm{j}} z_{\mathrm{j}}^{k+\frac{1}{2}}$. **(S.3.2):** the generated $z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ is "pushed back" to agent i^k itself and its out-neighbors. Specifically, out

Algorithm 3 P-ASY-SUM-PUSH (Global View)

```
**Data:** $z_{\mathrm{i}}^0 \in \mathbb{R}^d$, $\phi_{\mathrm{i}}^0 = 1$, $\tilde{\boldsymbol{\rho}}_{\mathrm{ij}}^0 = 0$, $\tilde{\sigma}_{\mathrm{ij}}^0 = 0$, $\tau_{\mathrm{ij}}^{-1} = -D$, for all $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$ and $\mathrm{i} \in \mathcal{V}$; $\sigma_{\mathrm{ij}}^t = 0$ and $\boldsymbol{\rho}_{\mathrm{ij}}^t = 0$, for all $t = -D, \ldots, 0$; and $\{\epsilon^k\}_{k \in \mathbb{N}_0}$. Set $k = 0$.
**While:** a termination criterion is not met **do**
    (S.1) Pick $(\mathrm{i}^k, d^k)$, with $d^k \triangleq (d_{\mathrm{j}}^k)_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}}$;
    (S.2) Set (purge out the old information):
        $$\tau_{\mathrm{i}^k\mathrm{j}}^k = \max\left(\tau_{\mathrm{i}^k\mathrm{j}}^{k-1}, k - d_{\mathrm{j}}^k\right), \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}; \tag{4.11}$$
    (S.3) Update the variables performing
    • (S.3.1) **Sum step:**
        $$z_{\mathrm{i}^k}^{k+\frac{1}{2}} = z_{\mathrm{i}^k}^k + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} \left(\rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k\right) + \epsilon^k \tag{4.12}$$
        $$\phi_{\mathrm{i}^k}^{k+\frac{1}{2}} = \phi_{\mathrm{i}^k}^k + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} \left(\sigma_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k} - \tilde{\sigma}_{\mathrm{i}^k\mathrm{j}}^k\right)$$
    • (S.3.2) **Push step:**
        $$\begin{aligned} z_{\mathrm{i}^k}^{k+1} &= a_{\mathrm{i}^k\mathrm{i}^k}\, z_{\mathrm{i}^k}^{k+\frac{1}{2}}, \quad \phi_{\mathrm{i}^k}^{k+1} = a_{\mathrm{i}^k\mathrm{i}^k}\, \phi_{\mathrm{i}^k}^{k+\frac{1}{2}} \\ \rho_{\mathrm{j}\mathrm{i}^k}^{k+1} &= \rho_{\mathrm{j}\mathrm{i}^k}^k + a_{\mathrm{j}\mathrm{i}^k}\, z_{\mathrm{i}^k}^{k+\frac{1}{2}}, \\ \sigma_{\mathrm{j}\mathrm{i}^k}^{k+1} &= \sigma_{\mathrm{j}\mathrm{i}^k}^k + a_{\mathrm{j}\mathrm{i}^k}\, \phi_{\mathrm{i}^k}^{k+\frac{1}{2}}, \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}} \end{aligned} \tag{4.13}$$
    • (S.3.3) **Mass-Buffer update:**
        $$\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k+1} = \rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}, \quad \tilde{\sigma}_{\mathrm{i}^k\mathrm{j}}^{k+1} = \sigma_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}, \qquad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}} \tag{4.14}$$
    • (S.3.4) **Set:**    $y_{\mathrm{i}^k}^{k+1} = z_{\mathrm{i}^k}^{k+1} / \phi_{\mathrm{i}^k}^{k+1}$.
    (S.4) Untouched state variables shift to state $k+1$
          while keeping the same value; $k \leftarrow k+1$.
```

of the total mass $z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ generated, agent i^k gets $a_{\mathrm{ii}}\, z_{\mathrm{i}}^{k+\frac{1}{2}}$, determining the update $z_{\mathrm{i}}^k \to z_{\mathrm{i}}^{k+1}$ while the remaining is allocated to the agents $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}$, with $a_{\mathrm{j}\mathrm{i}^k}\, z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ cumulating to the mass buffer $\rho_{\mathrm{j}\mathrm{i}^k}^k$ and generating the update $\rho_{\mathrm{j}\mathrm{i}^k}^k \to \rho_{\mathrm{j}\mathrm{i}^k}^{k+1}$, to be sent to agent j. **(S.3.3):** each local buffer variable $\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k$ is updated to account for the use of new information from $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$. The final information is then read on the y-variables [cf. **(S.3.4)**].

Remark 4.3.1. *(Global view description)* Note that each agent's update is fully defined, once i^k and d^k are given. The selection (i^k, d^k) in **(S.1)** is not performed by anyone; it is instead an *a-posteriori* description of agents' actions: All agents act asynchronously and continuously; the agent completing the "push" step and updating its own variables triggers *retrospectively* the iteration counter $k \to k+1$ and determines the pair (i^k, d^k) along with all quantities involved in the other steps. Differently from most of the current literature, this "global view" description of the agents' actions allows us to abstract from specific computation-communication protocols and asynchronous modus operandi and captures by a unified model a gamut of asynchronous schemes.

Convergence is given under the following assumptions.

Assumption 4.3.2 (On the asynchronous model)**.** *Suppose:*

a. $\exists\ 0 < T < \infty$ *such that* $\cup_{t=k}^{k+T-1} \mathrm{i}^t = \mathcal{V}$*, for all* $k \in \mathbb{N}_0$*;*

b. $\exists\ 0 < D < \infty$ *such that* $0 \leq d_{\mathrm{j}}^k \leq D$*, for all* $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{in}$ *and* $k \in \mathbb{N}_0$*.* □

Assumption 4.3.2(a) is an essentially cyclic rule stating that within T iterations all agents will have updated at least once, which guarantees that all of them participate "sufficiently often". Assumption 4.3.2(b) requires bounded delay–old information must eventually be purged by the system. This asynchronous model is general and imposes no coordination among agents or specific communication/activation protocol.

The next theorem studies convergence of P-ASY-SUM-PUSH, establishing geometric decay of the error $\|y_{\mathrm{i}}^k - (1/m) \cdot \mathfrak{m}_z^k\|$, even in the presence of unknown perturbations, where $\mathfrak{m}_z^k \triangleq \sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^k + \sum_{(\mathrm{j},\mathrm{i}) \in \mathcal{E}} (\rho_{\mathrm{ij}}^k - \tilde{\rho}_{\mathrm{ij}}^k)$ represents the "total mass" of the system at iteration k.

Theorem 4.3.3. *Let* $\{y^k \triangleq [y_1^k, \ldots, y_m^k]^\top$, $z^k \triangleq [z_1^k, \ldots, z_m^k]^\top$, $(\rho_{\mathrm{ij}}^k, \tilde{\rho}_{\mathrm{ij}}^k)_{(\mathrm{j},\mathrm{i}) \in \mathcal{E}}\}_{k \in \mathbb{N}_0}$ *be the sequence generated by Algorithm 3, under Assumption 4.2.1, 4.3.2, and with* $A \triangleq (a_{\mathrm{ij}})_{\mathrm{i},\mathrm{j}=1}^m$ *satisfying Assumption 4.2.2. Define* $K_1 \triangleq (2\,m - 1) \cdot T + m \cdot D$*. There exist constants* $\rho \in (0,1)$ *and* $C_1 > 0$*, such that*

$$\left\| y_{\mathrm{i}}^{k+1} - (1/m) \cdot \mathfrak{m}_z^{k+1} \right\| \leq C_1 \left(\rho^k \left\| z^0 \right\| + \sum_{l=0}^{k} \rho^{k-l} \left\| \epsilon^l \right\| \right), \tag{4.15}$$

for all $\mathrm{i} \in \mathcal{V}$ *and* $k \geq K_1 - 1$.

Furthermore, $\mathfrak{m}_z^k = \sum_{\mathrm{i}=1}^m z_\mathrm{i}^0 + \sum_{t=0}^{k-1} \epsilon^t$.

Proof. See Sec. 4.4. □

Discussion: Several comments are in order.

- On the asynchronous model

Algorithm 3 captures a gamut of asynchronous *parallel* schemes and architectures, through the mechanism of generation of (i^k, d^k). Assumption 4.3.2 on (i^k, d^k) is quite mild: (a) controls the frequency of the updates whereas (b) limits the age of the old information used in the computations; they can be easily enforced in practice. For instance, (a) is readily satisfied if each agent wakes up and performs an update whenever some independent internal clock ticks or it is triggered by some of the neighbors; (b) imposes conditions on the frequency and quality of the communications: information used by each agent cannot become infinitely old, implying that successful communications must occur sufficiently often. This however does not enforce any specific protocol on the activation/idle time/communication. For instance, i) agents need not perform the actions in Algorithm 3 sequentially or inside the same activation round; or ii) executing the "push" step does not mean that agents must broadcast their new variables in the same activation; this would just incur a delay (or packet loss) in the communication.

Note that the time-varying nature of the delays d^k permits to model also packet losses, as detailed next. Suppose that at iteration k_1 agent j sends its current ρ, σ-variables to its out-neighbor ℓ and they get lost; and let k_2 be the subsequent iteration when j updates again. Let t be the first iteration after k_1 when agent ℓ performs its update; it will use information from j such that $t - d_\mathrm{j}^t \notin [k_1 + 1, k_2]$, for some $d_\mathrm{j}^t \leq D < \infty$. If $t - d_\mathrm{j}^t < k_1 + 1$, no newer information from j has been used by ℓ; otherwise $t - d_\mathrm{j}^t \geq k_2 + 1$ (implying $k_2 < t$), meaning that agent ℓ has used information not older than $k_2 + 1$.

- Beyond average consensus By choosing properly the perturbation signal ϵ^k, P-ASY-SUM-PUSH can solve different problems. Some examples are discussed next.

(i) Error free: $\epsilon^k = 0$. P-ASY-SUM-PUSH solves the average consensus problem and (4.15) reads

$$\left\| y_{\mathrm{i}}^{k+1} - (1/m) \cdot \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{0} \right\| \leq C_1\, \rho^k \left\| z^0 \right\|.$$

(ii) Vanishing error: $\lim_{k\to\infty} \|\epsilon^k\| = 0$. Using [7, Lemma 7(a)], (4.15) reads $\lim_{k\to\infty} \|y_{\mathrm{i}}^{k+1} - \mathfrak{m}_z^{k+1}\| = 0$.

(iii) Asynchronous tracking. Each agent i owns a (time-varying) signal $\{u_{\mathrm{i}}^k\}_{k\in\mathbb{N}_0}$; the average tracking problem consists in asymptotically track the average signal $\bar{u}^k \triangleq (1/m)\cdot\sum_{\mathrm{i}=1}^m u_{\mathrm{i}}^k$, that is,

$$\lim_{k\to\infty} \left\| y_{\mathrm{i}}^{k+1} - \bar{u}^{k+1} \right\| = 0, \quad \forall \mathrm{i} \in \mathcal{V}. \tag{4.16}$$

Under mild conditions on the signal, this can be accomplished in a distributed and asynchronous fashion, using P-ASY-SUM-PUSH, as formalized next.

Corollary 4.3.3.1. *Consider, the following setting in* P-ASY-SUM-PUSH*:* $z_{\mathrm{i}}^0 = u_{\mathrm{i}}^0$*, for all* $\mathrm{i} \in \mathcal{V}$*;* $\epsilon^k = u_{\mathrm{i}^k}^{k+1} - \tilde{u}_{\mathrm{i}^k}^k$*, with*

$$\tilde{u}_{\mathrm{i}}^{k+1} = \begin{cases} u_{\mathrm{i}}^{k+1} & \textit{if } \mathrm{i} = \mathrm{i}^k; \\ \tilde{u}_{\mathrm{i}}^{k} & \textit{otherwise}; \end{cases} \qquad \tilde{u}_{\mathrm{i}}^0 = u_{\mathrm{i}}^0;$$

Then (4.15) *holds, with* $\mathfrak{m}_z^{k+1} = \sum_{\mathrm{i}=1}^m \tilde{u}_{\mathrm{i}}^{k+1}$*. Furthermore, if* $\lim_{k\to\infty} \sum_{\mathrm{i}=1}^m \left\| u_{\mathrm{i}}^{k+1} - u_{\mathrm{i}}^k \right\| = 0$*, then* (4.16) *holds.*

Proof. We know that $\mathfrak{m}_z^k = \sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^0 + \sum_{t=0}^{k-1} \epsilon^t$. Clearly $\mathfrak{m}_z^0 = \sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^0 = \sum_{\mathrm{i}=1}^m \tilde{u}_{\mathrm{i}}^0$. Suppose for $k = \ell$, we have that $\mathfrak{m}_z^\ell = \sum_{\mathrm{i}=1}^m \tilde{u}_{\mathrm{i}}^\ell$. Then we have that

$$\mathfrak{m}_z^{\ell+1} = \mathfrak{m}_z^\ell + \epsilon^\ell = \left(\sum_{\mathrm{i}=1}^m \tilde{u}_{\mathrm{i}}^\ell \right) + u_{\mathrm{i}^\ell}^{\ell+1} - \tilde{u}_{\mathrm{i}^\ell}^\ell = \sum_{\mathrm{j}\neq\mathrm{i}^\ell} \tilde{u}_{\mathrm{j}}^\ell + u_{\mathrm{i}^\ell}^{\ell+1} = \sum_{\mathrm{i}=1}^m \tilde{u}_{\mathrm{i}}^{\ell+1}.$$

Thus we have that $\mathfrak{m}_z^k = \sum_{\mathrm{i}=1}^m \tilde{u}_{\mathrm{i}}^k, \quad \forall k \in \mathbb{N}_0$.

Now we assume that $\lim_{k\to\infty} \sum_{\mathrm{i}=1}^m \left| u_{\mathrm{i}}^{k+1} - u_{\mathrm{i}}^k \right| = 0$. Notice that for $k \geq T$,

$$\left| \epsilon^k \right| = \left| u_{\mathrm{i}^k}^{k+1} - \tilde{u}_{\mathrm{i}^k}^k \right| \leq \sum_{t=k-T+1}^{k} \left| u_{\mathrm{i}^k}^{t+1} - u_{\mathrm{i}^k}^t \right| \leq \sum_{t=k-T+1}^{k} \sum_{\mathrm{i}=1}^{m} \left| u_{\mathrm{i}}^{t+1} - u_{\mathrm{i}}^t \right|.$$

Therefore we have that $\lim_{k\to\infty}\left|\epsilon^k\right| = 0$. According to Theorem 4.3.3 and [7, Lemma 7(a)], we have that

$$\lim_{k\to\infty}\left|\mathrm{tr}_{\mathrm{i}}^{k+1} - (1/m)\cdot\sum_{\mathrm{i}=1}^{m}\tilde{u}_{\mathrm{i}}^{k+1}\right| = 0.$$

On the other hand, we have that

$$\left|\sum_{\mathrm{i}=1}^{m} u_{\mathrm{i}}^{k+1} - \sum_{\mathrm{i}=1}^{m}\tilde{u}_{\mathrm{i}}^{k+1}\right| \leq \sum_{\mathrm{i}=1}^{m}\left|u_{\mathrm{i}}^{k+1} - \tilde{u}_{\mathrm{i}}^{k+1}\right| \leq \sum_{\mathrm{i}=1}^{m}\sum_{t=k-T+1}^{k}\left|u_{\mathrm{i}}^{t+1} - u_{\mathrm{i}}^{t}\right| \xrightarrow{k\to\infty} 0.$$

By triangle inequality, we get that

$$\lim_{k\to\infty}\left|\mathrm{tr}_{\mathrm{i}}^{k+1} - (1/m)\cdot\sum_{\mathrm{i}=1}^{m}u_{\mathrm{i}}^{k+1}\right| = 0.$$

□

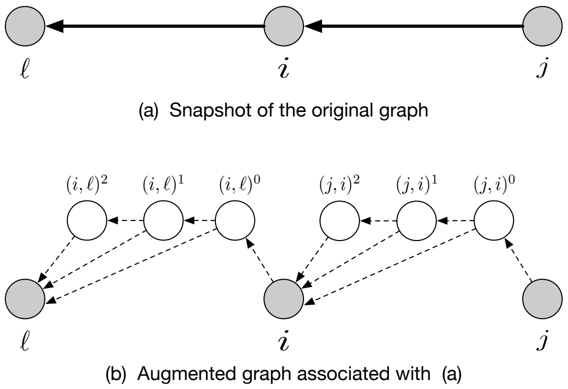
Remark 4.3.4 (Asynchronous average consensus)**.** To the best of our knowledge, the error-free instance of the P-ASY-SUM-PUSH discussed above is the first (stepsize-free) scheme that provably solves the *average* consensus problem at a linear rate, under the general asynchronous model described by Assumption 4.3.2. In fact, the existing asynchronous consensus schemes [75] [76] achieve an agreement among the agents' local variables whose value is not in general the average of their initial values, but instead some *unknown* function of them and the asynchronous modus operandi of the agents. Related to the P-ASY-SUM-PUSH is the ra-AC algorithm in [74], which enjoys the same convergence property but under a more restrictive and specific asynchronous model (no delays but packet losses and single-agent activation per iteration).

4.4 Convergence Analysis of P-ASY-SUM-PUSH

We prove Theorem 4.3.3; we assume $d = 1$, without loss of generality. The proof is organized in the following two steps. **Step 1:** We first reduce the asynchronous agent system to a synchronous "augmented" one with no delays. This will be done adding virtual agents to the graph $\mathcal{G}$ along with their state variables, so that P-ASY-SUM-PUSH will be rewritten as a (synchronous) perturbed push-sum algorithm on the augmented graph. While this idea was first explored in [73], [75], there are some important differences between the proposed enlarged systems and those used therein, see Remark 4.4.1. **Step 2:** We conclude the proof establishing convergence of the perturbed push-sum algorithm built in Step 1.

4.4.1 Step 1: Reduction to a synchronous perturbed push-sum

- The augmented graph We begin constructing the augmented graph–an enlarged agent system obtained adding virtual agents to the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Specifically, we associate to each edge $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$ an ordered set of virtual nodes (agents), one for each of the possible delay values, denoted with a slight abuse of notation by $(\mathrm{j}, \mathrm{i})^0, (\mathrm{j}, \mathrm{i})^1, \ldots, (\mathrm{j}, \mathrm{i})^D$; see Fig. 4.1. Roughly speaking, these virtual nodes store the "information on fly" based upon its associated delay, that is, the information that has been generated by $\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$ for i but not used (received) by i yet. Adopting the terminology in [75], nodes in the original graph $\mathcal{G}$ are termed *computing agents* while the virtual nodes will be called *noncomputing agents*. With a slight abuse of notation, we define the set of computing and noncomputing agents as $\widehat{\mathcal{V}} \triangleq \mathcal{V} \cup \{(\mathrm{i}, \mathrm{j})^d | \, (\mathrm{i}, \mathrm{j}) \in \mathcal{E}, \, d = 0, 1, \ldots, D\}$, and its cardinality as $S \triangleq \left|\widehat{\mathcal{V}}\right| = (m + (D+1)\,|\mathcal{E}|)$. We now identify the neighbors of each agent in this augmented systems. Computing agents no longer communicate among themselves; each $\mathrm{j} \in \mathcal{V}$ can only send information to the noncomputing nodes $(\mathrm{j}, \mathrm{i})^0$, with $\mathrm{i} \in \mathcal{N}_{\mathrm{j}}^{\mathrm{out}}$. Each noncomputing agent $(\mathrm{j}, \mathrm{i})^d$ can either send information to the next noncomputing agent, that is $(\mathrm{j}, \mathrm{i})^{d+1}$ (if any), or to the computing agent i; see Fig. 4.1(b).



(a) Snapshot of the original graph

(b) Augmented graph associated with (a)

Figure 4.1. Example of augmented graph, when the maximum delay $D = 2$; three noncomputing agents are added for each edge $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$.

To describe the information stored by the agents in the augmented system at each iteration, let us first introduce the following quantities: $\mathcal{T}_{\mathrm{i}} \triangleq \left\{k \,\middle|\, \mathrm{i}^k = \mathrm{i},\, k \in \mathbb{N}_0\right\}$ is the set of global iteration indices at which the computing agent $\mathrm{i} \in \mathcal{V}$ wakes up; and, given $k \in \mathbb{N}_0$, let $\mathcal{T}_{\mathrm{i}}^k \triangleq \left\{t \in \mathcal{T}_{\mathrm{i}} \,\middle|\, t \leq k\right\}$. It is not difficult to conclude from (4.13) and (4.14) that

$$\rho_{\mathrm{ij}}^k = \sum_{t \in \mathcal{T}_{\mathrm{j}}^{k-1}} a_{\mathrm{ij}} z_{\mathrm{j}}^{t+1/2} \text{ and } \tilde{\rho}_{\mathrm{ij}}^k = \rho_{\mathrm{ij}}^{\tau_{\mathrm{ij}}^{k-1}}, \quad (\mathrm{j},\mathrm{i}) \in \mathcal{E}. \tag{4.17}$$

At iteration $k = 0$, every computing agent i stores z_{i}^0, whereas the values of the noncomputing agents are initialized to 0. At the beginning of iteration k, every computing agent i will store z_{i}^k whereas every noncomputing agent $(\mathrm{j},\mathrm{i})^d$, with $0 \leq d \leq D-1$, stores the mass $a_{\mathrm{ij}} z_{\mathrm{j}}$ (if any) generated by j for i at iteration $k-d-1$ (thus $k-d-1 \in \mathcal{T}_{\mathrm{j}}^{k-1}$), i.e., $a_{\mathrm{ij}} z_{\mathrm{j}}^{k-(d+1)+1/2}$ (cf. Step 3.2), and not been used by i yet (thus $k - d > \tau_{\mathrm{ij}}^{k-1}$); otherwise it stores 0. Formally, we have

$$z_{(\mathrm{j},\mathrm{i})^d}^k \triangleq \ a_{\mathrm{ij}} z_{\mathrm{j}}^{t+1/2} \cdot \mathbb{1}\Big[t = k-d-1 \in \mathcal{T}_{\mathrm{j}}^{k-1} \ \& \ t+1 > \tau_{\mathrm{ij}}^{k-1}\Big]. \tag{4.18}$$

The virtual node $(\mathrm{j},\mathrm{i})^D$ cumulates all the masses $a_{\mathrm{ij}} z_{\mathrm{j}}^{k-(d+1)+1/2}$ with $d \geq D$, not received by i yet:

$$z_{(\mathrm{j},\mathrm{i})^D}^k \triangleq \sum_{t \in \mathcal{T}_{\mathrm{j}}^{k-D-1},\, t+1 > \tau_{\mathrm{ij}}^{k-1}} a_{\mathrm{ij}} z_{\mathrm{j}}^{t+1/2}. \tag{4.19}$$

We write next P-ASY-SUM-PUSH on the augmented graph in terms of the z-variables of both the computing and noncomputing agents, absorbing the $(\rho, \tilde{\rho})$-variables using (4.17)-(4.19).

The sum-step over the augmented graph. In the sum-step, the update of the z-variables of the computing agents reads:

$$z_{\mathrm{i}^k}^{k+\frac{1}{2}} = z_{\mathrm{i}^k}^k + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} \left(\rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k\right) + \epsilon^k \overset{(4.17)-(4.19)}{=} z_{\mathrm{i}^k}^k + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} \sum_{d=k-\tau_{\mathrm{i}^k\mathrm{j}}^k}^{D} z_{(\mathrm{j},\mathrm{i}^k)^d}^k + \epsilon^k; \tag{4.20a}$$

$$z_{\mathrm{j}}^{k+\frac{1}{2}} = z_{\mathrm{j}}^k, \quad \mathrm{j} \in \mathcal{V} \setminus \{\mathrm{i}^k\}. \tag{4.20b}$$

In words, node i^k builds the update $z_{\mathrm{i}^k}^k \to z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ based upon the masses transmitted by the noncomputing agents $(\mathrm{j},\mathrm{i}^k)^{k-\tau_{\mathrm{i}^k\mathrm{j}}^k}, (\mathrm{j},\mathrm{i}^k)^{k-\tau_{\mathrm{i}^k\mathrm{j}}^k+1}, \ldots, (\mathrm{j},\mathrm{i}^k)^D$ [cf. (4.20a)]. All the other com-

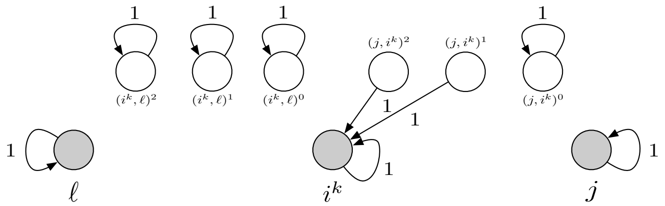


Figure 4.2. Sum step on the augmented graph: $\tau^k_{\mathrm{i}^k\mathrm{j}} = k - 1$ (delay one); the two noncomputing agents, $(\mathrm{j}, \mathrm{i}^k)^1$ and $(\mathrm{j}, \mathrm{i}^k)^2$, send their masses to i^k.

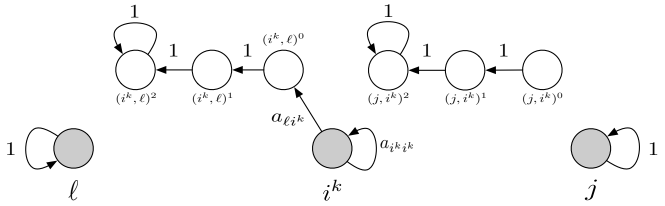


Figure 4.3. Push step on the augmented graph: Agent i^k keeps $a_{\mathrm{i}^k\mathrm{i}^k} z^{k+1/2}_{\mathrm{i}^k}$ while sending $a_{\ell\mathrm{i}^k} z^{k+1/2}_{\mathrm{i}^k}$ to the virtual nodes $(\mathrm{i}^k, \ell)^0$, $\ell \in \mathcal{N}^{\mathrm{out}}_{\mathrm{i}^k}$.

puting agents keep their masses unchanged [cf. (4.20b)]. The updates of the noncomputing agents is set to

$$z^{k+\frac{1}{2}}_{(\mathrm{j},\mathrm{i}^k)^d} \triangleq 0, \quad d = k - \tau^k_{\mathrm{i}^k\mathrm{j}}, \ldots, D, \quad \mathrm{j} \in \mathcal{N}^{\mathrm{in}}_{\mathrm{i}^k}; \tag{4.20c}$$

$$z^{k+\frac{1}{2}}_{(\mathrm{j}',\mathrm{i})^\tau} \triangleq z^k_{(\mathrm{j}',\mathrm{i})^\tau}, \quad \text{for all the other } (\mathrm{j}', \mathrm{i})^\tau \in \widehat{\mathcal{V}}. \tag{4.20d}$$

The noncomputing agents in (4.20c) set their variables to zero (as they transferred their masses to i^k) while the other noncomputing agents keep their variables unchanged [cf. (4.20d)]. Fig. 4.2 illustrates the sum-step over the augmented graph.

The push-step over the augmented graph. In the push-step, the update of the z-variables of the computing agents reads:

$$z^{k+1}_{\mathrm{i}^k} = a_{\mathrm{i}^k\mathrm{i}^k}\, z^{k+\frac{1}{2}}_{\mathrm{i}^k}; \tag{4.21a}$$

$$z^{k+1}_{\mathrm{j}} = z^{k+\frac{1}{2}}_{\mathrm{j}}, \qquad \text{for } \mathrm{j} \in \mathcal{V} \setminus \{\mathrm{i}^k\}. \tag{4.21b}$$

In words, agent i^k keeps the portion $a_{\mathrm{i}^k\mathrm{i}^k} z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ of the new generated mass [cf. (4.21a)] whereas the other computing agents do not change their variables [cf. (4.21b)]. The noncomputing agents update as:

$$z_{(\mathrm{i}^k,\ell)^0}^{k+1} \triangleq a_{\ell\mathrm{i}^k}\, z_{\mathrm{i}^k}^{k+1/2}, \quad \ell \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}; \tag{4.21c}$$

$$z_{(\mathrm{i},\mathrm{j})^0}^{k+1} \triangleq 0, \quad (\mathrm{i},\mathrm{j}) \in \mathcal{E}, \quad \mathrm{i} \neq \mathrm{i}^k; \tag{4.21d}$$

$$z_{(\mathrm{i},\mathrm{j})^d}^{k+1} \triangleq z_{(\mathrm{i},\mathrm{j})^{d-1}}^{k+\frac{1}{2}}, \quad d = 1, \ldots, D-1, \quad (\mathrm{i},\mathrm{j}) \in \mathcal{E}; \tag{4.21e}$$

$$z_{(\mathrm{i},\mathrm{j})^D}^{k+1} \triangleq z_{(\mathrm{i},\mathrm{j})^D}^{k+\frac{1}{2}} + z_{(\mathrm{i},\mathrm{j})^{D-1}}^{k+\frac{1}{2}}, \quad (\mathrm{i},\mathrm{j}) \in \mathcal{E}. \tag{4.21f}$$

In words, the computing agent i^k pushes its masses $a_{\ell\mathrm{i}^k} z_{\mathrm{i}^k}^{k+\frac{1}{2}}$ to the noncomputing agents $(\mathrm{i}^k,\ell)^0$, with $\ell \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}$ [cf. (4.21c)]. As the other noncomputing agents $(\mathrm{i},\mathrm{j})^0$, $\mathrm{i} \neq \mathrm{i}^k$, do not receive any mass for their associated computing agents, they set their variables to zero [cf. (4.21d)]. Finally the other noncomputing agents $(\mathrm{i},\mathrm{j})^d$, with $0 \leq d \leq D-1$, transfers their mass to the next noncomputing node $(\mathrm{j},\mathrm{i})^{d+1}$ [cf. (4.21f), (4.21e)]. This push-step is illustrated in Fig. 4.3.

The following result establishes the equivalence between the update of the enlarged system with that of Algorithm 3.

Proposition 4.4.1. *Consider the setting of Theorem 4.3.3. The values of the z-variables of the computing agents in* (4.20)-(4.21) *coincide with those of the z-variables generated by* P-ASY-SUM-PUSH *(Algorithm 3), for all iterations $k \in \mathbb{N}_0$.*

Proof. By construction, the updates of the computing agents as in (4.20a)-(4.20b) and (4.21a)-(4.21b) coincide with the z-updates in the sum- and push-steps of P-ASY-SUM-PUSH, respectively. Therefore, we only need to show that the updates of the noncomputing agents are consistent with those of the $(\rho, \tilde{\rho})$-variables in P-ASY-SUM-PUSH. This follows using (4.17) and noting that the updates (4.21c)-(4.21f) are compliant with (4.18) and (4.19). For instance, by (4.17)-(4.18), it must be $z_{(\mathrm{i}^k,\mathrm{j})^0}^{k+1} = a_{\mathrm{j}\mathrm{i}^k} z_{\mathrm{j}}^{t+1/2} \cdot \mathbb{1}[t = k \in \mathcal{T}_{\mathrm{i}^k}^k \text{ and } t+1 > \tau_{\mathrm{j}\mathrm{i}^k}^k] = a_{\mathrm{j}\mathrm{i}^k} z_{\mathrm{j}}^{k+1/2}$, which in fact coincides with (4.21c). The other equations (4.21d)–(4.21f) can be similarly validated. □

Proposition 4.4.1 opens the way to study convergence of P-ASY-SUM-PUSH via that of the synchronous perturbed push-sum algorithm (4.20)-(4.21). To do so, it is convenient to rewrite (4.20)-(4.21) in vector-matrix form, as described next.

We begin introducing an enumeration rule for the components of the z-vector in the augmented system. We enumerate all the elements of $\mathcal{E}$ as $1, 2, \ldots, |\mathcal{E}|$. The computing agents in $\widehat{\mathcal{V}}$ are indexed as in $\mathcal{V}$, that is, $1, 2, \ldots, m$. Each noncomputing agent $(\mathrm{j}, \mathrm{i})^d$ is indexed as $m + d\,|\mathcal{E}| + s$, where s is the index associated with (j, i) in $\mathcal{E}$; we will use interchangeably $z_{m+d|\mathcal{E}|+s}$ and $z_{(\mathrm{j},\mathrm{i})^d}$. We define the z-vector as $\widehat{z} = [z_\mathrm{i}]_{\mathrm{i}=1}^S$; and its value at iteration $k \in \mathbb{N}_0$ is denoted by $\widehat{z}^k$.

The transition matrix S^k of the sum step is defined as

$$S^k_{hm} \triangleq \begin{cases} 1, & \text{if } m \in \{(\mathrm{j}, \mathrm{i}^k)^d \mid k - \tau^k_{\mathrm{i}^k \mathrm{j}} \leq d \leq D\} \text{ and } h = \mathrm{i}^k; \\ 1, & \text{if } m \in \widehat{\mathcal{V}} \setminus \{(\mathrm{j}, \mathrm{i}^k)^d \mid k - \tau^k_{\mathrm{i}^k \mathrm{j}} \leq d \leq D\} \text{ and } h = m; \\ 0, & \text{otherwise.} \end{cases}$$

Let $\varepsilon^k \triangleq \epsilon^k \mathrm{e}_{\mathrm{i}^k}$ be the $S-$dimensional perturbation vector. The sum-step can be written in compact form as

$$\widehat{z}^{k+\frac{1}{2}} = S^k \widehat{z}^k + \varepsilon^k. \tag{4.22}$$

Define the transition matrix P^k of the push step as

$$P^k_{hm} \triangleq \begin{cases} a_{\mathrm{j}\mathrm{i}^k}, & \text{if } m = \mathrm{i}^k \text{ and } h = (\mathrm{j}, \mathrm{i}^k)^0, \mathrm{j} \in \mathcal{N}^{\text{out}}_{\mathrm{i}^k}; \\ a_{\mathrm{i}^k \mathrm{i}^k}, & \text{if } m = h = \mathrm{i}^k; \\ 1, & \text{if } m = h \in \mathcal{V} \setminus \mathrm{i}^k; \\ 1, & \text{if } m = (\mathrm{i}, \mathrm{j})^d,\ h = (\mathrm{i}, \mathrm{j})^{d+1},\ (\mathrm{i}, \mathrm{j}) \in \mathcal{E},\ 0 \leq d \leq D - 1; \\ 1, & \text{if } m = h = (\mathrm{i}, \mathrm{j})^D,\ (\mathrm{i}, \mathrm{j}) \in \mathcal{E}; \\ 0, & \text{otherwise} \end{cases}$$

Then, the push-step can be written as

$$\widehat{z}^{k+1} = P^k \widehat{z}^{k+\frac{1}{2}}. \tag{4.23}$$

Combing (4.22) and (4.23), yields

$$\widehat{z}^{k+1} = \widehat{A}^k \widehat{z}^k + p^k, \quad \widehat{A}^k \triangleq P^k S^k, \quad p^k \triangleq P^k \varepsilon^k. \tag{4.24}$$

The updates of the ϕ variables and the definition of the ϕ-vector are similar as above. In summary, the P-ASY-SUM-PUSH algorithm can be rewritten in compact form as

$$\widehat{z}^{k+1} = \widehat{A}^k \widehat{z}^k + p^k, \quad p^k = \epsilon^k \, P^k \, \mathrm{e}_{\mathrm{i}^k}; \tag{4.25a}$$

$$\widehat{\phi}^{k+1} = \widehat{A}^k \widehat{\phi}^k; \tag{4.25b}$$

with initialization: $z_{\mathrm{i}}^0 \in \mathbb{R}$ and $\phi_{\mathrm{i}}^0 = 1$, for $\mathrm{i} \in \mathcal{V}$; and $z_{\mathrm{i}}^0 = 0$ and $\phi_{\mathrm{i}}^0 = 0$, for $\mathrm{i} \in \widehat{\mathcal{V}} \setminus \mathcal{V}$.

Remark 4.4.1 (Comparison with [73]–[76])**.** *The idea of reducing asynchronous (consensus) algorithms into synchronous ones over an augmented system was already explored in [74]–[76]. However, there are several important differences between the models therein and the proposed augmented graph. First of all, [74] extends the analysis in [73] to deal with asynchronous activations, but both work consider only packet losses (no delays). Second, our augmented graph model departs from that in [75], [76] in the following aspects: i) in our model, the virtual nodes are associated with the edges of the original graph rather than the nodes; ii) the noncomputing nodes store the information on fly (i.e., generated by a sender but not received by the intended receiver yet), while in [75], [76], each noncomputing agent owns a delayed copy of the message generated by the associated computing agent; and iii) the dynamics* (4.25) *over the augmented graph used to describe the P-ASY-SUM-PUSH procedure is different from those of the asynchronous consensus schemes [75, (1)] and [76, (1)].*

4.4.2 Step 2: Proof of Theorem 4.3.3

- Preliminaries We begin studying some properties of the matrix product $\widehat{A}^{k:t}$, which will be instrumental to prove convergence of the perturbed push-sum scheme (4.25).

Lemma 4.4.2. *Let $\{\widehat{A}^k\}_{k \in \mathbb{N}_0}$ be the sequence of matrices in* (4.25)*, generated by Algorithm 3, under Assumption 4.3.2, and with $A \triangleq (a_{\mathrm{ij}})_{\mathrm{i,j}=1}^m$ satisfying Assumption 4.2.2. The following* *hold: for all $k \in \mathbb{N}_0$, a) $\widehat{A}^k$ is column stochastic; and b) the entries of the first m rows of $\widehat{A}^{k+K_1-1:k}$ are uniformly lower bounded by $\eta \triangleq \bar{m}^{K_1} \in (0,1)$, with $K_1 \triangleq (2I-1)\cdot T + m \cdot D$.*

Proof. The lemma essentially proves that $(\widehat{A}^{k+K_1-1:k})^\top$ is a SIA (Stochastic Indecomposable Aperiodic) matrix [76], by showing that for any time length of K_1 iterations, there exists a path from any node m in the augmented graph to any computing node h. While at a high level the proof shares some similarities with that of [75, Lemma 2] and [76, Lemma 5 (a)], there are important differences due to the distinct modeling of our augmented system.

We study any entry $\widehat{A}^{k+K_1-1}_{hm}$ with $m \in \widehat{\mathcal{V}}$ and $h \in \mathcal{V}$. We prove the result by considering the following four cases.

(i) Assume $h = m \in \mathcal{V}$. It is easy to check that $\widehat{A}^k_{hh} \geq \bar{m}$, for any $k \in \mathbb{N}_0$ and $h \in \mathcal{V}$. Therefore, $\widehat{A}^{k+s-1:k}_{hh} \geq \prod_{t=k}^{k+s-1} \widehat{A}^t_{hh} \geq \bar{m}^s$, for all $k, s \in \mathbb{N}_0$ and $h \in \mathcal{V}$.

(ii) Let $(m,h) \in \mathcal{E}$; and let s be the first time m wakes up in the interval $[k, k+T-1]$. We have $\widehat{A}^s_{(m,h)^0,m} = a_{hm}$. The information that node m sent to node $(m,h)^0$ at iteration s is received by node h when the information is on some virtual node $(m,h)^d$. We discuss separately the following three sub-cases for d: 1) $1 \leq d \leq D-1$; 2) $d = 0$; and 3) $d = D$.

1) $1 \leq d \leq D-1$: We have

$$\widehat{A}^{s+d:s+1}_{(m,h)^d,(m,h)^0} = \widehat{A}^{s+d}_{(m,h)^d,(m,h)^{d-1}} \cdots \widehat{A}^{s+1}_{(m,h)^1,(m,h)^0} = 1, \qquad \widehat{A}^{s+d+1}_{h,(m,h)^d} = a_{hh}.$$

Therefore, $\widehat{A}^{s+d+1:s}_{hm} = \widehat{A}^{s+d+1}_{h,(m,h)^d} \widehat{A}^{s+d:s+1}_{(m,h)^d,(m,h)^0}\ \widehat{A}^s_{(m,h)^0,m} = a_{hh}a_{hm} \geq \bar{m}^2$.

2) $d = 0$: We simply have

$$\widehat{A}^{s+1:s}_{hm} = \widehat{A}^{s+1}_{h,(m,h)^0} \widehat{A}^s_{(m,h)^0,m} = a_{hh}a_{hm} \geq \bar{m}^2.$$

Therefore, for $0 \leq d \leq D-1$,

$$\widehat{A}^{k+2T+D-1:k}_{hm} = \widehat{A}^{k+2T+D-1:s+d+2}_{hh} \widehat{A}^{s+d+1:s}_{hm} \widehat{A}^{s-1:k}_{mm} \geq \bar{m}^{k+2T+D-s-d-2} \bar{m}^2 \bar{m}^{s-k} \geq \bar{m}^{2T+D}.$$

3) $d = D$: Before agent j wakes up at time $s + D + \tau$, where $1 \leq \tau \leq T$, the information will stay on virtual nodes $(m, h)^D$. Once agent j wakes up, nodes $(m, h)^D$ will send all its information to it. Then we have

$$\widehat{A}^{s+D:s+1}_{(m,h)^D,(m,h)^0} = 1, \quad \widehat{A}^{s+D+\tau:s+D+1}_{h,(m,h)^D} = a_{hh}.$$

Similarly, we have

$$\widehat{A}^{k+2T+D-1}_{hm} = \widehat{A}^{k+2T+D-1:s+D+\tau+1}_{hh} \widehat{A}^{s+D+\tau:s+D+1}_{h,(m,h)^D} \widehat{A}^{s+D:s+1}_{(m,h)^D,(m,h)^0} \widehat{A}^{s}_{(m,h)^0,m} \widehat{A}^{s-1:k}_{mm} \geq \bar{m}^{2T+D}.$$

To summarize, in all of the three sub-cases, we have

$$\widehat{A}^{k+K_1-1}_{hm} \geq \widehat{A}^{k+K_1-1:k+2T+D}_{hh} \widehat{A}^{k+2T+D-1:k}_{hm} \geq \bar{m}^{K_1-2T-D} \bar{m}^{2T+D} = \bar{m}^{K_1}.$$

(iii) Let $m \neq h$ and $(m, h) \in \mathcal{V} \times \mathcal{V} \backslash \mathcal{E}$. Since the graph $(\mathcal{V}, \mathcal{E})$ is connected, there are mutually different agents $\mathrm{i}_1, \ldots, \mathrm{i}_r$, with $r \leq m - 2$, such that $(m, \mathrm{i}_1), (\mathrm{i}_1, \mathrm{i}_2), \ldots, (\mathrm{i}_{r-1}, \mathrm{i}_r), (\mathrm{i}_r, h) \subset \mathcal{E}$, which is actually a directed path from m to h in the graph $(\mathcal{V}, \mathcal{E})$. Then, by result proved in (ii), we have

$$\widehat{A}^{k+(m-1)(2T+D)-1:k}_{hm} \geq \widehat{A}^{k+(m-1)(2T+D)-1:k+(r+1)(2T+D)}_{hh} \widehat{A}^{k+(r+1)(2T+D)-1:k+r(2T+D)}_{h\mathrm{i}_r} \cdots \widehat{A}^{k+2T+D-1:k}_{\mathrm{i}_1 m}$$
$$\geq \bar{m}^{(m-r-2)(2T+D)} \bar{m}^{(r+1)(2T+D)} = \bar{m}^{(m-1)(2T+D)}.$$

We can then easily get $\widehat{A}^{k+K_1-1:k}_{hm} = \widehat{A}^{k+K_1-1:k+(m-1)(2T+D)}_{hh} \widehat{A}^{k+(m-1)(2T+D)-1:k}_{hm} \geq \bar{m}^{K_1}$.

(iv) If m is a virtual node, it must be associated with an edge $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$ and there exists $0 \leq d \leq D$ such that $m = (\mathrm{j}, \mathrm{i})^d$. A similar argument as in (ii) above shows that any information on m will eventually enter node i taking $1 \leq \tau \leq D + T$. That is, $\widehat{A}^{k+\tau-1:k}_{\mathrm{i}m} = a_{\mathrm{ii}}$, for some $1 \leq \tau \leq D + T$. On the other hand, by the above results, we know

$$\widehat{A}^{k+T+D+(m-1)(2T+D)-1:k+T+D}_{h\mathrm{i}} \geq \bar{m}^{(m-1)(2T+D)}.$$

Therefore,

$$\widehat{A}_{hm}^{k+K_1-1:k} \geq \widehat{A}_{h\mathrm{i}}^{k+K_1-1:k+T+D} \widehat{A}_{\mathrm{ii}}^{k+T+D-1:k+\tau} \widehat{A}_{\mathrm{i}m}^{k+\tau-1:k} \geq \bar{m}^{(m-1)(2T+D)} \bar{m}^{T+D-\tau} \bar{m} \geq \bar{m}^{K_1}$$

□

The key result of this section is stated next and shows that as $k-t$ increases, $\widehat{A}^{k:t}$ approaches a column stochastic rank one matrix at a linear rate. Given Lemma 4.4.2, the proof follows the path of [75, Lemma 4, Lemma 5], [76, Lemma 4, Lemma 5(b, c)] and thus is omitted.

Lemma 4.4.3. *In the setting above, there exists a sequence of stochastic vectors* $\{\xi^k\}_{k\in\mathbb{N}_0}$ *such that, for any* $k \geq t \in \mathbb{N}_0$ *and* $\mathrm{i}, \mathrm{j} \in \{1, \cdots, S\}$, *there holds*

$$\left|\widehat{A}_{\mathrm{ij}}^{k:t} - \xi_{\mathrm{i}}^k\right| \leq C\rho^{k-t}, \tag{4.26}$$

with

$$C \triangleq 2\frac{1+\bar{m}^{-K_1}}{1-\bar{m}^{K_1}}, \quad \rho \triangleq (1-\bar{m}^{K_1})^{\frac{1}{K_1}} \in (0,1).$$

Furthermore, $\xi_{\mathrm{i}}^k \geq \eta$, *for all* $\mathrm{i} \in \mathcal{V}$ *and* $k \in \mathbb{N}_0$.

- Proof of Theorem 4.3.3

Applying (4.25) telescopically, yields: $\widehat{z}^{k+1} = \widehat{A}^{k:0}\widehat{z}^0 + \sum_{l=1}^{k} \widehat{A}^{k:l}p^{l-1} + p^k$ and $\widehat{\phi}^{k+1} = \widehat{A}^{k:0}\widehat{\phi}^0$,which using the column stochasticity of $\widehat{A}^{k:t}$, yields

$$1^\top \widehat{z}^{k+1} = 1^\top \widehat{z}^0 + \sum_{l=0}^{k} 1^\top p^l, \quad 1^\top \widehat{\phi}^{k+1} = 1^\top \widehat{\phi}^0 = m. \tag{4.27}$$

Using (4.27) and $\phi_{\mathrm{i}}^{k+1} \geq m\eta$, for all $\mathrm{i} \in \mathcal{V}$ and $k \geq K_1 - 1$ [due to Lemma 4.4.2(b)], we have: for $\mathrm{i} \in \mathcal{V}$ and $k \geq K_1 - 1$,

$$\begin{aligned}
&\left|\frac{z_{\mathrm{i}}^{k+1}}{\phi_{\mathrm{i}}^{k+1}} - \frac{1^\top \widehat{z}^{k+1}}{m}\right| \leq \frac{1}{m\eta}\left|z_{\mathrm{i}}^{k+1} - \frac{\phi_{\mathrm{i}}^{k+1}}{m}(1^\top \widehat{z}^{k+1})\right| \leq \frac{1}{m\eta}\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k} 1^\top \widehat{z}^{k+1}\right| + \frac{1}{m\eta}\left|\left(\xi_{\mathrm{i}}^{k} - \frac{\phi_{\mathrm{i}}^{k+1}}{m}\right) 1^\top \widehat{z}^{k+1}\right| \\
&\leq \frac{1}{m\eta}\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k} 1^\top \widehat{z}^{k+1}\right| + \frac{1}{m\eta}\left|\xi_{\mathrm{i}}^{k} - \frac{\widehat{A}_{\mathrm{i},:}^{k:0}\widehat{\phi}^{0}}{m}\right| \cdot \left|1^\top \widehat{z}^{0} + \sum_{l=0}^{k} 1^\top p^{l}\right| \\
&\overset{(4.26)}{\leq} \frac{1}{m\eta}\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k} 1^\top \widehat{z}^{k+1}\right| + \frac{C\rho^{k}}{\sqrt{m}\eta}\left(\left\|z^{0}\right\| + \sum_{l=0}^{k}\left|\epsilon^{l}\right|\right) \qquad (4.28)
\end{aligned}$$

The next lemma provides a bound of $\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k} 1^\top \widehat{z}^{k+1}\right|$.

Lemma 4.4.4. *Let $\{\widehat{z}^{k}\}_{k=0}^{\infty}$ be the sequence generated by the perturbed system (4.25a), under Assumption 4.3.2, $A = (a_{\mathrm{ij}})_{\mathrm{i,j}=1}^{m}$ satisfying Assumption 4.2.2, and given $\{\epsilon^{k}\}_{k\in\mathbb{N}_0}$. For any $\mathrm{i} \in \mathcal{V}$ and $k \geq 0$, there holds*

$$\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k} 1^\top \widehat{z}^{k+1}\right| \leq C_0\left(\rho^{k}\left\|z^{0}\right\| + \sum_{l=0}^{k}\rho^{k-l}\left|\epsilon^{l}\right|\right), \qquad (4.29)$$

with $\{\xi^{k}\}_{d\in\mathbb{N}_0}$ defined in Lemma 4.4.3 and $C_0 \triangleq C\sqrt{2S}/\rho$.

Proof.

$$\begin{aligned}
&\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k} 1^\top \widehat{z}^{k+1}\right| \overset{(4.25a)}{=} \left|\left(\widehat{A}_{\mathrm{i},:}^{k:0}\widehat{z}^{0} + \sum_{l=1}^{k}\widehat{A}_{\mathrm{i},:}^{k:l}p^{l-1} + p_{\mathrm{i}}^{k}\right) - \xi_{\mathrm{i}}^{k}\left(1^\top \widehat{z}^{0} + \sum_{l=0}^{k} 1^\top p^{l}\right)\right| \\
&\leq \left|p_{\mathrm{i}}^{k}\right| + \left|1^\top p^{k}\right| + \left\|\widehat{A}_{\mathrm{i},:}^{k:0} - \xi_{\mathrm{i}}^{k} 1^\top\right\|\left\|\widehat{z}^{0}\right\| + \sum_{l=1}^{k}\left\|\widehat{A}_{\mathrm{i},:}^{k:l} - \xi_{\mathrm{i}}^{k} 1^\top\right\|\left\|p^{l-1}\right\| \\
&\overset{(4.26)}{\leq} \frac{\sqrt{S}}{\rho}C\left(\rho^{k}\left\|\widehat{z}^{0}\right\| + \sum_{l=0}^{k}\rho^{k-l}\left\|P^{l}\right\|\left|\epsilon^{l}\right|\right) \overset{(a)}{\leq} C_0\left(\rho^{k}\left\|z^{0}\right\| + \sum_{l=0}^{k}\rho^{k-l}\left|\epsilon^{l}\right|\right),
\end{aligned}$$

where in (a) we used $\left\|P^{l}\right\| \leq \sqrt{\left\|P^{l}\right\|_1\left\|P^{l}\right\|_\infty} \leq \sqrt{2}$. □

Combing Eq. (4.28) and (4.29) leads to

$$\left|\frac{z_{\mathrm{i}}^{k+1}}{\phi_{\mathrm{i}}^{k+1}} - \frac{1^\top \widehat{z}^{k+1}}{m}\right| \leq C_1\left(\rho^{k}\left\|z^{0}\right\| + \sum_{l=0}^{k}\rho^{k-l}\left|\epsilon^{l}\right|\right),$$

where we defined $C_1 \triangleq C_0 \cdot 2/(m\,\eta)$.

Recalling the definition of $\mathfrak{m}_z^k \triangleq \sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^k + \sum_{(\mathrm{j},\mathrm{i})\in\mathcal{E}}(\rho_{\mathrm{ij}}^k - \tilde{\rho}_{\mathrm{ij}}^k)$, to complete the proof, it remains to show that

$$\mathfrak{m}_z^k \overset{(m)}{=} \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^0 + \sum_{t=0}^{k-1} \epsilon^t \overset{(II)}{=} 1^\top \widehat{z}^k. \tag{4.30}$$

We prove next the equalities (m) and (II) separately.

Proof of (m): Since $\mathfrak{m}_z^0 = \sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^0$, it suffices to show that $\mathfrak{m}_z^{k+1} = \mathfrak{m}_z^k + \epsilon^k$ for all $k \in \mathbb{N}_0$. Since agent i^k triggers $k \to k+1$, we only need to show that

$$\begin{aligned}
&z_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k+1} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k+1}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k+1} - \tilde{\rho}_{\mathrm{ji}^k}^{k+1}) \\
=&z_{\mathrm{i}^k}^{k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k} - \tilde{\rho}_{\mathrm{ji}^k}^{k}) + \epsilon^k.
\end{aligned}$$

We have

$$\begin{aligned}
&z_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k+1} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k+1}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k+1} - \tilde{\rho}_{\mathrm{ji}^k}^{k+1}) \\
\overset{(a)}{=}&a_{\mathrm{i}^k\mathrm{i}^k} z_{\mathrm{i}^k}^{k+\frac{1}{2}} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k} + a_{\mathrm{ji}^k} z_{\mathrm{i}^k}^{k+\frac{1}{2}} - \tilde{\rho}_{\mathrm{ji}^k}^{k}) \\
\overset{(b)}{=}&z_{\mathrm{i}^k}^{k+\frac{1}{2}} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k} - \tilde{\rho}_{\mathrm{ji}^k}^{k}) \\
\overset{(c)}{=}&z_{\mathrm{i}^k}^{k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k}) + \epsilon^k + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k} - \tilde{\rho}_{\mathrm{ji}^k}^{k}) \\
=&z_{\mathrm{i}^k}^{k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} (\rho_{\mathrm{i}^k\mathrm{j}}^{k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k}) + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} (\rho_{\mathrm{ji}^k}^{k} - \tilde{\rho}_{\mathrm{ji}^k}^{k}) + \epsilon^k,
\end{aligned}$$

where in (a) we used: the definition of the push step, $\rho_{\mathrm{i}^k\mathrm{j}}^{k+1} = \rho_{\mathrm{i}^k\mathrm{j}}^{k}$ for all $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$, and $\tilde{\rho}_{\mathrm{ji}^k}^{k+1} = \tilde{\rho}_{\mathrm{ji}^k}^{k}$ for all $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}$; (b) follows from $a_{\mathrm{i}^k\mathrm{i}^k} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}} a_{\mathrm{ji}^k} = 1$; and in (c), we used the sum-step.

Proof of (II): Using (4.27), yields $1^\top \widehat{z}^{k+1} = 1^\top \widehat{z}^0 + \sum_{l=0}^k 1^\top p^l = 1^\top \widehat{z}^k + 1^\top \varepsilon^k = \sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^0 + \sum_{t=0}^k \epsilon^t$. □

5. ASYNCHRONOUS DECENTRALIZED ALGORITHM - PART II: ASY-SONATA

In this chapter, we study asynchronous multi-agent smooth unconstrained optimization, over static digraphs. Agents can perform their local computations as well as communicate with their immediate neighbors at any time, without any form of coordination or centralized scheduling; furthermore, when solving their local subproblems, they can use outdated information from their neighbors. The algorithm builds on the asynchronous tracking algorithm P-ASY-SUM-PUSH proposed in Chapter 4, whose goal is to estimate locally the sum of agents' gradients When applied to strongly convex functions, we prove that it converges at an R-linear (geometric) rate as long as the step-size is sufficiently small. A sublinear convergence rate is proved, when nonconvex problems and/or diminishing, *uncoordinated* step-sizes are considered. To the best of our knowledge, this is the first distributed algorithm with provable geometric convergence rate in such a general asynchronous setting. Numerical results demonstrate the efficacy of the proposed algorithm and validate our theoretical findings.

The novel results of this chapter have been published in

- Ye Tian, Ying Sun, and Gesualdo Scutari. "ASY-SONATA: Achieving linear convergence in distributed asynchronous multiagent optimization." In 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 543-551. IEEE, 2018.
- Ye Tian, Ying Sun, and Gesualdo Scutari. "Achieving Linear Convergence in Distributed Asynchronous Multiagent Optimization." IEEE Transactions on Automatic Control 65, no. 12 (2020): 5264-5279.

5.1 Introduction

We study convex and nonconvex distributed optimization over a network of agents, modeled as a directed fixed graph. Agents aim at cooperatively solving the optimization problem

$$\min_{x\in\mathbb{R}^d} F(x) = \sum_{\mathrm{i}=1}^{m} f_{\mathrm{i}}\big(x\big) \tag{5.1}$$

which is a special instance of the Problem (P), with $G = 0$ and $\mathcal{K} = \mathbb{R}^d$. Various asynchronous distributed/decentralized optimization algorithms have been studied in the literature–see Sec. 5.1.1 for an overview of related works. However, we are not aware of any distributed algorithm that is compliant to the asynchrony model (i)-(ii), discussed in Sec. 4.1, and distributed (nonconvex) setting above. Furthermore, when considering the special case of a strongly convex function F, it is not clear how to design a (first-order) *distributed asynchronous* algorithm (as specified above) that achieves *linear convergence* rate. This chapter answers these questions– see Sec. 5.1.2 and Table 5.1 for a summary of our contributions.

5.1.1 Literature Review

Since the seminal work [78], asynchronous parallelism has been applied to several *centralized* optimization algorithms, including block coordinate descent (e.g., [78]–[80]) and stochastic gradient (e.g., [81], [82]) methods. However, these schemes are not applicable to the networked setup considered in this chapter, because they would require the knowledge of the function F from each agent. *Distributed* methods exploring (some form of) asynchrony over networks with no centralized node have been studied in [33], [77], [83]–[98]. We group next these works based upon the asynchronous features (i)-(ii), discussed in Sec. 4.1.

(a) Random activations and no delays [33], [83]–[86]: These schemes considered distributed *convex* unconstrained optimization over *undirected* graphs. While substantially different in the form of the updates performed by the agents–[83], [84], [86] are instances of primal-dual (proximal-based) algorithms, [85] is an ADMM-type algorithm, while [33] is based on the distributed gradient tracking mechanism introduced in[5], [7], [99]–all these algorithms are asynchronous in the sense of feature (i) [but not (ii)]: at each iteration, a

Table 5.1. Comparison with state-of-art distributed asynchronous algorithms. Current schemes can deal with uncoordinated activations but only with some forms of delays. ASY-SONATA enjoys all the desirable features listed in the table.

Algorithm	Nonconvex Cost Function	No Idle Time	Arbitrary Delays	Parallel	Step Sizes		Digraph	Global Convergence to Exact Solutions	Rate Analysis	
					Fixed	Uncoordinated Diminishing			Linear Rate for Strongly Convex	Nonconvex
Asyn. Broadcast [93]				✓	✓	✓		In expectation (w. diminishing step)		
Asyn. Diffusion [94]					✓					
Asyn. ADMM [95]	✓				✓			Deterministic		
Dual Ascent in [96]		✓	Restricted	Restricted	✓					
ra-NRC [77]					✓		✓			
ARock [97]		✓	Restricted		✓			Almost surely	In expectation	
ASY-PrimalDual [98]		✓	Restricted		✓			Almost surely		
ASY-SONATA	✓	✓	✓	✓	✓	✓	✓	Deterministic	Deterministic	Deterministic

subset of agents [83], [84], [86] (or edge-connected agents [33], [85]), chosen at random, is activated, performing then their updates and communications with their immediate neighbors; between two activations, agents are assumed to be in *idle* mode (i.e., able to *continuously* receive information). However, *no form of delays* is allowed: every agent must perform its local computations/updates using the *most updated* information from its neighbors. This means that all the actions performed by the agent(s) in an activation must be completed before a new activation (agent) takes place (wakes-up), which calls for some coordination among the agents. Finally, no convergence rate was provided for the aforementioned schemes but [33], [85].

(b) Synchronous activations and delays [87]–[92]: These schemes considered distributed constrained *convex* optimization over *undirected* graphs. They study the impact of delayed gradient information [87], [88] or communication delays (fixed [89], uniform [88], [92] or time-varying [90], [91]) on the convergence rate of distributed gradient (proximal [87], [88] or projection-based [91], [92]) algorithms or dual-averaging distributed-based schemes [89], [90]. While these schemes are all synchronous [thus lacking of feature (i)], they can tolerate *communication delays* [an instantiation of feature (ii)], converging at a *sublinear rate* to an optimal solution. Delays must be such that no losses occur–every agent's message will eventually reach its destination within a finite time.

(c) Random/cyclic activations and some form of delays [77], [93]–[98]: The class of optimization problems along with the key features of the algorithms proposed in these papers are summarized in Table 5.1 and briefly discussed next. The majority of these works studied distributed (strongly) *convex* optimization over *undirected* graphs, with [94] assuming that all the functions f_i have the same minimizer, [95] considering also nonconvex objectives, and [77] being implementable also over digraphs. The algorithms in [93], [94] are gradient-based schemes; [95] is a decentralized instance of ADMM; [97] applies an asynchronous parallel ADMM scheme to distributed optimization; and [98] builds on a primal-dual method. The schemes in [77], [96] instead build on (approximate) second-order information. All these algorithms are asynchronous in the sense of feature (i): [93]–[95], [97], [98] considered random activations of the agents (or edges-connected agents) while [77], [96] studied deterministic, uncoordinated activation rules. As far as feature (ii) is concerned, some form of delays is allowed. More specifically, [77], [93]–[95] can deal with *packet losses*: the information sent by an agent to its neighbors either gets lost or received with *no delay*. They also assume that agents are *always in idle mode* between two activations. Closer to the proposed asynchronous framework are the schemes in [97], [98] wherein a probabilistic model is employed to describe the activation of the agents and the aged information used in their updates. The model requires that the random variables triggering the activation of the agents are i.i.d and *independent* of the delay vector used by the agent to performs its update. While this assumption makes the convergence analysis possible, in reality, there is a strong dependence of the delays on the activation index; see [80] for a detailed discussion on this issue and several counter examples. Other consequences of this model are: the schemes [97], [98] are *not parallel*–only one agent per time can perform the update–and a random self-delay must be used in the update of each agent (even if agents have access to their most recent information). Furthermore, [97] calls for the solution of a convex subproblem for each agent at every iteration. Referring to the convergence rate, [97] is the only scheme exhibiting linear convergence *in expectation*, when each f_i is strongly convex and the graph *undirected*. No convergence rate is available in any of the aforementioned papers, when F is nonconvex.

5.1.2 Summary of Contributions

The review of the literature clearly showed that there exits no distributed asynchronous [in the sense (i)-(ii)] scheme, even for convex instances of Problem (5.1) and undirected graphs. Furthermore, it is unknown whether one can design a geometric (globally) convergent scheme (when U is strongly convex).

This chapter proposes a general distributed, asynchronous algorithmic framework for (strongly) convex and *nonconvex* instances of Problem (5.1), over *directed* graphs. The algorithm leverages the P-ASY-SUM-PUSH algorithm, whose goal is to track locally the average of agents' gradients. To the best of our knowledge, the proposed framework is the first scheme combining the following attractive features (cf. Table 5.1): (a) it is *parallel and asynchronous [in the sense (i) and (ii)]*–multiple agents can be activated at the same time (with no coordination) and/or outdated information can be used in the agents' updates; our asynchronous setting (i) and (ii) is less restrictive than the one in [97], [98]; furthermore, in contrast with [97], our scheme avoids solving possibly complicated subproblems; (b) it is applicable to *nonconvex* problems, with probable convergence to stationary solutions of (5.1); (c) it is implementable over *digraph*; (d) it employs either a constant step-size or *uncoordinated* diminishing ones; (e) it *converges at an R-linear rate* (resp. sublinear) when F is strongly convex (resp. nonconvex) and a constant (resp. diminishing, uncoordinated) step-size(s) is employed; this contrasts [97] wherein each f_{i} needs to be strongly convex; and (f) it is "protocol-free", meaning that agents need not obey any specific communication protocols or asynchronous modus operandi (as long as delays are bounded and agents update/communicate uniformly infinitely often).

On the technical side, the rate analysis is employed putting forth a generalization of the small gain theorem (widely used in the literature [100] to analyze synchronous schemes), which is expected to be broadly applicable to other distributed algorithms.

5.2 Problem Setup and Preliminaries

5.2.1 Problem Setup

We study Problem (5.1) under the following assumptions.

Assumption 5.2.1 (On the optimization problem)**.**

a. Each $f_{\mathrm{i}} : \mathbb{R}^d \to \mathbb{R}$ *is proper, closed and* L_{i}*-Lipschitz differentiable;*

b. F is bounded from below. □

Note that f_{i} need not be convex. We also make the blanket assumption that each agent i knows only its own f_{i}, but not $\sum_{\mathrm{j}\neq\mathrm{i}} f_{\mathrm{j}}$. To state linear convergence, we will use the following extra condition on the objective function.

Assumption 5.2.2 (Strong convexity)**.** *Assumption 6.2.1(i) holds and, in addition, F is τ-strongly convex.* □

5.2.2 Preliminaries: The SONATA algorithm for smooth unconstrained optimization [9], [101]

The proposed asynchronous algorithmic framework builds on the synchronous SONATA algorithm, proposed in [9], [101] to solve (nonconvex) multi-agent optimization problems over time-varying digraphs. This is motivated by the fact that SONATA has the unique property of being provably applicable to both convex and nonconvex problems, and it achieves linear convergence when applied to strongly convex objectives F. We thus begin reviewing SONATA, tailored to (5.1); then we generalized it to the asynchronous setting (cf. Sec. 5.3).

Every agent controls and iteratively updates the tuple $(x_{\mathrm{i}}, y_{\mathrm{i}}, z_{\mathrm{i}}, \phi_{\mathrm{i}})$: x_{i} is agent i's copy of the shared variables x in (5.1); y_{i} acts as a local proxy of the sum-gradient ∇F; and z_{i} and ϕ_{i} are auxiliary variables instrumental to deal with communications over digraphs. Let $x_{\mathrm{i}}^k, z_{\mathrm{i}}^k$, ϕ_{i}^k, and y_{i}^k denote the value of the aforementioned variables at iteration $k \in \mathbb{N}_0$. The update of each agent i reads:

$$x_{\mathrm{i}}^{k+1} = \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}^{\mathrm{in}}\cup\{\mathrm{i}\}} w_{\mathrm{ij}} \left(x_{\mathrm{j}}^k - \alpha^k\, y_{\mathrm{j}}^k\right), \tag{5.2}$$

$$z_{\mathrm{i}}^{k+1} = \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}^{\mathrm{in}}\cup\{\mathrm{i}\}} a_{\mathrm{ij}} z_{\mathrm{j}}^k + \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{k+1}) - \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^k), \tag{5.3}$$

$$\phi_{\mathrm{i}}^{k+1} = \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}}^{\mathrm{in}}\cup\{\mathrm{i}\}} a_{\mathrm{ij}} \phi_{\mathrm{j}}^k, \tag{5.4}$$

$$y_{\mathrm{i}}^{k+1} = z_{\mathrm{i}}^{k+1}/\phi_{\mathrm{i}}^{k+1}, \tag{5.5}$$

with $z_{\mathrm{i}}^0 = y_{\mathrm{i}}^0 = \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^0)$ and $\phi_{\mathrm{i}}^0 = 1$, for all $\mathrm{i} \in \mathcal{V}$. In (5.2), y_{i}^k is a local estimate of the average-gradient $(1/m)\sum_{\mathrm{i}=1}^m \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^k)$. Therefore, every agent, first moves along the estimated gradient direction, generating $x_{\mathrm{i}}^k - \alpha^k\, y_{\mathrm{i}}^k$ (α^k is the step-size); and then performs a consensus step to force asymptotic agreement among the local variables x_{i}. Steps (5.3)-(5.5) represent a perturbed-push-sum update, aiming at tracking the gradient $(1/m)\,\nabla F$ [5], [7], [9]. The weight matrix $W \triangleq (w_{\mathrm{ij}})_{\mathrm{i,j}=1}^m$ satisfies the following assumption, and $A \triangleq (a_{\mathrm{ij}})_{\mathrm{i,j}=1}^m$ satisfies the Assumption 4.2.2.

Assumption 5.2.3. *The weight matrix* $W \triangleq (w_{\mathrm{ij}})_{\mathrm{i,j}=1}^m$ *satisfy:*

(i) $\exists\, \bar{m} > 0$ *such that:* $w_{\mathrm{ii}} \geq \bar{m}$, $\forall \mathrm{i} \in \mathcal{V}$*;* $w_{\mathrm{ij}} \geq \bar{m}$*, for all* $(\mathrm{j}, \mathrm{i}) \in \mathcal{E}$*; and* $w_{\mathrm{ij}} = 0$*, otherwise;*

(ii) W *is row-stochastic, that is,* $W\,\mathbf{1} = \mathbf{1}$.

In [100], a special instance of SONATA, was proved to converges at an R-linear rate when F is strongly convex. This result was extended to constrained, nonsmooth (composite), distributed optimization in [102]. A natural question is whether SONATA works also in an asynchronous setting still converging at a linear rate. Naive asynchronization of the updates (5.2)-(5.5)–such as using uncoordinated activations and/or replacing instantaneous information with a delayed one–would not work. For instance, the tracking (5.3)-(5.5) calls for the invariance of the averages, i.e., $\sum_{\mathrm{i}=1}^m z_{\mathrm{i}}^k = \sum_{\mathrm{i}=1}^m \nabla f_{\mathrm{i}}(x^k)$, for all $k \in \mathbb{N}_0$. It is not difficult to check that any perturbation in (5.3)-e.g., in the form of delays or packet losses–puts in jeopardy this property.

To cope with the above challenges, we robustify the gradient tracking component using P-ASY-SUM-PUSH, and we present in the next section the proposed distributed asynchronous optimization framework, ASY-SONATA.

5.3 Asynchronous SONATA (ASY-SONATA)

We are ready now to introduce our distributed asynchronous algorithm–ASY-SONATA. The algorithm combines SONATA (cf. Sec. 6.3.1) with P-ASY-SUM-PUSH (cf. Sec. 4.3), the latter replacing the synchronous tracking scheme (5.3)-(5.5). The "global view" of the scheme is given in Algorithm 4.

In ASY-SONATA, agents continuously and with no coordination perform: i) their local computations [cf. **(S.3)**], possibly using an out-of-sync estimate $z_{\mathrm{i}^k}^k$ of the average gradient;

Algorithm 4 ASY-SONATA (Global View)

```
Data: For all agent i and $\forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$, $x_{\mathrm{i}}^0 \in \mathbb{R}^d$, $z_{\mathrm{i}}^0 = \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^0)$, $\phi_{\mathrm{i}}^0 = 1$, $\tilde{\boldsymbol{\rho}}_{\mathrm{ij}}^0 = 0$, $\tilde{\sigma}_{\mathrm{ij}}^0 = 0$,
$\tau_{\mathrm{ij}}^{-1} = -D$. And for $t = -D, -D+1, \ldots, 0$, $\boldsymbol{\rho}_{\mathrm{ij}}^t = 0$, $\sigma_{\mathrm{ij}}^t = 0$, $v_{\mathrm{i}}^t = 0$. Set $k = 0$.
While: a termination criterion is not met do
    (S.1) Pick $(\mathrm{i}^k, d^k)$;
    (S.2) Set:
          $$\tau_{\mathrm{i}^k\mathrm{j}}^k = \max(\tau_{\mathrm{i}^k\mathrm{j}}^{k-1}, k - d_{\mathrm{j}}^k), \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}.$$
    (S.3) Local Descent:
          $$v_{\mathrm{i}^k}^{k+1} = x_{\mathrm{i}^k}^k - \gamma^k z_{\mathrm{i}^k}^k. \qquad (5.6)$$
    (S.4) Consensus:
          $$x_{\mathrm{i}^k}^{k+1} = w_{\mathrm{i}^k\mathrm{i}^k} v_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} w_{\mathrm{i}^k\mathrm{j}} v_{\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}.$$
    (S.5) Gradient Tracking:
    • (S.5.1) Sum step:
          $$z_{\mathrm{i}^k}^{k+\frac{1}{2}} = z_{\mathrm{i}^k}^k + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} \left( \rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k} - \tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k \right) + \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k+1}) - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)$$
    • (S.5.2) Push step:
          $$z_{\mathrm{i}^k}^{k+1} = a_{\mathrm{i}^k\mathrm{i}^k}\, z_{\mathrm{i}^k}^{k+\frac{1}{2}},$$
          $$\rho_{\mathrm{j}\mathrm{i}^k}^{k+1} = \rho_{\mathrm{j}\mathrm{i}^k}^k + a_{\mathrm{j}\mathrm{i}^k}\, z_{\mathrm{i}^k}^{k+\frac{1}{2}}, \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{out}}$$
    • (S.5.3) Mass-Buffer update:
          $$\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^{k+1} = \rho_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}, \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$$
    (S.6) Untouched state variables shift to state $k+1$
          while keeping the same value; $k \leftarrow k+1$.
```

in (5.6), γ^k is a step-size (to be properly chosen); ii) a consensus step on the x-variables, using possibly outdated information $v_{\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}$ from their in-neighbors [cf. **(S.4)**]; and iii) gradient tracking [cf. **(S.5)**] to update the local estimate $z_{\mathrm{i}^k}^k$, based on the current cumulative mass variables $\boldsymbol{\rho}_{\mathrm{i}^k\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}$, and buffer variables $\tilde{\rho}_{\mathrm{i}^k\mathrm{j}}^k$, $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$.

Note that in Algorithm 3, the tracking variable $y_{\mathrm{i}^k}^{k+1}$ is obtained rescaling $z_{\mathrm{i}^k}^{k+1}$ by the factor $1/\phi_{\mathrm{i}^k}^{k+1}$. In Algorithm 4, we absorbed the scaling $1/\phi_{\mathrm{i}^k}^{k+1}$ in the step size and use directly $z_{\mathrm{i}^k}^{k+1}$ as a proxy of the average gradient, eliminating thus the ϕ-variables (and the related σ-, $\tilde{\sigma}$-variables). Also, for notational simplicity and without loss of generality, we assumed that the v- and ρ- variables are subject to the same delays (e.g., they are transmitted within the same packet); same convergence results hold if different delays are considered.

We study now convergence of the scheme, under a constant step-size or diminishing, uncoordinated ones.

5.3.1 Constant Step-size

Theorem 6.4.1 below establishes *linear* convergence of ASY-SONATA when F is strongly convex.

Theorem 5.3.1 (Geometric convergence)**.** *Consider (P) under Assumption 5.2.2, and let $x^\star$ denote its unique solution. Let $\{(x_\mathrm{i}^k)_{\mathrm{i}=1}^m\}_{k\in\mathbb{N}_0}$ be the sequence generated by Algorithm 4, under Assumption 4.2.1, 4.3.2, and with weight matrices A and W satisfying Assumption 4.2.2 and 5.2.3. Then, there exists a constant $\bar{\gamma}_1 > 0$ [cf. (5.26)] such that if $\gamma^k \equiv \gamma \leq \bar{\gamma}_1$, it holds*

$$M_{sc}(x^k) \triangleq \|x^k - 1_m \otimes x^\star\| = \mathcal{O}(\lambda^k), \tag{5.7}$$

with $\lambda \in (0,1)$ given by

$$\lambda = \begin{cases} 1 - \frac{\tau \bar{m}^{2K_1}\gamma}{2} & \text{if } \gamma \in (0, \hat{\gamma}_1], \\ \rho + \sqrt{J_1 \gamma} & \text{if } \gamma \in (\hat{\gamma}_1, \hat{\gamma}_2), \end{cases} \tag{5.8}$$

where $\hat{\gamma}_1$ and $\hat{\gamma}_2$ are some constants strictly smaller than $\bar{\gamma}_1$, and $J_1 \triangleq (1-\rho)^2/\hat{\gamma}_2$.

Proof. See Sec. 5.3.3. □

When F is convex (resp. nonconvex), we introduce the following merit function to measure the progresses of the algorithm towards optimality (resp. stationarity) and consensus:

$$M_F(x^k) \triangleq \max\{\left\|\nabla F(\bar{x}^k)\right\|^2, \left\|x^k - 1_m \otimes \bar{x}^k\right\|^2\}, \tag{5.9}$$

where $x^k \triangleq [x_1^{k\top}, \cdots, x_m^{k\top}]^\top$ and $\bar{x}^k \triangleq (1/m) \cdot \sum_{\mathrm{i}=1}^m x_{\mathrm{i}}^k$. Note that M_F is a valid merit function, since it is continuous and $M_F(x) = 0$ if and only if all x_{i}'s are consensual and optimal (resp. stationary solutions).

Theorem 5.3.2 (Sublinear convergence)**.** *Consider (P) under Assumption 6.2.1 (thus possibly nonconvex). Let $\{(x_{\mathrm{i}}^k)_{\mathrm{i}=1}^m\}_{k\in\mathbb{N}_0}$ be the sequence generated by Algorithm 4, in the same setting of Theorem 6.4.1. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}_0$ such that $M_F(x^k) \le \delta$. Then, there exists a $\bar{\gamma}_2 > 0$ [cf. (5.35)], such that if $\gamma^k \equiv \gamma \le \bar{\gamma}_2$, $T_\delta = \mathcal{O}(1/\delta)$. The values of the above constants is given in the proof.*

Proof. See Sec. 5.3.4. □

Theorem 6.4.1 states that consensus and optimization errors of the sequence generated by ASY-SONATA vanish at a linear rate. We are not aware of any other scheme enjoying such a property in such a distributed, asynchronous computing environment. For general, possibly nonconvex instances of Problem (P), Theorem 6.4.2 shows that both consensus and optimization errors of the sequence generated by ASY-SONATA vanish at $\mathcal{O}(1/\delta)$ sublinear rate.

The choice of a proper stepsize calls for the estimates of $\bar{\gamma}_1$ and $\bar{\gamma}_2$ in Theorems 6.4.1 and 6.4.2, which depend on the following quantities: the optimization parameters L_{i} (Lipschitz constants of the gradients) and τ (strongly convexity constant), the network connectivity parameter ρ, and the constants D and T due to the asynchrony (cf. Assumption 4.3.2). Notice that the dependence of the stepsize on L_{i}, τ, and ρ is common to all the existing distributed synchronous algorithms and so is that on T and D to (even centralized) asynchronous algorithms [64]. While L_{i}, τ, and ρ can be acquired following approaches discussed in the literature (see, e.g., [100, Remark 4]), it is less clear how to estimate D and T, as they are related to the asynchronous model, generally not known to the agents. As an example, we address this question considering the following fairly general model for the agents' activations and asynchronous communications. Suppose that the length of any time window between consecutive "push" steps of any agent belongs to $[p_{\min}, p_{\max}]$, for some $p_{\max} \ge p_{\min} > 0$, and one agent always sends out its updated information immediately after the completion of its "push" step. The traveling time of each packet is at most D^{tv}. Also, at least one packet is successfully received every D^{ls} successive one-hop communications. Note that there is a vast literature on how to estimate D^{tv} and D^{ls}, based upon the specific channel model under consideration; see, e.g., [103], [104]. In this setting, it is not difficult to check that one can set $T = (m-1)\lceil p_{\max}/p_{\min}\rceil + 1$ and $D = m\lceil D^{\mathrm{tv}}/p_{\min}\rceil D^{\mathrm{ls}}$. To cope with the issue of estimating $\bar{\gamma}_1$ and $\bar{\gamma}_2$, in the next section we show how to employ in ASY-SONATA diminishing, uncoordinated stepsizes.

5.3.2 Uncoordinated diminishing step-sizes

The use of a diminishing stepsize shared across the agents is quite common in synchronous distributed algorithms. However, it is not clear how to implement such option in an asynchronous setting, without enforcing any coordination among the agents (they should know the global iteration counter k). In this section, we provide for the first time a solution to this issue. Inspired by [105], our model assumes that each agent, *independently* and with *no coordination* with the others, draws the step-size from a local sequence $\{\alpha^t\}_{t\in\mathbb{N}_0}$, according to its local clock. The sequence $\{\gamma^k\}_{k\in\mathbb{N}_0}$ in (5.6) will be thus the result of the "uncoordinated samplings" of the local out-of-sync sequences $\{\alpha^t\}_{t\in\mathbb{N}_0}$. The next theorem shows that in this setting, ASY-SONATA converges at a sublinear rate for both convex and nonconvex objectives.

Theorem 5.3.3. *Consider Problem (P) under Assumption 6.2.1 (thus possibly nonconvex). Let $\{(x_{\mathrm{i}}^k)_{\mathrm{i}=1}^m\}_{k\in\mathbb{N}_0}$ be the sequence generated by Algorithm 4, in the same setting of Theorem 6.4.1, but with the agents using a local step-size sequence $\{\alpha^t\}_{t\in\mathbb{N}_0}$ satisfying $\alpha^t \downarrow 0$ and $\sum_{t=0}^{\infty}\alpha^t = \infty$. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}_0$ such that $M_F(x^k) \leq \delta$. Then*

$$T_\delta \leq \inf\left\{k \in \mathbb{N}_0 \,\middle|\, \sum_{t=0}^{k}\gamma^t \geq c/\delta\right\}, \tag{5.10}$$

where c is a positive constant.

Proof. See Sec. 5.3.4. □

5.3.3 Proof of Theorem 6.4.1

We organize the proof in the following steps: **Step 1:** We introduce and study convergence of an auxiliary perturbed consensus scheme, which serves as a unified model for the descent and consensus updates in ASY-SONATA–the main result is summarized in Proposition 5.3.1; **Step 2:** We introduce the consensus and gradient tracking errors along with a suitably defined optimization error; and we derive bounds connecting these quantities, building on results in Step 1 and convergence of P-ASY-SUM-PUSH–see Proposition 5.3.2. The goal is to prove that the aforementioned errors vanish at a linear rate. To do so, **Step 3** introduces a general form of the small gain theorem–Theorem 5.3.7–along with some technical results, which allows us to establish the desired linear convergence through the boundedness of the solution of an associated linear system of inequalities. **Step 4** builds such a linear system for the error quantities introduced in Step 2 and proves the boundedness of its solution, proving thus Theorem 6.4.1. The rate expression (5.8) is derived in Appendix 5.6.3. Through the proof we assume $d = 1$ (scalar variables); and define $C_L \triangleq \max_{\mathrm{i}=1,\ldots,m} L_\mathrm{i}$ and $L \triangleq \sum_{\mathrm{i}=1}^m L_\mathrm{i}$.

Step 1: A perturbed asynchronous consensus scheme

We introduce a unified model to study the dynamics of the consensus and optimization errors in ASY-SONATA, which consists in pulling out the tracking update (Step 5) and treat the z-variables–the term $-\gamma^k z_{\mathrm{i}^k}^k$ in (5.6)–as an exogenous perturbation δ^k. More specifically, consider the following scheme (with a slight abuse of notation, we use the same symbols as in ASY-SONATA):

$$v_{\mathrm{i}^k}^{k+1} = x_{\mathrm{i}^k}^k + \delta^k, \tag{5.11a}$$

$$x_{\mathrm{i}^k}^{k+1} = w_{\mathrm{i}^k\mathrm{i}^k} v_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j}\in\mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} w_{\mathrm{i}^k\mathrm{j}} v_\mathrm{j}^{k-d_\mathrm{j}^k}, \tag{5.11b}$$

$$v_\mathrm{j}^{k+1} = v_\mathrm{j}^k,\; x_\mathrm{j}^{k+1} = x_\mathrm{j}^k, \quad \forall \mathrm{j} \in \mathcal{V} \setminus \{\mathrm{i}^k\}, \tag{5.11c}$$

with given $x_\mathrm{i}^0 \in \mathbb{R}$, $v_\mathrm{i}^t = 0$, $t = -D, -D+1, \ldots, 0$, for all $\mathrm{i} \in \tilde{\mathcal{V}}$. We make the blanket assumption that agents' activations and delays satisfy Assumption 4.3.2.

Let us rewrite (5.11) in a vector-matrix form. Define $x^k \triangleq [x_1^k, \cdots, x_m^k]^\top$ and $v^k \triangleq [v_1^k, \cdots, v_m^k]^\top$. Construct the $(D+2)m$ dimensional concatenated vectors

$$H^k \triangleq [{x^k}^\top, {v^k}^\top, {v^{k-1}}^\top, \cdots, {v^{k-D}}^\top]^\top, \quad \delta^k \triangleq \delta^k\, \mathrm{e}_{\mathrm{i}^k}; \tag{5.12}$$

and the augmented matrix $\widehat{W}^k$, defined as

$$\widehat{W}^k_{rh} \triangleq \begin{cases} w_{\mathrm{i}^k\mathrm{i}^k}, & \text{if } r = h = \mathrm{i}^k; \\ w_{\mathrm{i}^k\mathrm{j}}, & \text{if } r = \mathrm{i}^k,\ h = \mathrm{j} + (d_\mathrm{j}^k + 1)m; \\ 1, & \text{if } r = h \in \{1, 2, \ldots, 2m\} \setminus \{\mathrm{i}^k, \mathrm{i}^k + m\}; \\ 1, & \text{if } r \in \{2m+1, 2m+2, \ldots, (D+2)m\} \cup \{\mathrm{i}^k + m\} \text{ and } h = r - m; \\ 0, & \text{otherwise.} \end{cases}$$

System (5.11) can be rewritten in compact form as

$$H^{k+1} = \widehat{W}^k(H^k + \delta^k), \tag{5.13}$$

The following lemma captures the asymptotic behavior of $\widehat{W}^k$.

Lemma 5.3.4. *Let $\{\widehat{W}^k\}_{k\in\mathbb{N}_0}$ be the sequence of matrices in (6.18), generated by (5.11), under Assumption 4.3.2 and with W satisfying Assumption* **??** *(i), (ii). The following hold: for all $k \in \mathbb{N}_0$, a) $\widehat{W}^k$ is row stochastic; b) there exists a sequence of stochastic vectors $\{\psi^k\}_{k\in\mathbb{N}_0}$ such that*

$$\left\|\widehat{W}^{k:t} - 1{\psi^t}^\top\right\| \le C_2\rho^{k-t}, \quad C_2 \triangleq \frac{2\sqrt{(D+2)m}(1+\bar{m}^{-K_1})}{1-\bar{m}^{-K_1}} \tag{5.14}$$

Furthermore, $\psi_\mathrm{i}^k \ge \eta = \bar{m}^{K_1}$, for all $k \ge 0$ and $\mathrm{i} \in \mathcal{V}$.

Proof. The proof follows similar techniques as in [75], [76], and can be found in Appendix 5.6.5. □

We define now a proper consensus error for (6.18). Writing H^k in (6.18) recursively, yields

$$H^{k+1} = \widehat{W}^{k:0}H^0 + \sum_{l=0}^{k} \widehat{W}^{k:l}\delta^l. \tag{5.15}$$

Using Lemma 6.6.1, for any fixed $N \in \mathbb{N}_0$, we have

$$\lim_{k\to\infty} (\widehat{W}^{k:0}H^0 + \sum_{l=0}^{N} \widehat{W}^{k:l}\delta^l) = 1{\psi^0}^\top H^0 + \sum_{l=0}^{N} 1{\psi^l}^\top \delta^l.$$

Define

$$x_\psi^0 \triangleq {\psi^0}^\top H^0, \quad x_\psi^{k+1} \triangleq {\psi^0}^\top H^0 + \sum_{l=0}^{k} {\psi^l}^\top \delta^l,\ k \in \mathbb{N}_0. \tag{5.16}$$

Applying (5.16) inductively, it is easy to check that

$$x_\psi^{k+1} = x_\psi^k + {\psi^k}^\top \delta^k = x_\psi^k + \psi_{\mathrm{i}^k}^k \delta^k. \tag{5.17}$$

We are now ready to state the main result of this subsection, which is a bound of the consensus disagreement $\|H^{k+1} - 1x_\psi^{k+1}\|$ in terms of the magnitude of the perturbation.

Proposition 5.3.1. *In the above setting, the consensus error $\|H^{k+1} - 1x_\psi^{k+1}\|$ satisfies: for all $k \in \mathbb{N}_0$,*

$$\left\|H^{k+1} - 1x_\psi^{k+1}\right\| \le C_2\rho^k \left\|H^0 - 1x_\psi^0\right\| + C_2 \sum_{l=0}^{k} \rho^{k-l} \left|\delta^l\right|.$$

Proof. The proof follows readily from (5.15), (5.16), and Lemma 6.6.1; we omit further details. □

Step 2: Consensus, tracking, and optimization errors

1) Consensus disagreement: As anticipated, the updates of ASY-SONATA are also described by (5.11), if one sets therein $\delta^k = -\gamma^k z_{\mathrm{i}^k}^k$ (with $z_{\mathrm{i}^k}^k$ satisfying Step 5 of ASY-SONATA). Let h^k and x_ψ^k be defined as in (5.12) and (5.16), respectively, with $\delta^k = -\gamma^k z_{\mathrm{i}^k}^k$. The consensus error at iteration k is defined as

$$E_c^k \triangleq \left\|H^k - 1x_\psi^k\right\|. \tag{5.18}$$

2) Gradient tracking error: The gradient tracking step in ASY-SONATA is an instance of P-ASY-SUM-PUSH, with $\epsilon^k = \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k+1}) - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)$. By Proposition 4.4.1, P-ASY-SUM-PUSH is equivalent to (4.25). In view of Lemma 4.4.4 and the following property

$1^\top \widehat{z}^k = \sum_{\mathrm{i}=1}^m \nabla f_\mathrm{i}(x_\mathrm{i}^0) + \sum_{t=0}^{k-1} \left(\nabla f_{\mathrm{i}^t}(x_{\mathrm{i}^t}^{t+1}) - \nabla f_{\mathrm{i}^t}(x_{\mathrm{i}^t}^t)\right) = \sum_{\mathrm{i}=1}^m \nabla f_\mathrm{i}(x_\mathrm{i}^k)$ where the first equality follows from (4.30) and $\epsilon^k = \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k+1}) - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)$ while in the second equality we used $x_\mathrm{j}^{t+1} = x_\mathrm{j}^t$, for $\mathrm{j} \neq \mathrm{i}^t$, the tracking error at iteration k along with the magnitude of the tracking variables are defined as

$$E_t^k \triangleq \left| z_{\mathrm{i}^k}^k - \xi_{\mathrm{i}^k}^{k-1}\, \bar{g}^k \right|,\ E_z^k \triangleq \left| z_{\mathrm{i}^k}^k \right|,\ \bar{g}^k \triangleq \sum_{\mathrm{i}=1}^m \nabla f_\mathrm{i}(x_\mathrm{i}^k), \tag{5.19}$$

with $\xi_\mathrm{i}^{-1} \triangleq \eta$, $\mathrm{i} \in \mathcal{V}$. Let $g^k \triangleq [\nabla f_1(x_1^k), \ldots, \nabla f_I(x_I^k)]^\top$.

3) Optimization error: Let $x^\star$ be the unique minimizer of F. Given the definition of consensus disagreement in (5.18), we define the optimization error at iteration k as

$$E_o^k \triangleq \left| x_\psi^k - x^\star \right|. \tag{5.20}$$

Note that this is a natural choice as, if consensual, all agents' local variables will converge to a limit point of $\{x_\psi^k\}_{k\in\mathbb{N}_0}$.

4) Connection among E_c^k, E_t^k, E_z^k, and E_o^k: The following proposition establishes bounds on the above quantities.

Proposition 5.3.2. *Let $\{x^k, v^k, z^k\}_{k\in\mathbb{N}_0}$ be the sequence generated by ASY-SONATA, in the setting of Theorem 6.4.1, but possibly with a time-varying step-size $\{\gamma^k\}_{k\in\mathbb{N}_0}$. The error quantities E_c^k, E_t^k, E_z^k, and E_o^k satisfy: for all $k \in \mathbb{N}_0$,*

$$E_c^{k+1} \leq C_2 \rho^k E_c^0 + C_2 \sum_{l=0}^k \rho^{k-l} \gamma^l E_z^l. \tag{5.21a}$$

$$E_t^{k+1} \leq 3 C_0 C_L \sum_{l=0}^k \rho^{k-l} \left(E_c^l + \gamma^l E_z^l\right) + C_0 \rho^k \left\| g^0 \right\|; \tag{5.21b}$$

$$E_z^k \ \leq\ E_t^k + C_L \sqrt{m} E_c^k + L\, E_o^k \tag{5.21c}$$

Further assume $\gamma^k \leq 1/L$, $k \in \mathbb{N}_0$; then

$$E_o^{k+1} \leq \sum_{l=0}^k \left(\prod_{t=l+1}^k \left(1 - \tau\eta^2\gamma^t\right) \right) \left(C_L \sqrt{m} E_c^l + E_t^l\right) \gamma^l + \prod_{t=0}^k \left(1 - \tau\eta^2\gamma^t\right) E_o^0, \tag{5.21d}$$

where $\eta \in (0,1)$ is defined in Lemma 4.4.3 and τ is the strongly convexity constant of F.

Proof. Eq. (6.23) follows readily from Proposition 5.3.1.

We prove now (6.24). Recall $1^\top \widehat{z}^k = \bar{g}^k$. Using Lemma 4.4.4 with $\epsilon^k = \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k+1}) - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k})$, we obtain: for all $\mathrm{i} \in \mathcal{V}$,

$$
\begin{aligned}
&\left|z_{\mathrm{i}}^{k+1} - \xi_{\mathrm{i}}^{k}\, \bar{g}^{k+1}\right| \\
&\le C_0 \left(\rho^k \left\|g^0\right\| + \sum_{l=0}^{k} \rho^{k-l} \left|\nabla f_{\mathrm{i}^l}^{l+1} - \nabla f_{\mathrm{i}^l}^{l}\right|\right) \le C_0\, \rho^k \left\|g^0\right\| + C_0 C_L \sum_{l=0}^{k} \rho^{k-l} \left|x_{\mathrm{i}^l}^{l+1} - x_{\mathrm{i}^l}^{l}\right| \\
&\le C_0\, \rho^k \left\|g^0\right\| + C_0 C_L \sum_{l=0}^{k} \rho^{k-l} \left\|H^{l+1} - H^l\right\| = C_0 \rho^k \left\|g^0\right\| + C_0 C_L \sum_{l=0}^{k} \rho^{k-l} \left\|\widehat{W}^l \left(H^l + \delta^l\right) - H^l\right\| \\
&\overset{(a)}{=} C_0\, \rho^k \left\|g^0\right\| + C_0 C_L \sum_{l=0}^{k} \rho^{k-l} \left\|\left(\widehat{W}^l - m\right)\left(H^l - 1 x_\psi^l\right) - \gamma^l z_{\mathrm{i}^l}^l \widehat{W}^l \mathrm{e}_{\mathrm{i}^l}\right\| \\
&\le C_0 \rho^k \left\|g^0\right\| + C_0 C_L \sum_{l=0}^{k} \rho^{k-l} \left(\|\widehat{W}^l\| \gamma^l E_z^l + \left(\|\widehat{W}^l\| + \|m\|\right) E_c^l\right) \\
&\overset{(b)}{\le} C_0 \rho^k \left\|g^0\right\| + 3 C_0 C_L \sum_{l=0}^{k} \rho^{k-l} \left(E_c^l + \gamma^l E_z^l\right),
\end{aligned}
$$

where in (a) we used (6.18) and the row stochasticity of $\widehat{W}^k$ [Lemma 6.6.1(a)]; and (b) follows from $\|\widehat{W}^l\| \le \sqrt{\|\widehat{W}^l\|_1 \|\widehat{W}^l\|_\infty} \le \sqrt{3}$. This proves (6.24).

Eq. (5.21c) follows readily from

$$
E_z^k = \left|z_{\mathrm{i}^k}^k\right| \le \left|z_{\mathrm{i}^k}^k - \xi_{\mathrm{i}^k}^{k-1}\, \bar{g}^k\right| + \xi_{\mathrm{i}^k}^{k-1} \left|\bar{g}^k - \nabla F(x_\psi^k)\right| + \xi_{\mathrm{i}^k}^{k-1} \left|\nabla F(x_\psi^k) - \nabla F(x^\star)\right|.
$$

Finally, we prove (5.21d). Using (5.21c) and $x_\psi^{k+1} = x_\psi^k - \gamma \psi_{\mathrm{i}^k}^k z_{\mathrm{i}^k}^k$ [cf. (5.17) and recall $\delta^k = -\gamma z_{\mathrm{i}^k}^k$], we can write

$$
\begin{aligned}
E_o^{k+1} &= \left|x_\psi^k - \gamma^k \psi_{\mathrm{i}^k}^k z_{\mathrm{i}^k}^k - x^\star\right| \\
&\le \gamma^k \psi_{\mathrm{i}^k}^k \xi_{\mathrm{i}^k}^{k-1} \left|\nabla F(x_\psi^k) - \bar{g}^k\right| + \gamma^k \psi_{\mathrm{i}^k}^k \left|\xi_{\mathrm{i}^k}^{k-1}\, \bar{g}^k - z_{\mathrm{i}^k}^k\right| + \left|x_\psi^k - \gamma^k \psi_{\mathrm{i}^k}^k \xi_{\mathrm{i}^k}^{k-1} \nabla F(x_\psi^k) - x^\star\right| \\
&\overset{(a)}{\le} \left(1 - \tau \eta^2 \gamma^k\right) E_o^k + C_L \sqrt{m} \gamma^k \left\|H^k - 1 x_\psi^k\right\| + \gamma^k E_t^k \\
&\overset{(b)}{\le} \sum_{l=0}^{k} \left(\prod_{t=l+1}^{k} \left(1 - \tau \eta^2 \gamma^t\right)\right) \left(C_L \sqrt{m} E_c^l + E_t^l\right) \gamma^l + \prod_{t=0}^{k} \left(1 - \tau \eta^2 \gamma^t\right) E_o^0
\end{aligned}
$$

where in (a) we used $\eta^2 \leq \psi_{\mathrm{i}^k}^k \xi_{\mathrm{i}^k}^{k-1} < 1$ (cf. Lemma 4.4.3) and $|x - \gamma \nabla F(x) - x^\star| \leq (1 - \tau\gamma)\, |x - x^\star|$, which holds for $\gamma \leq 1/L$; (b) follows readily by applying the above inequality telescopically. □

Step 3: The generalized small gain theorem

The last step of our proof is to show that the error quantities E_c^k, E_t^k, E_z^k, and E_o^k vanish linearly. This is not a straightforward task, as these quantities are interconnected through the inequalities (5.21). This subsection provides tools to address this issue. The key result is a generalization of the small gain theorem (cf. Theorem 5.3.7), first used in [100].

Definition 5.3.1 ([100])**.** *Given the sequence $\{u^k\}_{k=0}^\infty$, a constant $\lambda \in (0,1)$, and $N \in \mathbb{N}$, let us define*

$$|u|^{\lambda,N} = \max_{k=0,\ldots,N} \frac{|u^k|}{\lambda^k}, \quad |u|^{\lambda} = \sup_{k\in\mathbb{N}_0} \frac{|u^k|}{\lambda^k}.$$

If $|u|^\lambda$ is upper bounded, then $u^k = \mathcal{O}(\lambda^k)$, for all $k \in \mathbb{N}_0$.

The following lemma shows how one can interpret the inequalities in (5.21) using the notions introduced in Definition 6.6.1.

Lemma 5.3.5. *Let $\{u^k\}_{k=0}^\infty, \{v_\mathrm{i}^k\}_{k=0}^\infty, \mathrm{i} = 1,\ldots,m,$ be nonnegative sequences; let $\lambda_0, \lambda_1, \ldots, \lambda_m \in (0,1)$; and let $R_0, R_1, \ldots, R_m \in \mathbb{R}_+$ such that*

$$u^{k+1} \leq R_0(\lambda_0)^k + \sum_{\mathrm{i}=1}^m R_\mathrm{i} \sum_{l=0}^k (\lambda_\mathrm{i})^{k-l} v_\mathrm{i}^l, \quad \forall k \in \mathbb{N}_0.$$

Then, there holds

$$|u|^{\lambda,N} \leq u^0 + \frac{R_0}{\lambda} + \sum_{\mathrm{i}=1}^m \frac{R_\mathrm{i}}{\lambda - \lambda_\mathrm{i}} |v_\mathrm{i}|^{\lambda,N},$$

for any $\lambda \in (\max_{\mathrm{i}=0,1,\ldots,m} \lambda_\mathrm{i}, 1)$ and $N \in \mathbb{N}$.

Proof. See Appendix 5.6.1. □

Lemma 5.3.6. *Let $\{u^k\}_{k=0}^\infty$ and $\{v^k\}_{k=0}^\infty$ be two nonnegative sequences. The following hold*

a. *$u^k \leq v^k$, for all $k \in \mathbb{N}_0 \Longrightarrow |u|^{\lambda,N} \leq |v|^{\lambda,N}$, for any $\lambda \in (0,1)$ and $N \in \mathbb{N}$;*

b.

$$|\beta_1 u + \beta_2 v|^{\lambda,N} \leq |\beta_1| \, |u|^{\lambda,N} + |\beta_2| \, |v|^{\lambda,N} \,,$$

for any $\beta_1, \beta_2 \in \mathbb{R}$, $\lambda \in (0,1)$, and positive integer N.

The major result of this section is the generalized small gain theorem, as stated next.

Theorem 5.3.7. *(Generalized Small Gain Theorem) Given nonnegative sequences $\{u_{\mathrm{i}}^k\}_{k=0}^{\infty}$, $\mathrm{i} = 1, \ldots, m$, a non-negative matrix $T \in \mathbb{R}^{m\times m}$, $\beta \in \mathbb{R}^m$, and $\lambda \in (0,1)$ such that*

$$u^{\lambda,N} \preccurlyeq T u^{\lambda,N} + \beta, \quad \forall N \in \mathbb{N}, \tag{5.22}$$

where $u^{\lambda,N} \triangleq [\, |u_1|^{\lambda,N}, \ldots, |u_m|^{\lambda,N} \,]^\top$. If $\rho(T) < 1$, then $|u_{\mathrm{i}}|^{\lambda}$ is bounded, for all $\mathrm{i} = 1, \ldots, m$. That is, each u_{i}^k vanishes at a R-linear rate $\mathcal{O}(\lambda^k)$.

Proof. See Appendix 5.6.2. □

Then following results are instrumental to find a sufficient condition for $\rho(T) < 1$.

Lemma 5.3.8. *Consider a polynomial $p(z) = z^m - a_1 z^{m-1} - a_2 z^{m-2} - \ldots - a_{m-1} z - a_m$, with $z \in \mathbb{C}$ and $a_{\mathrm{i}} \in \mathbb{R}_+$, $\mathrm{i} = 1, \ldots m$. Define $z_p \triangleq \max \Big\{ |z_{\mathrm{i}}| \, \Big| \, p(z_{\mathrm{i}}) = 0, \;\; \mathrm{i} = 1, \ldots, m \Big\}$. Then, $z_p < 1$ if and only if $p(1) > 0$.*

Proof. See Appendix 5.6.4. □

Step 4: Linear convergence rate (proof of Theorem 6.4.1)

Our path to prove linear convergence rate passes through Theorem 5.3.7: we first cast the set of inequalities (5.21) into a system in the form (5.22), and then study the spectral properties of the resulting coefficient matrix.

Given $\gamma < 1/L$, define $\mathcal{L}(\gamma) \triangleq 1 - \tau\eta^2\gamma$; and choose $\lambda \in \mathbb{R}$ such that

$$\max\left(\rho, \mathcal{L}(\gamma)\right) < \lambda < 1. \tag{5.23}$$

Note that $\mathcal{L}(\gamma) < 1$, as $\gamma < 1/L$; hence (5.23) is nonempty.

Applying Lemma 5.3.5 and Lemma 5.3.6 to the set of inequalities (5.21) with $\gamma^k \equiv \gamma$, we obtain the following with $b_1 \triangleq C_L\sqrt{m}, \quad b_2 \triangleq 3C_0C_L$:

$$\begin{bmatrix} |E_z|^{\lambda,N} \\ |E_c|^{\lambda,N} \\ |E_t|^{\lambda,N} \\ |E_o|^{\lambda,N} \end{bmatrix} \preccurlyeq \underbrace{\begin{bmatrix} 0 & b_1 & 1 & L \\ \frac{C_2\gamma}{\lambda-\rho} & 0 & 0 & 0 \\ \frac{b_2\gamma}{\lambda-\rho} & \frac{b_2}{\lambda-\rho} & 0 & 0 \\ 0 & \frac{b_2\gamma}{\lambda-\mathcal{L}(\gamma)} & \frac{\gamma}{\lambda-\mathcal{L}(\gamma)} & 0 \end{bmatrix}}_{\triangleq K} \begin{bmatrix} |E_z|^{\lambda,N} \\ |E_c|^{\lambda,N} \\ |E_t|^{\lambda,N} \\ |E_o|^{\lambda,N} \end{bmatrix} + \begin{bmatrix} 0 \\ \left(1+\frac{C_2}{\lambda}\right)E_c^0 \\ \frac{C_0\|g^0\|}{\lambda} + E_t^0 \\ \frac{1+\lambda}{\lambda}E_o^0 \end{bmatrix}. \tag{5.24}$$

By Theorem 5.3.7, to prove the desired linear convergence rate, it is sufficient to show that $\rho(K) < 1$. The characteristic polynomial $p_K(t)$ of T satisfies the conditions of Lemma 5.3.8; hence $\rho(K) < 1$ if and only if $p_K(1) > 0$, that is,

$$\left(\left(1+\frac{L\gamma}{\lambda-\mathcal{L}(\gamma)}\right)\frac{b_2}{\lambda-\rho}+b_1+\frac{Lb_2\gamma}{\lambda-\mathcal{L}(\gamma)}\right)\frac{C_2\gamma}{\lambda-\rho}+\left(1+\frac{L\gamma}{\lambda-\mathcal{L}(\gamma)}\right)\frac{b_2\gamma}{\lambda-\rho} \triangleq \mathfrak{B}(\lambda;\gamma) < 1. \tag{5.25}$$

By the continuity of $\mathfrak{B}(\lambda;\gamma)$ and (5.23), $\mathfrak{B}(1;\gamma) < 1$ is sufficient to claim the existence of some $\lambda \in (\max(\rho, \mathcal{L}(\gamma)), 1)$ such that $\mathfrak{B}(\lambda;\gamma) < 1$. Hence, setting $\mathfrak{B}(1;\gamma) < 1$, yields $0 < \gamma < \bar{\gamma}_1$, with

$$\bar{\gamma}_1 \triangleq \frac{\tau\eta^2(1-\rho)^2}{(\tau\eta^2+L)b_2(C_2+1-\rho)+(b_1\tau\eta^2+Lb_2)C_2(1-\rho)}. \tag{5.26}$$

It is easy to check that $\bar{\gamma}_1 < 1/L$. Therefore, $0 < \gamma < \bar{\gamma}_1$ is sufficient for $E_c^k, E_t^k, E_z^k, E_o^k$ to vanish with an R-Linear rate. The desired result, $\left|x_{\rm i}^k - x^\star\right| = \mathcal{O}(\lambda^k)$, $\rm i \in \mathcal{V}$, follows readily from $E_c^k = \mathcal{O}(\lambda^k)$ and $E_o^k = \mathcal{O}(\lambda^k)$. The explicit expression of the rate λ, as in (5.8), is derived in Appendix 5.6.3.

5.3.4 Proof of Theorems 6.4.2 and 5.3.3

Through the subsection, we use the same notation as in Sec.5.3.3. **- Preliminaries** We begin establishing a connection between the merit function M_F defined in (5.9) and the error quantities E_c^k, E_t^k, and E_z^k, defined in (5.18), (5.19), and (5.20) respectively.

Lemma 5.3.9. *The merit function M_F satisfies*

$$M_F(x^k) \leq C_3\,(E_c^k)^2 + 3\,\eta^{-2}\,\left((E_t^k)^2 + (E_z^k)^2\right), \tag{5.27}$$

with $C_3 \triangleq 3C_L^2 m + \frac{3L^2}{m} + 6C_L L + 4$.

Proof. Define $J \triangleq (1/m)\cdot \mathbf{1}\mathbf{1}^\top$ and $\bar{x}^k \triangleq (1/m)\cdot \mathbf{1}^\top x^k$; and recall the definition of ξ_{i}^k (cf. Lemma 4.4.3) and that $x_\psi^{k+1} = x_\psi^k - \gamma^k \psi_{\mathrm{i}^k}^k z_{\mathrm{i}^k}^k$. [cf. (5.17)]. We have

$$\begin{aligned}
M_F(x^k) &\leq \left|\nabla F(\bar{x}^k)\right|^2 + \left\|x^k - \mathbf{1}\bar{x}^k\right\|^2 && (5.28)\\
&\leq \left|\nabla F(\bar{x}^k)\right|^2 + 2\left\|x^k - \mathbf{1}x_\psi^k\right\|^2 + 2\left\|\mathbf{1}x_\psi^k - \mathbf{1}\bar{x}^k\right\|^2 && (5.29)\\
&\leq \left|\nabla F(\bar{x}^k)\right|^2 + 2\left\|x^k - \mathbf{1}x_\psi^k\right\|^2 + 2\left\|J\left(\mathbf{1}x_\psi^k - x^k\right)\right\|^2 &&\\
&\leq \left|\nabla F(\bar{x}^k)\right|^2 + 4\left\|x^k - \mathbf{1}x_\psi^k\right\|^2. && (5.30)
\end{aligned}$$

We bound now $\left|\nabla F(\bar{x}^k)\right|$; we have

$$\begin{aligned}
&\left|\nabla F(\bar{x}^k)\right| \leq \left|\nabla F(x_\psi^k)\right| + L\left|\bar{x}^k - x_\psi^k\right|\\
&\leq \left|\nabla F(x_\psi^k) - \bar{g}^k\right| + \left|\bar{g}^k - (\xi_{\mathrm{i}^k}^{k-1})^{-1} z_{\mathrm{i}^k}^k\right| + (\xi_{\mathrm{i}^k}^{k-1})^{-1}\left|z_{\mathrm{i}^k}^k\right| + \frac{L}{\sqrt{m}}\left\|J\left(x^k - \mathbf{1}x_\psi^k\right)\right\|\\
&\leq \left(C_L\sqrt{m} + \frac{L}{\sqrt{m}}\right)E_c^k + \eta^{-1}\,E_t^k + \eta^{-1}\,E_z^k,
\end{aligned} \tag{5.31}$$

where in the last inequality we used $\xi_{\mathrm{i}^k}^k \geq \eta$ for all k (cf. Lemma 4.4.3) and $\|J(x^k - \mathbf{1}x_\psi^k)\| \leq E_c^k$.

Eq. (5.27) follows readily from (5.30) and (5.31). □

Our ultimate goal is to show that the RHS of (5.27) is summable. To do so, we need two further results, Proposition 5.3.3 and Lemma 5.3.10 below. Proposition 5.3.3 establishes a connection between $F(x_\psi^{k+1})$ and E_c^k, E_t^k, and E_z^k.

Proposition 5.3.3. *In the above setting, there holds:* $k \in \mathbb{N}_0$,

$$
\begin{aligned}
F(x_{\psi}^{k+1}) \leq F(x_{\psi}^{0}) &+ \frac{1}{2}\left(L + \alpha^{-1} + \beta^{-1}\right)\sum_{t=0}^{k}(E_z^t)^2(\gamma^t)^2 \\
&- \eta\sum_{t=0}^{k}(E_z^t)^2\gamma^t + \frac{\alpha}{2}C_L^2 m\sum_{t=0}^{k}(E_c^t)^2 + \frac{\beta}{2}\eta^{-2}\sum_{t=0}^{k}(E_t^t)^2,
\end{aligned}
\tag{5.32}
$$

where α and β are two arbitrary positive constants.

Proof. By descent lemma, we get

$$
\begin{aligned}
&F(x_{\psi}^{k+1}) \leq F(x_{\psi}^{k}) + \gamma^k\psi_{\mathrm{i}^k}^k\left\langle\nabla F(x_{\psi}^k), -z_{\mathrm{i}^k}^k\right\rangle + \frac{L(\gamma^k\psi_{\mathrm{i}^k}^k)^2}{2}\left|z_{\mathrm{i}^k}^k\right|^2 \\
&\leq F(x_{\psi}^{k}) + \frac{L{\gamma^k}^2}{2}\left|z_{\mathrm{i}^k}^k\right|^2 + \gamma^k\psi_{\mathrm{i}^k}^k\left\langle(\xi_{\mathrm{i}^k}^{k-1})^{-1}z_{\mathrm{i}^k}^k, -z_{\mathrm{i}^k}^k\right\rangle + \gamma^k\psi_{\mathrm{i}^k}^k\left\langle\nabla F(x_{\psi}^k) - \bar{g}^k, -z_{\mathrm{i}^k}^k\right\rangle \\
&\qquad + \gamma^k\psi_{\mathrm{i}^k}^k\left\langle\bar{g}^k - (\xi_{\mathrm{i}^k}^{k-1})^{-1}z_{\mathrm{i}^k}^k, -z_{\mathrm{i}^k}^k\right\rangle \\
&\leq F(x_{\psi}^{k}) + \frac{L{\gamma^k}^2}{2}\left|z_{\mathrm{i}^k}^k\right|^2 - \gamma^k\eta\left|z_{\mathrm{i}^k}^k\right|^2 + \gamma^k C_L\sum_{\mathrm{j}=1}^{m}\left|x_{\psi}^k - x_{\mathrm{j}}^k\right|\left|z_{\mathrm{i}^k}^k\right| + \gamma^k\eta^{-1}E_t^k\left|z_{\mathrm{i}^k}^k\right| \\
&\leq F(x_{\psi}^{k}) + \frac{L{\gamma^k}^2}{2}\left|z_{\mathrm{i}^k}^k\right|^2 - \gamma^k\eta\left|z_{\mathrm{i}^k}^k\right|^2 + \gamma^k C_L\sqrt{m}E_c^k\left|z_{\mathrm{i}^k}^k\right| + \gamma^k\eta^{-1}E_t^k\left|z_{\mathrm{i}^k}^k\right| \\
&\leq F(x_{\psi}^{k}) + \frac{L{\gamma^k}^2}{2}\left|z_{\mathrm{i}^k}^k\right|^2 - \gamma^k\eta\left|z_{\mathrm{i}^k}^k\right|^2 + \frac{\alpha}{2}C_L^2 m(E_c^k)^2 + \frac{\alpha^{-1}}{2}\left|z_{\mathrm{i}^k}^k\right|^2{\gamma^k}^2 + \frac{\beta}{2}\eta^{-2}(E_t^k)^2 + \frac{\beta^{-1}}{2}\left|z_{\mathrm{i}^k}^k\right|^2(\gamma^k)^2 \\
&\leq F(x_{\psi}^{k}) + \frac{1}{2}\left(L + \alpha^{-1} + \beta^{-1}\right)(E_z^k)^2(\gamma^k)^2 - \eta(E_z^k)^2\gamma^k + \frac{\alpha}{2}C_L^2 m(E_c^k)^2 + \frac{\beta}{2}\eta^{-2}(E_t^k)^2.
\end{aligned}
$$

Applying the above inequality inductively one gets (5.32). □

The last result we need is a bound of $\sum_{t=0}^{k}(E_c^t)^2$ and $\sum_{t=0}^{k}(E_t^t)^2$ in (5.32) in terms of $\sum_{t=0}^{k}(E_z^t)^2(\gamma^t)^2$.

Lemma 5.3.10. *Define*

$$
\varrho_c \triangleq \frac{2C_2^2}{(1-\rho)^2} \quad \textit{and} \quad \varrho_t \triangleq \frac{36\left(C_0 C_L\right)^2\left(2C_2^2 + (1-\rho)^2\right)}{(1-\rho)^4}.
$$

The following holds: $k \in \mathbb{N}$,

$$
\begin{aligned}
\sum_{t=0}^{k}(E_c^t)^2 &\leq c_c + \varrho_c\sum_{t=0}^{k}(E_z^t)^2(\gamma^t)^2, \\
\sum_{t=0}^{k}(E_t^t)^2 &\leq c_t + \varrho_t\sum_{t=0}^{k}(E_z^t)^2(\gamma^t)^2,
\end{aligned}
\tag{5.33}
$$

where c_c and c_t are some positive constants.

Proof. The proof follows from Proposition 5.3.2 and Lemma 5.3.11 below, which is a variant of [106] (its proof is thus omitted).

Lemma 5.3.11. *Let $\{u^k\}_{k=0}^{\infty}, \{v_{\mathrm{i}}^k\}_{k=0}^{\infty}, \mathrm{i} = 1, \ldots, m,$ be nonnegative sequences; $\lambda \in (0,1)$; and $R_0 \in \mathbb{R}_+$ such that*

$$u^{k+1} \leq R\lambda^k + \sum_{l=0}^{k} \lambda^{k-l} v^l.$$

Then, there holds: $k \in \mathbb{N}$,

$$\sum_{l=0}^{k} (u^l)^2 \leq (u^0)^2 + \frac{2R^2}{1-\lambda^2} + \frac{2}{(1-\lambda)^2} \sum_{l=0}^{k} (v^l)^2.$$

□

Using (5.33) in (5.32), we finally obtain

$$\sum_{t=0}^{k} (E_z^t)^2 \gamma^t (\eta - \gamma^t C_4(\alpha, \beta)) \leq F(x_\psi^0) - F^{\inf} + C_5(\alpha, \beta) \tag{5.34}$$

with $C_4(\alpha,\beta) \triangleq (1/2)\,(L + \alpha^{-1} + \beta^{-1} + C_L^2 m\alpha\varrho_c + \eta^{-2}\beta\varrho_t)$ and $C_5(\alpha,\beta) = (1/2)\,(C_L^2 m\alpha c_c + \eta^{-2}\beta c_t)$; and $F^{\inf} > -\infty$ is the lower bound of F.

We are now ready to prove Theorems 6.4.2 and 5.3.3.

- Proof of Theorem 6.4.2 Set $\gamma^k \equiv \gamma$, for all $k \in \mathbb{N}_0$. By (5.34), one infers that $\sum_{t=0}^{\infty} {E_z^t}^2 < \infty$ if γ satisfies $0 < \gamma < \bar{\gamma}_2(\alpha, \beta)$, with $\bar{\gamma}_2(\alpha,\beta) \triangleq \eta / C_4(\alpha,\beta)$.Note that $\bar{\gamma}_2(\alpha,\beta)$ is maximized setting $\alpha = \alpha^\star = \left(C_L\sqrt{m\varrho_c}\right)^{-1}$ and $\beta = \beta^\star = \eta\varrho_t^{-1/2}$, resulting in

$$\bar{\gamma}_2(\alpha^\star, \beta^\star) = (2\eta)/(L + 2C_L\sqrt{m\varrho_c} + 2\eta^{-1}\sqrt{\varrho_t}). \tag{5.35}$$

Let $0 < \gamma < \bar{\gamma}_2(\alpha^\star, \beta^\star)$. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}_0$ such that $M_F(x^k) \leq \delta$. Then we have

$$T_\delta \cdot \delta < \sum_{k=0}^{T_\delta - 1} M_F(x^k) \leq \sum_{k=0}^{\infty} M_F(x^k) \overset{(5.27)}{\leq} C_3 \sum_{k=0}^{\infty} (E_c^k)^2 + 3\eta^{-2} \sum_{k=0}^{\infty} \left((E_t^k)^2 + (E_z^k)^2\right)$$
$$\overset{(5.33),(5.34)}{\leq} \frac{F(x_\psi^0) - F^{\inf} + C_5(\alpha^\star,\beta^\star)}{\gamma(\eta - \gamma C_4(\alpha^\star,\beta^\star))} \cdot C_6 + C_7 < \infty$$

where $C_6 \triangleq C_3 \varrho_c(\gamma)^2 + 3\eta^{-2}\left(\varrho_t(\gamma)^2+1\right)$ and C_7 is some constant. Therefore, $T_\delta = \mathcal{O}(1/\delta)$.

- Proof of Theorem 5.3.3.

We begin showing that the step-size sequence $\{\gamma^t\}_{t\in\mathbb{N}_0}$ induced by the local step-size sequence $\{\alpha^t\}_{t\in\mathbb{N}_0}$ and the asynchrony mechanism satisfying Assumption 4.3.2 is nonsummable. The proof is straightforward and is thus omitted.

Lemma 5.3.12. *Let $\{\gamma^t\}_{t\in\mathbb{N}_0}$ be the global step-size sequence resulted from Algorithm 4, under Assumption 4.3.2. Then, there hold:* $\lim_{t\to\infty}\gamma^t = 0$ *and* $\sum_{t=0}^{\infty}\gamma^t = \infty$.

Since $\lim_{t\to\infty}\gamma^t = 0$, there exists a sufficiently large $k \in \mathbb{N}$, say $\bar{k}$, such that $\eta - \gamma^k C_4(\alpha^\star, \beta^\star) \geq \eta/2$ for all $k > \bar{k}$. It is not difficult to check that this, together with (5.34), yields $\sum_{k=0}^{\infty}(E_z^k)^2\gamma^k < \infty$. We can then write

$$\sum_{k=0}^{\infty} M_F(x^k)\gamma^k \overset{(5.27)}{\leq} C_3\sum_{k=0}^{\infty}(E_c^k)^2\gamma^k + 3\eta^{-2}\sum_{k=0}^{\infty}\left((E_t^k)^2 + (E_z^k)^2\right)\gamma^k < C_8, \tag{5.36}$$

for some finite constant C_8, where in the last inequality we used (5.33), $\sum_{k=0}^{\infty}(E_z^k)^2\gamma^k < \infty$ and $\lim_{t\to\infty}\gamma^t = 0$.

Let $N_\delta \triangleq \inf\left\{k \in \mathbb{N}_0 \,:\, \sum_{t=0}^{k}\gamma^t \geq C_8/\delta\right\}$. Note that N_δ exists, as $\sum_{k=0}^{\infty}\gamma^k = \infty$ (cf. Lemma 5.3.12). Let $T_\delta \triangleq \inf\left\{k \in \mathbb{N}_0 \,:\, M_F(x^k) \leq \delta\right\}$. It must be $T_\delta \leq N_\delta$. In fact, suppose by contradiction that $T_\delta > N_\delta$; and thus $M_F(x^k) > \delta$, for $0 \leq k \leq N_\delta$. It would imply $\sum_{k=0}^{N_\delta} M_F(x^k)\gamma^k > \delta\sum_{k=0}^{N_\delta}\gamma^k \geq \delta\cdot(C_8/\delta) = C_8$, which contradicts (5.36). This proves (5.10).

5.4 Numerical Results

We test ASY-SONATA on the least square regression and the binary classification problems. The MATLAB code can be found at https://github.com/YeTian-93/ASY-SONATA.

5.4.1 Least square regression

In the LS problem, each agent i aims to estimate an unknown signal $x_0 \in \mathbb{R}^d$ through linear measurements $b_\mathrm{i} = M_\mathrm{i} x_0 + d_\mathrm{i}$, where $M_\mathrm{i} \in \mathbb{R}^{d_\mathrm{i}\times d}$ is the sensing matrix, and $d_\mathrm{i} \in \mathbb{R}^{d_\mathrm{i}}$

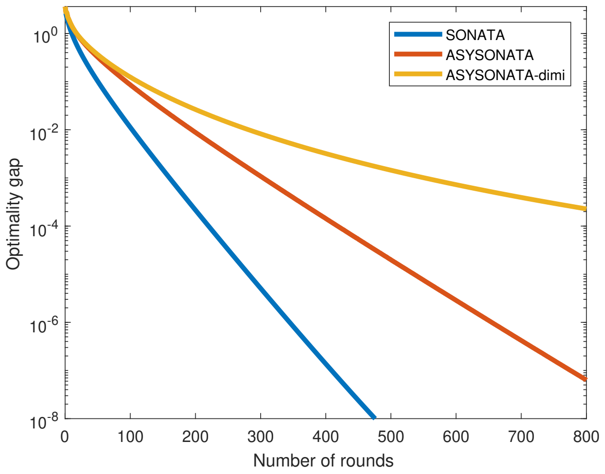


Figure 5.1. Directed graphs: optimality gap J^k versus number of rounds.

is the additive noise. The LS problem can be written in the form of (P), with each $f_{\mathrm{i}}(x) = \|M_{\mathrm{i}}x - b_{\mathrm{i}}\|^2$.

Data: We fix x_0 with its elements being i.i.d. random variables drawn from the standard normal distribution. For each M_{i}, we firstly generate all its elements as i.i.d. random variables drawn from the standard normal distribution, and then normalize the matrix by multiplying it with the reciprocal of its spectral norm. The elements of the additive noise d_{i} are i.i.d. Gaussian distributed, with zero mean and variance equal to 0.04. We set $d = 200$ and $d_{\mathrm{i}} = 30$ for each agent. **Network model:** We simulate a network of $m = 30$ agents. Each agent i has 3 out-neighbors; one of them belongs to a directed cycle graph connecting all the agents while the other two are picked uniformly at random. **Asynchronous model:** Agents are activated according to a cyclic rule where the order is randomly permuted at the beginning of each round. Once activated, every agent performs all the steps as in Algorithm 4 and then sends its updates to all its out-neighbors. Each transmitted message has (integer) traveling time which is drawn uniformly at random within the interval $[0, D^{\mathrm{tv}}]$. We set $D^{\mathrm{tv}} = 40$.

We test ASY-SONATA with a constant step size $\gamma = 3.5$, and also a diminishing step-size rule with each agent updating its local step size according to $\alpha^{t+1} = \alpha^t \left(1 - 0.001 \cdot \alpha^t\right)$ and $\alpha^0 = 3.5$; as benchmark, we also simulate its synchronous instance, with step size

$\gamma = 0.8$. In Fig. 5.1, we plot $J^k \triangleq (1/m)\sqrt{\sum_{\mathrm{i}=1}^m \|x_\mathrm{i}^k - x^\star\|_2^2}$ versus the number of rounds (one round corresponds to one update of all the agents). The curves are averaged over 100 Monte-Carlo simulations, with different graph and data instantiations. The plot clearly shows linear convergence of ASY-SONATAwith a constant step-size.

5.4.2 Binary classification

In this subsection, we consider a strongly convex and nonconvex instance of Problem (P) over digraphs, namely: the regularized logistic regression (RLR) and the robust classification (RC) problems. Both formulations can be abstracted as:

$$\min_x \frac{1}{|\mathcal{D}|}\sum_{\mathrm{i}=1}^m \sum_{\mathrm{j}\in\mathcal{D}_\mathrm{i}} V(y_\mathrm{j}\cdot \ell_x(u_\mathrm{j})) + \lambda\Big\|\nabla \ell_x(\cdot)\Big\|_2^2, \tag{5.37}$$

where $\mathcal{D} = \cup_{\mathrm{i}=1}^m \mathcal{D}_\mathrm{i}$ is the set of indices of the data distributed across the agents, with agent i owning $\mathcal{D}_\mathrm{i}$, and $\mathcal{D}_\mathrm{i} \cap \mathcal{D}_l = \emptyset$, for all $\mathrm{i} \neq l$; u_j and $y_\mathrm{j} \in \{-1, 1\}$ are the feature vector and associated label of the j-th sample in $\mathcal{D}$; $\ell_x(\cdot)$ is a linear function, parameterized by x; and V is the loss function. More specifically, if the RLR problem is considered, V reads $V(r) = \frac{1}{1+\mathrm{e}^{-r}}$ while for the RC problem, we have [107]

$$V(r) = \begin{cases} 0, & \text{if } r > 1; \\ \frac{1}{4}r^3 - \frac{3}{4}r + \frac{1}{2}, & \text{if } -1 \leq r \leq 1; \\ 1, & \text{if } r < -1. \end{cases}$$

Data: We use the following data sets for the RLR and RC problems. (RLR): We set $\ell_x(u) = x^\top u$, $d = 100$, each $|\mathcal{D}_\mathrm{i}| = 20$, and $\lambda = 0.01$. The underlying statistical model is the following: We generated the ground truth $\widehat{x}$ with i.i.d. $\mathcal{N}(0, 1)$ components; each training pair $(u_\mathrm{j}, y_\mathrm{j})$ is generated independently, with each element of u_j being i.i.d. $\mathcal{N}(0, 1)$ and y_j is set as 1 with probability $V(\ell_{\widehat{x}}(u_\mathrm{j}))$, and -1 otherwise. (RC): We use the Cleveland Heart Disease Data set with 14 features [37], preprocessing it by deleting observations with missing entries, scaling features between 0-1, and distributing the data to agents evenly. We set $\ell_x(u) = \mathrm{e}_{15}^\top x + \sum_{d=1}^{14} \mathrm{e}_d^\top x\, \mathrm{e}_d^\top u$. **Network model:** We simulated a digraph of $m = 30$ agents.

Each agent has 7 out-neighbors; one of them belongs to a directed cycle connecting all the agents while the other 6 are picked uniformly at random. One row and one column stochastic matrix with uniform weights are generated. **Asynchronous model:** a) Activation lists are generated by concatenating *random rounds.* To generate one round, we first sample its length uniformly from the interval $[m, T]$, with $T = 90$. Within a round, we first have each agent appearing exactly once and then sample agents uniformly for the remaining spots. Finally a random shuffle of the agents order is performed on each round; b) Each transmitted message has (integer) traveling time which is sampled uniformly from the interval $[0, D^{\mathrm{tv}}]$, with $D^{\mathrm{tv}} = 90$.

We compare the performance of our algorithm with AsySubPush [108] and AsySPA [109]. AsySubPush and AsySPA differ from ASY-SONATA in the following aspects: i) they do not employ any gradient tracking mechanism; ii) they cannot handle packet losses and purge out old information from the system (information is used as it is received); iii) when F is strongly convex, they provably converge at *sublinear* rate; and iv) they cannot handle nonconvex F. The step sizes of all algorithms are manually tuned to obtain the best practical performance. We run two instances of ASY-SONATA, one employing a constant step size $\gamma = 0.4$ and the other one using the diminishing step size rule $\alpha^{t+1} = \alpha^t\,(1 - 0.001 \cdot \alpha^t)$, where $\alpha^0 = 0.5$ and t is the local iteration counter. For AsySubPush (resp. AsySPA) we set, for each agent i, $\alpha_{\mathrm{i}} = 0.0001$ (resp. $\rho(k) = c/\sqrt{k}$ with $c = 0.01$) in RLC and $\alpha_{\mathrm{i}} = 0.00001$ (resp. $\rho(k) = c/\sqrt{k}$ with $c = 0.001$) in RC. The result is averaged over 20 Monte Carlo experiments with different digraph instances, and is presented in Fig. 5.2; for each algorithm, we plot the merit functions M_{sc} (left panel) and M_{F} (right panel) evaluated in the generated trajectory versus the global iteration counter k. Consistently with the convergence theory, ASY-SONATA with a constant step size exhibits a linear convergence rate. Also, ASY-SONATA outperforms the other two algorithms; this is mainly due to i) the presence in ASY-SONATA of an asynchronous gradient tracking mechanism which provides, at each iteration, a better estimate of ∇F; and ii) the possibility in ASY-SONATA to discard old information when received after a newer one [cf. (4.11)].

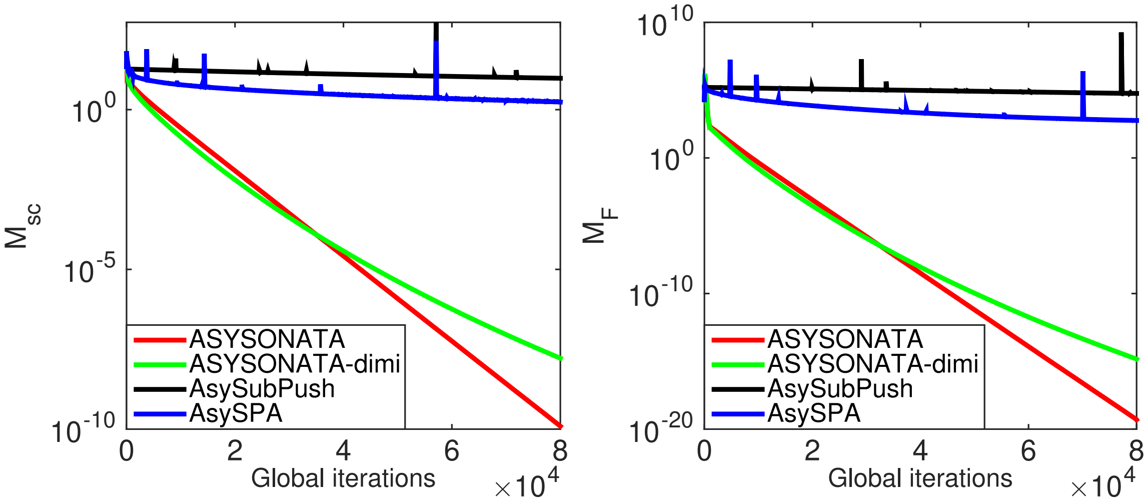


Figure 5.2. L: regularized logistic regression; R: robust classification.

5.5 Conclusions

We proposed ASY-SONATA, a distributed asynchronous algorithmic framework for convex and nonconvex (unconstrained, smooth) multi-agent problems, over digraphs. The algorithm is robust against uncoordinated agents' activation and (communication/computation) (time-varying) delays. When employing a constant step-size, ASY-SONATA achieves a linear rate for strongly convex objectives–matching the rate of a centralized gradient algorithm– and sublinear rate for (non)convex problems. Sublinear rate is also established when agents employ uncoordinated diminishing step-sizes, which is more realistic in a distributed setting. To the best of our knowledge, ASY-SONATA is the first distributed algorithm enjoying the above properties, in the general asynchronous setting described in the chapter.

5.6 Appendix: Proofs of Theorems

5.6.1 Proof of Lemma 5.3.5

Fix $N \in \mathbb{N}$, and let k such that $1 \leq k+1 \leq N$. We have:

$$\begin{aligned}\frac{u^{k+1}}{\lambda^{k+1}} &\leq \frac{R_0}{\lambda}\left(\frac{\lambda_0}{\lambda}\right)^k + \sum_{\mathrm{i}=1}^{m}\frac{R_\mathrm{i}}{\lambda}\sum_{l=0}^{k}\left(\frac{\lambda_\mathrm{i}}{\lambda}\right)^{k-l}\frac{v_\mathrm{i}^l}{\lambda^l}\\ &\leq \frac{R_0}{\lambda} + \sum_{\mathrm{i}=1}^{m}\frac{R_\mathrm{i}}{\lambda}\left|v_\mathrm{i}\right|^{\lambda,N}\sum_{l=0}^{k}\left(\frac{\lambda_\mathrm{i}}{\lambda}\right)^{k-l} \leq \frac{R_0}{\lambda} + \sum_{\mathrm{i}=1}^{m}\frac{R_\mathrm{i}}{\lambda-\lambda_\mathrm{i}}\left|v_\mathrm{i}\right|^{\lambda,N}.\end{aligned}$$

Hence,

$$|u|^{\lambda,N} \leq \max\left(u_0, \frac{R_0}{\lambda} + \sum_{\mathrm{i}=1}^{m}\frac{R_\mathrm{i}}{\lambda-\lambda_\mathrm{i}}\left|v_\mathrm{i}\right|^{\lambda,N}\right) \leq u^0 + \frac{R_0}{\lambda} + \sum_{\mathrm{i}=1}^{m}\frac{R_\mathrm{i}}{\lambda-\lambda_\mathrm{i}}\left|v_\mathrm{i}\right|^{\lambda,N}. \quad \square$$

5.6.2 Proof of Theorem 5.3.7

From [110, Ch. 5.6], we know that if $\rho(T) < 1$, then $\lim_{k\to\infty} T^k = 0$, the series $\sum_{k=0}^{\infty} T^k$ converges (wherein we define $T^0 \triangleq m$), $m - T$ is invertible and $\sum_{k=0}^{\infty} T^k = (m-T)^{-1}$.

Given $N \in \mathbb{N}$, using (5.22) recursively, yields: $u^{\lambda,N} \leq Tu^{\lambda,N} + \beta \leq T\left(Tu^{\lambda,N} + \beta\right) + \beta = T^2u^{\lambda,N} + (T+m)\,\beta \leq \cdots \leq T^\ell u^{\lambda,N} + \sum_{k=0}^{\ell-1} T^k\beta$, for any $\ell \in \mathbb{N}$. Let $\ell \to \infty$, we get $u^{\lambda,N} \leq (m-T)^{-1}\beta$. Since this holds for any given $N \in \mathbb{N}$, we have $u^\lambda \leq (m-T)^{-1}\beta$. Hence, u^λ is bounded, and thus each u_i^k vanishes at an R-linear rate $\mathcal{O}(\lambda^k)$. □

5.6.3 Proof of the rate decay (5.8) in Theorem 6.4.1

Let $\lambda \geq \mathcal{L}(\gamma) + \epsilon\gamma$, with $\epsilon > 0$ to be properly chosen. Then,

$$\mathfrak{B}(\lambda;\gamma) \leq \left(1 + \frac{L}{\epsilon}\right)\frac{b_2\gamma}{\lambda-\rho} + \left(\left(1+\frac{L}{\epsilon}\right)\frac{b_2}{\lambda-\rho} + b_1 + \frac{Lb_2}{\epsilon}\right)\frac{C_2\gamma}{\lambda-\rho}. \tag{5.38}$$

Using $\lambda - \rho < 1$, a sufficient condition for Eq. (5.25) is [RHS less than one]

$$\left(b_1C_2 + \frac{Lb_2C_2}{\epsilon} + \left(1+\frac{L}{\epsilon}\right)b_2(1+C_2)\right)\gamma \leq (\lambda-\rho)^2. \tag{5.39}$$

Now set $\epsilon = (\tau\eta^2)/2$. Since the RHS of the above inequality can be arbitrarily close to $(1-\rho)^2$, an upper bound of γ is

$$\hat{\gamma}_2 \triangleq (1-\rho)^2 \Big/ \underbrace{\left(b_1 C_2 + \frac{2Lb_2C_2}{\tau\eta^2} + \left(1 + \frac{2L}{\tau\eta^2}\right) b_2(1+C_2)\right)}_{\triangleq J_1}.$$

According to $\lambda \geq \mathcal{L}(\gamma) + \epsilon\gamma$ and (5.39), we get

$$\lambda = \max\left(1 - \frac{\tau\eta^2\gamma}{2}, \quad \rho + \sqrt{J_1\gamma}\right). \tag{5.40}$$

Notice that when γ goes from 0 to $\hat{\gamma}_2$, the first argument inside the max operator decreases from 1 to $1-(\tau\eta^2\hat{\gamma}_2)/2$, while the second argument increases from ρ to 1. Letting $1 - \frac{\tau\eta^2\gamma}{2} = \rho + \sqrt{J_1\gamma}$, we get the solution as $\hat{\gamma}_1 = \left(\frac{\sqrt{J_1+2\tau\eta^2(1-\rho)}-\sqrt{J_1}}{\tau\eta^2}\right)^2$. The expression of λ as in (5.8) follows readily. □

5.6.4 Proof of Lemma 5.3.8

"$\Longleftarrow$:" From $p(1) > 0$, we know that $\sum_{\mathrm{i}=1}^{m} a_{\mathrm{i}} < 1$. We prove by contradiction. Suppose there is a root z_* of $p(z)$ satisfying $|z_*| \geq 1$, then we have

$$z_*^m = a_1 z_*^{m-1} + a_2 z_*^{m-2} + \ldots + a_{m-1} z_* + a_m.$$

Clearly $z_* \neq 0$, so equivalently

$$1 = a_1 \frac{1}{z_*} + a_2 \frac{1}{z_*^2} + \ldots + a_{m-1} \frac{1}{z_*^{m-1}} + a_m \frac{1}{z_*^m}.$$

Further,

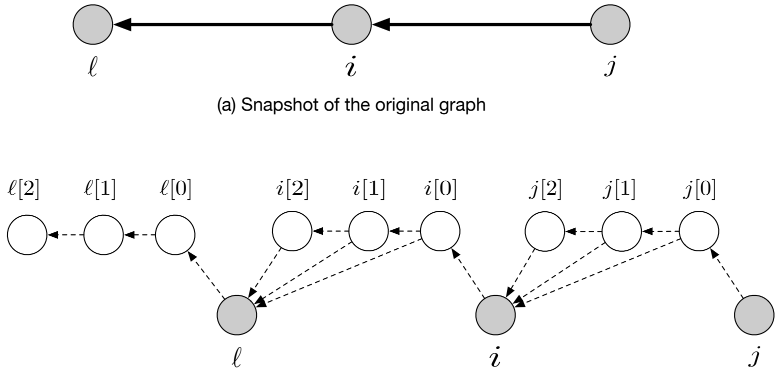
$$\begin{aligned}
1 &= \left| a_1 \frac{1}{z_*} + a_2 \frac{1}{z_*^2} + \ldots + a_{m-1} \frac{1}{z_*^{m-1}} + a_m \frac{1}{z_*^m} \right| \\
&\leq a_1 \frac{1}{|z_*|} + a_2 \frac{1}{|z_*|^2} + \ldots + a_{m-1} \frac{1}{|z_*|^{m-1}} + a_m \frac{1}{|z_*|^m} \\
&\leq a_1 + a_2 + \ldots + a_{m-1} + a_m < 1.
\end{aligned}$$

This is a contradiction.

"$\Longrightarrow$:" If $p(1) = 0$, we clearly have that $z_p \geq 1$. Now suppose $p(1) < 0$. Because $\lim_{z\in\mathbb{R}, z\to+\infty} p(z) = +\infty$ and $p(z)$ is continuous on $\mathbb{R}$, we know that $p(z)$ has a zero in $(1, +\infty) \subset \mathbb{R}$. Thus $z_p > 1$. □

5.6.5 Proof of Lemma 6.6.1

We interpret the dynamical system (6.18) over an augmented graph. We begin constructing the augmented graph obtained adding virtual nodes to the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We associate each node $\mathrm{i} \in \mathcal{V}$ with an ordered set of virtual nodes $\mathrm{i}[0], \mathrm{i}[1], \ldots, \mathrm{i}[D]$; see Fig. 5.3. We still call the nodes in the original graph G as *computing agents* and the virtual nodes as *noncomputing agents*. We now identify the neighbors of each agent in this augmented system. Any noncomputing agent $\mathrm{i}[d]$, $d = D, D-1, \cdots, 1$, can only receive information from the previous virtual node $\mathrm{i}[d-1]$; $\mathrm{i}[0]$ can only receive information from the real node i or simply keep its value unchanged; computing agents cannot communicate among themselves.



(a) Snapshot of the original graph

(b) Augmented graph associated with (a)

Figure 5.3. Example of augmented graph, when the maximum delay is $D = 2$; three noncomputing agents are added for each node $\mathrm{i} \in \mathcal{V}$.

At the beginning of each iteration k, every computing agent $\mathrm{i} \in \mathcal{V}$ will store the information x_{i}^k; whereas every noncomputing agent $\mathrm{i}[d]$, with $d = 0, 1, \cdots, D$, will store the delayed information v_{i}^{k-d}. The dynamics over the augmented graph happening in iteration k is described by (6.18). In words, any noncomputing agent $\mathrm{i}[d]$ with $\mathrm{i} \in \mathcal{V}$ and $d = D, D-1, \cdots, 1$ receives the information from $\mathrm{i}[d-1]$; the noncomputing agent $\mathrm{i}^k[0]$ receives the perturbed information $x_{\mathrm{i}^k}^k + \delta^k$ from node i^k; the values of noncomputing agents $\mathrm{j}[0]$ for $\mathrm{j} \in \mathcal{V} \setminus \{\mathrm{i}^k\}$ remain the same; node i^k sets its new value as a weighted average of the perturbed information $x_{\mathrm{i}^k}^k + \delta^k$ and $v_{\mathrm{j}}^{k-d_{\mathrm{j}}^k}$'s received from the virtual nodes $\mathrm{j}[d_{\mathrm{j}}^k]$'s for $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$; and the values of the other computing agents remain the same. The dynamics is further illustrated in Fig. 5.4. The following Lemma shows that the product of a sufficiently large number of any instantiations of the matrix $\widehat{W}^k$, under Assumption 4.3.2, is a scrambling matrix.

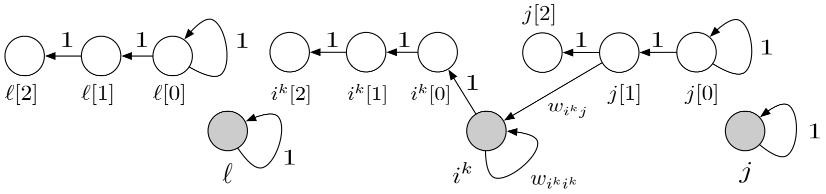


Figure 5.4. The dynamics in iteration k. Agent i^k uses the delayed information v_{j}^{k-1} from the virtual node j[1].

Lemma 5.6.1. *Let $\{\widehat{W}^k\}_{k\in\mathbb{N}_0}$ be the sequence of augmented matrices generated according to the dynamical system (5.11), under Assumption 4.3.2, and with W satisfying Assumption* **??** *(i), (ii). Then, for any $k \in \mathbb{N}_0$, $\widehat{W}^k$ is row stochastic and $\widehat{W}^{k+K_1-1:k}$ has the property that all entries of its first m columns are uniformly lower bounded by η.*

Proof. We study any entry $\widehat{W}_{hm}^{k+K_1-1}$ with $m \in \mathcal{V}$ and $h \in \widehat{\mathcal{V}}$. We prove the result by considering the following four cases.

(i) Assume $h = m \in \mathcal{V}$. Since $\widehat{W}_{hh}^k \geq \bar{m}$ for any $k \in \mathbb{N}_0$ and any $h \in \mathcal{V}$, we have $\widehat{W}_{hh}^{k+s-1:k} \geq \prod_{t=k}^{k+s-1} \widehat{W}_{hh}^t \geq \bar{m}^s$ for $\forall\, k \in \mathbb{N}_0$, $\forall\, s \in \mathbb{N}$ and $\forall\, h \in \mathcal{V}$.

(ii) Assume that $(m, h) \in \mathcal{E}$. Suppose that the first time when agent h wakes up during the time interval $[k+T+D, k+2T+D-1]$ is s, and agent h uses the information v_m^{s-d} from the noncomputing agent $m[d]$. Then we have

$$\widehat{W}_{h,m[0]}^{s:s-d} \geq \widehat{W}_{h,m[d]}^{s} \cdots \widehat{W}_{m[1],m[0]}^{s-d} = w_{hm} \geq \bar{m}^{d+1}.$$

Then suppose that the last time when agent m wakes up during the time interval $[s-d-T, s-d-1]$ is $s-d-t$. The noncomputing agent $m[0]$ receives some perturbed information from agent m at iteration $s-d-t$ and then performs self-loop (i.e., keep its value unchanged) during the time interval $[s-d-t+1, s-d-1]$. Thus we have

$$\widehat{W}_{m[0],m}^{s-d-1:s-d-t} = \widehat{W}_{m[0],m[0]}^{s-d-1:s-d-t+1}\widehat{W}_{m[0],m}^{s-d-t} = 1 \cdot 1 \geq \bar{m}^{t-1}.$$

Therefore we have

$$\begin{aligned}&\widehat{W}_{hm}^{k+2T+D-1:k} \geq \widehat{W}_{hh}^{k+2T+D-1:s+1}\widehat{W}_{h,m[0]}^{s:s-d}\widehat{W}_{m[0],m}^{s-d-1:s-d-t}\widehat{W}_{mm}^{s-d-t-1:k}\\ &\geq \bar{m}^{k+2T+D-s-1}\bar{m}^{d+1}\bar{m}^{t-1}\bar{m}^{s-d-t-k} \geq \bar{m}^{2T+D}.\end{aligned}$$

Further we have

$$\widehat{W}_{hm}^{k+K_1-1} \geq \widehat{W}_{hh}^{k+K_1-1:k+2T+D}\widehat{W}_{hm}^{k+2T+D-1:k} \geq \bar{m}^{K_1-2T-D}\bar{m}^{2T+D} = \bar{m}^{K_1}.$$

(iii) Assume that $m \neq h$ and $(m,h) \in \mathcal{V} \times \mathcal{V} \setminus \mathcal{E}$. Because the graph $(\mathcal{V}, \mathcal{E})$ is connected, there are mutually different agents $\mathrm{i}_1, \ldots, \mathrm{i}_r$ with $r \leq m-2$ such that

$$(m, \mathrm{i}_1), (\mathrm{i}_1, \mathrm{i}_2), \ldots, (\mathrm{i}_{r-1}, \mathrm{i}_r), (\mathrm{i}_r, h) \subset \mathcal{E},$$

which is actually a directed path from m to h. Then, by result proved in (ii), we have

$$\begin{aligned}&\widehat{W}_{hm}^{k+(m-1)(2T+D)-1:k}\\ &= \widehat{W}_{hh}^{k+(m-1)(2T+D)-1:k+(r+1)(2T+D)}\widehat{W}_{h\mathrm{i}_r}^{k+(r+1)(2T+D)-1:k+r(2T+D)} \cdots \widehat{W}_{\mathrm{i}_2\mathrm{i}_1}^{k+2(2T+D)-1:k+2T+D}\widehat{W}_{\mathrm{i}_1 m}^{k+2T+D-1:k}\\ &\geq \bar{m}^{(m-r-2)(2T+D)}\bar{m}^{(r+1)(2T+D)} = \bar{m}^{(m-1)(2T+D)}.\end{aligned}$$

Then we can easily get

$$\begin{aligned}&\widehat{W}_{hm}^{k+K_1-1:k} = \widehat{W}_{hh}^{k+K_1-1:k+(m-1)(2T+D)}\widehat{W}_{hm}^{k+(m-1)(2T+D)-1:k}\\ &\geq \bar{m}^{K_1-(m-1)(2T+D)}\bar{m}^{(m-1)(2T+D)} = \bar{m}^{K_1}.\end{aligned}$$

(iv) If h is a noncomputing node, it must be affiliated with a computing agent $\mathrm{j} \in \mathcal{V}$, i.e., there exists $0 \le d \le D$ such that $h = \mathrm{j}[d]$. Then we have

$$\widehat{W}_{h,\mathrm{j}[0]}^{k+K_1-1:k+K_1-d} = \widehat{W}_{\mathrm{j}[d],\mathrm{j}[d-1]}^{k+K_1-1} \cdots \widehat{W}_{\mathrm{j}[1],\mathrm{j}[0]}^{k+K_1-d} = 1.$$

Suppose that the last time when agent j wakes up during the time interval $[k + K_1 - d - T, k + K_1 - d - 1]$ is s. We have

$$\widehat{W}_{\mathrm{j}[0],\mathrm{j}}^{k+K_1-d-1:s} = \widehat{W}_{\mathrm{j}[0],\mathrm{j}[0]}^{k+K_1-d-1:s+1} \widehat{W}_{\mathrm{j}[0],\mathrm{j}}^{s} = 1.$$

By results proved before, we have

$$\begin{aligned}&\widehat{W}_{hm}^{k+K_1-1:k} \ge \widehat{W}_{h,\mathrm{j}[0]}^{k+K_1-1:k+K_1-d} \widehat{W}_{\mathrm{j}[0],\mathrm{j}}^{k+K_1-d-1:s} \widehat{W}_{\mathrm{jj}}^{s-1:k+(m-1)(2T+D)} \widehat{W}_{\mathrm{j}m}^{k+(m-1)(2T+D)-1:k} \\ &\ge 1 \cdot 1 \cdot \bar{m}^{s-k-(m-1)(2T+D)} \bar{m}^{(m-1)(2T+D)} \ge \bar{m}^{K_1}.\end{aligned}$$

□

Based on Lemma 5.6.1, we get the following result according to the discussion in [75].

Lemma 5.6.2. *In the setting above, there exists a sequence of stochastic vectors* $\{\psi^k\}_{k\in\mathbb{N}_0}$ *such that for any* $k \ge t \ge 0$*,*

$$\left\|\widehat{W}^{k:t} - 1{\psi^t}^\top\right\|_\infty \le \frac{2(1 + \bar{m}^{-K_1})}{1 - \bar{m}^{-K_1}} \rho^{k-t}.$$

Furthermore, $\psi_{\mathrm{i}}^k \ge \eta = \bar{m}^{K_1}$ *for all* $k \ge 0$ *and* $\mathrm{i} \in \mathcal{V}$*.*

The above result leads to Lemma 6.6.1 by noticing that

$$\left\|\widehat{W}^{k:t} - 1{\psi^t}^\top\right\| \le \sqrt{(D+2)m}\left\|\widehat{W}^{k:t} - 1{\psi^t}^\top\right\|_\infty \le C_2 \rho^{k-t}.$$

6. ASYNCHRONOUS DECENTRALIZED ALGORITHM - PART III: ASY-DSCA

In this chapter, we propose the asynchronous distributed algorithm ASY-DSCA for multi-agent optimization over static digraphs. Compared to the algorithm ASY-SONATA proposed in the previous chapter, ASY-DSCA is meant for general nonsmooth constrained problems. When the objective function is nonconvex, ASY-DSCA provably converges to a stationary solution at a sublinear rate. ASY-DSCA converges at an R-linear rate to the optimal solution when the problem is convex and satisfies the Luo-Tseng error bound condition, which is weaker than the strong convexity. This is another improvement on the result of the previous chapter, as a strongly convex objective function is required for ASY-SONATA to converge linearly. The Luo-Tseng error bound condition is satisfied by several non-strongly-convex functions arising from machine learning applications; examples include LASSO and logistic regression problems. ASY-DSCA is the first distributed algorithm provably achieving linear rate for such a class of problems.

The novel results of this chapter are available online at

- Ye Tian, Ying Sun, and Gesualdo Scutari. ”Asynchronous decentralized successive convex approximation.” arXiv preprint arXiv:1909.10144 (2019).

6.1 Introduction

In this chapter, we introduce ASY-DSCA, the first distributed asynchronous algorithm [in the sense (i) and (ii) discussed in Sec. 4.1] applicable to the *composite, constrained* optimization (P). ASY-DSCA builds on successive convex approximation techniques (SCA) [111]–[114]–agents solve strongly convex approximations of (P)–coupled with a suitably defined perturbed push-sum mechanism that is robust against asynchrony, whose goal is to track locally and asynchronously the average of agents' gradients. No specific activation mechanism for the agents' updates, coordination, or communication protocol is assumed, but only some mild conditions ensuring that information used in the updates does not become infinitely old. We remark that SCA offers a unified umbrella to deal efficiently with convex and nonconvex problems [111]–[114]: for several problems (P) of practical interest (cf. Sec. 6.2.1), a proper choice of the agents' surrogate functions to minimize leads to subproblems that admit a closed form solution (e.g., soft-thresholding and/or projection to the Euclidean ball). ASY-DSCA generalizes ASY-SONATA, by i) enabling SCA models in the agents' local updates; and ii) enlarging the class of optimization problems to include constraints and nonsmooth (convex) objectives.

We are not aware of any provably convergence scheme applicable to the envisioned decentralized asynchronous setting and Problem (P)–specifically in the presence of constraints or the nonsmooth term G–see Sec. 6.1.1 for a discussion of related works. This chapter fills exactly this gap.

6.1.1 Literature Review

On the asynchronous model: The literature on asynchronous methods is vast; based upon agents' activation rules and assumptions on delays, existing algorithms can be roughly grouped in three categories. **1)** Algorithms in [87]–[92] tolerate delayed information but require synchronization among agents, thus fail to meet the asynchronous requirement (i) above. **2)** On the other hand, schemes in [33], [83]–[86], [115], [116] accounts for agents' random (thus uncoordinated) activation; however, upon activation, they must use the most updated information from their neighbors, i.e., no delays are allowed; hence, they fail to meet requirement (ii). **3)** Asynchronous activations and delays are considered in [93]–[95], [97], [98] and [77], [96], [108], [117], [118], with the former (resp. latter) schemes employing random (resp. deterministic) activations. Some restrictions on the form of delays are imposed. Specifically, [77], [93]–[95] can only tolerate packet losses (either the information gets lost or is received with no delay); [108] handles only communication delays (eventually all the transmitted information is received by the intended agent); and [97], [98] assume that the agents' activation and delay as independent random variables, which is not realistic and hard to enforce in practice [80].

The only schemes we are aware of that are compliant with the asynchronous model (i) and (ii) are those in [117], [118]; however, they are applicable only to *smooth unconstrained* problems. Furthermore, all the aforementioned algorithms but [95], [117] are designed only for *convex* objectives U.

On the convergence rate: Referring to convergence rate guarantees, none of the aforementioned methods is proved to converge linearly in the *asynchronous* setting and when applied to *nonsmooth constrained* problems in the form (P). Furthermore, even restricting the focus to *synchronous* distributed methods or smooth unconstrained instances of (P), we are not aware of any distributed scheme that provably achieves linear rate without requiring U to be strongly convex; we refer to [10] for a recent literature review of synchronous distributed schemes belonging to this class. In the centralized setting, linear rate can be proved for first order methods under the assumption that U satisfies some error bound conditions, which are weaker than strongly convexity; see, e.g., [119]–[122]. A natural question is whether such results can be extended to (asynchronous) decentralized methods. This chapter provides a positive answer to this open question.

6.1.2 Summary of Contributions

• **Convergence rate:** Our convergence results are the following: **i)** For general nonconvex F in (P), a sublinear rate is established for a suitably defined merit function measuring both distance of the (average) iterates from stationary solutions and consensus disagreement; ii) When (P) satisfies the Luo-Tseng (LT) error bound condition [121], we establish *R-linear* convergence of the sequence generated by ASY-DSCA to an optimal solution. Notice that the LT condition is weaker than strong convexity, which is the common assumption used in the literature to establish linear convergence of distributed (even synchronous) algorithms. Our interest in the LT condition is motivated by the fact that several popular objective functions arising from machine learning applications are nonstrongly convex but satisfy the LT error bound; examples include popular empirical losses in high-dimensional statistics such as quadratic and logistic losses–see Sec.6.2.2 for more details. ASY-DSCA is the first *asynchronous distributed* algorithm with provably linear rate for such a class of problems over networks; this result is new even in the synchronous distributed setting.

• **New line of analysis:** We put forth novel convergence proofs, whose main novelties are highlighted next.

- New Lyapunov function for descent Our convergence analysis consists in carefully analyzing the interaction among the consensus, the gradient tracking and the nonconvex-nonsmooth-constrained optimization processes *in the asynchronous environment*. This interaction can be seen as a perturbation that each of these processes induces on the dynamics of the others. The challenge is proving that the perturbation generated by one system on the others is of a sufficiently small order (with respect to suitably defined metrics), so that convergence can be established and a convergence rate of suitably defined quantities be derived. Current techniques from centralized (nonsmooth) SCA optimization methods [111]–[114], error-bound analysis [121], and (asynchronous) consensus algorithms, alone or brute-forcely put together, do not provide a satisfactory answer: they would generate "too large" perturbation errors and do not exploit the interactions among different processes. On the other hand, existing approaches proposed for distributed algorithms are not applicable too (see Sec. 6.1.1 for a detailed review of the state of the art): they can neither deal with asynchrony (e.g., [10]) or be applicable to optimization problems with a nonsmooth function in the objective and/or constraints.

To cope with the above challenges our analysis builds on two new Lyapunov functions, one for nonconvex instances of (P) and one for convex ones. These functions are carefully crafted to combine objective value dynamics with consensus and gradient errors while accounting for asynchrony and outdated information in the agents' updates. Apart from the specific expression of these functions, a major novelty here is the use in the Lyapunov functions of weighting vectors that *endogenously vary based upon the asynchrony trajectory of the algorithm*–see Sec.6.6 (Step 2) and Sec.6.7 (Remark 6.7.1) for technical details. The descent property of the Lyapunov functions is the key step to prove that consensus and tracking errors vanish and further establish the desired converge rate of valid optimality/stationarity measures.

- Linear rate under the LT condition The proof of linear convergence of ASY-DSCA under the LT condition is a new contribution of this work. Existing proofs establishing linear rate of distributed synchronous and asynchronous algorithms [4], [8], [10], [30], [100] (including the previous convergence proof for ASY-SONATA) are not applicable here, as they all leverage strong convexity of F, a property that we do not assume. On the other hand, existing techniques showing linear rate of *centralized* first-order methods under the LT condition [121], [123] do not customize to our *distributed*, asynchronous setting. Roughly speaking, this is mainly due to the fact that use of the LT condition in [121], [123] is subject to proving descent on the objective function along the algorithm iterates, a property that can no longer be guaranteed in the distributed setting, due to the perturbations generated by the consensus and the gradient tracking errors. Asynchrony complicates further the analysis, as it induces unbalanced updating frequency of agents and the presence of the outdated information in agents' local computation. Our proof of linear convergence leverages the descent property of the proposed Lyapunov function to be able to invoke the LT condition in our distributed, asynchronous setting (see Sec.6.6 for a technical discussion on this matter).

6.2 Problem setup

We study Problem (P) under the following assumptions.

Assumption 6.2.1 (On Problem (P))**.** *The following hold:*

(i) The set $\mathcal{K} \subset \mathbb{R}^d$ is nonempty, closed, and convex;

(ii) Each $f_\mathrm{i} : \mathcal{O} \to \mathbb{R}$ is proper, closed and l-smooth, where $\mathcal{O} \supset \mathcal{K}$ is open; F is L-smooth with $L \triangleq m \cdot l$;

(iii) $G : \mathcal{K} \to \mathbb{R}$ is convex but possibly nonsmooth;

(iv) U is lower bounded on $\mathcal{K}$.

Note that each f_{i} need not be convex, and each agent i knows only its own f_{i} but not $\sum_{\mathrm{j}\neq\mathrm{i}} f_{\mathrm{j}}$. The regularizer G and the constraint set $\mathcal{K}$ are common knowledge to all agents.

6.2.1 Case study: Collaborative supervised learning

A timely application of the described decentralized setting and optimization Problem (P) is collaborative supervised learning. Consider a training data set $\{(u_s, y_s)\}_{s\in\mathcal{D}}$, where u_s is the input feature vector and y_s is the outcome associated to item s. In the envisioned decentralized setting, data $\mathcal{D}$ are partitioned into m subsets $\{\mathcal{D}_{\mathrm{i}}\}_{\mathrm{i}\in[m]}$, each of which belongs to an agent $\mathrm{i} \in [m]$. The goal is to learn a mapping $p(\cdot\,; x)$ parameterized by $x \in \mathbb{R}^d$ using *all* samples in $\mathcal{D}$ by solving $\min_{x\in\mathcal{K}} 1/|\mathcal{D}| \sum_{s\in\mathcal{D}} \ell\left(p(u_s; x), y_s\right) + G(x)$, wherein ℓ is a loss function that measures the mismatch between $p(u_s; x)$ and y_s; and G and $\mathcal{K}$ play the role of regularizing the solution. This problem is an instance of (P) with $f_{\mathrm{i}}(x) \triangleq 1/|\mathcal{D}| \sum_{s\in\mathcal{D}_{\mathrm{i}}} \ell\left(p(u_s; x), y_s\right)$. Specific examples of loss functions and regularizers are give next.

1) **Elastic net regularization for log linear models:** $\ell\left(p(u_s; x), y_s\right) \triangleq \Phi(u_s^\top x) - y_s \cdot (u_s^\top x)$ with Φ convex, $u_s \in \mathbb{R}^d$ and $y_s \in \mathbb{R}$; $G(x) \triangleq \lambda_1\big\|x\big\|_1 + \lambda_2\big\|x\big\|_2^2$ is the elastic net regularizer, which reduces to the LASSO regularizer when $(\lambda_1, \lambda_2) = (\lambda, 0)$ or the ridge regression regularizer when $(\lambda_1, \lambda_2) = (0, \lambda)$;

2) **Sparse group LASSO friedman2010note:** The loss function is the same as that in example 1), with $\Phi(t) = t^2/2$; $G(x) = \sum_{S\in\mathcal{J}} w_S\big\|x_S\big\|_2 + \lambda\big\|x\big\|_1$, where $\mathcal{J}$ is a partition of $[d]$;

3) **Logistic regression:** $\ell\left(p(u_s; x), y_s\right) \triangleq \ln(1 + \mathrm{e}^{-y_s\cdot u_s^\top x})$; popular choices of $G(x)$ are $G(x) \triangleq \lambda\big\|x\big\|_1$ or $G(x) \triangleq \lambda\big\|x\big\|_2^2$. The constraint set $\mathcal{K}$ is generally assumed to be bounded.

For large scale data sets, solving such learning problems is computationally challenging even if F is convex. When the problem dimension d is larger than the sample size $|\mathcal{D}|$, the Hessian of the empirical risk loss F is typically rank deficient and hence F is not strongly convex. Since linear convergence rate for decentralized methods is established in the literature only under strong convexity, it is unclear whether such a fast rate can be achieved under less restrictive conditions, e.g., embracing popular high-dimensional learning problems as those mentioned above. We show next that a positive answer to this question can be obtained leveraging the renowned LT error bound, a condition that has been wide explored in the literature of centralized optimization methods.

6.2.2 The Luo-Tseng error bound

Assumption 6.2.2. *(Error-bound conditions [121], [124], [125]):*

(i) *F is convex;*

(ii) *For any $\eta > \inf_{x\in\mathcal{K}} U(x)$, there exists $\epsilon, \kappa > 0$ such that:*

$$U(x) \leq \eta \quad \textit{and} \quad \left\| x - \textit{prox}_G(x - \nabla F(x)) \right\| \leq \epsilon \tag{6.1}$$

$$\Downarrow$$

$$\textit{dist}(x, \mathcal{K}^*) \leq \kappa \left\| x - \textit{prox}_G(x - \nabla F(x)) \right\|. \tag{6.2}$$

Assumption 6.2.2(ii) is a local growth condition on U around $\mathcal{K}^*$, crucial to prove linear rate. Note that for convex F, condition 6.2.2(ii) is equivalent to other renowned error bound conditions, such as the Polyak-Łojasiewicz [126], [127], the quadratic growth [128], and the Kurdyka-Łojasiewicz [120] conditions. A broad class of functions satisfying Assumption 6.2.2 is in the form $U(x) = F(x) + G(x)$, with F and G such that (cf. [129, Theorem 4], [119, Theorem 1]):

(i) $F(x) = h(Ax)$ is L-smooth, where h is strongly convex and A is any linear operator;

(ii) G is either a polyhedral convex function (i.e., its epigraph is a polyhedral set) or has a specific separable form as $G(x) = \sum_{S\in\mathcal{J}} w_S \left\| x_S \right\|_2 + \lambda \left\| x \right\|_1$, where $\mathcal{J}$ is a partition of the set $[d]$, and λ and w_S's are nonnegative weights (we used x_S to denote the vector whose component i is x_{i} if i $\in S$, and 0 otherwise);

(iii) $U(x)$ is coercive.

It follows that all examples listed in Section 6.2.1 satisfy Assumption 6.2.2. Hence, the proposed decentralized asynchronous algorithm, to be introduced, will provably achieve linear rate for such a general classes of problems.

6.3 Algorithmic development

Solving Problem (P) over $\mathcal{G}$ poses the following challenges: i) U is nonconvex/nonsmooth; ii) each agent i only knows its local loss f_i but not the global F; and iii) agents perform updates in an asynchronous fashion. Furthermore, it is well established that, when f_i are nonconvex (or convex only in some variable), using convex surrogates for f_i in the agents' subproblems rather than just linearization (as in gradient algorithms) provides more flexibility in the algorithmic design and can enhance practical convergence [111]–[114]. This motivated us to equip our distributed asynchronous design with SCA models.

To address these challenges, we develop our algorithm building on SONATA [9], [10], as to our knowledge it is the only synchronous decentralized algorithm for (P) capable to handle challenges i) and ii) and incorporating SCA techniques. Moreover, when employing a constant step size, it converges linearly to the optimal solution of (P) when F is strongly convex; and sublinearly to the set of stationary points of (P), when F is nonconvex. We begin briefly reviewing SONATA.

6.3.1 Preliminaries: the SONATA algorithm for nonsmooth constrained optimization [9], [10]

Each agent i maintains a local estimate x_i of the common optimization vector x, to be updated at each iteration; the k-th iterate is denoted by x_i^k. The specific procedure put forth by SONATA is given in Algorithm 5 and briefly described next.

(S.1): Local optimization. At each iteration k, every agent i locally solves a strongly convex approximation of Problem (P) at x_i^k, as given in (6.3a), where $\widetilde{f}_\mathrm{i} : \mathcal{K} \times \mathcal{K} \to \mathbb{R}$ is a so-called SCA surrogate of f_i, that is, satisfies Assumption 6.3.1 below. The second term in (6.3a), $\left(m y_\mathrm{i}^k - \nabla f_\mathrm{i}(x_\mathrm{i}^k)\right)^\top \left(x - x_\mathrm{i}^k\right)$, serves as a first order approximation of $\sum_{\mathrm{j}\neq\mathrm{i}} f_\mathrm{j}(x)$

```
Algorithm 5 The SONATA Algorithm
Data: For all agent i and $\forall j \in \mathcal{N}_i^{in}$, $x_i^0 \in \mathbb{R}^d$, $z_i^0 = y_i^0 = \nabla f_i(x_i^0)$, $\phi_i^0 = 1$. Set $k = 0$.
  While: a termination criterion is not met, each agent $i \in [m]$ do
    (S.1) Local optimization:
        $\widetilde{x}_i^k = \operatorname*{argmin}_{x \in \mathcal{K}} \Big\{ \widehat{U}_i\Big(x; x_i^k, m\, y_i^k - \nabla f_i(x_i^k)\Big) \triangleq \widetilde{f}_i(x; x_i^k) + (m\, y_i^k - \nabla f_i(x_i^k))^\top \Big(x - x_i^k\Big) + G(x) \Big\},$   (6.3a)
        $v_i^{k+1} = x_i^k + \gamma \Big(\widetilde{x}_i^k - x_i^k\Big).$   (6.3b)
    (S.2) Consensus step:
        $x_i^{k+1} = w_{ii} v_i^{k+1} + \sum_{j \in \mathcal{N}_i^{in}} w_{ij} v_j^{k+1}.$   (6.4)
    (S.3) Gradient tracking:
        $z_i^{k+1} = \sum_{j=1}^{m} a_{ij} \Big(z_j^k + \nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k)\Big),$
        $\phi_i^{k+1} = \sum_{j=1}^{m} a_{ij} \phi_j^k, \quad y_i^{k+1} = \frac{z_i^{k+1}}{\phi_i^{k+1}}.$   (6.5)
    $k \leftarrow k + 1$
```

unknown to agent i, wherein $m y_i^k$ tracks the sum gradient $\sum_{j=1}^m \nabla f_j(x_i^k)$ (see step (S.3)). We then employ a relaxation step (6.3b) with step size γ.

Assumption 6.3.1. $\widetilde{f}_i : \mathcal{K} \times \mathcal{K} \to \mathbb{R}$ *satisfies:*

(i) $\nabla \widetilde{f}_i(x; x) = \nabla f_i(x)$ *for all* $x \in \mathcal{K}$*;*

(ii) $\widetilde{f}_i(\cdot; y)$ *is uniformly strongly convex on* $\mathcal{K}$ *with constant* $\widetilde{\mu} > 0$*;*

(iii) $\nabla \widetilde{f}_i(x; \cdot)$ *is uniformly Lipschitz continuous on* $\mathcal{K}$ *with constant* $\widetilde{l}$*.*

The choice of $\widetilde{f}_i$ is quite flexible. For example, one can construct a proximal gradient type update (6.3a) by linearizing f_i and adding a proximal term; if f_i is a DC function, $\widetilde{f}_i$ can retain the convex part of f_i while linearizing the nonconvex part. We refer to [111]–[114] for more details on the choices of $\widetilde{f}_i$, and Sec. 6.5 for specific examples used in our experiments.

(S.2): Consensus. This steps aims at enforcing consensus on the local variables x_{i} via gossiping. Specifically, after the local optimization step, each agent i performs a consensus update (6.4) with mixing matrix $W = (w_{\mathrm{ij}})_{\mathrm{i,j}=1}^{m}$ satisfying the Assumption 5.2.3.

Note that SONATA uses a row-stochastic matrix W for the consensus update and a column-stochastic matrix A for the gradient tracking. In fact, for general digraph, a doubly stochastic matrix compliant with the graph might not exist while one can always build compliant row or column stochastic matrices. These weights can be determined locally by the agents, e.g., once its in- and out-degree can be estimated.

(S.3): Gradient tracking. This step updates y_{i} by employing a perturbed push-sum algorithm with weight matrix A satisfying Assumption 4.2.2. This step aims to track the average gradient $(1/m)\sum_{\mathrm{i}=1}^{m}\nabla f_{\mathrm{i}}(x_{\mathrm{i}})$ via y_{i}. In fact, using the column stochasticity of A and applying the telescopic cancellation, one can check that the following holds:

$$\sum_{\mathrm{i}=1}^{m}\phi_{\mathrm{i}}^{k} = \sum_{\mathrm{i}=1}^{m}\phi_{\mathrm{i}}^{0} = m, \quad \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{k} = \sum_{\mathrm{i}=1}^{m}\nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{k}). \tag{6.6}$$

It can be shown that for all i $\in [m]$, z_{i}^{k} and ϕ_{i}^{k} converges to $\xi_{\mathrm{i}}^{k}\cdot\sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^{k}$ and $\xi_{\mathrm{i}}^{k}\cdot\sum_{\mathrm{i}=1}^{m}\phi_{\mathrm{i}}^{k}$, respectively, for some $\xi_{\mathrm{i}}^{k} > 0$. Hence, $y_{\mathrm{i}}^{k} = z_{\mathrm{i}}^{k}/\phi_{\mathrm{i}}^{k}$ converges to $(1/m)\sum_{\mathrm{i}=1}^{m}\nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{k})$, employing the desired gradient tracking.

Notice that the extension of the gradient tracking to the asynchronous setting is not trivial, as the ratio consensus property discussed above no longer holds if agents naively perform their updates using in (6.5) delayed information. In fact, packets sent by an agent, corresponding to the summand in (6.5), may get lost. This breaks the equalities in (6.6). Consequently, the ratio y_{i}^{k} cannot correctly track the average gradient. To cope with this issue, our approach is to replace step (S.3) by the asynchronous gradient tracking mechanism developed in [117].

6.3.2 Asynchronous decentralized SCA (ASY-DSCA)

We now break the synchronism in SONATA and propose ASY-DSCA (cf. Algorithm 6). All agents update asynchronously and continuously without coordination, possibly using delayed information from their neighbors. More specifically, a global iteration counter k,

Algorithm 6 The ASY-DSCA Algorithm

```
Data: For all agent i and $\forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}$, $x_{\mathrm{i}}^{0} \in \mathbb{R}^{d}$, $z_{\mathrm{i}}^{0} = y_{\mathrm{i}}^{0} = \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{0})$, $\phi_{\mathrm{i}}^{0} = 1$, $\tilde{\rho}_{\mathrm{ij}}^{0} = 0$, $\tilde{\sigma}_{\mathrm{ij}}^{0} = 0$, $\tau_{\mathrm{ij}}^{-1} = -D$. And for $t = -D, -D+1, \ldots, 0$, $\rho_{\mathrm{ij}}^{t} = 0$, $\sigma_{\mathrm{ij}}^{t} = 0$, $v_{\mathrm{i}}^{t} = 0$. Set $k = 0$.
  While: a termination criterion is not met do
    Pick:   $(\mathrm{i}^k, d^k)$;
    Set:    $\tau_{\mathrm{i}^k \mathrm{j}}^{k} = \max(\tau_{\mathrm{i}^k \mathrm{j}}^{k-1}, k - d_{\mathrm{j}}^{k}), \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$;
    (S.1) Local optimization:
        $\tilde{x}_{\mathrm{i}^k}^{k} = \underset{x \in \mathcal{K}}{\operatorname{argmin}} \quad \widehat{U}_{\mathrm{i}^k}\left(x;\, x_{\mathrm{i}^k}^{k},\, m\, y_{\mathrm{i}^k}^{k} - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k})\right),$
        $v_{\mathrm{i}^k}^{k+1} = x_{\mathrm{i}^k}^{k} + \gamma\left(\tilde{x}_{\mathrm{i}^k}^{k} - x_{\mathrm{i}^k}^{k}\right);$      (6.7)
    (S.2) Consensus step (using delayed information):
        $x_{\mathrm{i}^k}^{k+1} = w_{\mathrm{i}^k \mathrm{i}^k} v_{\mathrm{i}^k}^{k+1} + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}} w_{\mathrm{i}^k \mathrm{j}} v_{\mathrm{j}}^{\tau_{\mathrm{i}^k \mathrm{j}}^{k}};$      (6.8)
    (S.3) Robust gradient tracking:
        $y_{\mathrm{i}^k}^{k+1} = \mathcal{F}\Big(\mathrm{i}^k,\, k,\, (\rho_{\mathrm{i}^k \mathrm{j}}^{\tau_{\mathrm{i}^k \mathrm{j}}^{k}})_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}},\, (\sigma_{\mathrm{i}^k \mathrm{j}}^{\tau_{\mathrm{i}^k \mathrm{j}}^{k}})_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}},\, \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k+1}) - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^{k})\Big)$      (6.9)
    Untouched state variables shift to state $k+1$ while keeping the same value; $k \leftarrow k+1$.

  procedure $\mathcal{F}(\mathrm{i}, k, (\rho_{\mathrm{ij}})_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}}, (\sigma_{\mathrm{ij}})_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}}, \epsilon)$
    Sum step:
        $z_{\mathrm{i}}^{k+\frac{1}{2}} = z_{\mathrm{i}}^{k} + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}} \left(\rho_{\mathrm{ij}} - \tilde{\rho}_{\mathrm{ij}}^{k}\right) + \epsilon,$
        $\phi_{\mathrm{i}}^{k+\frac{1}{2}} = \phi_{\mathrm{i}}^{k} + \sum_{\mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}}} \left(\sigma_{\mathrm{ij}} - \tilde{\sigma}_{\mathrm{ij}}^{k}\right);$      (6.10)
    Push step:
        $z_{\mathrm{i}}^{k+1} = a_{\mathrm{ii}}\, z_{\mathrm{i}}^{k+\frac{1}{2}}, \quad \phi_{\mathrm{i}}^{k+1} = a_{\mathrm{ii}}\, \phi_{\mathrm{i}}^{k+\frac{1}{2}}; \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{out}},$
        $\rho_{\mathrm{ji}}^{k+1} = \rho_{\mathrm{ji}}^{k} + a_{\mathrm{ji}}\, z_{\mathrm{i}}^{k+\frac{1}{2}}, \quad \sigma_{\mathrm{ji}}^{k+1} = \sigma_{\mathrm{ji}}^{k} + a_{\mathrm{ji}}\, \phi_{\mathrm{i}}^{k+\frac{1}{2}};$      (6.11)
    Mass-Buffer update:
        $\tilde{\rho}_{\mathrm{ij}}^{k+1} = \rho_{\mathrm{ij}}, \quad \tilde{\sigma}_{\mathrm{ij}}^{k+1} = \sigma_{\mathrm{ij}}, \quad \forall \mathrm{j} \in \mathcal{N}_{\mathrm{i}}^{\mathrm{in}};$      (6.12)
    return   $z_{\mathrm{i}}^{k+1} / \phi_{\mathrm{i}}^{k+1}$.
```

unknown to the agents, is introduced, which increases by 1 whenever a variable of the multiagent system changes. Let i^k be the agent triggering iteration $k \to k+1$; it executes Steps (S1)-(S.3) (no necessarily withih the same activation), as described below.

(S.1): Local optimization. Agent i^k solves the strongly convex optimization problem (6.7) based on the local surrogate $\widehat{U}_{\mathrm{i}^k}$. It is tacitly assumed that $\widehat{U}_{\mathrm{i}^k}$ is chosen so that (6.7) is simple to solve (i.e., the solution can be computed in closed form or efficiently). Given the solution $\widetilde{x}_{\mathrm{i}^k}^k$, $v_{\mathrm{i}^k}^{k+1}$ is generated.

(S.2): Consensus. Agent i^k may receive delayed variables from its in-neighbors $\mathrm{j} \in \mathcal{N}_{\mathrm{i}^k}^{\mathrm{in}}$, whose iteration index is $k - d_{\mathrm{j}}^k$. To perform its update, agent i^k first sorts the "age" of all the received variables from agent j since $k = 0$, and then picks the most recently generated one. This is implemented maintaining a local counter $\tau_{\mathrm{i}^k\mathrm{j}}$, updated recursively as $\tau_{\mathrm{i}^k\mathrm{j}}^k = \max(\tau_{\mathrm{i}^k\mathrm{j}}^{k-1}, k - d_{\mathrm{j}}^k)$. Thus, the variable agent i^k uses from j has iteration index $\tau_{\mathrm{i}^k\mathrm{j}}^k$. Since the consensus algorithm is robust against asynchrony [117], we simply adopt the update of SONATA [cf. (6.4)] and replace v_{j}^k by its delayed version $v_{\mathrm{j}}^{\tau_{\mathrm{i}^k\mathrm{j}}^k}$.

(S.3): Robust gradient tracking. As anticipated in Sec. 6.3.1, the packet loss caused by asynchrony breaks the sum preservation property (6.6) in SONATA. If treated in the same way as the x variable in (6.8), y_{i} would fail to track $(1/m)\sum_{\mathrm{i}=1}^m \nabla f_{\mathrm{i}}(x_{\mathrm{i}})$. To cope with this issue, we leverage the asynchronous sum-push scheme P-ASY-SUM-PUSH introduced in Chapter 4.

6.4 Convergence of ASY-DSCA

We study ASY-DSCA under the asynchronous assumption – Assumption 4.3.2. The convergence of ASY-DSCA is established under two settings, namely: i) convex F and error bound Assumption 6.2.2 (cf. Theorem 6.4.1); and ii) general nonconvex F (cf. Theorem 6.4.2).

Theorem 6.4.1 (Linear convergence)**.** *Consider (P) under Assumption 6.2.1 and 6.2.2, and let $U^\star$ denote the optimal function value. Let $\{(x_{\mathrm{i}}^k)_{\mathrm{i}=1}^m\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 6, under Assumption 4.2.1, 4.3.2, and with weight matrices W and A satisfying*

Assumption 5.2.3 and 4.2.2. Then, there exist a constant $\bar{\gamma}_{cvx} > 0$ and a solution $x^\star$ of (P) such that if $\gamma \leq \bar{\gamma}_{cvx}$, it holds

$$\|U(x_{\mathrm{i}}^k) - U(x^\star)\| = \mathcal{O}(\lambda^k), \quad \|x_{\mathrm{i}}^k - x^\star\| = \mathcal{O}\left((\sqrt{\lambda})^k\right),$$

for all $\mathrm{i} \in \mathcal{V}$ *and some* $\lambda \in (0,1)$. □

Theorem 6.4.1 establishes the first linear convergence result of a distributed (synchronous or asynchronous) algorithm over networks without requiring strong convexity but the weaker LT condition. Linear convergence is achieve on both function values and sequence iterates.

We consider now the nonconvex setting. To measure the progress of ASY-DSCA towards stationarity, we introduce the merit function

$$M_F(x^k) \triangleq \max\left\{\|\bar{x}^k - \mathrm{prox}_G(\bar{x}^k - \nabla F(\bar{x}^k))\|^2, \sum_{\mathrm{i}=1}^m \|x_{\mathrm{i}}^k - \bar{x}^k\|^2\right\}, \tag{6.13}$$

where $\bar{x}^k \triangleq (1/m) \cdot \sum_{\mathrm{i}=1}^m x_{\mathrm{i}}^k$, and prox_G is the prox operator (cf. Sec. 6.2.2). M_F is a valid merit function since it is continuous and $M_F(x^k) = 0$ if and only if all the x_{i}'s are consensual and stationary. The following theorem shows that $M_F(x^k)$ vanishes at sublinear rate.

Theorem 6.4.2 (Sublinear convergence)**.** *Consider (P) under Assumption 6.2.1 (thus possibly nonconvex). Let $\{(x_{\mathrm{i}}^k)_{\mathrm{i}=1}^m\}_{k \in \mathbb{N}_0}$ be the sequence generated by Algorithm 6, in the same setting of Theorem 6.4.1. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}$ such that $M_F(x^k) \leq \delta$. Then, there exists a $\bar{\gamma}_{ncvx} > 0$, such that if $\gamma \leq \bar{\gamma}_{ncvx}$, $T_\delta = \mathcal{O}(1/\delta)$.* □

The expression of the step-size can be found in (6.55).

6.5 Numerical Results

We test ASY-DSCA on a LASSO problem (a convex instance of (P)) and an M-estimation problem (a constrained nonconvex formulation) over both directed and undirected graphs. The experiments were performed using MATLAB R2018b on a cluster computer with two 22-cores Intel E5-2699Av4 processors (44 cores in total) and 512GB of RAM each. The setting of our simulations is the following.

(i) Network graph. We simulated both undirected and directed graph, generated according to the following procedures. `Undirected graph:` An undirected graph is generated according to the Erdos-Renyi model with parameter p = 0.3 (which represents the probability of having an edge between any two nodes). Doubly stochastic weight matrices are used, with weights generated according to the Metropolis-Hasting rule. `Directed graph:` We first generate a directed cycle graph to guarantee strong connectivity. Then we randomly add a fixed number of out-neighbors for each node. The row-stochastic weight matrix W and the column-stochastic weight matrix A are generated using uniform weights.

(ii) Surrogate functions of ASY-DSCA and SONATA. We consider two surrogate functions: $\widetilde{f}_{\mathrm{i}}^{1}(x; x_{\mathrm{i}}^{k}) = \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{k})^{\top}(x - x_{\mathrm{i}}^{k}) + \frac{\widetilde{\mu}}{2}\|x - x_{\mathrm{i}}^{k}\|^2$ and $\widetilde{f}_{\mathrm{i}}^{2}(x; x_{\mathrm{i}}^{k}) = \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^{k})^{\top}(x - x_{\mathrm{i}}^{k}) + \frac{1}{2}(x - x_{\mathrm{i}}^{k})^{\top} H (x - x_{\mathrm{i}}^{k}) + \frac{\widetilde{\mu}}{2}\|x - x_{\mathrm{i}}^{k}\|^2$, where H is a diagonal matrix having the same diagonal entries as $\nabla^2 f_{\mathrm{i}}(x_{\mathrm{i}}^{k})$. We suffix SONATA and ASY-DSCA with “-L” if the former surrogate functions are employed and with “-DH” if the latter are adopted.

(iii) Asynchronous model. Each agent sends its updated information to its out-neighbors and starts a new computation round, immediately after it finishes one. The length of each computation time is sampled from a uniform distribution over the interval $[p_{\min}, p_{\max}]$. The communication time/traveling time of each packet follows an exponential distribution $\exp(\frac{1}{D_{\mathrm{tv}}})$. Each agent uses the most recent information among the arrived packets from its in-neighbors, which in general is subject to delays. In all our simulations, we set $p_{\min} = 5$, $p_{\max} = 15$, and $D_{\mathrm{tv}} = 30$ (ms is the default time unit).

(iv) Comparison with state of arts schemes. We compare the convergence rate of ASY-DSCA, AsyPrimalDual [98] and synchronous SONATA in terms of time. The parameters are manually tuned to yield the best empirical performance for each–the used setting is reported in the caption of the associated figure. Note that AsyPrimalDual is the only asynchronous decentralized algorithm able to handle constraints and nonsmoothness additive functions in the objective and constraints, but only over undirected graphs and under restricted assumptions of asynchrony; also AsyPrimalDual is provably convergence only when applied to convex problems.

6.5.1 LASSO

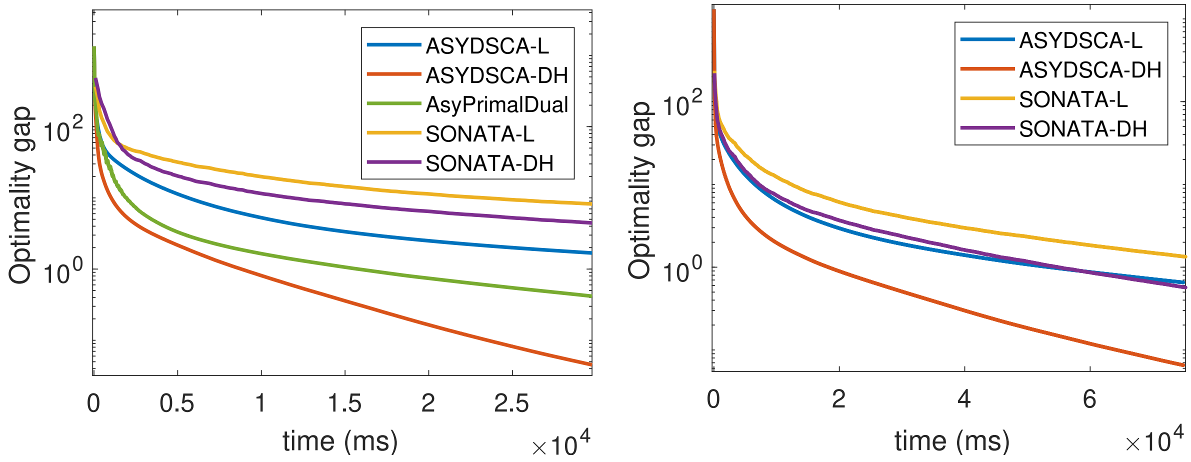


Figure 6.1. LASSO. **Left:** undirected graph. We set $\tilde{\mu} = 8$ and $\gamma = 0.008$ in ASY-DSCA-L; $\tilde{\mu} = 1$ and $\gamma = 0.008$ in ASY-DSCA-DH; $\alpha = 0.06$ and $\eta = 0.6$ in AsyPrimalDual; $\tilde{\mu} = 1$ and $\gamma = 0.002$ in SONATA-L; and $\tilde{\mu} = 1$ and $\gamma = 0.005$ in SONATA-DH. **right:** directed graph (each agent is of 10 out-neighbors). We set $\tilde{\mu} = 10$ and $\gamma = 0.01$ in ASY-DSCA-L; $\tilde{\mu} = 10$ and $\gamma = 0.03$ in ASY-DSCA-DH; $\tilde{\mu} = 10$ and $\gamma = 0.03$ in SONATA-L; and $\tilde{\mu} = 10$ and $\gamma = 0.05$ in SONATA-DH.

The decentralized LASSO problem reads

$$\min_{x \in \mathbb{R}^d} U(x) \triangleq \sum_{\mathrm{i} \in [m]} \|M_\mathrm{i} x - b_\mathrm{i}\|^2 + \lambda \left\| x \right\|_1. \tag{6.14}$$

Data $(M_\mathrm{i}, b_\mathrm{i})_{\mathrm{i} \in [m]}$ are generated as follows. We choose $x_0 \in \mathbb{R}^d$ as a ground truth sparse vector, with $density * d$ nonzero entries drawn i.i.d. from $\mathcal{N}(0,1)$. Each row of $M_\mathrm{i} \in \mathbb{R}^{r \times d}$ is drawn i.i.d. from $\mathcal{N}(0, \Sigma)$ with Σ as a diagonal matrix such that $\Sigma_{\mathrm{i},\mathrm{i}} = \mathrm{i}^{-\omega}$. We use ω to control the conditional number of Σ. Then we generate $b_\mathrm{i} = M_\mathrm{i} x_0 + \delta_\mathrm{i}$, with each entry of δ_i drawn i.i.d. from $\mathcal{N}(0, 0.01)$. We set $r = 10$, $d = 300$, $m = 20$, $\lambda = 2$, $\omega = 1.1$ and $density = 0.3$. Since the problem satisfies the LT condition, we use $\frac{1}{m} \sum_{\mathrm{i} \in [m]} U\left(x_\mathrm{i}^k\right) - U^\star$ as the optimality measure. The result are reported in Fig. 6.1.

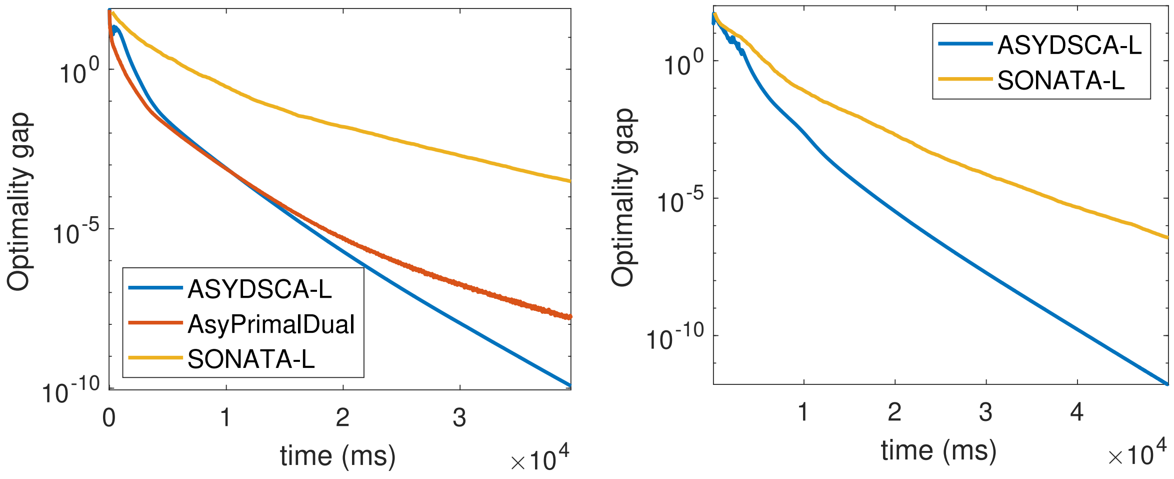


Figure 6.2. Logistic regression. **Left:** undirected graph. We set $\tilde{\mu} = 10$ and $\gamma = 0.06$ in ASY-DSCA-L; $\alpha = 0.1$ and $\eta = 0.7$ in AsyPrimalDual; and $\tilde{\mu} = 10$ and $\gamma = 0.08$ in SONATA-L. **right:** directed graph (each agent is of 10 out-neighbors). We set $\tilde{\mu} = 10$ and $\gamma = 0.05$ in ASY-DSCA-L; and $\tilde{\mu} = 10$ and $\gamma = 0.1$ in SONATA-L.

6.5.2 Sparse logistic regression

We consider the decentralized sparse logistic regression problem in the following form

$$\min_{x\in\mathbb{R}^d} \sum_{\mathrm{i}\in[m]} \sum_{s\in\mathcal{D}_\mathrm{i}} \log(1+\exp(-y_s\, u_s^\top x)) + \lambda\big\|x\big\|_1,$$

Data (u_s, y_s), $s \in \cup_{\mathrm{i}\in[m]}\mathcal{D}_\mathrm{i}$, are generated as follows. We first choose $x_0 \in \mathbb{R}^d$ as a ground truth sparse vector with $density * d$ nonzero entries drawn i.i.d. from $\mathcal{N}(0,1)$. We generate each sample feature u_s independently, with each entry drawn i.i.d. from $\mathcal{N}(0,1)$; then we set $y_s = 1$ with probability $1/(1+\exp(-u_s^\top x_0))$, and $y_s = -1$ otherwise. We set $|\mathcal{D}_\mathrm{i}| = 3, \forall \mathrm{i} \in [m]$, $d = 100$, $m = 20$, $\lambda = 0.01$ and $density = 0.3$. We use the same optimality measure as that for the LASSO problem. The results and the tuning of parameters are reported in Fig. 6.2.

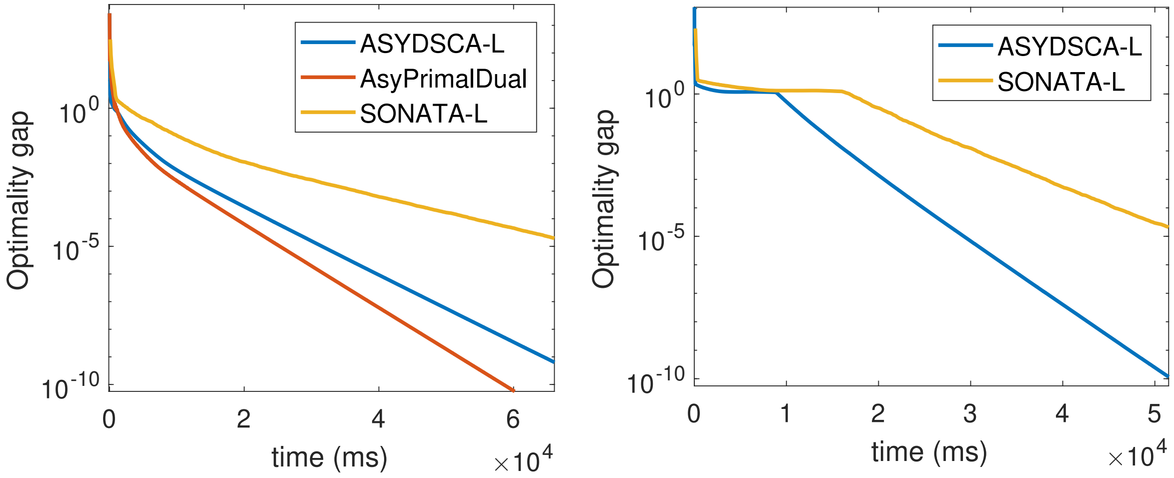


Figure 6.3. m-estimator. **Left:** undirected graph. We set $\tilde{\mu} = 300$ and $\gamma = 0.1$ in ASY-DSCA-L; $\alpha = 0.01$ and $\eta = 0.6$ in AsyPrimalDual; and $\tilde{\mu} = 100$ and $\gamma = 0.1$ in SONATA-L. **right:** directed graph (each agent is of 7 out-neighbors). We set $\tilde{\mu} = 1000$ and $\gamma = 0.08$ in ASY-DSCA-L; and $\tilde{\mu} = 1000$ and $\gamma = 0.2$ in SONATA-L.

6.5.3 M-estimator

As nonconvex (constrained, nonsmooth) instance of problem (P), we consider the following M-estimation task [130, (17)]:

$$\min_{\|x\|_2 \leq r} \frac{1}{|\mathcal{D}|} \sum_{\mathrm{i}\in[m]} \sum_{s\in\mathcal{D}_\mathrm{i}} \rho_\alpha(u_s^\top x - y_s) + \lambda \big\|x\big\|_1, \tag{6.15}$$

where $\rho_\alpha(t) = (1 - \mathrm{e}^{-\alpha t^2/2})/\alpha$ is the nonconvex Welsch's exponential squared loss and $\mathcal{D} \triangleq \cup_{\mathrm{i}\in[m]}\mathcal{D}_\mathrm{i}$. We generate $x_0 \in \mathbb{R}^d$ as unit norm sparse vector with *density* $* d$ nonzero entries drawn i.i.d. from $\mathcal{N}(0,1)$. Each entry of $u_s \in \mathbb{R}^d$ is drawn i.i.d. from $\mathcal{N}(0,1)$; we generate $y_s = {u_s}^\top x_0 + 0.1 * \epsilon_s$, with $\epsilon_s \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We set $|\mathcal{D}_\mathrm{i}| = 10$, for all $\mathrm{i} \in [m]$, $d = 100$, $m = 30$, $\alpha = 0.1$, $r = 2$, $\lambda = 0.01$, and *density* $= 0.1$. Since (6.15) is nonconvex, progresses towards stationarity and consensus are measured using the merit function $M_F(\cdot)$ in (6.13). The result and tuning of parameters are reported in Fig. 6.3.

6.5.4 Discussion

All the experiments clearly show that ASY-DSCA achieves linear rate on LASSO and Logistic regression, with nonstrongly convex objectives, both over undirected and directed graphs–this supports our theoretical findings (Theorem 6.4.1). The flexibility in choosing the surrogate functions provides us the chance to better exploit the curvature of the objective function than plain linearization-based choices. For example, in the LASSO experiment, ASY-DSCA-DH outperforms all the other schemes due to its advantage of better exploiting second order information. Also, ASY-DSCA compares favorably with AsyPrimalDual. ASY-DSCA exhibits good performance also in the nonconvex setting (recall that no convergence proof is available for AsyPrimalDual applied to nonconvex problems). In our experiments, asynchronous algorithms turned to be faster than synchronous ones. The reason is that, at each iteration, agents in synchronous algorithms must wait for the slowest agent receiving the information and finishing its computation (no delays are allowed), before proceeding to the next iteration. This is not the case of asynchronous algorithms wherein agents communicate and update continuously with no coordination.

6.6 Proof of Theorem 6.4.1

6.6.1 Roadmap of the proof

We begin introducing in this section the roadmap of the proof. Define $x^k \triangleq [x_1^k, \cdots, x_I^k]^\top$, $v^k \triangleq [v_1^k, \cdots, v_I^k]^\top \in \mathbb{R}^{m\times d}$; and let $S \triangleq (D+2)m$. Construct the two $S \times d$ matrices:

$$\delta^k \triangleq \mathrm{e}_{\mathrm{i}^k} \left(\Delta x^k\right)^\top, \quad \text{with} \quad \Delta x^k \triangleq \widetilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k,$$
$$H^k \triangleq [(x^k)^\top, (v^k)^\top, (v^{k-1})^\top, \cdots, (v^{k-D})^\top]^\top,$$

with $v^t = 0$, for $t \leq 0$. Our proof builds on the following quantities that monitor the progress of the algorithm.

- Optimality gaps:

$$E_z^k \triangleq \|\tilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k\|, \quad E_o^k \triangleq \max_{\mathrm{i}\in[S]} U(H_{\mathrm{i}}^k) - U^*; \tag{6.16a}$$

- Consensus errors (x^k_ψ is some weighted average of row vectors of H^k and will be defined in Sec. 6.6.2):

$$E^k_c \triangleq \left\| H^k - \mathbf{1} \cdot x^k_\psi \right\|, \quad E^k_t \triangleq \left\| y^k_{\mathrm{i}^k} - \bar{g}^k \right\|; \tag{6.16b}$$

- Tracking error:

$$E^k_t \triangleq \left\| I y^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k}) \right\|^2. \tag{6.16c}$$

Specifically, E^k_z and E^k_o measure the distance of the x^k_{i}'s from optimality in terms of step-length and objective value. E^k_c and E^k_t represents the consensus error of x_{i}'s and y_{i}'s, respectively while E^k_t is the tracking error of y^k_{i}. Our goal is to show that the above quantities vanish at a linear rate, implying convergence (at the same rate) of the iterates generated by the algorithm to a solution of Problem (P). Since each of them affects the dynamics of the others, our proof begins establishing the following set of inequalities linking these quantities (the explicit expression of the constants below will be given in the forthcoming sections):

$$E^{k+1}_t \leq 3C_1 l \sum_{l=0}^{k} \rho^{k-l} \left(E^l_c + \gamma E^l_z \right) + C_1 \rho^k \left\| g^0 \right\|, \tag{6.17a}$$

$$E^{k+1}_c \leq C_2 \rho^k E^0_c + C_2 \sum_{l=0}^{k} \rho^{k-l} \gamma E^l_z, \tag{6.17b}$$

$$E^k_t \leq 8m\, l^2 (E^k_c)^2 + 2m^2 (E^k_t)^2, \tag{6.17c}$$

$$E^{k+1}_o \leq C_4(\gamma)\, \zeta(\gamma)^k E^0_o + \frac{C_3(\gamma) C_4(\gamma)}{\zeta(\gamma)} \sum_{\ell=0}^{k} \zeta(\gamma)^{k-\ell} E^\ell_t, \tag{6.17d}$$

$$(E^k_z)^2 \leq \frac{1}{\gamma \left(\tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2} \right)} E^k_o + \frac{1}{2\,\epsilon \left(\tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2} \right)} E^k_t. \tag{6.17e}$$

We then show that E^k_z, E^k_o, E^k_c, E^k_t and E^k_t vanish at linear rate chaining the above inequalities by means of the generalized small gain theorem [117].

The main steps of the proof are summarized next.

• **Step 1: Proof of** (6.17a)**-**(6.17c) **via P-ASY-SUM-PUSH.** We rewrite (S.2) and (S.3) in ASY-DSCA (Algorithm 6) as instances of the perturbed asynchronous consensus scheme and the perturbed asynchronous sum-push scheme (the P-ASY-SUM-PUSH) introduced in Chapter 4. By doing so, we can bound the consensus errors E^k_c and E^k_t in terms of Δ^k and then prove (6.17a)-(6.17b)–see Lemma 6.6.2 and Lemma 6.6.3. Eq. (6.17c) follows readily from (6.17a)-(6.17b)–see Lemma 6.6.4.

• **Step 2: Proof of (6.17d)-(6.17e) under the LT condition.** Proving (6.17d)–contraction of the optimality measure E_o^k up to the tracking error–poses several challenges. To prove contraction of some form of optimization errors, existing techniques developed in the literature of *distributed* algorithms [4], [8], [10], [30], [100], including the convergence proof of ASY-SONATA, leverage strong convexity of F, a property that is replaced here by the weaker local growing condition (6.2) in the LT error bound. Hence, they are not applicable to our setting. On the other hand, existing proofs showing linear rate of *centralized* first-order methods under the LT condition [121] do not readily customize to our distributed, asynchronous setting, for the reasons elaborated next. To invoke the local growing condition (6.2), one needs first to show that the sequences generated by the algorithm enters (and stays into) the region where (6.1) holds, namely: a) the function value remains bounded; and b) the proximal operator residual is sufficiently small. A standard path to prove a) and b) in the centralized setting is showing that the objective function sufficiently descents along the trajectory of the algorithm. Asynchrony apart, in the distributed setting, function values on the agents' iterates do not monotonically decrease provably, due to consensus and gradient tracking errors. To cope with these issues, in this Step 2, we put forth a new analysis. Specifically, i) Sec. 6.6.3: we build a novel Lyapunov function [cf. (6.28)] that linearly combines objective values of current and past (up to D) iterates (all the elements of H^k); notice that the choice of the weights (cf. ψ^k in Lemma 6.6.1) is very peculiar and represents a major departure from existing approaches, including the result of ASY-SONATA–ψ^k *endogenously vary according to the asynchrony trajectory of the algorithm.* The Lyapunov function is proved to "sufficiently" descent over the asynchronous iterates of ASY-DSCA (cf. Proposition 6.6.1); ii) Sec. 6.6.3: building on such descent properties, we manage to prove that $x_{i^k}^k$ will eventually satisfy the aforementioned conditions (6.1) (cf. Lemma 6.6.6 & Corollary 6.6.5.1), so that the LT growing property (6.2) can be invoked at $x_{i^k}^k$ (cf. Corollary 6.6.6.1); iii) Sec. 6.6.3: Finally, leveraging this local growth, we uncover relations between E_o^k and E_t^k and prove (6.17d) (cf. Proposition 6.6.2). Eq. (6.17e) is proved in Sec. 6.6.3 by product of the derivations above.

• **Step 3: R-linear convergence via the generalized small gain theorem.** We complete the proof of linear convergence by applying [117, Th. 23] to the inequality system (6.17), and conclude that all the local variables $\{x_{\mathrm{i}}\}_{\mathrm{i}\in[m]}$ converge to the set of optimal solutions $\mathcal{K}^*$ R-linearly.

6.6.2 Step 1: Proof of (6.17a)-(6.17c)

We interpret the consensus step (S.2) in Algorithm 2 as an instance of the perturbed asynchronous consensus scheme [117]: (6.8) can be rewritten as

$$H^{k+1} = \widehat{W}^k (H^k + \gamma \delta^k), \tag{6.18}$$

where $\widehat{W}^k$ is a time-varying augmented matrix induced by the update order of the agents and the delay profile. The specific expression of $\widehat{W}^k$ can be found in [117] and is omitted here, as it is not relevant to the convergence proof. We only need to recall the following properties of $\widehat{W}^k$.

Lemma 6.6.1. *[117, Lemma 17] Let $\{\widehat{W}^k\}_{k\in\mathbb{N}_+}$ be the sequence of matrices in the dynamical system (6.18), generated under Assumption 4.3.2, and with W satisfying Assumption 5.2.3. Define $K_1 \triangleq (2I-1)\cdot T + m\cdot D$, $C_2 \triangleq \frac{2\sqrt{(D+2)m}(1+\bar{m}^{-K_1})}{1-\bar{m}^{-K_1}}$, $\eta \triangleq \bar{m}^{K_1}$ and $\rho \triangleq (1-\eta)^{\frac{1}{K_1}}$. Then we have for any $k \geq 0$:*

a. $\widehat{W}^k$ is row stochastic;

b. all the entries in the first m columns of $\widehat{W}^{k+K_1-1:k}$ are uniformly bounded below by η;

c. there exists a sequence of stochastic vectors $\{\psi^k\}_{k\geq 0}$ such that: i) for any $\ell \geq t \geq 0$, $\left\|\widehat{W}^{\ell:t} - 1{\psi^t}^\top\right\|_2 \leq C_2\rho^{\ell-t}$; ii) $\psi^k_{\mathrm{i}} \geq \eta$ for all $\mathrm{i} \in \mathcal{V}$.

Note that Lemma 6.6.1 implies

$$1{\psi^t}^\top = \lim_{d\to\infty} \widehat{W}^{d:t} = (\lim_{d\to\infty} \widehat{W}^{d:t+1})\widehat{W}^t = 1{\psi^{t+1}}^\top \widehat{W}^t, \tag{6.19}$$

and thus ${\psi^{t+1}}^\top \widehat{W}^t = {\psi^t}^\top$, for all $t \geq 0$. Then we define

$$x_\psi^k = {\psi^k}^\top H^k; \tag{6.20}$$

x_ψ^k evolves according to the following dynamics:

$$x_\psi^{k+1} = {\psi^0}^\top H^0 + \sum_{l=0}^{k} \gamma {\psi^l}^\top \Delta H^l. \tag{6.21}$$

This can be shown by applying (6.18) recursively, so that

$$H^{k+1} = \widehat{W}^{d:0} H^0 + \sum_{l=0}^{k} \widehat{W}^{d:l} \gamma \Delta H^l, \tag{6.22}$$

and multiplying (6.22) from the left by ${\psi^{k+1}}^\top$ and using (6.19). Taking the difference between (6.21) and (6.22) and applying Lemma 6.6.1 the consensus error E_c^k can be bound as follows.

Lemma 6.6.2. *Under the condition of Lemma 6.6.1, $\{E_c^k\}$ satisfies*

$$E_c^{k+1} \leq C_2\, \rho^k\, E_c^0 + C_2 \sum_{l=0}^{k} \rho^{k-l} \gamma\, E_z^l, \quad \forall k \geq 0. \tag{6.23}$$

To establish similar bounds for E_y^k, we build on the fact that the gradient tracking update (6.9) is an instance of the P-ASY-SUM-PUSH in [117], as shown next. Define

$$\begin{aligned} g^k &= [\nabla f_1(x_1^k), \nabla f_2(x_2^k), \cdots, \nabla f_I(x_I^k)]^\top, \\ \bar{g}^k &= (1/m) \cdot (g^k)^\top \mathbf{1}, \qquad E_t^k \triangleq \left\| y_{\mathrm{i}^k}^k - \bar{g}^k \right\|. \end{aligned}$$

We can prove the following bound for E_t^k.

Lemma 6.6.3. *Let $\{x^k, y_{\mathrm{i}^k}^k\}_{k=0}^\infty$ be the sequence generated by the Algorithm 6 under Assumption 4.2.1, 4.2.2, 5.2.3, and 4.3.2. Then, there exists a constant $C_1 = \frac{4\sqrt{2S}(1+\bar{m}^{-K_1})}{m\,\eta\,\rho(1-\bar{m}^{K_1})}$ such that*

$$E_t^{k+1} \leq 3\, C_1\, l \sum_{l=0}^{k} \rho^{k-l} \left(E_c^l + \gamma E_z^l \right) + C_1 \rho^k \left\| g^0 \right\|. \tag{6.24}$$

Proof. See Appendix 6.9.1. □

Finally, using Lemma 6.6.2 and Lemma 6.6.3, we can bound $\sum_{t=0}^{k}(E_c^t)^2$ and $\sum_{t=0}^{k}(E_t^t)^2$ in terms of $\sum_{t=0}^{k}\gamma^2(E_z^t)^2$, and E_t^k in terms of E_c^k and E_t^k, as given below.

Lemma 6.6.4. *Under the setting of Lemma 6.6.2 and Lemma 6.6.3, we have: for any $k \geq 1$,*

$$\begin{aligned}
&\sum_{t=0}^{k}(E_c^t)^2 \leq c_x + \varrho_x \sum_{t=0}^{k}\gamma^2(E_z^t)^2, \\
&\sum_{t=0}^{k}(E_t^t)^2 \leq c_y + \varrho_y \sum_{t=0}^{k}\gamma^2(E_z^t)^2, \\
&E_t^k \leq 2m^2(E_t^k)^2 + 8m\, l^2\, (E_c^k)^2.
\end{aligned} \tag{6.25}$$

with $\varrho_x \triangleq \frac{2C_2^2}{(1-\rho)^2}$, and $\varrho_y \triangleq \frac{36(C_1L)^2\left(2C_2^2+(1-\rho)^2\right)}{(1-\rho)^4}$. (The expressions of the constants c_x and c_y are omitted as they are not relevant).

Proof. The proof of the first two results follows similar steps as in that of [117, Lemma 26] and thus is omitted. We prove only the last inequality, as follows:

$$\begin{aligned}
&E_t^k = \left\|my_{\mathrm{i}^k}^k \pm m\bar{g}^k - \nabla F(x_{\mathrm{i}^k}^k)\right\|^2 \leq 2m^2(E_t^k)^2 + 2\left\|\sum_{\mathrm{j}=1}^{m} f_{\mathrm{j}}(x_{\mathrm{j}}^k) \pm F(x_{\psi}^k) - \nabla F(x_{\mathrm{i}^k}^k)\right\|^2 \\
&\leq 2m^2(E_t^k)^2 + 8m\, l^2\, (E_c^k)^2.
\end{aligned}$$

□

6.6.3 Step 2: Proof of (6.17d)-(6.17e) under the LT condition

- A new Lyapunov function and its descent We begin studying descent of the objective function U along the trajectory of the algorithm; we have the following result.

Lemma 6.6.5. *Let $\{(x^k, y^k)\}$ be the sequence generated by Algorithm 6 under Assumptions 6.2.1 and 6.3.1, it holds*

$$U(v_{\mathrm{i}^k}^{k+1}) \leq U(x_{\mathrm{i}^k}^k) - \gamma\left(\tilde{\mu} - \frac{\gamma L}{2}\right)\left\|\Delta x^k\right\|^2 + \gamma \cdot \left(\nabla F(x_{\mathrm{i}^k}^k) - my_{\mathrm{i}^k}^k\right)^\top \Delta x^k. \tag{6.26}$$

Proof. Applying the first order optimality condition to (6.7) and invoking the strong convexity of $\tilde{f}_{\mathrm{i}^k}$ (Assumption 6.3.1) we have

$$\begin{aligned}
&-(\Delta x^k)^\top I y^k_{\mathrm{i}^k} + G(x^k_{\mathrm{i}^k}) - G(\tilde{x}^k_{\mathrm{i}^k}) \geq -(\Delta x^k)^\top (\nabla f_{\mathrm{i}^k}(x^k_{\mathrm{i}^k}) - \nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}^k_{\mathrm{i}^k}; x^k_{\mathrm{i}^k})) \\
&= (\Delta x^k)^\top \Big(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}^k_{\mathrm{i}^k}; x^k_{\mathrm{i}^k}) - \nabla \widetilde{f}_{\mathrm{i}^k}(x^k_{\mathrm{i}^k}; x^k_{\mathrm{i}^k})\Big) \geq \tilde{\mu} \cdot \big\|\Delta x^k\big\|^2.
\end{aligned} \tag{6.27}$$

As F is L-smooth, applying the descent lemma gives

$$\begin{aligned}
&F(v^{k+1}_{\mathrm{i}^k}) \leq F(x^k_{\mathrm{i}^k}) + \gamma \cdot \nabla F(x^k_{\mathrm{i}^k})^\top \Delta x^k + \frac{L}{2}\gamma^2 \big\|\Delta x^k\big\|^2 \\
&= F(x^k_{\mathrm{i}^k}) + \gamma \cdot (m y^k_{\mathrm{i}^k})^\top \Delta x^k + \gamma \cdot \Big(\nabla F(x^k_{\mathrm{i}^k}) - m y^k_{\mathrm{i}^k}\Big)^\top \Delta x^k + \frac{L}{2}\gamma^2 \big\|\Delta x^k\big\|^2 \\
&\overset{(6.27)}{\leq} F(x^k_{\mathrm{i}^k}) + \gamma \Big(G(x^k_{\mathrm{i}^k}) - G(\tilde{x}^k_{\mathrm{i}^k}) - \tilde{\mu}\big\|\Delta x^k\big\|^2\Big) + \frac{L}{2}\gamma^2 \big\|\Delta x^k\big\|^2 + \gamma \cdot \Big(\nabla F(x^k_{\mathrm{i}^k}) - m y^k_{\mathrm{i}^k}\Big)^\top \Delta x^k.
\end{aligned}$$

By the convexity of G, we have

$$\gamma \Big(G(x^k_{\mathrm{i}^k}) - G(\tilde{x}^k_{\mathrm{i}^k})\Big) \leq G(x^k_{\mathrm{i}^k}) - G(v^{k+1}_{\mathrm{i}^k}).$$

Combining the above two results proves (6.26). □

We build now on (6.26) and establish descent on a suitable defined Lyapunov function. Define the mapping $\widetilde{U} : \mathbb{R}^{S\times d} \to \mathbb{R}^S$ as $\widetilde{U}(H) = [U(h_1), \cdots, U(h_S)]^\top$ for $H = [h_1, \cdots, h_S]^\top \in \mathbb{R}^{S\times d}$. That is, $\widetilde{U}(H)$ is a vector constructed by stacking the value of the objective function U evaluated at each local variable h_{i}. Recalling the definition of the weights ψ^k (cf. Lemma 6.6.1), we introduce the Lyapunov function

$$L^k \triangleq {\psi^k}^\top \widetilde{U}(H^k), \tag{6.28}$$

and study next its descent properties.

Proposition 6.6.1. *Let $\{(x^k, v^k, y^k)\}$ be the sequence generated by Algorithm 6 under Assumptions 6.2.1, 4.2.1, 6.3.1, 4.2.2, and 5.2.3. Then,*

$$L^{k+1} \leq L^0 - \sum_{t=0}^{k} (E^t_z)^2 \gamma \left(\eta\tilde{\mu} - \gamma\left(\frac{L}{2} + l\, m^{\frac{3}{2}}\sqrt{\varrho_x} + m\sqrt{\varrho_y}\right)\right) + C, \tag{6.29}$$

for all $k \geq 0$, where C is some constant independent of γ and k; and ϱ_x and ϱ_y are defined in Lemma 6.6.4.

Proof. By the row stochasticity of $\widehat{W}$ and the convexity of U:

$$\begin{aligned}\widetilde{U}(H^{k+1}) &= \widetilde{U}\left(\widehat{W}^k(H^k + \gamma\Delta H^k)\right) \preccurlyeq \widehat{W}^k\,\widetilde{U}\left(H^k + \gamma\Delta H^k\right)\\ &\preccurlyeq \widehat{W}^k\left(\widetilde{U}(H^k) - \left(\gamma\left(\tilde{\mu} - \frac{\gamma L}{2}\right)\left\|\Delta x^k\right\|^2 - \gamma\cdot\left(\nabla F(x^k_{\mathrm{i}^k}) - m y^k_{\mathrm{i}^k}\right)^\top \Delta x^k\right)\mathrm{e}_{\mathrm{i}^k}\right),\end{aligned}$$

where in the last inequality we applied Lemma 6.6.5. Using now Lemma 6.6.1, we have

$$\begin{aligned}&L^{k+1}\\ &\leq L^k - \psi^k_{\mathrm{i}^k}\left(\gamma\left(\tilde{\mu} - \frac{\gamma L}{2}\right)\left\|\Delta x^k\right\|^2 - \gamma\left(\nabla F(x^k_{\mathrm{i}^k}) - m y^k_{\mathrm{i}^k}\right)^\top \Delta x^k\right)\\ &\leq L^k - \gamma\eta\tilde{\mu}\left\|\Delta x^k\right\|^2 + \frac{L(\gamma)^2}{2}\left\|\Delta x^k\right\|^2 + \psi^k_{\mathrm{i}^k}\,\gamma\left(\nabla F(x^k_{\mathrm{i}^k}) - m y^k_{\mathrm{i}^k}\right)^\top \Delta x^k\\ &\leq L^k - \gamma\left(\eta\tilde{\mu} - \frac{\gamma L}{2}\right)\left\|\Delta x^k\right\|^2 + \psi^k_{\mathrm{i}^k}\gamma\left(\nabla F(x^k_{\mathrm{i}^k}) \pm m\bar{g}^k - m\,y^k_{\mathrm{i}^k}\right)^\top \Delta x^k\\ &\leq L^k - \gamma\left(\eta\tilde{\mu} - \frac{\gamma L}{2}\right)\left\|\Delta x^k\right\|^2 + \gamma\,m\,l\sum_{\mathrm{j}=1}^{m}\left\|x^k_\psi - x^k_{\mathrm{j}}\right\|\left\|\Delta x^k\right\| + \gamma\,m E^k_t\cdot\left\|\Delta x^k\right\|\\ &\leq L^k - \gamma\left(\eta\tilde{\mu} - \frac{\gamma L}{2}\right)\left\|\Delta x^k\right\|^2 + \gamma\,l\,m^{\frac{3}{2}}E^k_c\left\|\Delta x^k\right\| + \gamma\,m E^k_t\cdot\left\|\Delta x^k\right\|^2\\ &\overset{(*)}{\leq} L^k - \gamma\left(\eta\tilde{\mu} - \gamma\left(\frac{L}{2} + \frac{1}{2\epsilon_1} + \frac{1}{2\epsilon_2}\right)\right)\left\|\Delta x^k\right\|^2 + \frac{\epsilon_1}{2}\,l^2\,m^3(E^k_c)^2 + \frac{\epsilon_2}{2}\,m^2(E^k_t)^2\\ &\leq L^0 - \gamma\left(\eta\tilde{\mu} - \gamma\left(\frac{L}{2} + \frac{1}{2\epsilon_1} + \frac{1}{2\epsilon_2}\right)\right)\sum_{t=0}^{k}\left\|\Delta x^t\right\|^2 + \frac{\epsilon_1}{2}\,l^2\,m^3\sum_{t=0}^{k}(E^t_c)^2 + \frac{\epsilon_2}{2}\,m^2\sum_{t=0}^{k}(E^t_t)^2,\end{aligned}$$

where $(*)$ follows from the Young's inequality with $\epsilon_{1,2} > 0$. Invoking Lemma 6.6.4 and setting $\gamma^l \equiv \gamma$ gives (6.29), where the free parameters $\epsilon_{1,2}$ are chosen as $\epsilon_1 = 1/(l\,m^{\frac{3}{2}}\sqrt{\varrho_x})$ and $\epsilon_2 = 1/(m\sqrt{\varrho_y})$, respectively. □

- Leveraging the LT condition We build now on Proposition 6.6.1 and show next that the two conditions in (6.1) holds at x^k_{i}, for sufficiently large k; this will permit to invoke the LT growing property (6.2).

The first condition–$U(x^k_{\mathrm{i}})$ bounded for large k–is a direct consequence of Proposition 6.6.1 and the facts that U is bounded from below (Assumption 6.2.1) and $\psi^k_{\mathrm{i}} \geq \eta$, for all $\mathrm{i} \in [m]$ and $k \geq 0$. Formally, we have the following.

Corollary 6.6.5.1. *Under the setting of Proposition 6.6.1 and step-size* $0 < \gamma < \bar{\gamma} \triangleq \frac{2\eta\tilde{\mu}}{L+2l\,m^{\frac{3}{2}}\sqrt{\varrho_x}+2I\sqrt{\varrho_y}}$*, it holds:*

a. $U(x_{\mathrm{i}}^k)$ *is uniformly upper bounded, for all* $\mathrm{i} \in \mathcal{V}$ *and* $k \geq 0$*;*

b. $\sum_{t=0}^{\infty}(E_z^t)^2 < \infty$, $\sum_{t=0}^{\infty}(E_c^t)^2 < \infty$, *and* $\sum_{t=0}^{\infty}(E_t^t)^2 < \infty$.

We prove now that the second condition in (6.1) holds for large k–the residual of the proximal operator at $x_{\mathrm{i}^k}^k$, that is $\left\|x_{\mathrm{i}^k}^k - \mathrm{prox}_G(x_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k))\right\|$, is sufficiently small. Since E_z^k and the gradient tracking error E_t^k are vanishing [as a consequence of Corollary 6.6.5.1ii) and Lemma 6.6.4], it is sufficient to bound the aforementioned residual by E_z^k and E_t^k. This is done in the lemma below.

Lemma 6.6.6. *The proximal operator residual on* $x_{\mathrm{i}^k}^k$ *satisfies* $\left\|x_{\mathrm{i}^k}^k - \mathit{prox}_G(x_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k))\right\|^2 \leq 4\Big(1 + (l + \tilde{l})^2\Big)(E_z^k)^2 + 5E_t^k$.

Proof. For simplicity, we denote $\hat{x}^k = \mathrm{prox}_G(x_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k))$. According to the variational characterization of the proximal operator, we have, for all $w \in \mathcal{K}$,

$$\left(\hat{x}^k - \left(x_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k)\right)\right)^\top (\hat{x}^k - w) + G(\hat{x}^k) - G(w) \leq 0.$$

The first order optimality condition of $\tilde{x}_{\mathrm{i}^k}^k$ implies

$$\left(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) + I y_{\mathrm{i}^k}^k - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)\right)^\top (\tilde{x}_{\mathrm{i}^k}^k - z) + G(\tilde{x}_{\mathrm{i}^k}^k) - G(z) \leq 0, \quad \forall z \in \mathcal{K}. \tag{6.30}$$

Setting $z = \hat{x}^k$ and $w = \tilde{x}_{\mathrm{i}^k}^k$ and adding the above two inequalities yields

$$\begin{aligned}
0 &\geq \left(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) + I y_{\mathrm{i}^k}^k - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k) - \hat{x}^k + x_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k)\right)^\top (\tilde{x}_{\mathrm{i}^k}^k - \hat{x}^k) \\
&= \left(I y_{\mathrm{i}^k}^k - \hat{x}^k + x_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k)\right)^\top (\tilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k) + \left(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)\right)^\top (\tilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k) \\
&\quad + \left\|\hat{x}^k - x_{\mathrm{i}^k}^k\right\|^2 + \left(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) + I y_{\mathrm{i}^k}^k - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k) - \nabla F(x_{\mathrm{i}^k}^k)\right)^\top (x_{\mathrm{i}^k}^k - \hat{x}^k) \\
&\geq -\frac{1}{2}\left\|I y_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k)\right\|^2 - \frac{1}{2}\left\|\Delta x^k\right\|^2 - \frac{1}{4}\left\|\hat{x}^k - x_{\mathrm{i}^k}^k\right\|^2 \\
&\quad - \left\|\Delta x^k\right\|^2 + \tilde{\mu}\left\|\Delta x^k\right\|^2 + \left\|\hat{x}^k - x_{\mathrm{i}^k}^k\right\|^2 - \frac{1}{4}\left\|\hat{x}^k - x_{\mathrm{i}^k}^k\right\|^2 \\
&\quad - 2\left((l + \tilde{l})^2\left\|\Delta x^k\right\|^2 + \left\|I y_{\mathrm{i}^k}^k - \nabla F(x_{\mathrm{i}^k}^k)\right\|^2\right).
\end{aligned}$$

Rearranging terms proves the desired result. □

Corollary 6.6.5.1 in conjunction with Lemma 6.6.6 and Lemma 6.6.4 show that both conditions in (6.1) hold at $\{x^k\}$, for large k. We can then invoke the growing condition (6.2).

Corollary 6.6.6.1. *Let $\{x^k\}$ be the sequence generated by Algorithm 6 under the setting of Corollary 6.6.5.1. Then, there exists a constant $\kappa > 0$ and a sufficiently large $\bar{k}$ such that, for $k \geq \bar{k}$,*

$$\mathit{dist}(x^k_{\mathrm{i}^k}, \mathcal{K}^*) \leq \kappa \left\| x^k_{\mathrm{i}^k} - \mathit{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k})) \right\|. \tag{6.31}$$

Proof. It is sufficient to show that (6.1) holds at $x^k_{\mathrm{i}^k}$. By Corollary 6.6.5.1(i), $U(x^k_{\mathrm{i}^k}) \leq B$, for all $k \geq 0$ and some $B < +\infty$. Lemma 6.6.6 in conjunction with Corollary 6.6.5.1(ii) and Lemma 6.6.4 yields $\lim\limits_{k\to\infty} \left\| x^k_{\mathrm{i}^k} - \mathrm{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k})) \right\| = 0$. □

- Proof of (6.17d) Define

$$C_3(\gamma) \triangleq \frac{\gamma \left(c_6(\tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2}) + \frac{c_7}{2\epsilon} \right)}{c_7 + \tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2}}, \tag{6.32}$$

$$C_4(\gamma) \triangleq \left(1 - \Big(1 - \sigma(\gamma) \Big) \eta \right)^{-1}, \tag{6.33}$$

$$\zeta(\gamma) \triangleq \left(1 - \Big(1 - \sigma(\gamma) \Big) \eta \right)^{\frac{1}{K_1}}, \tag{6.34}$$

$$\sigma(\gamma) \triangleq \frac{c_7 + \left(\tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2} \right) (1 - \gamma)}{c_7 + \tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2}}, \tag{6.35}$$

where $K_1 = (2I - 1) \cdot T + m \cdot D$, and c_6, c_7 are polynomials in $(1, l, \tilde{l}, L, \kappa)$ whose expressions are given in (6.60) and (6.42); and $\epsilon \in (0, 2\tilde{\mu})$ is a free parameter (to be chosen).

In this section, we prove (6.17d), which is formally stated in the proposition below.

Proposition 6.6.2. *Let $\{(x^k, y^k)\}$ be the sequence generated by Algorithm 6 under Assumptions 6.2.1, 4.2.1, 6.2.2, 6.3.1, 5.2.3, and 4.2.2. Then, for $k \geq \bar{k}$, it holds*

$$E_o^{k+1} \leq C_4(\gamma)\, \zeta(\gamma)^k E_o^0 + \frac{C_3(\gamma) C_4(\gamma)}{\zeta(\gamma)} \sum_{\ell=0}^{k} \zeta(\gamma)^{k-\ell} E_t^\ell. \tag{6.36}$$

Since $\sigma(\gamma) < 1$ for $0 < \gamma < \sup_{\epsilon \in (0, 2\tilde{\mu})} \frac{2\tilde{\mu} - \epsilon}{L} = \frac{2\tilde{\mu}}{L}$ and $\eta \in (0, 1]$, Proposition 6.6.2 shows that, for sufficiently small $\gamma > 0$, the optimality gap E_o^k converges to zero R-linearly if E_t^k

does so. The proof of Proposition 6.6.2 follows from Proposition 6.6.3 and Lemma 6.6.8 below.

Proposition 6.6.3. *Let $\{(x^k, y^k)\}$ be the sequence generated by Algorithm 6 in the setting of Proposition 6.6.2. Let $p^k \triangleq \widetilde{U}(H^k) - U(x^*)\mathbf{1}$; let Σ^k be the diagonal matrix with all diagonal entries 1 and $\Sigma^k_{\mathrm{i}^k \mathrm{i}^k} = \sigma(\gamma)$; and let $(\widehat{W}\Sigma)^{k:\ell} \triangleq \widehat{W}^k \Sigma^k \cdots \widehat{W}^\ell \Sigma^\ell$. Then, for $k \geq \bar{k}$,*

$$p^{k+1} \preccurlyeq \left(\widehat{W}\Sigma\right)^{k:0} p^0 + C_3(\gamma) \sum_{\ell=1}^{k} \left(\widehat{W}\Sigma\right)^{k:\ell} \widehat{W}^{\ell-1} \mathrm{e}_{\mathrm{i}^{\ell-1}} E_t^{\ell-1} + C_3(\gamma) \widehat{W}^k \mathrm{e}_{\mathrm{i}^k} E_t^k, \tag{6.37}$$

where $C_3(\gamma)$ is defined in (6.32).

Proof. By convexity of U and (6.18), we have

$$p^{k+1} = \widetilde{U}(H^{k+1}) - U(x^*)\mathbf{1} \preccurlyeq \widehat{W}^k \left(\widetilde{U}\left(H^k + \gamma \Delta H^k\right) - U(x^*)\mathbf{1}\right). \tag{6.38}$$

Since $\widetilde{U}\left(H^k + \gamma \Delta H^k\right)$ differs from $\widetilde{U}\left(H^k\right)$ only by its i^k-th row, we study descent occurred at this row, which is $(v^{k+1}_{\mathrm{i}^k})^\top = (x^k_{\mathrm{i}^k} + \gamma\left(\widetilde{x}^k_{\mathrm{i}^k} - x^k_{\mathrm{i}^k}\right))^\top$. Recall that by applying the descent lemma on F and using the convexity of G we proved

$$U(v^{k+1}_{\mathrm{i}^k}) - U(x^k_{\mathrm{i}^k}) \leq \frac{L}{2}\gamma^2 \left\|\Delta x^k\right\|^2 + \gamma \underbrace{\left(\nabla F(x^k_{\mathrm{i}^k})^\top \left(\widetilde{x}^k_{\mathrm{i}^k} - x^k_{\mathrm{i}^k}\right) + G(\widetilde{x}^k_{\mathrm{i}^k}) - G(x^k_{\mathrm{i}^k})\right)}_{T_1}. \tag{6.39}$$

The above inequality establishes a connections between $U(v^{k+1}_{\mathrm{i}^k})$ and $U(x^k_{\mathrm{i}^k})$. However, it is not clear whether there is any contraction (up to some error) going from the optimality gap $U(v^{k+1}_{\mathrm{i}^k}) - U^*$ to $U(x^k_{\mathrm{i}^k}) - U^*$. To investigate it, we derive in the lemma below two upper bounds of T_1 in (6.39), in terms of $U(v^{k+1}_{\mathrm{i}^k}) - U(x^*)$ and $\left\|\Delta x^k\right\|$ (up to the tracking error). Building on these bounds and (6.39) we can finally prove the desired contraction, as stated in (6.43).

Lemma 6.6.7. *T_1 in (6.39) can be bounded in the following two alternative ways: for $k \geq \bar{k}$,*

$$T_1 \leq \left(-\tilde{\mu} + \frac{\epsilon}{2}\right) \cdot \left\|\Delta x^k\right\|^2 + \frac{1}{2\epsilon} E_t^k, \tag{6.40}$$

$$T_1 \leq -\frac{1}{1-\gamma}\left(U(v^{k+1}_{\mathrm{i}^k}) - U(x^*)\right) + \frac{1}{1-\gamma}\left(c_5 \left\|\Delta x^k\right\|^2 + c_6 E_t^k\right), \tag{6.41}$$

where c_5 and c_6 are polynomials in $(1, l, \tilde{l}, L, \kappa)$ whose expressions are given in (6.60)*.*

Proof. See Appendix 6.9.2. □

Using Lemma 6.6.7 in (6.39) yields

$$\begin{aligned} U(v_{\mathrm{i}^k}^{k+1}) - U^* &\leq (1-\gamma)\left(U(x_{\mathrm{i}^k}^k) - U^*)\right) + \left(\frac{L}{2}\gamma(1-\gamma) + c_5\right)\gamma\left\|\Delta x^k\right\|^2 + c_6 \cdot \gamma E_t^k \\ &\leq (1-\gamma)\left(U(x_{\mathrm{i}^k}^k) - U(x^*(x_{\mathrm{i}^k}^k))\right) + \underbrace{(c_5 + L/8)}_{c_7}\gamma\left\|\Delta x^k\right\|^2 + c_6 \cdot \gamma E_t^k, \end{aligned} \tag{6.42a}$$

and

$$U(v_{\mathrm{i}^k}^{k+1}) - U^* \leq U(x_{\mathrm{i}^k}^k) - U^* - \left(\tilde{\mu} - \frac{\gamma L}{2} - \frac{\epsilon}{2}\right)\gamma\left\|\Delta x^k\right\|^2 + \frac{\gamma}{2\epsilon}E_t^k. \tag{6.42b}$$

Canceling out $\left\|\Delta x^k\right\|^2$ in (6.42a)-(6.42b) yields: for $k \geq \bar{k}$,

$$U(v_{\mathrm{i}^k}^{k+1}) - U(x^*) \leq \sigma(\gamma)\left(U(x_{\mathrm{i}^k}^k) - U(x^*)\right) + C_3(\gamma)E_t^k, \tag{6.43}$$

where $\sigma(\gamma)$ and $C_3(\gamma)$ are defined in (6.32). Thus we observed a contraction from $\left(U(x_{\mathrm{i}^k}^k) - U(x^*)\right)$ to $U(v_{\mathrm{i}^k}^{k+1}) - U(x^*)$. Continuing from (6.38), we have

$$\begin{aligned} p^{k+1} &\overset{(6.43)}{\preccurlyeq} \widehat{W}^k\left(\Sigma^k p^k + C_3(\gamma)\, E_t^k\, \mathrm{e}_{\mathrm{i}^k}\right) \\ &\preccurlyeq \left(\widehat{W}\Sigma\right)^{k:0} p^0 + C_3(\gamma)\sum_{\ell=1}^{k}\left(\widehat{W}\Sigma\right)^{k:\ell}\widehat{W}^{\ell-1}\mathrm{e}_{\mathrm{i}^{\ell-1}}E_t^{\ell-1} + C_3(\gamma)\widehat{W}^k\mathrm{e}_{\mathrm{i}^k}E_t^k. \end{aligned}$$

□

The lemma below shows that the operator norm of $(\widehat{W}\Sigma)^{k:\ell}$ induced by the ℓ_∞ norm decays at a linear rate.

Lemma 6.6.8. *For any $k \geq \ell \geq 0$,*

$$\left\|(\widehat{W}\Sigma)^{k:\ell}\right\|_\infty \leq C_4(\gamma)\,\zeta(\gamma)^{k-\ell},$$

where the expression of $\zeta(\gamma)$, $C_4(\gamma)$, and K_1 are given in (6.32)*.*

Proof. See Appendix 6.9.3. □

- Proof of (6.17e) Eq. (6.17e) follows directly from the second inequality of (6.42) and the fact that $U(x_{i^k}^k) - U(v_{i^k}^{k+1}) \leq E_o^k$. This completes the proof of the inequality system (6.17).

6.6.4 Step 3: R-linear convergence via the generalized small gain theorem

The last step is to show that all the error quantities in (6.17) vanish at a linear rate. To do so, we leverage the generalized small gain theorem [117, Th. 17]. We use the following.

Definition 6.6.1 ([100])**.** *Given the sequence* $\{u^k\}_{k=0}^{\infty}$*, a constant* $\lambda \in (0,1)$*, and* $N \in \mathbb{N}$*, let us define*

$$|u|^{\lambda,N} = \max_{k=0,\ldots,N} \frac{\left|u^k\right|}{\lambda^k}, \quad |u|^{\lambda} = \sup_{k\in\mathbb{N}_0} \frac{\left|u^k\right|}{\lambda^k}.$$

If $|u|^{\lambda}$ *is upper bounded, then* $u^k = \mathcal{O}(\lambda^k)$*, for all* $k \in \mathbb{N}_0$*.*

Invoking [117, Lemma 20 & Lemma 21], if we choose λ such that $\max\left(\rho^2, \zeta(\gamma)\right) < \lambda < 1$, by (6.17) we get

$$|E_t|^{\sqrt{\lambda},N} \leq \frac{3C_1 l}{\sqrt{\lambda}-\rho}(|E_c|^{\sqrt{\lambda},N} + \gamma\left|E_z^k\right|^{\sqrt{\lambda},N}) + E_t^0 + \frac{C_1\left\|g^0\right\|}{\sqrt{\lambda}} \tag{6.44}$$

$$|E_c|^{\sqrt{\lambda},N} \leq \frac{C_2\gamma}{\sqrt{\lambda}-\rho}\left|E_z^k\right|^{\sqrt{\lambda},N} + E_c^0 + \frac{C_2 E_c^0}{\sqrt{\lambda}} \tag{6.45}$$

$$|E_o|^{\lambda,N} \leq \frac{C_3(\gamma)C_4(\gamma)}{\zeta(\gamma)\left(\lambda-\zeta(\gamma)\right)}|E_t|^{\lambda,N} + E_o^0 + \frac{C_4(\gamma)E_o^0}{\lambda} \tag{6.46}$$

$$|E_t|^{\lambda,N} \leq 8\,m\,l^2\left|(E_c)^2\right|^{\lambda,N} + 2\,m^2\left|(E_t)^2\right|^{\lambda,N} \tag{6.47}$$

$$\left|(E_z^k)^2\right|^{\lambda,N} \leq \frac{1}{2\,\epsilon\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)}|E_t|^{\lambda,N} + \frac{1}{\gamma\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)}|E_o|^{\lambda,N} \tag{6.48}$$

Taking the square on both sides of (6.44) & (6.45) while using $\left(|u|^{q,N}\right)^2 = |(u)^2|^{q^2,N}$, and writing the result in matrix form we obtain:

$$\begin{bmatrix} |(E_t)^2|^{\lambda,N} \\ |(E_c)^2|^{\lambda,N} \\ |E_o|^{\lambda,N} \\ |E_t|^{\lambda,N} \\ \left|(E_z^k)^2\right|^{\lambda,N} \end{bmatrix} \preccurlyeq \underbrace{\begin{bmatrix} 0 & \frac{36C_1^2 l^2}{(\sqrt{\lambda}-\rho)^2} & 0 & 0 & \frac{36C_1^2 l^2\gamma^2}{(\sqrt{\lambda}-\rho)^2} \\ 0 & 0 & 0 & 0 & \frac{3C_2^2\gamma^2}{(\sqrt{\lambda}-\rho)^2} \\ 0 & 0 & 0 & \frac{C_3(\gamma)C_4(\gamma)}{\zeta(\gamma)(\lambda-\zeta(\gamma))} & 0 \\ 2I^2 & 8I\,l^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\gamma\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)} & \frac{1}{2\,\epsilon\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)} & 0 \end{bmatrix}}_{\triangleq G} \begin{bmatrix} |(E_t)^2|^{\lambda,N} \\ |(E_c)^2|^{\lambda,N} \\ |E_o|^{\lambda,N} \\ |E_t|^{\lambda,N} \\ \left|(E_z^k)^2\right|^{\lambda,N} \end{bmatrix} + \epsilon^N. \tag{6.49}$$

We are now ready to apply [117, Th. 17]: a sufficient condition for E_t, E_c, E_o, E_t, and E_z^2 to vanish at an R-linear rate is $\rho(G) < 1$. By [117, Lemma 23], this is equivalent to requiring $p_G(1) > 0$, where $p_G(z)$ is the characteristic polynomial of G, This leads to the following condition:

$$\begin{aligned} &\mathcal{B}(\lambda;\gamma) \\ &= \left(\frac{72\,m^2\,C_1^2\,l^2\,\gamma^2}{(\sqrt{\lambda}-\rho)^2} + \frac{24\,m\,l^2\,C_2^2\gamma^2}{(\sqrt{\lambda}-\rho)^2} + \frac{216\,m^2\,C_1^2\,C_2^2\,l^2\,\gamma^2}{(\sqrt{\lambda}-\rho)^4}\right) \\ &\cdot\left(\frac{1}{2\epsilon\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)} + \frac{C_3(\gamma)C_4(\gamma)}{\zeta(\gamma)\,(\lambda-\zeta(\gamma))}\frac{1}{\gamma\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)}\right) < 1. \end{aligned}$$

It is not hard to see that $\mathcal{B}(\lambda;\gamma)$ is continuous at $\lambda = 1$, for any $\gamma \in (0, \frac{2\tilde{\mu}-\epsilon}{L})$. Therefore, as long as

$$\begin{aligned} \mathcal{B}(1;\gamma) = &\left(\frac{72\,m^2\,C_1^2\,l^2}{(1-\rho)^2} + \frac{24\,m\,l^2\,C_2^2}{(1-\rho)^2} + \frac{216\,m^2\,C_1^2\,C_2^2\,l^2}{(1-\rho)^4}\right)\gamma\cdot \\ &\left(\frac{\gamma}{2\epsilon\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)} + \frac{C_3(\gamma)C_4(\gamma)}{\zeta(\gamma)\,(1-\zeta(\gamma))}\frac{1}{\left(\tilde{\mu}-\frac{\epsilon}{2}-\frac{\gamma L}{2}\right)}\right) < 1, \end{aligned} \tag{6.50}$$

there will exist some $\lambda \in (0,1)$ such that $\mathcal{B}(\lambda;\gamma) < 1$.

We show now that $\mathcal{B}(1;\gamma) < 1$, for sufficiently small γ. We only need to prove boundedness of the following quantity when $\gamma \downarrow 0$:

$$\frac{C_3(\gamma)C_4(\gamma)}{\zeta(\gamma)\,(1-\zeta(\gamma))} = \underbrace{\frac{c_6(\tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2}) + \frac{c_7}{2\epsilon}}{\left(c_7 + \tilde{\mu} - \frac{\epsilon}{2} - \frac{\gamma L}{2}\right)\zeta(\gamma)^{K_1+1}}}_{\triangleq h(\gamma)} \cdot \frac{\gamma}{1-\zeta(\gamma)}.$$

It is clear that $h(\gamma)$ is right-continuous at 0 and thus $\lim_{\gamma\downarrow 0} h(\gamma) < \infty$. Hence, it is left to check that $\frac{\gamma}{1-\zeta(\gamma)}$ is bounded when $\gamma \downarrow 0$. According to L'Hôpital's rule,

$$\lim_{\gamma\downarrow 0} \frac{\gamma}{1-\zeta(\gamma)} = -\frac{K_1}{\left(1-\left(1-\sigma(\gamma)\right)\eta\right)^{\frac{1}{K_1}-1}} \frac{1}{\eta\sigma(\gamma)}\bigg|_{\gamma=0} = \frac{K_1\left(c_7 + \tilde{\mu} - \frac{\epsilon}{2}\right)}{\eta\left(\tilde{\mu} - \frac{\epsilon}{2}\right)} < \infty.$$

Finally, we prove that all $(x_{\mathrm{i}}^k)_{k\geq\bar{k}}$ converge linearly to some $x^\star$. By the definition of the augmented matrix H and the update (6.18), we have: for $k \geq \bar{k}$,

$$\begin{aligned} &\|h^{k+1} - h^k\| = \|(\widehat{W} - m)h^k + \gamma\delta^k\| \\ &\leq \|(\widehat{W} - m)(H^k - \mathbf{1}\cdot(x_\psi^k)^\top)\| + \gamma\|\delta^k\| \leq 3E_c^k + \gamma E_z^k. \end{aligned}$$

Since both E_c^k and E_z^k are $\mathcal{O}\left((\sqrt{\lambda})^k\right)$, $\sum_{k=0}^\infty \|h^{k+1} - h^k\| < +\infty$; thus $\{H^k\}_{k\in\mathbb{N}}$ is Cauchy and converges to some $\mathbf{1}(x^\star)^\top$, implying all x_{i}^k converges to $x^\star$. We prove next that x_{i}^k converges to $x^\star$ R-linearly. For any $k > k \geq \bar{k}$, we have $\|H^k - H^k\| \leq \sum_{t=k}^{k-1}\|H^t - H^{t+1}\| \leq \sum_{t=k}^{k-1}(3E_c^t + \gamma E_z^t) = \mathcal{O}\left((\sqrt{\lambda})^k\right)$. Taking $k \to \infty$ completes the proof.

6.7 Proof of Theorem 6.4.2

In this section we prove the sublinear convergence of ASY-DSCA. We organize the proof in two steps. Step 1: we prove $\sum_{k=0}^\infty (E_z^k)^2 < +\infty$ by showing the descent of a properly constructed Lyapunov function. This function represents a major novelty of our analysis– see Remark 6.7.1. Step 2: we connect the decay rate of E_z^k and that of the merit function $M_F(x^k)$.

6.7.1 Step 1: E_z^k is square summable

In Sec. 6.6.2 we have shown that the weighted average of the local variables x_ψ evolves according to the dynamics Eq. (6.21). Using $x_\psi^0 = {\psi^0}^\top H^0$, (6.21) can be rewritten recursively as

$$x_\psi^{k+1} = x_\psi^k + \gamma {\psi^k}^\top \Delta H^k = x_\psi^k + \gamma \psi_{\mathrm{i}^k}^k \Delta x^k. \tag{6.51}$$

Invoking the descent lemma while recalling $E_z^k = \left\| \Delta x^k \right\|$, yields

$$\begin{aligned}
F(x_\psi^{k+1}) &\leq F(x_\psi^k) + \gamma \psi_{\mathrm{i}^k}^k \nabla F(x_\psi^k)^\top \Delta x^k + \frac{L(\gamma \psi_{\mathrm{i}^k}^k)^2}{2} (E_z^k)^2 \\
&\overset{(6.27)}{\leq} F(x_\psi^k) + \frac{L\gamma^2}{2}(E_z^k)^2 - \gamma \psi_{\mathrm{i}^k}^k \left(\tilde{\mu}(E_z^k)^2 + G(\tilde{x}_{\mathrm{i}^k}^k) - G(x_{\mathrm{i}^k}^k) \right) \\
&\quad + \gamma \psi_{\mathrm{i}^k}^k \left(\nabla F(x_\psi^k) - m\bar{g}^k \right)^\top \Delta x^k + \gamma \psi_{\mathrm{i}^k}^k \left(m\bar{g}^k - I y_{\mathrm{i}^k}^k \right)^\top \Delta x^k \\
&\leq F(x_\psi^k) + \frac{L\gamma^2}{2}(E_z^k)^2 - \gamma \psi_{\mathrm{i}^k}^k \left(\tilde{\mu}(E_z^k)^2 + G(\tilde{x}_{\mathrm{i}^k}^k) - G(x_{\mathrm{i}^k}^k) \right) + \gamma l \sqrt{m} E_c^k \, E_z^k + \gamma m E_t^k \, E_z^k.
\end{aligned} \tag{6.52}$$

Introduce the Lyapunov function

$$L^k \triangleq F(x_\psi^k) + {\psi^k}^\top \widetilde{G}(H^k) \tag{6.53}$$

where $\widetilde{G} : \mathbb{R}^{S \times d} \to \mathbb{R}^S$ is defined as $\widetilde{G}(H) \triangleq [G(h_1), \cdots, G(h_S)]^\top$, for $H = [h_1, \cdots, h_S]^\top \in \mathbb{R}^{S \times d}$.

Remark 6.7.1. *Note that L^k contrasts with the functions used in the literature of distributed algorithms to study convergence in the nonconvex setting. Existing choices either cannot deal with asynchrony [9], [10] (e.g. the unbalance in the update frequency of the agents and the use of outdated information) or cannot handle nonsmooth functions in the objective and constraints [117]. A key feature of L^k is to combine current and past information throughout suitable dynamics, $\{x_\psi^k\}$, and weights averaging via $\{\psi^k\}$.*

Using the dynamics of H^k as in (6.18), we get

$$\widetilde{G}(H^{k+1}) \preccurlyeq \widehat{W}^k \left((1-\gamma)\widetilde{G}(H^k) + \gamma \widetilde{G}(H^k + \delta^k) \right).$$

where we used the convexity of G and the row-stochasticity of $\widehat{W}^k$. Thus

$$\begin{aligned}\psi^{k+1^\top}\widetilde{G}(H^{k+1}) &\leq \psi^{k+1^\top}\widehat{W}^k\left((1-\gamma)\widetilde{G}(H^k)+\gamma\widetilde{G}(H^k+\delta^k)\right)\\ &= \psi^{k^\top}\left((1-\gamma)\widetilde{G}(H^k)+\gamma\widetilde{G}(H^k+\delta^k)\right),\end{aligned}$$

where in the last equality we used $\psi^{t+1^\top}\widehat{W}^t = \psi^{t^\top}$ [cf. (6.19)]. Therefore,

$$\gamma\psi^k_{\mathrm{i}^k}\left(G(x^k_{\mathrm{i}^k})-G(\tilde{x}^k_{\mathrm{i}^k})\right) = \gamma\left(\psi^{k^\top}\widetilde{G}(H^k)-\psi^{k^\top}\widetilde{G}(H^k+\delta^k)\right) \leq \psi^{k^\top}\widetilde{G}(H^k)-\psi^{k+1^\top}\widetilde{G}(H^{k+1}).$$

Combining the above inequality with (6.52), we get

$$\begin{aligned}&L^{k+1} \leq L^k - \eta\tilde{\mu}(E^k_z)^2\gamma + \frac{L}{2}(E^k_z)^2\gamma^2 + \frac{\epsilon_1}{2}l^2m(E^k_c)^2 + \frac{1}{2\epsilon_1}\gamma^2(E^k_z)^2 + \frac{\epsilon_2}{2}m^2(E^k_t)^2 + \frac{1}{2\epsilon_2}\gamma^2(E^k_z)^2\\ &= L^k - (E^k_z)^2\gamma\left(\eta\tilde{\mu}-\gamma\left(\frac{L}{2}+\frac{1}{2\epsilon_1}+\frac{1}{2\epsilon_2}\right)\right) + \frac{\epsilon_1}{2}l^2m(E^k_c)^2 + \frac{\epsilon_2}{2}m^2(E^k_t)^2\\ &\leq L^0 - \sum_{t=0}^{k}(E^t_z)^2\gamma\left(\eta\tilde{\mu}-\gamma\left(\frac{L}{2}+\frac{1}{2\epsilon_1}+\frac{1}{2\epsilon_2}\right)\right) + \frac{\epsilon_1}{2}l^2m\sum_{t=0}^{k}{E^t_c}^2 + \frac{\epsilon_2}{2}m^2\sum_{t=0}^{k}{E^t_t}^2.\end{aligned} \tag{6.54}$$

To bound the last two terms in (6.54), we apply Proposition 6.6.4:

$$\begin{aligned}L^{k+1} &\leq L^0 - \sum_{t=0}^{k}(E^t_z)^2\gamma\left(\eta\tilde{\mu}-\gamma\left(\frac{L}{2}+\frac{1}{2\epsilon_1}+\frac{1}{2\epsilon_2}+\frac{\epsilon_1}{2}l^2m\varrho_x+\frac{\epsilon_2}{2}m^2\varrho_y\right)\right) + \frac{\epsilon_1}{2}l^2mc_x + \frac{\epsilon_2}{2}m^2c_y\\ &= L^0 - \sum_{t=0}^{k}(E^t_z)^2\gamma\left(\eta\tilde{\mu}-\gamma\left(\frac{L}{2}+\sqrt{l^2m\varrho_x}+\sqrt{m^2\varrho_y}\right)\right) + \frac{l^2mc_x}{2\sqrt{l^2m\varrho_x}} + \frac{m^2c_y}{2\sqrt{m^2\varrho_y}},\end{aligned}$$

where in the last equality we set $\epsilon_1 = 1/\sqrt{l^2m\varrho_x}$ and $\epsilon_2 = 1/\sqrt{m^2\varrho_y}$. Note that

$$L^k = F(x^k_\psi) + \psi^{k^\top}\widetilde{G}(H^k) \geq F(x^k_\psi) + G(\psi^{k^\top}H^k) = U(x^k_\psi) \geq U^*,$$

for all $k \in \mathbb{N}_+$. Thus, for sufficiently small γ, such that

$$\gamma \leq \bar{\gamma}_{ncvx} \triangleq \eta\tilde{\mu}\left(L+2\sqrt{l^2m\varrho_x}+2\sqrt{m^2\varrho_y}\right)^{-1}, \tag{6.55}$$

we can obtain the following bound

$$\sum_{t=0}^{\infty}(E_z^t)^2 \leq \frac{2L^0 - 2U^* + \frac{l^2 m c_x}{\sqrt{l^2 m \varrho_x}} + \frac{m^2 c_y}{\sqrt{m^2 \varrho_y}}}{\gamma\eta\tilde{\mu}}. \tag{6.56}$$

6.7.2 Step 2: $M_F(x^k)$ vanishes at sublinear rate

In this section we establish the connection between $M_F(x^k)$ and E_z^k, E_c^k, and E_t^k. Invoking Lemma 6.6.6 we can bound $\|\bar{x}^k - \mathrm{prox}_G(\bar{x}^k - \nabla F(\bar{x}^k))\|$ as

$$\begin{aligned}
&\|\bar{x}^k - \mathrm{prox}_G(\bar{x}^k - \nabla F(\bar{x}^k))\|^2 \\
\leq\, & 3\|\bar{x}^k - x^k_{\mathrm{i}^k}\|^2 + 3\|x^k_{\mathrm{i}^k} - \mathrm{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k}))\|^2 \\
&\quad + 3\|\mathrm{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k})) - \mathrm{prox}_G(\bar{x}^k - \nabla F(\bar{x}^k))\|^2 \\
\overset{(*)}{\leq} & 3\|\bar{x}^k - x^k_{\mathrm{i}^k}\|^2 + 3\|x^k_{\mathrm{i}^k} - \mathrm{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k}))\|^2 + \|x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k}) - (\bar{x}^k - \nabla F(\bar{x}^k))\|^2 \\
\leq & (5 + 2L^2)\|\bar{x}^k - x^k_{\mathrm{i}^k}\|^2 + 3\|x^k_{\mathrm{i}^k} - \mathrm{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k}))\|^2 \\
\leq & 4(5 + 2L^2)(E_c^k)^2 + 3\|x^k_{\mathrm{i}^k} - \mathrm{prox}_G(x^k_{\mathrm{i}^k} - \nabla F(x^k_{\mathrm{i}^k}))\|^2 \\
\leq & 4(5 + 2L^2)(E_c^k)^2 + 3\left(4\left(1 + (l + \tilde{l})^2\right)(E_z^k)^2 + 5E_t^k\right),
\end{aligned}$$

where (*) follows from the nonexpansiveness of a proximal operator. Further applying Lemma 6.6.4 and (6.17c), yields:

$$\begin{aligned}
&\sum_{t=0}^{k} M_F(x^t) \\
\leq & \sum_{t=0}^{k} \|\bar{x}^k - \mathrm{prox}_G(\bar{x}^k - \nabla F(\bar{x}^k))\|^2 + \sum_{t=0}^{k}(E_c^t)^2 \\
\leq & \sum_{t=0}^{k} \left((21 + 8L^2)(E_c^t)^2 + 3\left(4(1 + (l + \tilde{l})^2)(E_z^t)^2 + 5E_t^t\right)\right) \\
\leq & \sum_{t=0}^{k} \left((21 + 8L^2)(E_c^t)^2 + 15\left(8Il^2(E_c^t)^2 + 2m^2(E_t^t)^2\right)\right) + 12(1 + (l + \tilde{l})^2)\sum_{t=0}^{k}(E_z^t)^2
\end{aligned}$$

$$
\begin{aligned}
&\leq \Big(21 + 8L^2 + 120ml^2\Big) \left(c_x + \varrho_x \sum_{t=0}^{k} \gamma^2 (E_z^t)^2\right) + 30m^2 \left(c_y + \varrho_y \sum_{t=0}^{k} \gamma^2 (E_z^t)^2\right) \\
&\qquad + 12(1 + (l + \tilde{l})^2) \sum_{t=0}^{k} (E_z^t)^2 \\
&= \Big(\Big(21 + 8L^2 + 120ml^2\Big) \varrho_x \gamma^2 + 30\kappa^2 m^2 \varrho_y \gamma^2 + 12(1 + (l + \tilde{l})^2) \Big) \\
&\quad \sum_{t=0}^{k} (E_z^t)^2 + \Big(21 + 8L^2 + 120ml^2\Big) c_x + 30\kappa^2 m^2 c_y \\
&\overset{(6.56)}{\leq} \Big(\Big(21 + 8L^2 + 120ml^2\Big) \varrho_x \gamma^2 + 30\kappa^2 m^2 \varrho_y \gamma^2 + 12(1 + (l + \tilde{l})^2) \Big) \left(\frac{2L^0 - 2U^* + \frac{l^2 m c_x}{\sqrt{l^2 m \varrho_x}} + \frac{m^2 c_y}{\sqrt{m^2 \varrho_y}}}{\gamma \eta \tilde{\mu}} \right) \\
&\quad + \Big(21 + 8L^2 + 120ml^2\Big) c_x + 30\kappa^2 m^2 c_y \triangleq B_{opt},
\end{aligned}
$$

where ϱ_x and ϱ_y are defined in Lemma 6.6.4.

Let $T_\delta = \inf\{k \in \mathbb{N} \mid M_F(x^k) \leq \delta\}$. Then it holds: $T_\delta \cdot \delta < \sum_{k=0}^{T_\delta - 1} M_F(x^k) \leq B_{opt}$ and thus $T_\delta = \mathcal{O}(B_{opt}/\delta)$.

6.8 Conclusion

We proposed ASY-DSCA, an asynchronous decentralized method for multiagent convex/nonconvex composite minimization problems over (di)graphs. The algorithm employs SCA techniques and is robust against agents' uncoordinated activations and use of outdated information (subject to arbitrary but bounded delays). For convex (not strongly convex) objectives satisfying the LT error bound condition, ASY-DSCA achieves R-linear convergence rate while sublinear convergence is established for nonconvex objectives.

6.9 Appendix: Proofs of Theorems

6.9.1 Proof of Lemma 6.6.3

Applying [117, Th. 6] with the identifications: $\epsilon^t = \nabla f_{\mathrm{i}^t}(x_{\mathrm{i}^t}^{t+1}) - \nabla f_{\mathrm{i}^t}(x_{\mathrm{i}^t}^{t})$ and

$$\mathfrak{m}_z^k = \sum_{\mathrm{i}=1}^{m} z_{\mathrm{i}}^0 + \sum_{t=0}^{k-1} \epsilon^t = \sum_{\mathrm{i}=1}^{m} \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^0) + \sum_{t=0}^{k-1} \Big(\nabla f_{\mathrm{i}^t}(x_{\mathrm{i}^t}^{t+1}) - \nabla f_{\mathrm{i}^t}(x_{\mathrm{i}^t}^{t})\Big) \overset{(*)}{=} m \cdot \underbrace{\frac{1}{m} \sum_{\mathrm{i}=1}^{m} \nabla f_{\mathrm{i}}(x_{\mathrm{i}}^k)}_{\triangleq \bar{g}^k},$$

we arrive at $E_t^{k+1} \leq C_1 \left(\rho^k \left\|g^0\right\| + \sum_{l=0}^{k} \rho^{k-l} \|\epsilon^l\|\right)$, where in $(*)$ we have used $x_{\mathrm{j}}^{t+1} = x_{\mathrm{j}}^t$ for $\mathrm{j} \neq \mathrm{i}^t$. The rest of the proof follows the same argument as in [117, Prop. 18]. □

6.9.2 Proof of Lemma 6.6.7

Using (6.27), we have: for any $\epsilon > 0$,

$$\begin{aligned} T_1 &= \Big(\nabla F(x_{\mathrm{i}^k}^k) \pm I y_{\mathrm{i}^k}^k\Big)^\top \Big(\tilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k\Big) + G(\tilde{x}_{\mathrm{i}^k}^k) - G(x_{\mathrm{i}^k}^k) \\ &\leq - \tilde{\mu} \cdot \left\|\Delta x^k\right\|^2 + \frac{1}{2\epsilon}\, E_t^k + \frac{\epsilon}{2} \left\|\Delta x^k\right\|^2. \end{aligned}$$

Next we prove (6.41). For any $z \in \mathcal{K}$, let $x^*(z) \in \mathcal{P}_{\mathcal{K}^*}(z)$. By the Mean Value Theorem, there exists $\xi^k = \beta\, x^*(x_{\mathrm{i}^k}^k) + (1-\beta) v_{\mathrm{i}^k}^{k+1}$, with $\beta \in (0,1)$, such that

$$U(v_{\mathrm{i}^k}^{k+1}) - U(x^*(x_{\mathrm{i}^k}^k)) = \nabla F(\xi^k)^\top \Big(v_{\mathrm{i}^k}^{k+1} - x^*(x_{\mathrm{i}^k}^k)) + G(v_{\mathrm{i}^k}^{k+1}\Big) - G(x^*(x_{\mathrm{i}^k}^k)). \tag{6.57}$$

To deal with the inner product term, we invoke the algorithmic update (6.7) and the first order optimality condtion (6.30) (with $z = x^*(x_{\mathrm{i}^k}^k)$):

$$\begin{aligned} &\Big(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) + I y_{\mathrm{i}^k}^k - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)\Big)^\top \Big(v_{\mathrm{i}^k}^{k+1} - x^*(x_{\mathrm{i}^k}^k)\Big) \\ =& \Big(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) + I y_{\mathrm{i}^k}^k - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)\Big)^\top \Big(\tilde{x}_{\mathrm{i}^k}^k - x^*(x_{\mathrm{i}^k}^k) + (\gamma - 1)(\tilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k)\Big) \\ \leq& - (1-\gamma) \Big(\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}_{\mathrm{i}^k}^k; x_{\mathrm{i}^k}^k) + I y_{\mathrm{i}^k}^k - \nabla f_{\mathrm{i}^k}(x_{\mathrm{i}^k}^k)\Big)^\top (\tilde{x}_{\mathrm{i}^k}^k - x_{\mathrm{i}^k}^k) + G(x^*(x_{\mathrm{i}^k}^k)) - G(\tilde{x}_{\mathrm{i}^k}^k). \end{aligned}$$

Therefore

$$
\begin{aligned}
&U(v_{i^k}^{k+1}) - U(x^*(x_{i^k}^k)) \\
&= \Big(\nabla F(\xi^k) \pm \Big(\nabla \widetilde{f}_{i^k}(\tilde{x}_{i^k}^k; x_{i^k}^k) + I y_{i^k}^k - \nabla f_{i^k}(x_{i^k}^k)\Big)\Big)^\top \Big(v_{i^k}^{k+1} - x^*(x_{i^k}^k)) + G(v_{i^k}^{k+1}\Big) - G(x^*(x_{i^k}^k)) \\
&\leq \Big(\nabla \widetilde{f}_{i^k}(\tilde{x}_{i^k}^k; x_{i^k}^k) + I y_{i^k}^k - \nabla f_{i^k}(x_{i^k}^k)\Big)\,(v_{i^k}^{k+1} - x^*(x_{i^k}^k)) + G(v_{i^k}^{k+1}) - G(x^*(x_{i^k}^k)) \\
&\quad + \underbrace{\Bigg(\Big\|\nabla F(\xi^k) - \nabla F(x_{i^k}^k)\Big\| + \Big\|\nabla F(x_{i^k}^k) - I y_{i^k}^k\Big\| + \Big\|\nabla \widetilde{f}_{i^k}(\tilde{x}_{i^k}^k; x_{i^k}^k) - \nabla f_{i^k}(x_{i^k}^k)\Big\|\Bigg)\Big\|v_{i^k}^{k+1} - x^*(x_{i^k}^k)\Big\|}_{R_1} \\
&\leq -\,(1-\gamma)\Big(\nabla \widetilde{f}_{i^k}(\tilde{x}_{i^k}^k; x_{i^k}^k) + I y_{i^k}^k - \nabla f_{i^k}(x_{i^k}^k)\Big)^\top (\tilde{x}_{i^k}^k - x_{i^k}^k) - (1-\gamma)(G(\tilde{x}_{i^k}^k) - G(x_{i^k}^k)) + R_1,
\end{aligned}
\tag{6.58}
$$

where in the last inequality we have used the convexity of G. We thus arrive at the following bound on T_1:

$$
\begin{aligned}
T_1 &= \Big(\nabla F(x_{i^k}^k) \pm \Big(\nabla \widetilde{f}_{i^k}(\tilde{x}_{i^k}^k; x_{i^k}^k) + I y_{i^k}^k - \nabla f_{i^k}(x_{i^k}^k)\Big)\Big)^\top \Big(\tilde{x}_{i^k}^k - x_{i^k}^k\Big) + G(\tilde{x}_{i^k}^k) - G(x_{i^k}^k) \\
&\leq -\frac{1}{1-\gamma}\Big(U(v_{i^k}^{k+1}) - U(x^*(x_{i^k}^k))\Big) + \frac{1}{1-\gamma}\cdot R_1 \\
&\quad + \underbrace{\Bigg(\Big\|\nabla \widetilde{f}_{i^k}(\tilde{x}_{i^k}^k; x_{i^k}^k) - \nabla f_{i^k}(x_{i^k}^k)\Big\| + \Big\|\nabla F(x_{i^k}^k) - I y_{i^k}^k\Big\|\Bigg)\Big\|\Delta x^k\Big\|}_{R_2}.
\end{aligned}
\tag{6.59}
$$

It remains to bound the remainder terms R_1 and R_2. Note that

$$
\begin{aligned}
&\Big\|v_{i^k}^{k+1} - x^*(x_{i^k}^k)\Big\| = \Big\|v_{i^k}^{k+1} \pm x_{i^k}^k - x^*(x_{i^k}^k)\Big\| \leq \operatorname{dist}(x_{i^k}^k, \mathcal{K}^*) + \gamma\,\Big\|\Delta x^k\Big\|, \\
&\Big\|\xi^k - x_{i^k}^k\Big\| \leq \beta\Big\|x_{i^k}^k - x^*(x_{i^k}^k)\Big\| + (1-\beta)\Big\|v_{i^k}^{k+1} - x_{i^k}^k\Big\| \leq \operatorname{dist}(x_{i^k}^k, \mathcal{K}^*) + \gamma\,\Big\|\Delta x^k\Big\|.
\end{aligned}
$$

Applying Lemma 6.6.6 and Corollary 6.6.6.1, the following holds: for $k \geq \bar{k}$,

$$
\Big(\operatorname{dist}(x_{i^k}^k, \mathcal{K}^*)\Big)^2 \leq \kappa^2\Big\|x_{i^k}^k - \operatorname{prox}_G(x_{i^k}^k - \nabla F(x_{i^k}^k))\Big\|^2 \leq \kappa^2\left(4\left(1 + (l + \tilde{l})^2\right)\Big\|\Delta x^k\Big\|^2 + 5E_t^k\right).
$$

With the above inequalities and using the fact that $\gamma \leq 1$ we can bound R_1 as

$$
\begin{aligned}
R_1 \leq & \left\|\nabla F(\xi^k) - \nabla F(x^k_{\mathrm{i}^k})\right\|^2 + \left\|\nabla F(x^k_{\mathrm{i}^k}) - I y^k_{\mathrm{i}^k}\right\|^2 + \left\|\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}^k_{\mathrm{i}^k}; x^k_{\mathrm{i}^k}) - \nabla f_{\mathrm{i}^k}(x^k_{\mathrm{i}^k})\right\|^2 + \left\|v^{k+1}_{\mathrm{i}^k} - x^*(x^k_{\mathrm{i}^k})\right\|^2 \\
\leq & L^2 \left\|\xi^k - x^k_{\mathrm{i}^k}\right\|^2 + E^k_t + (l + \widetilde{l})^2 \left\|\Delta x^k\right\|^2 + 2\,\mathrm{dist}(x^k_{\mathrm{i}^k}, \mathcal{K}^*)^2 + 2\gamma^2 \left\|\Delta x^k\right\|^2 \\
\leq & \left(2L^2 + 2\right) \mathrm{dist}(x^k_{\mathrm{i}^k}, \mathcal{K}^*)^2 + E^k_t + \left(2L^2\gamma^2 + 2\gamma^2 + (l + \widetilde{l})^2\right) \left\|\Delta x^k\right\|^2 \\
\leq & \underbrace{\left(8\kappa^2(L^2+1)\left(1 + (l + \tilde{l})^2\right) + 2L^2 + 2 + (l + \widetilde{l})^2\right)}_{c_3} \left\|\Delta x^k\right\|^2 + \underbrace{\left(10\kappa^2(L^2+1) + 1\right)}_{c_4} E^k_t.
\end{aligned}
$$

Similarly, R_2 can be bounded as

$$
R_2 \leq \left\|\nabla F(x^k_{\mathrm{i}^k}) - I y^k_{\mathrm{i}^k}\right\|^2 + \left\|\nabla \widetilde{f}_{\mathrm{i}^k}(\tilde{x}^k_{\mathrm{i}^k}; x^k_{\mathrm{i}^k}) - \nabla f_{\mathrm{i}^k}(x^k_{\mathrm{i}^k})\right\|^2 + \left\|\Delta x^k\right\|^2 \leq E^k_t + \left(1 + (l + \widetilde{l})^2\right) \left\|\Delta x^k\right\|^2.
$$

Substituting the bounds of R_1 and R_2 in (6.59) yields

$$
\begin{aligned}
& T_1 \leq -\frac{1}{1-\gamma}\left(U(v^{k+1}_{\mathrm{i}^k}) - U(x^*(x^k_{\mathrm{i}^k}))\right) + E^k_t + \frac{1}{1-\gamma}\cdot\left(c_3\left\|\Delta x^k\right\|^2 + c_4 E^k_t\right) + \left(1 + (l + \widetilde{l})^2\right)\left\|\Delta x^k\right\|^2 \\
& \leq -\frac{1}{1-\gamma}\left(U(v^{k+1}_{\mathrm{i}^k}) - U(x^*(x^k_{\mathrm{i}^k}))\right) + \frac{1}{1-\gamma}\left(c_5\left\|\Delta x^k\right\|^2 + c_6 E^k_t\right),
\end{aligned}
$$

where

$$
\begin{aligned}
c_5 &\triangleq 8\kappa^2(L^2+1)\left(1 + (l + \tilde{l})^2\right) + 2L^2 + 2 + (l + \widetilde{l})^2 + 1 + (l + \widetilde{l})^2, \\
c_6 &\triangleq 10\kappa^2(L^2+1) + 2.
\end{aligned}
\tag{6.60}
$$

□

6.9.3 Proof of Lemma 6.6.8

We know from Lemma 6.6.1 (ii): for all $k \geq 0$, all elements in the first m columns of $\widehat{W}^{k+K_1-1:k}$ are no less than η. Since $\widehat{W}^{k+K_1-1:k}\Sigma^k$ is nonnegative, we have for each $\mathrm{i} \in [S]$

$$
\left\|\widehat{W}^{k+K_1-1:k}\Sigma^k\right\|_\infty \leq \max_{\mathrm{i}=1,\ldots,S}\left\{1 - \left(1 - \sigma(\gamma)\right)\widehat{W}^{k+K_1-1:k}_{\mathrm{i},\mathrm{i}^k}\right\} \leq 1 - \left(1 - \sigma(\gamma)\right)\eta.
$$

On the other hand, because $0 \preccurlyeq \Sigma^k \preccurlyeq m$ for all k, we know $\left(\widehat{W}\Sigma\right)^{m:k} \preccurlyeq \widehat{W}^{m:k}\,\Sigma^k,\ \forall m \geq k$. Thus

$$
\left\|(\widehat{W}\Sigma)^{k+K_1-1:k}\right\|_\infty \leq \left\|\widehat{W}^{k+K_1-1:k}\Sigma^k\right\|_\infty \leq 1 - (1 - \sigma(\gamma))\,\eta.
$$

Finally for any $k \geq \ell \geq 0$,

$$\begin{aligned}
&\left\|(\widehat{W}\Sigma)^{k:\ell}\right\|_{\infty} \leq \left(\prod_{t=1}^{\lfloor\frac{k+1-\ell}{K_1}\rfloor} \left\| \left(\widehat{W}\Sigma\right)^{\ell+t\,K_1-1:\ell+(t-1)K_1} \right\|_{\infty}\right) \left\| \left(\widehat{W}\Sigma\right)^{k:\ell+\lfloor\frac{k+1-\ell}{K_1}\rfloor K_1} \right\|_{\infty} \\
&\leq \prod_{t=1}^{\lfloor\frac{k+1-\ell}{K_1}\rfloor} \left\| \left(\widehat{W}\Sigma\right)^{\ell+t\,K_1-1:\ell+(t-1)K_1} \right\|_{\infty} \\
&\leq \Big(1-\Big(1-\sigma(\gamma)\Big)\eta\Big)^{\lfloor\frac{k+1-\ell}{K_1}\rfloor} \leq \Big(1-\Big(1-\sigma(\gamma)\Big)\eta\Big)^{\frac{k-\ell}{K_1}-1} \\
&= \frac{1}{1-\Big(1-\sigma(\gamma)\Big)\eta}\left(\Big(1-\Big(1-\sigma(\gamma)\Big)\eta\Big)^{\frac{1}{K_1}}\right)^{k-\ell},
\end{aligned}$$

where we defined $\prod_{t=1}^{0} x^t = 1$, for any sequence $\{x^t\}$.

7. SUMMARY

This dissertation provides a unified distributed algorithmic framework, which encapsulates the majority of existing first-order distributed algorithms. We also propose optimal distributed optimization algorithm, to reach the min-max lower complexity bound of first-order distributed algorithms. Finally, we break synchronism in the networked system and propose asynchronous distributed algorithms for practical large-scale optimization. Extensive simulation results validate our theoretical findings and the efficacy of the proposed algorithms.

Our accelerated distributed optimization algorithmic framework OPTRA provides a class of algorithms which are optimal for smooth convex optimization. One research question that remains open is, how to design distributed optimization algorithms which are optimal in terms of both computation and communication, for strongly convex problems. Although algorithms proposed in [51], [131] can achieve the optimal computation complexity, they are not optimal in terms of the communication complexity.

For asynchronous distributed/decentralized algorithms, one question is whether one can break the Assumption 4.3.2 of the partial asynchrony, and still design asynchronous distributed algorithms which converge linearly in the deterministic sense.

Note that an instance of particular interest to the Problem (P) is the distributed empirical risk minimization: a training data set $\{(u_s, y_s)\}_{s\in\mathcal{D}}$, with u_s being the input feature vector and y_s the outcome associated to item s, is partitioned into m subsets $\{\mathcal{D}_{\mathrm{i}}\}_{\mathrm{i}\in[m]}$, each of which is assigned to a machine $\mathrm{i} \in [m]$. The goal is to learn a mapping $p(\cdot\,; x)$ parameterized by $x \in \mathbb{R}^d$ using all samples in $\mathcal{D}$ by minimizing the empirical risk $\sum_{s\in\mathcal{D}} \ell\left(p(u_s; x), y_s\right) + G(x)$, with each agent having access to only $f_{\mathrm{i}}(x) = 1/|\mathcal{D}| \sum_{s\in\mathcal{D}_{\mathrm{i}}} \ell\left(p(u_s; x), y_s\right)$. Real applications usually impose high-dimensional variables, leading to the challenge of designing communication-efficient distributed algorithms. In the meantime, when the data are i.i.d. among machines, the Hessian matrices of local functions are related (cf. Sec. 2.5.4). As discussed partially in Sec. 2.5.4 and Sec. 2.6.3, the statistical similarity/homogeneity of local functions significantly impacts the convergence rate of distributed optimization algorithms. Therefore, another promising research direction is to exploit such statistical similarity in the algorithmic design, and further study the accelerated/optimal distributed algorithms in such a setting, in order to minimize the communication cost.

REFERENCES

[1] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Machine learning proceedings 1994*, Elsevier, 1994, pp. 157–163.

[2] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, “Fully decentralized multi-agent reinforcement learning with networked agents,” *arXiv preprint arXiv:1802.08757*, 2018.

[3] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, “Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

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

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

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

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

[8] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *IEEE Trans. Contr. of Netw. Syst.*, 2017.

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

[10] Y. Sun, A. Daneshmand, and G. Scutari, “Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation,” *arXiv preprint arXiv:1905.0263* 2019.

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

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

[13] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning–part i: Algorithm development,” *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708–723, 2018.

[14] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, “Optimal algorithms for smooth and strongly convex distributed optimization in networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3027–3036.

[15] D. Jakovetić, “A unification and generalization of exact distributed first-order methods,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 31–46, 2018.

[16] F. Mansoori and E. Wei, “A general framework of exact primal-dual first order algorithms for distributed optimization,” *arXiv:1903.12601*, 2019.

[17] E. Wei and A. E. Ozdaglar, “Distributed alternating direction method of multipliers,” in *Proceedings of IEEE 51st Annual Conference on Decision and Control (CDC)*, 2012, pp. 5445–5450.

[18] S. A. Alghunaim, E. Ryu, K. Yuan, and A. H. Sayed, “Decentralized proximal gradient algorithms with linear convergence rates,” *IEEE Transactions on Automatic Control*, 2020.

[19] W. Shi, Q. Ling, G. Wu, and W. Yin, “A proximal gradient algorithm for decentralized composite optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[20] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, “A bregman splitting scheme for distributed optimization over networks,” *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3809–3824, 2018.

[21] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, “Distributed linearized alternating direction method of multipliers for composite convex consensus optimization,” *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 5–20, 2017.

[22] S. Pu, W. Shi, J. Xu, and A. Nedic, “Push-pull gradient methods for distributed optimization in networks,” *IEEE Transactions on Automatic Control*, 2020.

[23] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[24] A. Sundararajan, B. Van Scoy, and L. Lessard, "A canonical form for first-order distributed optimization algorithms," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 4075–4080.

[25] A. Sundararajan, B. Hu, and L. Lessard, "Robust convergence analysis of distributed optimization algorithms," in *Proceedings of the 55th Annual Allerton Conference on Communication, Control, and Computing*, 2017, pp. 2740–2749.

[26] J. Xu, Y. Tian, Y. Sun, and G. Scutari, "Accelerated primal-dual algorithms for distributed smooth convex optimization over networks," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020.

[27] 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.

[28] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning–part ii: Convergence analysis," *IEEE Transactions on Signal Processing*, no. 3, pp. 724–739, 2018.

[29] Y. Nesterov, *Introductory lectures on convex optimization: A basic course.* Springer Science & Business Media, 2013, vol. 87.

[30] S. A. Alghunaim, E. K. Ryu, K. Yuan, and A. H. Sayed, "Decentralized proximal gradient algorithms with linear convergence rates," *arXiv preprint arXiv:1909.06479*, 2019.

[31] A. Wien, *Iterative solution of large linear systems.* Lecture Notes, TU Wien, 2011.

[32] J. Xu, Y. Sun, Y. Tian, and G. Scutari, "A unified contraction analysis of a class of distributed algorithms for composite optimization," *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2019.

[33] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 434–448, 2017.

[34] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge University Press, 2012.

[35] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proceedings of International Conference on Machine Learning*, 2014, pp. 1000–1008.

[36] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[37] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[38] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[39] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. on Optim.*, vol. 26, no. 3, pp. 1835–1854, 2016.

[40] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[41] A. Nedic and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv:1406.2075*, Jun. 2014.

[42] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process*, vol. 62, no. 7, pp. 1750–1761, 2014.

[43] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process*, vol. 63, no. 15, pp. 4051–4064, 2015.

[44] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process*, vol. 63, no. 2, pp. 482–497, 2015.

[45] K. Scaman, F. Bach, S. Bubeck, L. Massoulié, and Y. T. Lee, "Optimal algorithms for non-smooth distributed optimization in networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 2740–2749.

[46] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2018, pp. 38–42.

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

[48] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," *Mathematical Programming*, pp. 1–48, 2017.

[49] O. Shamir, "Fundamental limits of online and distributed algorithms for statistical learning and estimation," in *Advances in Neural Information Processing Systems*, 2014, pp. 163–171.

[50] Y. Arjevani and O. Shamir, "Communication complexity of distributed convex learning and optimization," in *Advances in neural information processing systems*, 2015, pp. 1756–1764.

[51] H. Li, C. Fang, W. Yin, and Z. Lin, "Decentralized accelerated gradient methods with increasing penalty parameters," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4855–4870, 2020.

[52] G. Qu and N. Li, "Accelerated distributed nesterov gradient descent," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2566–2581, 2019.

[53] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[54] Y. Chen, G. Lan, and Y. Ouyang, "Optimal primal-dual methods for a class of saddle point problems," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 1779–1814, 2014.

[55] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[56] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[57] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.

[58] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Auto. Contr.*, vol. 54, no. 1, pp. 48–61, 2009.

[59] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, 2012.

[60] A. Agarwal, S. Negahban, and M. J. Wainwright, "Fast global convergence of gradient methods for high-dimensional statistical recovery," *The Annals of Statistics*, pp. 2452–2482, 2012.

[61] D. P. Bertsekas, *Convex analysis and optimization*. Belmont, MA: Athena Scientific, 2003.

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

[63] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal–dual algorithm," *Mathematical Programming*, vol. 159, no. 1-2, pp. 253–287, 2016.

[64] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[65] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE transactions on automatic control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[66] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[67] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. of FOCS 2003*, IEEE, pp. 482–491.

[68] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 2996–3000.

[69] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM journal on control and optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[70] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 6698–6703.

[71] W. Ren, "Consensus seeking in multi-vehicle systems with a time-varying reference state," in *2007 American Control Conference*, IEEE, 2007, pp. 717–722.

[72] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[73] C. N. Hadjicostis, N. H. Vaidya, and A. D. Dominguez-Garcia, "Robust distributed average consensus via exchange of running sums," *IEEE Trans. Auto. Contr.*, vol. 31, no. 6, pp. 1492–1507, 2016.

[74] N. Bof, R. Carli, and L. Schenato, "Average consensus with asynchronous updates and unreliable communication," in *Proc. of the IFAC Word Congress 2017*, pp. 601–606.

[75] A. Nedić and A. Ozdaglar, "Convergence rate for consensus with delays," *J. Glob. Optim.*, vol. 47, no. 3, pp. 437–456, 2010.

[76] P. Lin and W. Ren, "Constrained consensus in unbalanced networks with communication delays," *IEEE Trans. Auto. Contr.*, vol. 59, no. 3, pp. 775–781, 2013.

[77] N. Bof, R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Newton-raphson consensus under asynchronous and lossy communications for peer-to-peer networks," *arXiv:1707.09178*, 2017.

[78] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Auto. Contr.*, vol. 31, no. 9, pp. 803–812, 1986.

[79] J. Liu and S. J. Wright, "Asynchronous stochastic coordinate descent: Parallelism and convergence properties," *SIAM J. Optim.*, vol. 25, no. 1, pp. 351–376, 2015.

[80] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari, "Asynchronous parallel algorithms for nonconvex optimization," *Math. Prog.*, Jun. 2019.

[81] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. of NIPS 2011*, pp. 693–701.

[82] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. of NIPS 2015*, pp. 2719–2727.

[83] I. Notarnicola and G. Notarstefano, "Asynchronous distributed optimization via randomized dual proximal gradient," *IEEE Trans. Auto. Contr.*, vol. 62, no. 5, pp. 2095–2106, 2017.

[84] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *Proc. of CDC 2013*, pp. 3671–3676.

[85] E. Wei and A. Ozdaglar, “On the $o(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers,” in *Proc. of GlobalSIP 2013*, pp. 551–554.

[86] P. Bianchi, W. Hachem, and F. Iutzeler, “A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization,” *IEEE Trans. Auto. Contr.*, vol. 61, no. 10, pp. 2947–2957, 2016.

[87] H. Wang, X. Liao, T. Huang, and C. Li, “Cooperative distributed optimization in multiagent networks with delays,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 45, no. 2, pp. 363–369, 2015.

[88] J. Li, G. Chen, Z. Dong, and Z. Wu, “Distributed mirror descent method for multi-agent optimization with delay,” *Neurocomputing*, vol. 177, pp. 643–650, 2016.

[89] K. I. Tsianos and M. G. Rabbat, “Distributed dual averaging for convex optimization under communication delays,” in *Proc. of ACC 2012*, pp. 1067–1072.

[90] K. I. Tsianos and M. G. Rabbat, “Distributed consensus and optimization under communication delays,” in *Proc. of Allerton 2011*, pp. 974–982.

[91] P. Lin, W. Ren, and Y. Song, “Distributed multi-agent optimization subject to non-identical constraints and communication delays,” *Automatica*, vol. 65, pp. 120–131, 2016.

[92] T. T. Doan, C. L. Beck, and R. Srikant, “Impact of communication delays on the convergence rate of distributed optimization algorithms,” *arXiv:1708.03277*, 2017.

[93] A. Nedić, “Asynchronous broadcast-based convex optimization over a network,” *IEEE Trans. Auto. Contr.*, vol. 56, no. 6, pp. 1337–1351, 2011.

[94] X. Zhao and A. H. Sayed, “Asynchronous adaptation and learning over networks–Part I/Part II/Part III: Modeling and stability analysis/Performance analysis/Comparison analysis,” *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 811–858, 2015.

[95] S. Kumar, R. Jain, and K. Rajawat, “Asynchronous optimization over heterogeneous networks via consensus admm,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 114–129, 2017.

[96] M. Eisen, A. Mokhtari, and A. Ribeiro, “Decentralized quasi-newton methods,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2613–2628, 2017.

[97] Z. Peng, Y. Xu, M. Yan, and W. Yin, "Arock: An algorithmic framework for asynchronous parallel coordinate updates," *SIAM J. Sci. Comput.*, vol. 38, no. 5, A2851–A2879, 2016.

[98] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," *IEEE Trans. Signal Inf. Process. Netw.*, vol. PP, no. 99, 2017.

[99] P. Di Lorenzo and G. Scutari, "Distributed nonconvex optimization over networks," in *Proc. of IEEE CAMSAP 2015*, Dec. 2015.

[100] A. Nedić, 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.

[101] Y. Sun, G. Scutari, and D. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," in *Proc. of Asilomar 2016*, IEEE, pp. 788–794.

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

[103] T. S. Rappaport, *Wireless Communications: Principles & Practice*. Prentice Hall, 2002.

[104] S. M. Kay, *Fundamentals of Statistical Signal Processing–Estimation Theory*. Prentice Hall, 1993.

[105] L. Cannelli, F. Facchinei, and G. Scutari, "Multi-agent asynchronous nonconvex large-scale optimization," in *Proc. of IEEE CAMSAP 2017*, pp. 1–5.

[106] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *Proc. of CDC 2015*, pp. 2055–2060.

[107] L. Zhao, M. Mammadov, and J. Yearwood, "From convex to nonconvex: A loss function analysis for binary classification," in *2010 IEEE ICDM Workshops*, pp. 1281–1288.

[108] M. Assran and M. Rabbat, "Asynchronous subgradient-push," *arXiv:1803.08950*, 2018.

[109] J. Zhang and K. You, "Asyspa: An exact asynchronous algorithm for convex optimization over digraphs," *arXiv:1808.04118*, 2018.

[110] R. A. Horn and C. R. Johnson, *Matrix analysis.* Cambridge university press, 1990.

[111] F. Facchinei, G. Scutari, and S. Sagratella, “Parallel selective algorithms for nonconvex big data optimization,” *IEEE Trans. on Signal Process.*, vol. 63, no. 7, pp. 1874–1889, 2015.

[112] G. Scutari and Y. Sun, “Parallel and distributed successive convex approximation methods for big-data optimization,” in *Multi-Agent Optimization*, F. Facchinei and J.-S. Pang, Eds., Springer, C.I.M.E. Foundation Subseries (Lecture Notes in Mathematics), 2018, pp. 141–308.

[113] G. Scutari, F. Facchinei, and L. Lampariello, “Parallel and distributed methods for constrained nonconvex optimization–Part I: Theory,” *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.

[114] G. Scutari, F. Facchinei, L. Lampariello, S. Sardellitti, and P. Song, “Parallel and distributed methods for constrained nonconvex optimization–Part II: Applications in communications and machine learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1945–1960, Apr. 2017.

[115] H. Hendrikx, F. Bach, and L. Massoulié, “Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums,” *arXiv preprint arXiv:1901.09865*, 2019.

[116] H. Hendrikx, L. Massoulié, and F. Bach, “Accelerated decentralized optimization with local updates for smooth and strongly convex objectives,” *arXiv preprint arXiv:1810.02660*, 2018.

[117] Y. Tian, Y. Sun, and G. Scutari, “Asy-sonata: Achieving geometric convergence for distributed asynchronous optimization,” *arXiv:1803.10359*, Mar. 2018.

[118] A. Olshevsky, I. C. Paschalidis, and A. Spiridonoff, “Robust asynchronous stochastic gradient-push: Asymptotically optimal and network-independent performance for strongly convex functions,” *arXiv preprint arXiv:1811.03982*, 2018.

[119] H. Zhang, J. Jiang, and Z.-Q. Luo, “On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems,” *Journal of the Operations Research Society of China*, vol. 1, no. 2, pp. 163–186, 2013.

[120] J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter, “From error bounds to the complexity of first-order descent methods for convex functions,” *Math. Prog.*, vol. 165, no. 2, pp. 471–507, 2017.

[121] Z.-Q. Luo and P. Tseng, "Error bounds and convergence analysis of feasible descent methods: A general approach," *Ann. of Oper. Res.*, vol. 46, no. 1, pp. 157–178, 1993.

[122] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2016, pp. 795–811.

[123] P. Tseng, "On the rate of convergence of a partially asynchronous gradient projection algorithm," *SIAM J. Optimiz.*, vol. 1, no. 4, pp. 603–619, 1991.

[124] U. T. R.S. Dembo, "Local convergence analysis for successive inexact quadratic programming methods," *Working Paper, School of Organization and Management, Yale University, New Haven, CT*, 1984.

[125] J.-S. Pang, "Inexact newton methods for the nonlinear complementarity problem," *Math. Prog.*, vol. 36, no. 1, pp. 54–71, 1986.

[126] B. T. Polyak, "Gradient methods for solving equations and inequalities," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 6, pp. 17–32, 1964.

[127] S. Lojasiewicz, "A topological property of real analytic subsets," *Coll. du CNRS, Les équations aux dérivées partielles*, vol. 117, pp. 87–89, 1963.

[128] D. Drusvyatskiy and A. S. Lewis, "Error bounds, quadratic growth, and linear convergence of proximal methods," *Math. Oper. Res.*, vol. 43, no. 3, pp. 919–948, 2018.

[129] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Math. Prog.*, vol. 117, no. 1-2, pp. 387–423, 2009.

[130] R. Zhang, Y. Mei, J. Shi, and H. Xu, "Robustness and tractability for non-convex m-estimators," *arXiv preprint arXiv:1906.02272*, 2019.

[131] H. Ye, L. Luo, Z. Zhou, and T. Zhang, "Multi-consensus decentralized accelerated gradient descent," *arXiv preprint arXiv:2005.00797*, 2020.

VITA

[Ye Tian received his B.S. degree in mathematics from Nanjing University, China, in 2016, and his Ph.D. degree in industrial engineering from Purdue University, West Lafayette, USA, in 2021. His research interests include optimization algorithms and their applications in machine learning.]