ASYNCHRONOUS PARALLEL ALGORITHMS FOR BIG-DATA NONCONVEX OPTIMIZATION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Loris Cannelli

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

To my parents, Patrizia & Renzo.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Cannelli, Loris PhD, Purdue University, August 2019. Asynchronous Parallel Algorithms for Big-Data Nonconvex Optimization. Major Professor: Gesualdo Scutari.

The focus of this Dissertation is to provide a unified and efficient solution method for an important class of nonconvex, nonsmooth, constrained optimization problems. Specifically, we are interested in problems where the objective function can be written as the sum of a smooth, nonconvex term, plus a convex, but possibly nonsmooth, regularizer. It is also considered the presence of nonconvex constraints. This kind of structure arises in many large-scale applications, as diverse as information processing, genomics, machine learning, or imaging reconstruction.

We design the first parallel, asynchronous, algorithmic framework with convergence guarantees to stationary points of the class of problems under exam. The method we propose is based on Successive Convex Approximation techniques; it can be implemented with both fixed and diminishing stepsizes; and enjoys sublinear convergence rate in the general nonconvex case, and linear convergence case under strong convexity or under less stringent standard error bound conditions. The algorithmic framework we propose is very abstract and general and can be applied to different computing architectures (e.g., message-passing systems, cluster of computers, shared-memory environments), always converging under the same set of assumptions.

In the last Chapter we consider the case of distributed multi-agent systems. Indeed, in many practical applications the objective function has a favorable separable structure. In this case, we generalize our framework to take into consideration the presence of different agents, where each one of them knows only a portion of the overall function, which they want cooperatively to minimize. The result is the first fully decentralized asynchronous method for the setting described above. The proposed method achieve sublinear convergence rate in the general case, and linear convergence rate under standard error bound conditions.

Extensive simulation results on problems of practical interest (MRI reconstruction, LASSO, matrix completion) show that the proposed methods compare favorably to state-of-the art-schemes.

1. INTRODUCTION

1.1 Motivation

In the last years a surge of interest has arisen in parallel and distributed optimization methods for large-scale systems. Nowadays, nonconvex large-scale optimization problems are found in a wide range of applications in different engineering areas as diverse as information processing over networks (e.g., parameter estimation, detection, localization, graph signal processing), communication networks (e.g., resource allocation in peer-to-peer/multi-cellular systems), sensor networks, recommendation systems (including Facebook, Google, Amazon, and YouTube), swarm robotic, and machine learning (e.g., nonlinear least squares, dictionary learning, PCA, tensor factorization), just to name a few - see Figure 1.1. To design optimization models and methods in order to deal with such complex systems requires the ability to solve a number of different *challenges*.

• **Big-Data -** The applications mentioned above usually involve a massive amount of data, leading to optimization problems with a huge number of variables. These problems are often referred to as Big-Data. In order to solve these large problems in a reasonable amount of time (real-time) and in order to cope with the curse of dimensionality, it becomes mandatory to rely on solution methods that leverage hierarchical computational architectures, and are parallel (e.g., clusters of computers, multi-cores environments, cloud-based networks), if available. From this point of view, the main challenge lies in the fact that the problems of interest are in general not separable in the optimization variables, which makes the design of a parallel method not a trivial goal.

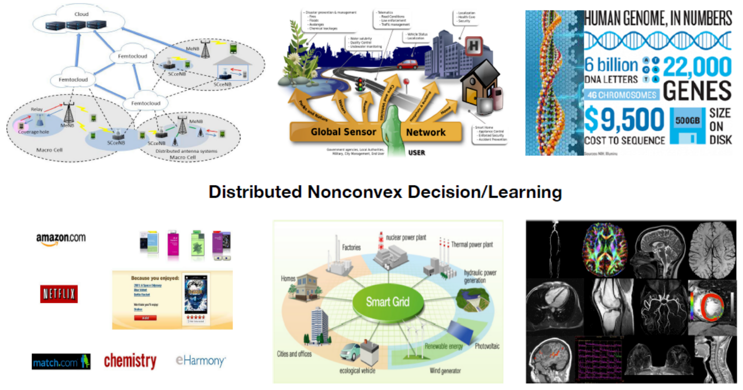

Figure 1.1. A general view of some relevant applications, which it is possible to write as nonconvex large-scale optimization problems over networks.

• **Nonconvexity -** Many of the aforementioned applications give rise to nonconvex optimization problems, where the nonconvexity can be present both at the level of the objective function, and both at the level of the constraints. In general, computing the global optimum solution of a nonconvex problem can be prohibitive in many applications of interest. Thus, the goal becomes to design (parallel/distributed) solution methods capable to find stationary solutions (e.g., local optimal points) in an efficient way, i.e. without requiring too many computational capabilities by the workers. In order to deal with nonconvexity, a powerful tool is represented by the so-called Successive Convex Approximation (SCA) techniques: according to this technique, a complex nonconvex optimization problem is transformed in a sequence of convex subproblem, easier to be solved. The convex subproblems are proper approximations of the original nonconvex problem, and it can be proved that solving a sequence of these subproblems leads to local solutions of the starting problem. Given the fact that the approximating subproblems can be constructed in several different ways, SCA techniques are a flexible and powerful tool in order to tackle the complexity coming from nonconvexity.

• **In-network Optimization -** Many applications of interest assume the presence of several agents, spatially/virtually distributed over a large area, that try to solve a problem cooperatively. Due to privacy/security reasons and due to the huge amount of data that should be communicated, usually each agent in these networks knows only a local portion of the information involved in the problem. The absence of a central coordinator avoids also that the overall system will collapse, if a single point crashes or goes under external attacks. So, the goal becomes to develop solution schemes where the agents compensate for the lack of global knowledge by exchanging information with their neighbors. The solution method should be flexible, in order to be applicable to different topologies, possibly time-varying

• **Asynchronous Optimization -** In a multi-agent system the presence of a central coordinator/clock can easily be a source of inefficiency. Indeed, what usually happens in multi-agent environments is that the slowest agent is the bottleneck and determines the global speed of the overall system-see Figure 1.2. In order to face this issue and avoid latencies and/or waiting times, *asynchrony* is needed. In an asynchronous environment no agent must wait for the others in order to perform its operations. The waiting times are highly reduced and the resulting system becomes more flexible without the need for a central clock.

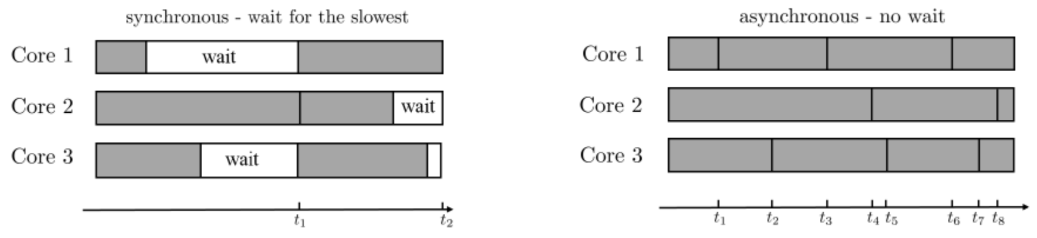

Figure 1.2. Multi-agent system. Left - synchronous environment: Core 3 and Core 1 have finished their own computations, but before going ahead they must wait time t_1 for Core 2 to finish its own computations too. Right - asynchronous environment: the cores are not forced to wait for each other anymore. Any core can perform its own computation without coordination with the other. Waiting times disappear in the asynchronous setting.

As a concrete example, consider the emerging field of in-network big-data analytics: the goal is to perform some, generally nonconvex, analytic tasks from a sheer volume of data, distributed over a network-see Fig. 1.3-examples include machine learning problems such as nonlinear least squares, dictionary learning, matrix completion, and tensor factorization, just to name a few. In these data-intensive applications, the huge volume and spatial/temporal disparity of data render centralized processing and storage a formidable task. This happens, for instance, whenever the volume of data overwhelms the storage capacity of a single computing device. Moreover, collecting sensor-network data, which are observed across a large number of spatially scattered centers/servers/agents, and routing all this local information to centralized processors, under energy, privacy constraints and/or link/hardware failures, is often infeasible or inefficient.

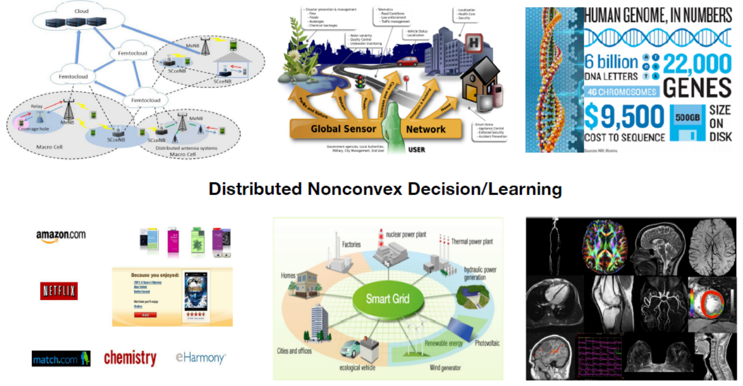

Figure 1.3. In-network big-data analytics: recent years witnessed the spread of large-scale networks of agents that try to solve huge problems in a distributed way. Classic optimization methods might be inapplicable in these settings, and need to be rethought in a fully decentralized way.

1.2 Research Contribution

The above challenges make clear that the classic optimization schemes are inapplicable or inefficient, thus calling for the development of new models and optimization schemes that support parallel, asynchronous, nonconvex optimization over decentralized architectures.

The major contribution of this Dissertation is to put forth a general, unified, algorithmic framework, based on SCA techniques, for the parallel, asynchronous, and distributed solution of a general class of nonconvex, nonsmooth, and constrained problems. Key novel features of the proposed framework are: i) is the first asynchronous optimization method to achieve provable convergence to stationary solutions of the class of problems under exam; ii) is guaranteed to achieve *linear* convergence rate on a large class of interesting problems; iii) generalizes existing SCA methods, opening the possibility of their parallel and/or distributed implementation and allowing the control of stepsizes, computation/communication trade-offs, and choices of approximant functions; iv) outperforms existing state-of-the-art methods on experiments of practical interest.

1.3 Outline of the Dissertation

This Dissertation is composed of three main Chapters, where we gradually build an algorithmic framework capable of solving nonconvex and nonsmooth constrained problems in a parallel, asynchronous, and distributed fashion.

In Chapter 2 the theory of SCA techniques, which will be a building block of the proposed optimization method, is briefly reviewed. We contribute to the existing literature of SCA along the following directions: i) linear convergence rate of FLEXA, a SCA-based parallel algorithm developed in [1], is proved under standard error bound conditions or under strong convexity of the objective function; ii) a specific instance of FLEXA is designed, tailored for the Magnetic Resonance Imaging (MRI) reconstruction problem. Experimental results show that the proposed version of FLEXA outperforms state-of-the-art schemes and paves the way towards real-time MRI.

Building on the FLEXA framework, in Chapter 3, AsyFLEXA, the first asynchronous, parallel algorithmic framework for nonconvex, nonsmooth optimization problems is presented. Convergence to stationary solutions is proved when a diminishing stepsize sequence is employed; while iteration complexity is provided when a fixed stepsize is used. Finally, near linear speed-up of AsyFLEXA, with respect to the number of agents involved in the computations, is demonstrated. Experimental results on convex and nonconvex practical problems show that AsyFLEXA compares favorably to existing asynchronous approaches.

In Chapter 4 we generalize AsyFLEXA to distributed optimization problems with a partitioned structure. Instances of this setting are multi-agent systems where, due to the size of the problems, or due to privacy issues, each agent has knowledge of a local portion of the objective function only. This kind of setting arises in, e.g., network utility maximization and resource allocation problems, cooperative localization in wireless network, or machine learning problems when a favorable sparsity structure is present. DAsyFLEXA, the first distributed and asynchronous algorithmic framework for nonconvex, nonsmooth, partitioned optimization problems, is presented in this Chapter. The main contributions of this Chapter are: i) asymptotic convergence to stationary solutions of DAsyFLEXA is derived when diminishing uncoordinated stepsize sequences are employed; iteration complexity is provided too, for stepsizes fixed and sufficiently small; ii) linear convergence rate of DAsyFLEXA is proved, when a standard error bound condition is satisfied; iii) experimental results on convex and nonconvex practical problems show that DAsyFLEXA performs more efficiently than existing asynchronous, decentralized approaches.

• **Notation -** Throughout the Dissertation we will use the following notation.

Small bold letters $\mathbf{x}$ will denote vectors, while capital bold letters $\mathbf{X}$ will be used for matrices. Subscripts $\mathbf{x}_i$, $\mathbf{X}_{ij}$ will indicate, respectively, the ith component of a vector $\mathbf{x}$, and the element on the ith row and jth column of a matrix $\mathbf{X}$. The gradient of a smooth function f will be denoted by ∇f. A subscript will specify the block-variable with respect to the gradient is taken. For instance, $\nabla_{\mathbf{x}} f(\mathbf{x})$ means that the gradient vector is taken with respect to the whole vector $\mathbf{x}$, while $\nabla_{\mathbf{x}_i} f(\mathbf{x})$ means that the gradient is computed with respect to the block-variable $\mathbf{x}_i$ only. For smooth functions depending on multiple arguments, i.e. $f(\cdot;\cdot)$, we use the convention that $\nabla f(\cdot;\cdot)$ will denote the partial gradient of f with respect to the first argument. For a nonsmooth function $g(\mathbf{x})$, $\partial g(\mathbf{x})$ will denote the Clarke subdifferential of G [2].

2. FLEXIBLE PARALLEL ALGORITHM (FLEXA)

In this Chapter we present the building blocks of the optimization methods that will be developed in Chapter 3 and Chapter 4. Specifically, Section 2.1 describes the class of problems of interest, together with some motivating example. In Section 2.2 we first briefly review FLEXA [3], a parallel algorithmic framework based on SCA techniques for nonconvex, and nonsmooth optimization. Then, we prove new convergence results of FLEXA, namely: i) linear convergence rate under strong convexity; and ii) linear convergence rate under a standard error bound assumption. Finally in Section 2.3 we design an instance of FLEXA tailored for a setting of practical interest, namely the MRI reconstruction problem, and we show that FLEXA has a convergence speed much higher than state-of-the-art schemes. The Appendix of this Chapter contains a review of standard error bound conditions, commonly assumed in the literature to prove linear convergence rate of iterative schemes.

The novel results of this Chapter have been published in

- Cannelli, L. and Scarponi, P. and Scutari, G. and Ying, L., *"A Parallel Algorithm for Compressed Sensing Dynamic MRI Reconstruction"*, Proceedings of the International Society for Magnetic Resonance in Medicine (ISMRM) Scientific Meeting Exhibition (Toronto), 2015 [4].

A sample code, written in C++, of FLEXA for LASSO problems is available at `https://github.com/lcannell/Parallel-Asynchronous-Algorithms-For-Nonconvex-Problems`.

2.1 Problem Formulation and Motivating Examples

The following optimization problem will be considered in the rest of this Chapter

$$\min_{\mathbf{x}\triangleq[\mathbf{x}_i]_{i=1}^N\in\mathcal{X}\subseteq\mathbb{R}^n} \quad V(\mathbf{x}) \triangleq F(\mathbf{x}) + \underbrace{\sum_{i=1}^N g_i(\mathbf{x})}_{\triangleq G(\mathbf{x})}, \tag{P}$$

satisfying the following assumptions

Assumption A (On Problem (P)).

(A1) $\mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_N$, and each set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is nonempty, closed, and convex;

(A2) F is C^1 and its gradients $\nabla_{\mathbf{x}_i} F$ are globally $L-$Lipschitz on $\mathcal{X}$;

(A3) Each $g_i : \mathcal{X}_i \to \mathbb{R}$ is convex;

(A4) V is bounded from below on $\mathcal{X}$.

Assumption A is very general and encompasses a gamut of formulations, arising from applications in several fields; many problems in areas as diverse as sensor networks, imaging, machine learning, data analysis, genomics, and geophysics, can be formulated as Problem (P) and satisfy Assumption A. Some illustrative examples are documented next.

Example #1−LASSO: Consider a linear regression task with p feature/outcome pairs $\{(z_i, \mathbf{a}_i)\}_{i=1}^p$, where $\mathbf{a}_i \in \mathbb{R}^m$ is a vector of features, and z_i is the associated outcome. Let $\mathbf{z} = (z_1, \ldots, z_p)^T$ denote the p-dimensional vector of outcomes, and $\mathbf{A} \in \mathbb{R}^{p\times m}$ be the matrix with $\mathbf{a}_i$ in its ith row. Then, the renowned LASSO problem [5] aims at finding a (sparse) vector of weights $\mathbf{x} \in \mathbb{R}^m$, and can be written as an instance of Problem (P), by setting $F(\mathbf{x}) = \|\mathbf{z} - \mathbf{A}\mathbf{x}\|_2^2$ and $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$, with $\lambda > 0$.

Example #2−Group LASSO: In many regression tasks it is of interest to select or omit all the coefficient within a group together. The *group* LASSO promotes this structure by using sums of (un-squared) ℓ_2-norm losses. Specifically, consider the regression vector $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T]^T$, $\mathbf{x} \in \mathbb{R}^m$, possessing a group sparse pattern , i.e., all the elements of $\mathbf{x}_i$ either take large values either they are close to zero [6]. A popular group LASSO formulation is Problem (P), where $F(\mathbf{x}) = \|\mathbf{z} - \mathbf{A}\mathbf{x}\|_2^2$ and $G(\mathbf{x}) = \lambda \sum_{i=1}^n \|\mathbf{x}_i\|_2$, with $\lambda > 0$.

Example #3−Sparse logistic regression: Given a training set $\{\mathbf{z}_i, w_i\}_{i=1}^p$, where $\mathbf{z}_i \in \mathbb{R}^m$ is the feature vector and $w_i \in \{-1, 1\}$ is the label of the ith sample, the logistic regression problem consists in minimizing the negative log likelihood [7, 8] $F(\mathbf{x}) = (1/p) \cdot \sum_{i=1}^p \log(1 + e^{-w_i \cdot \mathbf{z}_i^T \mathbf{x}})$.It is possible to also use some regularizer, e.g., $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$ (or $G(\mathbf{x}) = \lambda \sum_{i=1}^n \|\mathbf{x}_i\|_2$), with $\lambda > 0$. It is straightforward to see that this formulation falls into the framework (P).

Example #4−Dictionary Learning (DL): Dictionary learning is an unsupervised learning problem that consists in finding a matrix $\mathbf{D} \in \mathbb{R}^{p \times m}$ (the *dictionary*) such that data $\mathbf{z}_i \in \mathbb{R}^p$ can be represented in a sparse fashion by coefficients $\mathbf{x}_i \in \mathbb{R}^m$. Let $\mathbf{X} \in \mathbb{R}^{m \times I}$ and $\mathbf{Z} \in \mathbb{R}^{p \times I}$ be the representation and data matrix whose columns are the coefficients $\mathbf{x}_i$ and the data vectors $\mathbf{z}_i$, respectively. The DL problem is an instance of Problem (P), with $F(\mathbf{D}, \mathbf{X}) = \|\mathbf{Z} - \mathbf{D}\mathbf{X}\|_F^2$ and $G(\mathbf{X}) = \lambda\|\mathbf{X}\|_1$, $\mathcal{X} = \{(\mathbf{D}, \mathbf{X}) \in \mathbb{R}^{p \times m} \times \mathbb{R}^{m \times I} : \|\mathbf{D}\,\mathbf{e}_i\|^2 \leq \alpha_i, \forall i = 1, \ldots, m\}$, where $\mathbf{e}_i$ is the ith canonical vector, and $\|\mathbf{X}\|_F$ and $\|\mathbf{X}\|_1$ denote the Frobenius norm and the ℓ_1 matrix norm of $\mathbf{X}$, respectively. Note that $F(\mathbf{D}, \mathbf{X})$ is nonconvex, in fact is biconvex, i.e., convex in $\mathbf{D}$ and $\mathbf{X}$ separately.

Example #5−(Sparse) empirical risk minimization: Given a training set $\{D_i\}_{i=1}^I$, the goal in an empirical risk minimization problem is to find a model $h : \mathbb{R}^m \ni \mathcal{X} \to \mathbb{R}^p$, parameterized by the vector of variables $\mathbf{x}$, that minimizes the risk $\sum_{i=1}^I \ell\left(h\left(\mathbf{x}; D_i\right)\right)$, where $\ell : \mathbb{R}^p \to \mathbb{R}$ is a loss function, measuring the mismatch between the data and the model. This setting falls into the framework (P), with $F(\mathbf{x}) = \sum_{i=1}^I f_i(\mathbf{x})$ and $f_i(\mathbf{x}) \triangleq \ell\big(h\left(\mathbf{x}; D_i\right)\big)$. In order to impose some structure on the solution, as, for example, sparsity, is it possible to add a regularizer function G in the objective. Note that this setting contains the previous examples as special cases.

2.2 The FLEXA Algorithm

The most natural approach for solving (P) in a *parallel* way is to act *blockwise*: given a point $\mathbf{x}^k$, we update all the block-variables $\mathbf{x}_i$'s *simultaneously* in parallel by solving the following subproblems

$$\mathbf{x}_i^{k+1} \in \operatorname*{arg\,min}_{\mathbf{x}_i \in \mathcal{X}_i} \left\{ F(\mathbf{x}_i, \mathbf{x}_{-i}^k) + G(\mathbf{x}_i, \mathbf{x}_{-i}^k) \right\}, \qquad \forall i \in \mathcal{N} \triangleq \{1, \ldots, N\}, \tag{2.1}$$

Nevertheless, this approach has no convergence guarantees [9], even in the absence of the nonsmooth term G. Furthermore, solving (2.1) is in general difficult, due to the presence of the nonconvex term F. To cope with these two issues, FLEXA, the approach presented in [1], consists in solving for each block $\mathbf{x}_i$ the following subproblem

$$\widehat{\mathbf{x}}_i(\mathbf{x}^k) \triangleq \operatorname*{argmin}_{\mathbf{x}_i \in \mathcal{X}_i} \widetilde{F}_i(\mathbf{x}_i; \mathbf{x}^k) + g_i(\mathbf{x}_i), \tag{2.2}$$

and then setting

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \gamma^k \left(\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k \right). \tag{2.3}$$

In (2.2), $\widetilde{F}_i\left(\cdot; \mathbf{x}^k\right)$ is a strongly convex surrogate function approximating $F(\cdot, \mathbf{x}_{-i}^k)$, and in (2.3) a stepsize $\gamma^k \in (0, 1]$ is introduced to control the "length" of the update. A stepsize is needed to have convergence, when $\widetilde{F}_i\left(\cdot; \mathbf{x}^k\right)$ is not a global upper bound of $F(\cdot, \mathbf{x}_{-i}^k)$ (as it is, instead, in the Majorization-Minimization algorithms [10, 11]).

The surrogate function $\widetilde{F}_i$ must satisfy the following assumptions (we recall that $\nabla\tilde{F}_i$ denotes the partial gradient of $\tilde{F}_i$ with respect to the first argument).

Assumption B (On the surrogate functions $\tilde{F}_i$'s). Each $\tilde{F}_i : \mathcal{X}_i \times \mathcal{X} \to \mathbb{R}$, is chosen so that:

(B1) $\tilde{F}_i(\cdot; \mathbf{y})$ is C^1 and τ-strongly convex on $\mathcal{X}_i$, for all $\mathbf{y} \in \mathcal{X}$;

(B2) $\nabla\tilde{F}_i(\mathbf{y}_i; \mathbf{y}) = \nabla_{\mathbf{y}_i} F(\mathbf{y})$, for all $\mathbf{y} \in \mathcal{X}$;

(B3) $\nabla\tilde{F}_i(\mathbf{y}; \cdot)$ is L_A-Lipschitz continuous on $\mathcal{X}$, for all $\mathbf{y} \in \mathcal{X}_i$.

Assumption B requires that $\widetilde{F}_i$ should be a (simple) strongly convex approximation of F at the current point $\mathbf{x}^k$ that preserves the first order properties of F. Some examples of possible common choices for the surrogate functions $\widetilde{F}_i$'s are discussed next.

• **On the choice of the surrogate $\widetilde{F}_i$ -** FLEXA represents a gamut of parallel optimization schemes, each of them corresponding to a different choice of the surrogates $\widetilde{F}_i$ and stepsize rule.
Some examples of surrogate functions satisfying Assumption B are the following.

1) Block-wise convexity. Suppose $F(\mathbf{x})$ is convex in each block $\mathbf{x}_i$ separately, but not necessarily jointly convex. A natural surrogate for F in this case can be constructed by exploring its "partial" convexity: given $\mathbf{y} = [\mathbf{y}_i]_{i=1}^N \in \mathcal{X}$,

$$\widetilde{F}(\mathbf{x}; \mathbf{y}) \triangleq \sum_{i=1}^{N} \widetilde{F}_i(\mathbf{x}_i; \mathbf{y}), \tag{2.4}$$

with each $\widetilde{F}_i(\mathbf{x}_i; \mathbf{y})$ defined as

$$\widetilde{F}_i(\mathbf{x}_i; \mathbf{y}) \triangleq F(\mathbf{x}_i, \mathbf{y}_{-i}) + \frac{\alpha_i}{2} \langle \mathbf{x}_i - \mathbf{y}_i, \mathbf{H}_i (\mathbf{x}_i - \mathbf{y}_i) \rangle, \tag{2.5}$$

where $\alpha_i > 0$, and $\mathbf{H}_i$ is any $n_i \times n_i$ positive definite matrix (one can always choose $\mathbf{H}_i = \mathbf{I}$). The quadratic term in (2.5) can be set to zero if $F(\cdot, \mathbf{y}_{-i})$ is strongly convex on $\mathcal{X}_i$, for all $\mathbf{y}_{-i} \in \mathcal{X}_{-i} \triangleq \mathcal{X}_1 \times \cdots \times \mathcal{X}_{i-1} \times \mathcal{X}_{i+1} \times \cdots \times \mathcal{X}_N$.

An alternative choice to (2.5) could be

$$\begin{aligned} &\tilde{F}_i(\mathbf{x}_i; \mathbf{y}) \\ &= F(\mathbf{y}) + \langle \nabla_{\mathbf{x}_i} F(\mathbf{y}), \mathbf{x}_i - \mathbf{y}_i \rangle + \frac{1}{2} \left\langle \mathbf{x}_i - \mathbf{y}_i, \nabla^2_{\mathbf{x}_i \mathbf{x}_i} F(\mathbf{y})(\mathbf{x}_i - \mathbf{y}_i) \right\rangle + \alpha \|\mathbf{x}_i - \mathbf{y}_i\|_2^2, \end{aligned} \tag{2.6}$$

with $\alpha > 0$, which, with respect to (2.5), should lead to simpler subproblems (2.2), given the fact that in (2.6) we are saving less structure of the original F, by taking a second-order approximation of it.

2) (Proximal) gradient-like. If no convexity is present in F, a valid choice of $\widetilde{F}$ that can always be constructed, is the first order approximation of F (plus a quadratic regularization). In this case, $\widetilde{F}$ is given by (2.4), with:

$$\widetilde{F}_i(\mathbf{x}_i\,;\,\mathbf{y}) \triangleq \langle \nabla_{\mathbf{x}_i} F(\mathbf{y}), \mathbf{x}_i - \mathbf{y}_i \rangle + \frac{\alpha_i}{2} \|\mathbf{x}_i - \mathbf{y}_i\|_2^2, \tag{2.7}$$

where $\alpha_i > 0$. Note that when $\alpha_i < L$, $\widetilde{F}_i$ in (2.7) is not a global upper bound of $F(\cdot, \mathbf{x}_{-i})$.

3) Difference of Convex (DC) functions. Suppose that F is the difference of two convex functions $f^{(1)}$ and $f^{(2)}$, i.e., $F(\mathbf{x}) = f^1(\mathbf{x}) - f^2(\mathbf{x})$. One can preserve the partial convexity in F setting

$$\tilde{F}_i(\mathbf{x}_i; \mathbf{y}) = f^{(1)}(\mathbf{x}_i, \mathbf{y}_{-i}) - \left\langle \nabla_{\mathbf{x}_i} f^{(2)}(\mathbf{y}), \mathbf{x}_i - \mathbf{y}_i \right\rangle + \tau_i \|\mathbf{x}_i - \mathbf{y}_i\|_2^2, \tag{2.8}$$

where τ_i can be any positive number. As we will see in detail in Section 3.6.3, this structure arises in regression problems where it is of interest to estimate sparse solutions. Indeed, in order to impose sparsity in the solution it is possible to use nonconvex regularizers (e.g., SCAD [12] or MCP [13]), which can be written as DC functions.

4) Sum-utility. Suppose that $F(\mathbf{x}) \triangleq \sum_{i=1}^{I} f_i(\mathbf{x}_1, \ldots, \mathbf{x}_N)$. This kind of structure arises naturally in multi-agent systems wherein f_i is the local function of each agent i, which controls and updates its own block variables $\mathbf{x}_i \in \mathcal{X}_i$. In many practical cases the cost functions f_i are convex with respect to the block-variables of some agent. In order to leverage this partial convexity, we introduce the following set

$$\tilde{\mathcal{C}}_i \triangleq \{ j \,:\, f_j(\cdot, \mathbf{x}_{-i}) \text{ is convex}, \forall\, \mathbf{x}_{-i} \in \mathcal{X}_{-i} \}\,, \tag{2.9}$$

which collects the indexes of all the functions f_j that are convex in $\mathbf{x}_i$, for any $\mathbf{x}_{-i} \in \mathcal{X}_{-i}$; and let $\mathcal{C}_i \subseteq \tilde{\mathcal{C}}_i$ be any subset of $\tilde{\mathcal{C}}_i$. The following surrogate satisfies Assumption B and preserves the partial convexity of F (if any): given $\mathbf{y} = [\mathbf{y}_i]_{i=1}^N \in \mathcal{X}$,

$$\widetilde{F}(\mathbf{x}; \mathbf{y}) = \sum_{i=1}^{I} \widetilde{F}_{\mathcal{C}_i}(\mathbf{x}_i; \mathbf{y}),$$

with each $\widetilde{F}_{\mathcal{C}_i}$ defined as

$$\begin{aligned}\widetilde{F}_{\mathcal{C}_i}(\mathbf{x}_i;\mathbf{y}) \triangleq \sum_{j\in\mathcal{C}_i} f_j(\mathbf{x}_i,\mathbf{y}_{-i}) + \sum_{\ell\notin\mathcal{C}_i}\left\langle \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y}), \mathbf{x}_i - \mathbf{y}_i\right\rangle \\ + \frac{\alpha_i}{2}\left\langle \mathbf{x}_i - \mathbf{y}_i, \mathbf{H}_i\left(\mathbf{x}_i - \mathbf{y}_i\right)\right\rangle,\end{aligned} \tag{2.10}$$

where $\alpha_i > 0$, and $\mathbf{H}_i$ is any $n_i \times n_i$ positive definite matrix. The above surrogate function preserves the convex part of F w.r.t. $\mathbf{x}_i$ while linearizes the nonconvex part.

For further examples we refer the interested reader to [1, 14], where these types of approximations were extensively used.

• **Updating only some blocks -** When dealing with large-scale problems, updating all the block variables $\mathbf{x}_i$, $i \in \mathcal{N}$, in (2.2)-(2.3) at each iteration could be very inefficient, so FLEXA allows the parallel updates of *subsets* of block at a time. Specifically, at each iteration k a properly chosen subset of blocks–say $\mathcal{S}^k \subseteq \mathcal{N}$–is selected and updated by computing the solution $\widehat{\mathbf{x}}_i(\mathbf{x}^k)$ of the subproblem (2.2): given $\mathbf{x}^k$ and $\mathcal{S}^k$, let

$$\mathbf{x}_i^{k+1} = \begin{cases} \mathbf{x}_i^k + \gamma^k\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\right), & \text{if } i \in \mathcal{S}^k, \\ \mathbf{x}_i^k & \text{if } i \notin \mathcal{S}^k. \end{cases}$$

Several options are possible for the block selection rule $\mathcal{S}^k$ (see [15] for details). Specifically, in the rest of this work we will be interested in greedy-like schemes that update at each iteration only the blocks that are "far away" from the optimum - an idea which have been shown to be quite effective in practical applications.

Assumption C (On the block selection rule).

(C1) Greedy rule: Each $\mathcal{S}^k$ contains at least one index i such that

$$E_i\big(\mathbf{x}^k\big) \geq \rho \max_{j\in\mathcal{N}} E_j(\mathbf{x}^k),$$

where $\rho \in (0, 1]$ and $E_i\left(\mathbf{x}^k\right)$ is an error bound function satisfying

$$\underline{s}_i \cdot \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|_2 \leq E_i(\mathbf{x}^k) \leq \bar{s}_i \cdot \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|_2, \tag{2.11}$$

for some $0 < \underline{s}_i \leq \bar{s}_i < +\infty$.

$E_i\left(\mathbf{x}^k\right)$ in C1 can be viewed as a local measure of the distance of block i from optimality. Specifically, according to C1, one block that is within a fraction ρ from the largest distance $E_i\left(\mathbf{x}^k\right)$ should be updated. Some examples of valid error bound functions E_i are discussed in [15].

We remark that it is always possible to select $\mathcal{S}^k = \mathcal{N}$, resulting in the simultaneous update of all the block-variables at each iteration. At the other extreme, one can update only one block-variable at each iteration, thus obtaining a Gauss-Southwell kind of method. One can also explore all the possibilities "in between", by choosing properly $\mathcal{S}^k$ and ρ in C1 to control the degree of parallelism.

The selection of the most efficient updating rule depends on the specific application of interest, including the problem dimension, the computational environment, as well as the communication network among the agents.

Th parallel selective SCA method described in this Section is summarized in Algorithm 1, and termed "FLEXible parallel sca Algorithm" (FLEXA).

Algorithm 1: Flexible Parallel SCA algorithm (FLEXA)

Data : $\mathbf{x}^0 \in \mathcal{X}$, $\{\gamma^k\}$. Set $k = 0$.

`(S.1)` If $\mathbf{x}^k$ satisfies a termination criterion: STOP;

`(S.2)` Choose a set $\mathcal{S}^k$ satisfying C1;

`(S.3)` For all $i \in \mathcal{S}^k$, solve (2.2) and then set

$$\widehat{\mathbf{z}}_i^k = \begin{cases} \widehat{\mathbf{z}}_i(\mathbf{x}^k) & i \in \mathcal{S}^k, \\ \mathbf{x}_i^k & i \notin \mathcal{S}^k; \end{cases}$$

`(S.4)` Compute $\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \gamma^k\left(\widehat{\mathbf{z}}_i^k - \mathbf{x}_i^k\right)$, for any $i \in \mathcal{N}$;

`(S.5)` Update $k \leftarrow k + 1$, and go to `(S.1)`.

To complete the description of the algorithm, we need to specify how to choose the stepsize γ^k in `(S.4)`. The following assumption is required.

Assumption D (On the stepsize). The sequence $\{\gamma^k\} \in (0,1]$ satisfies any of the following rules:

(D1) **Bounded stepsize**: $0 < \liminf_{k\to\infty} \gamma^k \le \limsup_{k\to\infty} \gamma^k < 2\,\tau/L$;

(D2) **Diminishing stepsize**: $\sum_{k=0}^{\infty} \gamma^k = +\infty$ and $\sum_{k=0}^{\infty} \left(\gamma^k\right)^2 < +\infty$.

Assumption D is quite standard in the optimization literature. In scenarios where the knowledge of global problem parameters, e.g., L, is not available, one can use a diminishing stepsize satisfying D2. Two examples of diminishing stepsize rules are:

$$\gamma^k = \gamma^{k-1}\left(1 - \epsilon\,\gamma^{k-1}\right), \quad k = 1, \ldots, \quad \gamma^0 < 1/\epsilon; \tag{2.12}$$

$$\gamma^k = \frac{\gamma^{k-1} + \alpha(k)}{1 + \beta(k)}, \quad k = 1, \ldots, \quad \gamma^0 = 1; \tag{2.13}$$

where in (2.12) $\epsilon \in (0,1)$ is a given constant, whereas in (2.13) $\alpha(k)$ and $\beta(k)$ are two nonnegative real functions of $k \ge 1$ such that: i) $0 \le \alpha(k) \le \beta(k)$; and ii) $\alpha(k)/\beta(k) \to 0$ as $k \to \infty$ while $\sum_k (\alpha(k)/\beta(k)) = \infty$. Standard choices for $\alpha(k)$ and $\beta(k)$ are: $\alpha(k) = \alpha$ or $\alpha(k) = \log(k)^\alpha$, and $\beta(k) = \beta \cdot k$ or $\beta(k) = \beta \cdot \sqrt{k}$, with $\alpha \in (0,1)$, $\beta \in (0,1)$, and $\alpha \le \beta$.

2.2.1 Convergence Analysis

The following Proposition characterizes the best-response map $\widehat{\mathbf{x}}(\cdot) \triangleq [\widehat{\mathbf{x}}_i(\cdot)]_{i=1}^N$, with $\widehat{\mathbf{x}}_i(\cdot)$ defined in (2.2). This characterization will be used to derive the convergence properties of FLEXA.

Proposition 2.2.1 *Under Assumptions A-B, for any $i \in \mathcal{N}$ and $\mathbf{y}, \mathbf{z} \in \mathcal{X}$*

(a) [Optimality]

$$\left\langle \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k \right\rangle + g_i(\widehat{\mathbf{x}}_i(\mathbf{x}^k)) - g_i(\mathbf{x}_i^k) \le -\tau \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|_2^2. \tag{2.14}$$

(b) [Lipschitz continuity]

$$\|\widehat{\mathbf{x}}_i(\mathbf{y}) - \widehat{\mathbf{x}}_i(\mathbf{z})\|_2 \le \frac{L_A}{\tau} \|\mathbf{y} - \mathbf{z}\|_2; \tag{2.15}$$

(c) [Fixed-point characterization] *The set of fixed-points of* $\widehat{\mathbf{x}}(\cdot)$ *coincides with the set of stationary solutions of Problem* (P). *Therefore* $\widehat{\mathbf{x}}(\cdot)$ *has at least one fixed point.*

Proof See [1]. ■

The main convergence result of FLEXA is stated next.

Theorem 2.2.1 *Given Problem* (P) *under Assumption A. Let* $\{\mathbf{x}^k\}$ *be the sequence generated by FLEXA, under Assumptions B-C. Then, the following holds:*
(a) If the stepsize sequence $\{\gamma^k\}$ *satisfies D1, then it exists a subsequence of* $\{\mathbf{x}^k\}$ *such that any limit point of this subsequence is a stationary solution of problem P;*
(b) If the stepsize sequence $\{\gamma^k\}$ *satisfies D2, then any limit point of* $\{\mathbf{x}^k\}$ *is a stationary solution of problem P.*

Proof See [1, 15]. ■

The existence of a limit point of $\{\mathbf{x}^k\}$ is guaranteed under standard extra conditions on the feasible set $\mathcal{X}$ - e.g., boundedness - or on the objective function V -e.g., coercivity on $\mathcal{X}$.

Linear Convergence Rate of FLEXA under strong convexity

When the objective function V possesses more structure, it is possible to obtain stronger convergence results than those stated in Theorem 2.2.1. Specifically, in the following we will prove *linear* convergence rate of FLEXA under the assumption of V being strongly convex.

In the following we will assume that at each iteration k, all the block variables $\mathbf{x}_i$, $i \in \mathcal{N}$, are updated. This means $\mathcal{S}^k \equiv \mathcal{N}$, which satisfies C1.

We will assume the presence of a fixed stepsize $\gamma^k = \gamma$, for any $k \geq 0$, satisfying D1 (see more details later in the proof).

• **Notation -** For ease of presentation, we will assume $\mathcal{X} \equiv \mathbb{R}^n$, unless otherwise specified. In order to include the presence of constraints, then, we will assume that G is an extended value function, so it can express the presence of constraints by means of indicator functions. For ease of notation, we will use the following definition: $\Delta\widehat{\mathbf{x}}^k \triangleq \widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k$, for any $k \geq 0$.

In the rest of the proof we will use the following results, which hold true in the aforementioned setting.

To prove linear convergence rate of FLEXA under strong convexity we rely on the following two Lemmas.

Lemma 2.2.2 *Let $\{\mathbf{x}^k\}$ be the sequence generated by the FLEXA algorithm, under Assumptions A-B. Suppose further that V is $\tilde{\tau}$-strongly convex. Set $\mathcal{S}^k \equiv \mathcal{N}$ and $\gamma^k = \gamma \in (0;1]$ for any $k \geq 0$. Then, the following holds:*

$$V(\mathbf{x}^{k+1}) - V^\star \leq (1-\gamma)\left(V(\mathbf{x}^k) - V^\star\right) + \frac{\gamma\,(L^2 + NL_A^2)}{\tilde{\tau}}\|\Delta\widehat{\mathbf{x}}^k\|_2^2, \tag{2.16}$$

with $\mathbf{x}^\star$ being the optimal solution of problem (P)*, and $V^\star = V(\mathbf{x}^\star)$.*

Proof Since F is strongly convex, and by defining $\widehat{\mathbf{g}}^k \in \partial G(\widehat{\mathbf{x}}(\mathbf{x}^k))$ we have, for any $k \geq 0$

$$\begin{aligned}
V^\star &\geq V(\widehat{\mathbf{x}}(\mathbf{x}^k)) + \left\langle \nabla_{\mathbf{x}} F(\widehat{\mathbf{x}}(\mathbf{x}^k)) + \widehat{\mathbf{g}}^k, \mathbf{x}^\star - \widehat{\mathbf{x}}(\mathbf{x}^k) \right\rangle + \frac{\tilde{\tau}}{2}\|\mathbf{x}^\star - \widehat{\mathbf{x}}(\mathbf{x}^k)\|_2^2 \\
&\overset{(a)}{\geq} V(\widehat{\mathbf{x}}(\mathbf{x}^k)) + \sum_{i=1}^N \left\langle \nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \widehat{\mathbf{x}}(\mathbf{x}^k)\right) - \nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \mathbf{x}^k\right), \mathbf{x}_i^\star - \widehat{\mathbf{x}}_i(\mathbf{x}^k) \right\rangle \\
&+ \frac{\tilde{\tau}}{2}\|\mathbf{x}^\star - \widehat{\mathbf{x}}^k(\mathbf{x}^k)\|_2^2 \\
&= V(\widehat{\mathbf{x}}(\mathbf{x}^k)) + \sum_{i=1}^N \left(\frac{\tilde{\tau}}{2}\|\mathbf{x}_i^\star - \widehat{\mathbf{x}}_i(\mathbf{x}^k) + \frac{1}{\tilde{\tau}}\Big(\nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \widehat{\mathbf{x}}(\mathbf{x}^k)\right) \right. \\
&- \nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \mathbf{x}^k\right) \Big)\|_2^2 - \left. \frac{1}{2\tilde{\tau}}\|\nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \widehat{\mathbf{x}}(\mathbf{x}^k)\right) - \nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \mathbf{x}^k\right)\|_2^2 \right) \\
&\geq V(\widehat{\mathbf{x}}(\mathbf{x}^k)) - \frac{1}{2\tilde{\tau}} \sum_{i=1}^N \|\nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \widehat{\mathbf{x}}(\mathbf{x}^k)\right) - \nabla \tilde{F}_i\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \mathbf{x}^k\right)\|_2^2 \\
&\overset{(b)}{\geq} V(\widehat{\mathbf{x}}(\mathbf{x}^k)) - \frac{(L^2 + NL_A^2)}{\tilde{\tau}}\|\Delta\widehat{\mathbf{x}}^k\|_2^2, \qquad (2.17)
\end{aligned}$$

where (a) follows from B2 and the optimality condition of $\widehat{\mathbf{x}}_i(\mathbf{x}^k)$, i.e.,

$$\widehat{\mathbf{g}}_i^k \in -\nabla \tilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{x}^k); \mathbf{x}^k), \qquad (2.18)$$

while (b) follows from A2, B2, and B3. ■

Note that the following Lemma does not require strong convexity of V.

Lemma 2.2.3 *Let $\{\mathbf{x}^k\}$ be the sequence generated by the FLEXA algorithm, under Assumptions A-B. Set $\mathcal{S}^k \equiv \mathcal{N}$ and $\gamma^k = \gamma \in (0;1]$ for any $k \geq 0$. Then, the following holds:*

$$V(\mathbf{x}^{k+1}) - V(\mathbf{x}^\star) \leq V(\mathbf{x}^k) - V^\star - \gamma\left(\tau - \frac{L\gamma}{2}\right)\|\Delta\widehat{\mathbf{x}}^k\|_2^2. \qquad (2.19)$$

Proof For any $k \geq 0$, the following holds:

$$\begin{aligned}
V(\mathbf{x}^{k+1}) &= F(\mathbf{x}^{k+1}) + G(\mathbf{x}^{k+1}) \\
&\overset{\text{(a)}}{\leq} F(\mathbf{x}^k) + G(\mathbf{x}^{k+1}) + \gamma\left\langle \nabla F(\mathbf{x}^k), \Delta\widehat{\mathbf{x}}^k \right\rangle + \frac{L\gamma^2}{2}\|\Delta\widehat{\mathbf{x}}^k\|_2^2 \\
&\overset{(b)}{\leq} V(\mathbf{x}^k) - \tau\gamma\|\Delta\widehat{\mathbf{x}}^k\|_2^2 + \frac{L\gamma^2}{2}\|\Delta\widehat{\mathbf{x}}^k\|_2^2 \\
&\leq V(\mathbf{x}^k) - \gamma\left(\tau - \frac{L\gamma}{2}\right)\|\Delta\widehat{\mathbf{x}}^k\|_2^2,
\end{aligned}$$

where (a) follows from the Descent Lemma [9, Proposition A32] and A2; and (b) follows from Proposition 2.2.1(a) and A3 ■

By standard arguments and the fact that (P) is a strongly convex optimization problem, for $\frac{\gamma}{\tau} < \frac{2}{L}$ we can see from Lemma 2.2.3 that: i) $\left\{V(\mathbf{x}^k)\right\}$ converges to $V^\star \in \mathbb{R}$; ii) the sequences $\{V(\mathbf{x}^k)\}$, $\{\mathbf{x}^k\}$ are bounded; and iii)

$$\lim_{k\to+\infty} \|\Delta\widehat{\mathbf{x}}^k\|_2 = 0. \tag{2.20}$$

Thanks to the continuity of $\widehat{\mathbf{x}}(\cdot)$ (see Proposition 2.2.1(b)) and Proposition 2.2.1(c), the above results implies that the limit point $\mathbf{x}^\infty$ of $\{\mathbf{x}^k\}$ is the optimal solution of (P).

The linear convergence rate of FLEXA under strong convexity is established in the following Theorem.

Theorem 2.2.4 (Convergence rates) *Let $\{\mathbf{x}^k\}$ be the sequence generated by the FLEXA algorithm, under Assumptions A-B. Suppose further that V is $\tilde{\tau}$-strongly convex. Set $\mathcal{S}^k \equiv \mathcal{N}$. If $\gamma^k = \gamma \in (0;1]$, and $\frac{\gamma}{\tau} < \frac{2}{L}$, the following hold:*

(a) (Q-linear convergence of $\{V(\mathbf{x}^k)\}$):

$$V(\mathbf{x}^{k+1}) - V^\star \leq \beta_\gamma \left(V(\mathbf{x}^k) - V^\star\right), \tag{2.21}$$

with

$$\beta_\gamma \triangleq 1 - \frac{\tilde{\tau}\gamma\,(2\tau - L\gamma)}{2(L^2 + NL_A^2) + \tilde{\tau}\,(2\tau - L\gamma)}$$

(b) (R-linear convergence of $\{\mathbf{x}^k\}$):

$$\|\mathbf{x}^k - \mathbf{x}^\star\| = \mathcal{O}(\phi^k),$$

with

$$\phi = \sqrt{\beta_\gamma}.$$

Proof (a): By multiplying (2.19) and (2.16) for $\frac{\gamma(L^2+NL_A^2)}{\tilde{\tau}}$ and $\gamma\left(\tau - \frac{L\gamma}{2}\right)$ respectively, and then summing them together, we get

$$\begin{aligned}&\left(\frac{\gamma(L^2+NL_A^2)}{\tilde{\tau}} + \gamma\left(\tau - \frac{L\gamma}{2}\right)\right)\left(V(\mathbf{x}^{k+1}) - V^\star\right)\\ &\leq \left((1-\gamma)\gamma\left(\tau - \frac{L\gamma}{2}\right) + \frac{\gamma(L^2+NL_A^2)}{\tilde{\tau}}\right)\left(V(\mathbf{x}^{k}) - V^\star\right).\end{aligned}$$

By rearranging the terms, the result easily follows.

(b): Given that $\{V(\mathbf{x}^k)\}$ is monotonically decreasing (cf. (2.19)), it must be

$$V(\mathbf{x}^k) \geq V^\star, \quad \forall k. \tag{2.22}$$

Then, we can write: for any $k \geq 0$:

$$\begin{aligned}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 &= \gamma\|\Delta\widehat{\mathbf{x}}^k\|_2\\ &\overset{(2.19)}{\leq} \gamma\sqrt{\left(\gamma\left(\tau - \frac{L\gamma}{2}\right)\right)^{-1}|V(\mathbf{x}^k) - V(\mathbf{x}^{k+1})|}\\ &\leq \gamma\sqrt{\left(\gamma\left(\tau - \frac{L\gamma}{2}\right)\right)^{-1}(V(\mathbf{x}^k) - V^\star + V(\mathbf{x}^{k+1}) - V^\star)},\\ &\overset{(2.21)}{\leq} \gamma\sqrt{\left(\gamma\left(\tau - \frac{L\gamma}{2}\right)\right)^{-1}(1+\beta_\gamma)\left(V(\mathbf{x}^k) - V^\star\right)}\\ &\leq \gamma\sqrt{\left(\gamma\left(\tau - \frac{L\gamma}{2}\right)\right)^{-1}(1+\beta_\gamma)\,\beta_\gamma^k\left(V(\mathbf{x}^0) - V^\star\right)}.\end{aligned} \tag{2.23}$$

The above result proves that $\{\mathbf{x}^k\}$ is a Cauchy sequence. We already proved that its unique limit point $\mathbf{x}^\infty$ is the optimal solution of (P). Finally, invoking again (2.23), we have: for any $k' > k \geq 0$,

$$\begin{aligned}\|\mathbf{x}^k - \mathbf{x}^{k'}\|_2 &\leq \sum_{l=k}^{k'-1}\|\mathbf{x}^l - \mathbf{x}^{l+1}\|_2\\ &\leq \gamma\sqrt{\left(\gamma\left(\tau - \frac{L\gamma}{2}\right)\right)^{-1}(1+\beta_\gamma)(V(\mathbf{x}^0) - V^\star)}\sum_{l=k}^{k'-1}\sqrt{\beta_\gamma^l}\\ &= \mathcal{O}(\sqrt{\beta_\gamma^k}).\end{aligned}$$

By taking $k' \to +\infty$ and using $\mathbf{x}^k \to \mathbf{x}^\infty$, we get:

$$\|\mathbf{x}^k - \mathbf{x}^\infty\|_2 = \mathcal{O}\left(\sqrt{\beta_\gamma^k}\right),$$

which proves (b). ■

2.2.2 Convergence under Error Bounds

Error bounds refer to inequalities that bound the distance of vectors in a test set to a given set, by a residual function. Many structured optimization problems possess objective functions which can be characterized by error bounds. This characterization has been successfully leveraged as a tool to analyze the convergence rate of specific iterative methods.

Specifically, if an objective function V satisfies some error bound property, then many first order iterative schemes are guaranteed to reach a stationary point of V at a *linear* rate. We want to remark that the error bound properties are less stringent than strong convexity, which is usually required to prove linear convergence rate of optimization algorithms.

In the Appendix of this Chapter we report an extensive review about the most common error bounds, analyzing connections and relationships among them, and listing the practical applications satisfying at least one of these conditions.

In the rest of this Section we will assume that V satisfies the well-known Kurdyka-Łojasiewicz (KL) property and we will prove linear convergence rate of FLEXA under this additional assumption.

Definition 2.2.1 (KL property [16]) *A closed function $\phi : \mathbb{R}^n \to (-\infty;\infty]$ has the KL property at $\bar{\mathbf{x}} \in dom\, \partial\phi$ with an exponent of $\theta \in (0;1]$ if there exist a neighborhood $\mathcal{C}$ of $\bar{\mathbf{x}}$, $\eta \in (0,+\infty]$, such that:*

$$dist\,(0, \partial\phi(\mathbf{x})) \geq c\,(\phi(\mathbf{x}) - \phi(\bar{\mathbf{x}}))^{\theta} \tag{2.24}$$

whenever $\mathbf{x} \in \mathcal{C}$ and $\phi(\bar{\mathbf{x}}) < \phi(\mathbf{x}) < \phi(\bar{\mathbf{x}}) + \eta$.

If a function ϕ satisfies the KL property at all points in dom $\partial\phi$, then ϕ is called a KL function. If a function ϕ is a KL function and has the same exponent θ at any $\bar{\mathbf{x}} \in \partial\phi$, then ϕ is a KL function with an exponent of θ.

The KL property is satisfied by many functions of practical interest, both in the convex (i.e., quadratic loss functions, logistic function, l_1 penalty function) and in the nonconvex (i.e., SCAD penalty function, MCP penalty function) case. We refer the reader to the Appendix of this Chapter for more details about the KL property and the classes of problems covered.
As done in the strongly convex case, in the following we will assume that at each iteration k, all the block variables $\mathbf{x}_i$, $i \in \mathcal{N}$, are updated. This means $\mathcal{S}^k \equiv \mathcal{N}$, which satisfies C1.

We will also assume $\gamma = 1$. Consequently, we require the strong convexity parameter τ of the surrogate functions F_i's to be greater than $L/2$, in order to satisfy D1.

Finally, in order to claim convergence of the sequence $\{\mathbf{x}^k\}$ we will also require V to be coercive, which is slightly more stringent than A4.

• **Notation -** For ease of presentation, we will assume $\mathcal{X} \equiv \mathbb{R}^n$, unless otherwise specified. In order to include the presence of constraints, then, we will assume that G is an extended value function, so it can express the presence of constraints by means of indicator functions. For ease of notation, we will use the following definition: $\Delta\widehat{\mathbf{x}}^k \triangleq \mathbf{x}^{k+1} - \mathbf{x}^k$, for any $k \geq 0$.

In order to prove linear rate we will need the following two Lemmas.

Lemma 2.2.5 *Let $\{\mathbf{x}^k\}$ be the sequence generated by the FLEXA algorithm, under Assumptions A-B. Set $\gamma = 1$. There holds:*

$$\|\mathbf{v}^{k+1}\|_2^2 \leq N L_A^2 \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2, \tag{2.25}$$

where $\mathbf{v}^{k+1} \in \partial V(\mathbf{x}^{k+1})$.

Proof According to the optimality condition of $\widehat{\mathbf{x}}_i(\mathbf{x}^k)$, the following holds

$$\mathbf{g}_i^{k+1} \in -\nabla \tilde{F}_i(\mathbf{x}_i^{k+1}; \mathbf{x}^k), \tag{2.26}$$

where $\mathbf{g}_i^{k+1} \in \partial g_i(\mathbf{x}_i^{k+1})$. This result and B3 imply:

$$\|\nabla_{\mathbf{x}_i} F(\mathbf{x}^{k+1}) + \mathbf{g}_i^{k+1}\|_2 \leq L_A \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2.$$

The desired result easily follows. ■

Lemma 2.2.6 *Let $\{\Lambda^k\}$, with $\Lambda^k \geq 0$, be a decreasing sequence. Assume that there exist $\bar{k} > 0$, $\theta \in [0;1)$ and $\alpha \in (0;+\infty)$, such that, for every $k \geq \bar{k}$:*

$$\Lambda^k \leq \Lambda^{k-1} - \Lambda^k + \alpha\left(\Lambda^{k-1} - \Lambda^k\right)^{\frac{1-\theta}{\theta}}.$$

Then,

(i) If $\theta \in (0; \frac{1}{2}]$

$$\Lambda^k \leq \lambda \tau^k,$$

with $\lambda \triangleq \Lambda^0$ and $\tau \triangleq \frac{1+\alpha}{2+\alpha}$.

(ii) If $\theta \in (0;1)$

$$\Lambda^k \leq \beta k^{-\frac{1-\theta}{2\theta-1}},$$

with $\beta > 0$.

Proof See [17], Theorem 2. ■

As a first step, we demonstrate that FLEXA converges to stationary points of Problem (P).

Theorem 2.2.7 *Let $\{\mathbf{x}^k\}$ be the sequence generated by the FLEXA algorithm, under Assumptions A-B. Suppose further that V is coercive and satisfies the KL property* (2.24) *with exponent θ. Set $\mathcal{S}^k \equiv \mathcal{N}$ and $\gamma^k = 1$ for any $k \geq 0$. If $\tau > \frac{L}{2}$, the sequence converges to a stationary point $\mathbf{x}^\star$ of Problem* (P)*.*

Proof The statement of Lemma 2.2.3 still holds in setting, with $\gamma = 1$.

By standard arguments and the coercivity of V, for $\tau > \frac{L}{2}$ we can see from Lemma (2.2.3) that: i) $\left\{V(\mathbf{x}^k)\right\}$ converges to $V(\mathbf{x}^\star) \in \mathbb{R}$; ii) the sequences $\{V(\mathbf{x}^k)\}$, $\{\mathbf{x}^k\}$ are bounded; and iii) (2.20) holds.

Thanks to the continuity of $\widehat{\mathbf{x}}(\cdot)$ (see Proposition 2.2.1(b)) and Proposition 2.2.1(c), (2.20) implies that the limit point of $\{\mathbf{x}^k\}$ is a stationary point of (P).

From Lemma 2.2.3 we also have:

$$\left(\tau - \frac{L}{2}\right) \|\Delta\widehat{\mathbf{x}}^k\|_2^2 \leq \left(V(\mathbf{x}^k) - V(\mathbf{x}^\star)\right) - \left(V(\mathbf{x}^{k+1}) - V(\mathbf{x}^\star)\right). \tag{2.27}$$

Note that for a convex function $\phi(u) : [0; +\infty) \to [0; +\infty), \phi(u) \triangleq u^{\frac{1}{1-\theta}}$, with $\theta \in [0, 1)$, the following is true for any $y, z \in [0; +\infty)$:

$$\phi(y) - \phi(z) \leq \phi'(y)(y - z), \tag{2.28}$$

which, after a change of variables, can be rewritten as

$$y - z \leq (1 - \theta)^{-1} y^\theta \left(y^{1-\theta} - z^{1-\theta}\right). \tag{2.29}$$

By choosing $y = V(\mathbf{x}^k) - V(\mathbf{x}^\star)$ and $z = V(\mathbf{x}^{k+1}) - V(\mathbf{x}^\star)$, (2.29) gives us:

$$(V(\mathbf{x}^k) - V(\mathbf{x}^\star)) - (V(\mathbf{x}^{k+1}) - V(\mathbf{x}^\star)) \leq (1 - \theta)^{-1}(V(\mathbf{x}^k) - V(\mathbf{x}^\star))^\theta \Delta^k, \tag{2.30}$$

with $\Delta^k \triangleq (V(\mathbf{x}^k) - V(\mathbf{x}^\star))^{1-\theta} - (V(\mathbf{x}^{k+1}) - V(\mathbf{x}^\star))^{1-\theta}$. Combining this latter result with (2.27) we get

$$\|\Delta\widehat{\mathbf{x}}^k\|_2^2 \leq \left(\tau - \frac{L}{2}\right)^{-1} (1 - \theta)^{-1}(V(\mathbf{x}^k) - V(\mathbf{x}^\star))^\theta \Delta^k.$$

Due to the KL inequality (2.24) and the fact that $\{V(\mathbf{x}^k)\}$ is converging, we also have:

$$\|\Delta\widehat{\mathbf{x}}^k\|_2^2 \leq \left(\tau - \frac{L}{2}\right)^{-1} (1-\theta)^{-1} c^{-1} \|\mathbf{v}^k\|_2 \Delta^k,$$

where $\mathbf{v}^k \in \partial V(\mathbf{x}^k)$. From Lemma 2.2.5 we also have:

$$\|\Delta\widehat{\mathbf{x}}^k\|_2^2 \leq \left(\tau - \frac{L}{2}\right)^{-1} (1-\theta)^{-1} c^{-1} \sqrt{N} L_A \|\Delta\widehat{\mathbf{x}}^{k-1}\|_2 \Delta^k. \tag{2.31}$$

For any two scalars $y, z \geq 0$ the following relationships is true: $\sqrt{yz} \leq \frac{1}{2}(y+z)$, so we get from (2.31):

$$\|\Delta\widehat{\mathbf{x}}^k\|_2 \leq \frac{1}{2}\left(\tau - \frac{L}{2}\right)^{-1} (1-\theta)^{-1} c^{-1} \sqrt{N} L_A \Delta^k + \frac{1}{2}\|\Delta\widehat{\mathbf{x}}^{k-1}\|_2. \tag{2.32}$$

According to:

$$\sum_{k=\bar{k}}^{+\infty} \Delta^k = \left(V(\mathbf{x}^{\bar{k}}) - V(\mathbf{x}^\star)\right)^{1-\theta}, \tag{2.33}$$

$\{\Delta^k\}$ is a summable sequence, and, due to (2.32), $\{\Delta\widehat{\mathbf{x}}^k\}$ is a summable sequence too. This implies that $\{\mathbf{x}^k\}$ is a Cauchy sequence. ■

Theorem 2.2.8 (Convergence rate) *Let $\{\mathbf{x}^k\}$ be the sequence generated by the FLEXA algorithm, under Assumptions A-B. Suppose further that V is coercive and satisfies the KL property (2.24) with exponent θ. Set $\mathcal{S}^k \equiv \mathcal{N}$ and $\gamma^k = 1$ for any $k \geq 0$. Consider a stationary point $\mathbf{x}^\star$ of Problem (P) and define $V^\star \triangleq V(\mathbf{x}^\star)$. If $\tau > \frac{L}{2}$, for k sufficiently large, there hold:*

(i) For $\theta \in (0; \frac{1}{2}]$

$$\|\mathbf{x}^k - \mathbf{x}^\star\|_2 \leq C_1 \lambda^k, \tag{2.34}$$

$$V(\mathbf{x}^k) - V^\star \leq C_2 \lambda^{\frac{k}{\theta}}; \tag{2.35}$$

with

$$C_1 \triangleq \sum_{t=0}^{+\infty} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2 < +\infty, \tag{2.36}$$

$$\lambda \triangleq \frac{1 + \left(\tau - \frac{L}{2}\right)^{-1} (1-\theta)^{-1} c^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} L_A^{\frac{1}{\theta}}}{2 + \left(\tau - \frac{L}{2}\right)^{-1} (1-\theta)^{-1} c^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} L_A^{\frac{1}{\theta}}}, \tag{2.37}$$

and

$$C_2 \triangleq \lambda^{-\frac{1}{\theta}} C_1 c^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} L_A^{\frac{1}{\theta}}. \tag{2.38}$$

(ii) For $\theta \in (\frac{1}{2}; 1)$

$$\|\mathbf{x}^k - \mathbf{x}^\star\|_2 \leq C_3 k^{-\frac{1-\theta}{2\theta-1}}, \tag{2.39}$$

$$V(\mathbf{x}^k) - V^\star \leq C_4 (k-1)^{-\frac{1-\theta}{\theta(2\theta-1)}}; \tag{2.40}$$

with $C_3 > 0$*, and*

$$C_4 \triangleq C_3 c^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} L_A^{\frac{1}{\theta}}. \tag{2.41}$$

(iii) For $\theta = 0$ $\{\mathbf{x}^k\}$ *converges in a finite number of steps.*

Proof For any $k \geq 0$ the following hold:

$$\|\mathbf{x}^k - \mathbf{x}^\star\|_2 = \|\sum_{t=k}^{+\infty} (\mathbf{x}^{t+1} - \mathbf{x}^t)\|_2 \leq \sum_{t=k}^{+\infty} \|\Delta\widehat{\mathbf{x}}^t\| \triangleq \Lambda^k. \tag{2.42}$$

Note that:

$$\begin{aligned}
&\Lambda^k \\
&\overset{(2.32)}{\leq} \sum_{t=k}^{+\infty} \left(\frac{1}{2} \left(\tau - \frac{L}{2} \right)^{-1} (1-\theta)^{-1} c^{-1} \sqrt{N} L_A \Delta^t + \frac{1}{2} \|\Delta\widehat{\mathbf{x}}^{t-1}\|_2 \right) \\
&\overset{(2.33)}{\leq} \sum_{t=k}^{+\infty} \frac{1}{2} \|\Delta\widehat{\mathbf{x}}^{t-1}\|_2 + \frac{1}{2} \left(\tau - \frac{L}{2} \right)^{-1} (1-\theta)^{-1} c^{-1} \sqrt{N} L_A \left(V(\mathbf{x}^k) - V^\star \right)^{1-\theta};
\end{aligned}$$

and consequently:

$$\Lambda^k \leq \Lambda^{k-1} - \Lambda^k + \left(\tau - \frac{L}{2} \right)^{-1} (1-\theta)^{-1} c^{-1} \sqrt{N} \tilde{L} \left(V(\mathbf{x}^k) - V^\star \right)^{1-\theta}. \tag{2.43}$$

From Lemma 2.2.5 and the KL inequality (2.24) we know that:

$$c \left(V(\mathbf{x}^k) - V^\star \right)^\theta \leq N^{\frac{1}{2}} L_A \|\Delta\widehat{\mathbf{x}}^{k-1}\|_2,$$

which implies:

$$\left(V(\mathbf{x}^k) - V^\star \right)^{1-\theta} \leq c^{-\frac{1-\theta}{\theta}} N^{\frac{1-\theta}{2\theta}} L_A^{\frac{1-\theta}{\theta}} \|\Delta\widehat{\mathbf{x}}^{k-1}\|_2^{\frac{1-\theta}{\theta}}. \tag{2.44}$$

Combining this latter result with (2.43), we finally get:

$$\Lambda^k \leq \Lambda^{k-1} - \Lambda^k + \left(\tilde{\tau} - \frac{L}{2} \right)^{-1} (1-\theta)^{-1} \kappa^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} \tilde{L}^{\frac{1}{\theta}} \|\Delta\widehat{\mathbf{x}}^{k-1}\|_2^{\frac{1-\theta}{\theta}}. \tag{2.45}$$

From the above result, Lemma 2.2.6 and (2.42), the convergence rates of $\{\mathbf{x}^k\}$ stated in (i) and (ii) easily follow.

Furthermore, we also have

$$\begin{aligned} & V(\mathbf{x}^k) - V^\star \\ & \overset{(2.44)}{\leq} c^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} L^{\frac{1}{\theta}} \|\Delta\widehat{\mathbf{x}}^{k-1}\|_2^{\frac{1}{\theta}} \\ & = c^{-\frac{1}{\theta}} N^{\frac{1}{2\theta}} L^{\frac{1}{\theta}} (\Lambda^{k-1})^{\frac{1}{\theta}}. \end{aligned} \tag{2.46}$$

From the above result, (2.44) and Lemma 2.2.6 the convergence rates of $\{V(\mathbf{x}^k)\}$ in (i) and (ii) easily follow.

Let us now consider the case $\theta = 0$. Let us define the set $I \triangleq \{t \geq 0 : \mathbf{x}^{t+1} \neq \mathbf{x}^t\}$ and let k be in I. We have:

$$\begin{aligned} & V(\mathbf{x}^{k+1}) \\ & \overset{(2.19)}{\leq} V(\mathbf{x}^k) - \left(\frac{L}{2} - \tau\right) \|\Delta\widehat{\mathbf{x}}^k\|_2^2 \\ & \overset{(a)}{\leq} V(\mathbf{x}^k) - NL_A^2 \left(\frac{L}{2} - \tau\right) \|\mathbf{v}^{k+1}\|_2^2 \\ & \overset{(2.24)}{\leq} V(\mathbf{x}^k) - c^2 NL_A^2 \left(\frac{L}{2} - \tau\right), \end{aligned} \tag{2.47}$$

where in (a) we used Lemma 2.2.5 and we defined $\mathbf{v}^{k+1} \in \partial V(\mathbf{x}^{k+1})$. Since we already proved that $\{V(\mathbf{x}^k)\}$ is converging to $V^\star$, (iii) follows. ■

2.3 Case study: MRI

In this Section we apply FLEXA to a problem of practical interest: the MRI reconstruction problem. Specifically, we show how to reconstruct a sequence of images of a cardiac cycle from the noisy measurements collected by a MR scanner.

This Section is organized as follows: in Section 2.3.1 we first briefly review [18] the physics behind the MRI procedure and we present a formulation of the MRI reconstruction problem where FLEXA can be applied. In Section 2.3.2 we explain in detail how to design an instance of FLEXA for the MRI reconstruction problem. Finally, in Section 3.38 we show experimental results and comparisons, obtained by running FLEXA and other state-of-the-art schemes on a multi-core computing architecture.

2.3.1 Problem Description

The main operating principle behind the MRI procedure is that the MR scanner generates a large magnetic field $\mathbf{b}(\mathbf{r},t) = b_0(\mathbf{r},t)\widehat{\mathbf{z}}$ in order to induce a net magnetization $\mathbf{m}(\mathbf{r},t) \triangleq m_x(\mathbf{r},t)\widehat{\mathbf{x}} + m_y(\mathbf{r},t)\widehat{\mathbf{y}} + m_z(\mathbf{r},t)\widehat{\mathbf{z}}$ on the points of the body that we are interested to represent; here $\mathbf{r} = (x\widehat{\mathbf{x}}, y\widehat{\mathbf{y}}, z\widehat{\mathbf{z}})$ denotes the Cartesian coordinates in $\mathbb{R}^3$, $\widehat{\mathbf{x}}\widehat{\mathbf{y}}\widehat{\mathbf{z}}$ are the axis unit vectors, t denotes the time dependency, while $b_0(\mathbf{r})$ is the field strength generated by the MR scanner. In principle $b_0(\mathbf{r})$ should be a uniform field, but in practice it is spatially variable due to inhomogeneities and non-uniformities. During the scan the magnetic field $\mathbf{b}(\mathbf{r},t)$ is varied in time from the user in order to induce specific time-varying perturbations in the magnetization $\mathbf{m}(\mathbf{r},t)$. The effects of this time-varying magnetic field $\mathbf{b}(\mathbf{r},t)$ on the magnetization $\mathbf{m}(\mathbf{r},t)$ will be different according to the tissue under exam and according to what the user is interested to represent. The goal is to collect the values of the magnetization $\mathbf{m}(\mathbf{r},t)$ and reconstruct an image from it.
The MRI procedure is basically composed of two steps: excitation and readout. These two steps are periodically repeated, because, usually, the region of interest is too big to be scanned with a single excitation-readout step, so at each repetition the machine collects samples from a different slice of the object.
During the excitation step the time varying magnetic field $\mathbf{b}(\mathbf{r},t)$ is applied in order to bend the magnetization vector $\mathbf{m}(\mathbf{r},t)$ into the transversal (x,y) plane. It is in this step that the position of the scanned slice is selected and changed. It is convenient to describe the transverse magnetization as a complex function:

$$\bar{\mathbf{m}}(\mathbf{r},t) = m_x(\mathbf{r},t) + j m_y(\mathbf{r},t). \tag{2.48}$$

During the readout step the magnetic field $\mathbf{b}(\mathbf{r}, t)$ is manipulated in order to generate the desired effects on the transverse magnetization $\bar{\mathbf{m}}(\mathbf{r}, t)$, according to what is the specific image that we are interested to obtain. Only the longitudinal component of the magnetic field is usually changed, so it is possible to write $\mathbf{b}(\mathbf{r}, t)$ as: $\mathbf{b}(\mathbf{r}, t) = b_z(\mathbf{r}, t)\widehat{\mathbf{z}}$.

The relationship between $\bar{\mathbf{m}}(\mathbf{r}, t)$ and $\mathbf{b}(\mathbf{r}, t)$ has been characterized in detail by the Bloch equation [19], but for our purposes we can rely on some commonly used simplification. The Bloch equation states that the transverse magnetization $\bar{\mathbf{m}}(\mathbf{r}, t)$ will precess around the axis of the applied field with the so-called Larmor frequency $\omega(\mathbf{r}, t)$

$$\omega(\mathbf{r}, t) = \gamma b_z(\mathbf{r}, t), \tag{2.49}$$

where γ is called gyromagnetic ratio and its value for hydrogen protons (which are preponderant in the human body) is about 42.6 MHz/T. If we set $t = 0$ as the starting time of the readout step, it is possible to describe evolution of the transverse magnetization as

$$\begin{aligned} \bar{\mathbf{m}}(\mathbf{r}, t) &= \bar{\mathbf{m}}(\mathbf{r}, 0) e^{-\frac{t}{T(\mathbf{r})}} \exp\left(-j \int_0^t \omega(\mathbf{r}, t')\mathrm{d}t'\right) \\ &= f(\mathbf{r}) e^{-\frac{t}{T(\mathbf{r})}} \exp\left(-j\gamma \int_0^t b_z(\mathbf{r}, t')\mathrm{d}t'\right), \end{aligned} \tag{2.50}$$

where $f(\mathbf{r}) \triangleq \bar{\mathbf{m}}(\mathbf{r}, 0)$ is the value of the transverse magnetization at the start of the readout step, and $T(\mathbf{r})$ is a known decay factor that depends on the kind of tissue being scanned. $T(\mathbf{r})$ varies spatially, but usually can be approximated as a constant of the order of about 10 ms. Equation (2.50) is composed by three factors: the term $e^{-\frac{t}{T(\mathbf{r})}}$ it is a decaying factor that describes as the transverse magnetization disappears as the time passes; the other exponential factor describes how the phase of the transverse magnetization changes according to its instantaneous frequency (2.49); finally the term $f(\mathbf{r})$ is what we are interested in. $f(\mathbf{r})$ depends on the features of the tissue being scanned, so we are interested in obtaining an image of it. It is worth to be noticed that in this work it is assumed that $f(\mathbf{r})$ is not time-varying, but in some recent applications also this case is considered.

According to the Faraday's law of induction, the time-varying transverse magnetization will generate an electromotive force in a coil which is inserted in the MRI device to collect this signal. This potential $v(t)$ may be written as

$$v(t) = \text{real}\left(\int_{\mathbf{r} \in \text{body}} c(\mathbf{r}) \frac{\partial}{\partial t} \bar{\mathbf{m}}(\mathbf{r}, t) \mathrm{d}\mathbf{r} \right), \tag{2.51}$$

where $\int_{\mathbf{r}\in\text{body}}(\cdot)$ is a volume integral on the body being scanned, real$(\cdot)$ is the real part of a complex number and $c(\mathbf{r})$ is a function that describes the coil response pattern. It will be assumed in the following that only one coil will be used, even if it is possible to generalize this process to the case of multiple coils. Given that $T(\mathbf{r})$ is in the order of milliseconds, while the frequency variations in (2.50) are many MHz, it is possible [18] to do a narrow-band approximation of the time derivative in (2.51) as: $\frac{\partial}{\partial t}\bar{\mathbf{m}}(\mathbf{r}, t) \approx c_0 \bar{\mathbf{m}}(\mathbf{r}, t)$. c_0 is a constant that it is possible to absorb into $c(\mathbf{r})$; furthermore it will be assumed that the overall coil response pattern is uniform, i.e. $c(\mathbf{r}) = 1$. So, after all these simplifications equation (2.51) becomes

$$v(t) = \text{real}\left(\int_{\mathbf{r} \in \text{body}} \bar{\mathbf{m}}(\mathbf{r}, t) \mathrm{d}\mathbf{r} \right). \tag{2.52}$$

The potential $v(t)$ is then demodulated with center frequency $\omega_0 = \gamma b_0$ in order to obtain the signal

$$s(t) = \int_{\mathbf{r} \in \text{body}} f(\mathbf{r}) e^{-\frac{t}{T(\mathbf{r})}} e^{-j\phi(\mathbf{r},t)} \mathrm{d}\mathbf{r}, \tag{2.53}$$

where $\phi(\mathbf{r}, t)$ is the space- and time-varying phase

$$\phi(\mathbf{r}, t) = \int_0^t (\gamma b_z(\mathbf{r}, t') - \omega_0) \mathrm{d}t'. \tag{2.54}$$

Signal (2.53) is then sampled at different sampling times t_i's, $i = 1, \ldots, m$, evenly spaced, in order to obtain the data

$$y_i = s(t_i) + \epsilon_i \qquad i = 1, \ldots, m; \tag{2.55}$$

where m denotes the total number of samples, and ϵ_i represents the measurement error, which is commonly described as complex additive white Gaussian noise (AWGN) noise.

In order to reconstruct the function $f(\mathbf{r})$ from the samples $\mathbf{y} \triangleq [y_i]_{i=1}^m$, $f(\mathbf{r})$ will be represented through a finite series expansion, according to

$$f(\mathbf{r}) = \sum_{j=1}^{n} x_j l(\mathbf{r} - \mathbf{r}_j), \tag{2.56}$$

where $l(\cdot)$ is the object basis function, $\mathbf{r}_j$ is the position in which the jth basis function is centered, and n is the total number of coefficients x_j's used for the series expansion. One common choice, the one that will be considered here, is to use rectangular basis functions

$$l(\mathbf{r}) = \mathrm{rect}(\mathbf{r}/\Delta), \tag{2.57}$$

where Δ is the dimension of a square pixel. According to this series expansion it is possible to rewrite the signal (2.53) in a discrete way as

$$s(t_i) = \sum_{j=1}^{n} A_{ij}\tilde{x}_j, \tag{2.58}$$

with:

$$A_{ij} = \int l(\mathbf{r} - \mathbf{r}_j) e^{-\frac{t_i}{T(\mathbf{r})}} e^{-j\phi(\mathbf{r},t_i)} \mathrm{d}\mathbf{r}. \tag{2.59}$$

In order to simplify the mathematical formulation, equation (2.59) is usually approximated as

$$A_{ij} \approx \Delta e^{-\frac{t_i}{T(\mathbf{r})}} e^{-j\phi(\mathbf{r},t_i)}, \tag{2.60}$$

which is the value at the center of the jth pixel. By denoting with $\mathbf{A}$ the $m \times n$ matrix which contains all the elements A_{ij}'s (i is the row index and j the column one), and by putting together equations (2.55) and (2.58), the sampled signal can be represented in a compact form as

$$\mathbf{y} = \mathbf{A}\tilde{\mathbf{x}} + \boldsymbol{\epsilon}, \tag{2.61}$$

where $\boldsymbol{\epsilon} \triangleq [\epsilon_i]_{i=1}^n$ is the Gaussian noise vector, and $\tilde{\mathbf{x}} \triangleq [\tilde{x}_i]_{i=1}^n$ is the image we want to reconstruct from the measurement vector $\mathbf{y}$.

It is possible to investigate in more detail the nature of the elements A_{ij}'s in (2.60) in order to understand better the structure of $\mathbf{A}$. First of all, it is possible to assume that the total readout time $t_n - t_1$ is small with respect to $T(\mathbf{r})$: this allows to make the approximation $e^{-\frac{t_i}{T(\mathbf{r})}} \approx e^{-\frac{t_1}{T}}$. Thus, it is possible to absorb the resulting constant term $e^{-\frac{t_1}{T}}$ directly into the image $f(\mathbf{r})$. For what concerns the time-varying magnetic field $\mathbf{b}(\mathbf{r}, t)$ during the readout step, it is possible to represent it as composed by two main terms:

$$\mathbf{b}(\mathbf{r}, t) = b_z(\mathbf{r}, t) = b_0 + \mathbf{g}(t) \cdot \mathbf{r}, \tag{2.62}$$

where $\mathbf{g}(t) \triangleq g_x(t)\widehat{\mathbf{x}} + g_y(t)\widehat{\mathbf{y}} + g_z(t)\widehat{\mathbf{z}}$ is the vector of the field gradients. The field gradients are three functions, controlled by the user that permits to modify the magnetic field in order to get specific results. By substituting (2.62) into (2.54), the phase becomes:

$$\phi(\mathbf{r}, t) = \int_0^t \gamma \mathbf{g}(t) \cdot \mathbf{r} \mathrm{d}t. \tag{2.63}$$

According to this, it is possible to rewrite the elements of the matrix $\mathbf{A}$ in (2.60) as:

$$A_{ij} \approx e^{-j2\pi \mathbf{k}(t_i) \cdot \mathbf{r}_j}, \tag{2.64}$$

where $\mathbf{k}(t)$ is the so-called k-space trajectory:

$$\mathbf{k}(t) \triangleq \frac{1}{2\pi} \int_0^t \gamma \mathbf{g}(t) \mathrm{d}t. \tag{2.65}$$

If this trajectory is opportunely chosen, thanks to the gradient field vector $\mathbf{g}(t)$, then the $\mathbf{A}$ matrix becomes a FFT matrix. From a practical point of view, it is fast, easy, and efficient to perform FFTs on the data vectors.

According to the model (2.61), the MRI reconstruction problem may be casted as a regularized least-squares optimization problem:

$$\min_{\tilde{\mathbf{x}}} \|\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}\|_2^2 + \lambda R(\tilde{\mathbf{x}}), \tag{2.66}$$

where $R(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a regularizer, and $\lambda > 0$ is the regularizing parameter. Different type of regularizing functions have been proposed in order to deal with different kinds of situations that may arise in the context of MRI (undersampling, artifacts remotion, a priori knowledge etc.).

In the experiment showed in this Chapter we are interested in reconstructing not only a single image, but a series of images from a cardiac cycle. Consequently, $\mathbf{y}$ and $\tilde{\mathbf{x}}$ will denote in the following the stacked measurement vector and the stacked discretized image content from all the different images to be reconstructed, respectively.
Given that the sequence of images to reconstruct comes from a cardiac cycle, it is possible to assume a sort of time-periodicity in this representation. It is known that a signal which has some periodicity in time is somehow sparse in the frequency domain. For this reason it is useful to choose a regularizer $R(\cdot)$ which imposes some sparsity in the optimal solution. As proposed in [5], the renowned l_1 norm is a function which meets this criterion and, being, convex it is easy to deal with. According to this, the problem (2.66) may be reformulated as:

$$\min_{\tilde{\mathbf{x}}} \|\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}\|_2^2 + \lambda \|\mathbf{F}_t \tilde{\mathbf{x}}\|_1, \tag{2.67}$$

where $\mathbf{F}_t$ denotes a Fourier transformation with respect to the time-direction. In order to obtain a more canonical representation of problem (2.67) it is convenient apply the change of variable $\mathbf{x} \triangleq \mathbf{F}_t \tilde{\mathbf{x}}$ in order to obtain the well-known LASSO formulation:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{F}_t^{-1}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1, \tag{2.68}$$

where $\mathbf{F}_t^{-1}$ is the inverse Fourier transform with respect to the frequency domain.

2.3.2 Solution algorithm

It is easy to recognize that the convex problem (2.68) falls into the framework (P) with $F(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{F}_t^{-1}\mathbf{x}\|_2^2$, $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$, $g_i(\mathbf{x}_i) = \lambda\|\mathbf{x}_i\|_1$, and $\mathcal{X} = \mathbb{R}^n$ (see Example #1 in Section 2.1). In this Section we explain in detail how to tailor FLEXA to efficiently solve problem (2.68).

• **Computing architecture -** Let us consider that we want to solve problem (2.68) on a multi-core computing architecture with P cores that can work in parallel. Then, for the specific formulation of problem (2.68), it is convenient to split the data matrix $\tilde{\mathbf{A}} \triangleq \mathbf{A}\mathbf{F}_t^{-1}$ among the P cores, so that each core has its own matrix $\tilde{\mathbf{A}}_i$, where $\tilde{\mathbf{A}} \triangleq \left[\tilde{\mathbf{A}}_1 \ldots \tilde{\mathbf{A}}_i \ldots \tilde{\mathbf{A}}_P\right]$, and each $\tilde{\mathbf{A}}_i$ has n/P columns. In this way, common operations as computing the product $\tilde{\mathbf{A}}\mathbf{x}$ (resp. the l_1 norm $\|\mathbf{x}\|_1$) may be performed in parallel by computing separately $\tilde{\mathbf{A}}_i\mathbf{x}_i$ ($\|\mathbf{x}_i\|_1$) and then summing up the partial terms coming from the different cores.

We want to remark that, as we already noticed in Section 2.3.1, $\mathbf{A}$ is an FFT matrix. So, all the matrix-matrix or matrix-vector products involving this matrix can be performed efficiently in C++, using the specific libraries for signal processing.

• **Surrogate functions -** Let us denote with k the iteration index. We will use the following structure for the surrogate functions $\tilde{F}_i$'s (see also Example 1) in Section 2.2): for a given $\mathbf{x}^k \in \mathbb{R}^n$ and $i = 1, \ldots, n$

$$\tilde{F}_i(\mathbf{x}_i, \mathbf{x}^k) = P_i(\mathbf{x}_i; \mathbf{x}^k) + \frac{\tau_i^k}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_2^2 + g_i(\mathbf{x}_i), \tag{2.69}$$

where $P_i(\cdot; \mathbf{x}^k)$ must be a convex function satisfying B2-B3, while $\tau_i^k > 0$, are proximal parameters, tunable by the user in order to improve the speed of the algorithm.

An efficient choice for the surrogate functions is to set $n_i = 1$ and consider $P_i(x_i; \mathbf{x}^k) = F(x_i, \mathbf{x}_{-i}^k)$ and $g_i(x_i) =$. In words, this means that each subproblem i is a scalar instance of problem (2.68)), where the minimization is performed with respect to a scalar variable x_i only, assuming the other variables $\mathbf{x}_{-i}$ to be fixed at the current value $\mathbf{x}_{-i}^k$.

• **Solving the subproblems -** It is well known that the minimization of $\tilde{F}_i$, having the structure described above, can be computed in closed form through the so-called soft thresholding operator [20] $\mathcal{S}_\alpha : \mathbb{R} \to \mathbb{R}$

$$\mathcal{S}_\alpha(x) \triangleq (|x| - \alpha)^+ \text{sign}(x), \tag{2.70}$$

where $\alpha \in \mathbb{R}$, and, for any $z \in \mathbb{R}$, $(z)^+ = \max(0, z)$, and

$$\operatorname{sign}(z) \triangleq \begin{cases} -1 & \text{if } z < 0; \\ 0 & \text{if } z = 0; \\ 1 & \text{if } z > 0. \end{cases} \tag{2.71}$$

Specifically, for any given $\mathbf{x}^k \in \mathbb{R}^n$, the solution $\widehat{x}_i(\mathbf{x}^k)$ of each subproblem becomes

$$\begin{aligned} \widehat{x}_i(\mathbf{x}^k) &= \underset{x_i \in \mathbb{R}}{\arg\min} \left\{ P_i(x_i; \mathbf{x}^k) + \frac{\tau_i^k}{2} \|x_i - x_i^k\|_2^2 + g_i(x_i) \right\} \\ &= \underset{x_i \in \mathbb{R}}{\arg\min} \left\{ F(x_i, \mathbf{x}_{-i}^k) + \frac{\tau_i^k}{2} \|x_i - x_i^k\|_2^2 + g_i(x_i) \right\} \\ &= \underset{x_i \in \mathbb{R}}{\arg\min} \left\{ \|\mathbf{y} - \sum_{j \neq i} \tilde{\mathbf{a}}_j x_j^k - \tilde{\mathbf{a}}_i x_i\|_2^2 + \frac{\tau_i^k}{2} \|x_i - x_i^k\|_2^2 + \lambda |x_i| \right\} \\ &= \mathcal{S}_{\frac{\lambda}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k}} \left(x_i^k - \frac{\nabla_{x_i} F(\mathbf{x}^k)}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k} \right) \\ &= \begin{cases} x_i^k - \frac{\nabla_{x_i} F(\mathbf{x}^k) - \lambda}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k}, & \text{if} \quad x_i^k - \frac{\nabla_{x_i} F(\mathbf{x}^k)}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k} \geq \frac{\lambda}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k}; \\ x_i^k - \frac{\nabla_{x_i} F(\mathbf{x}^k) + \lambda}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k}, & \text{if} \quad x_i^k - \frac{\nabla_{x_i} F(\mathbf{x}^k)}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k} \leq -\frac{\lambda}{2\|\tilde{\mathbf{a}}_i\|_2^2 + \tau_i^k}; \\ \quad 0 \quad , & \text{otherwise}; \end{cases} \end{aligned} \tag{2.72}$$

where $\tilde{\mathbf{a}}_i$, $i = 1, \ldots, n$, is the ith column of $\tilde{\mathbf{A}}$ and

$$\nabla_{x_i} F(\mathbf{x}^k) = 2x_i^k \|\tilde{\mathbf{a}}_i\|_2^2 + 2 \sum_{j \neq i} x_j \langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{a}}_j \rangle + 2 \sum_{j=1}^{m} y_j \tilde{a}_{ji}. \tag{2.73}$$

• **Tunable parameters -** For the stepsize sequence $\{\gamma^k\}$, we used the following rule, similar to (2.12), which satisfies D2 and works well in practice:

$$\gamma^{k+1} = \gamma^k \left(1 - \min\left\{ 1, \frac{10^{-4}}{\|Z(\mathbf{x}^k)\|_\infty} \right\} \theta_\gamma \gamma^k \right), \tag{2.74}$$

with $\gamma^0 = 0.9$, $\theta_\gamma = 10^{-7}$, and $Z : \mathbb{R}^n \to \mathbb{R}^n$ defined as: for any $\mathbf{x} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$

$$Z(\mathbf{x}) \triangleq \nabla_{\mathbf{x}} F(\mathbf{x}) - \prod_{[-\alpha,\alpha]^n} (\nabla_{\mathbf{x}} F(\mathbf{x}) - \mathbf{x}) \tag{2.75}$$

where $\prod_{[-\alpha,\alpha]^n}(\cdot)$ is the Euclidean projection operator. Note that i) Z is a continuous function which is 0 if and only if its argument is a stationary point of problem

(2.68), so $\|Z(\mathbf{x}^k)\|_\infty$ is a valid measure of stationarity; and ii) the Euclidean projection $\prod_{[-\alpha,\alpha]^n}(\mathbf{z})$, for any $\mathbf{x} \in \mathbb{R}^n$, can be easily computed by noticing that it acts componentwise on $\mathbf{z}$, given that $[-\alpha, \alpha]^n = [-\alpha, \alpha] \times \ldots \times [-\alpha, \alpha]$.

For the proximal gains sequence $\{\tau_i^k\}$ we used the following heuristic: each τ_i is initialized as the half of the mean of the eigenvalues of $\nabla^2_{\mathbf{xx}} F(\mathbf{x})$. Then, each τ^i is doubled if at a certain iteration the objective function does not decrease (the associated iteration is discarded if this happens), while they are all halved if the objective function decreases for ten consecutive iterations or $\|Z(\mathbf{x}^k)\|_\infty$ is less than 10^{-2}. See Table 2 for the pseudocode relative to this heuristic.

Algorithm 2: Heuristic for the update of the proximal gains τ_i

```
Data: $\tau_i^0 = \text{trace}\frac{(\mathbf{AF}_t^{-1})^T(\mathbf{AF}_t^{-1})}{2n}$   $\forall i,\ k, p \geq 0,\ \mathbf{x}^k \in \mathbb{R}^n$.
(S.1) If $V(\mathbf{x}^{k+1}) \geq V(\mathbf{x}^k)$
          then $\tau_i^{k+1} = 2\tau_i^k \quad \forall i$
          $p = 0$;
      Else if $(\|Z(\mathbf{x}^{k+1})\|_\infty \leq 10^{-2} \vee p == 9)$
          then $\tau_i^{k+1} = \tau_i^k/2 \quad \forall i$
          $p = 0$;
(S.2) Update $p \leftarrow p + 1$.
```

• **Block selection rule -** As already discussed in Section 2.2, in order to speed up the convergence rate of FLEXA, it is possible to implement a greedy selection of the variables to update at each iteration. More specifically, at each iteration k, not all the n variables are updated, but only those that are furthest from the optimal solution, according to some metric. The metric used in this work, which satisfies C1 and it is easy computable, is the distance of each variable x_i^k at the current iterate from the best-response (2.72) $\widehat{x}_i^k(\mathbf{x}^k)$.

The overall algorithm is showed in Table 3 below.

```
**Algorithm 3: FLEXA for MRI reconstruction**
Data: $\{\tau_i^k\}$ given in Table 2, for $i = 1, \ldots, n$, $\rho \in [0,1]$, $\mathbf{x}^0 \in \mathbb{R}^n$, $\{\gamma^k\}$ given in (2.74). Set $k = 0$.
(S.1) If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) Compute each $\widehat{x}_i(\mathbf{x}^k)$ according to (2.72);
(S.3) Compute the maximum distance $M^k \triangleq \max_{i \in \mathcal{N}}\{\widehat{x}_i(\mathbf{x}^k) - x_i^k\}$;
(S.4) Define the set $\mathcal{S}^k = \{i \,|\widehat{x}_i(\mathbf{x}^k) - x_i^k \geq \rho M^k\}$;
(S.5) Compute $\mathbf{x}^{k+1}$ according to:
          $x_i^{k+1} = x_i^k + \gamma^k \left(\widehat{x}_i(\mathbf{x}^k) - x_i^k\right)$        $i \in \mathcal{S}^k$
          $x_i^{k+1} = x_i^k$        $i \notin \mathcal{S}^k$
(S.6) Update $k \leftarrow k + 1$ and go to (S.1)
```

2.3.3 Experimental Results

All codes have been written in C++, and all algebra is performed by using the Intel Math Kernel Library (MKL). The algorithms were tested on the General Compute Cluster of the Center for Computational Research at the SUNY Buffalo. In particular, we used a partition composed of 372 DELL 16x2.20GHz Intel E5-2660 "Sandy Bridge" Xeon Processor computer nodes with 128 Gb of DDR4 main memory and QDR InfiniBand 40Gb/s network card.

• **Algorithms in the literature -** We compared the instance of FLEXA presented in Table 3 with the most competitive parallel and sequential algorithms proposed in the literature to solve the MRI reconstruction problem. More specifically, we consider the following schemes.

- k-t FOCUSS: k-t FOCUSS [21] is a sequential algorithm specifically developed in order to solve the MRI reconstruction problem. Key feature of the scheme is to use a the covariance matrix of the data in order to construct a weight matrix that tries to predict the sparsity of the solution in order to speed up the optimization procedure. I
- FISTA: The Fast Iterative Shrinkage-Thresholding algorithm (FISTA) proposed in [20] is a first order method and can be regarded as the benchmark algorithm for LASSO problems. Building on the separability of the terms in the objective function, this method can be easily parallelized and thus take advantage of a parallel architecture. We implemented the parallel version that use a backtracking procedure to estimate the Lipschitz constant of $\nabla_{\mathbf{x}} F$
- SpaRSA: This is the first order method proposed in [22]; it is a popular spectral projected gradient method that uses a spectral step length together with a nonmonotone line search to enhance convergence. Also this method can be easily parallelized, which is the version implemented in our tests. In the experiments, we set the parameters of SpaRSA as in [22]: $M = 5$, $\sigma = 0.01$, $\alpha_{\max} = 1e30$, and $\alpha_{\min} = 1e - 30$.
- GRock: GRock is a parallel algorithm proposed in [23] that performs well on sparse least-squares problems. We tested the instance of GRock where the number of variables simultaneously updated is equal to the number of the parallel processors. It is important to remark that the theoretical convergence properties of GRock are in jeopardy as the number of variables updated in parallel increase.

Parallel algorithms have been tested using 16 parallel processes (2 nodes with 8 cores per each), while k-t FOCUSS ran on a single process (using thus a single core).

• **Experiments -** The experiment consists in the reconstruction of a series of 25 images collected by a 1.5T Phillips MR scanner from a cardiac cycle (66 bpm). Each image is composed of 256×256 samples. The fully sampled data have been used as a reference. To make the problem harder, the data vector $\mathbf{y}$ has been generated by randomly undersampling the original data with a 2/3 factor. The performance will be showed in terms of the Normalized Mean Square Error (NMSE), which, at any iteration k, is defined as:

$$\text{NMSE} \triangleq \frac{\|\mathbf{x}^k - \mathbf{x}_{\text{true}}\|_2^2}{\|\mathbf{x}_{\text{true}}\|_2^2}, \tag{2.76}$$

where $\mathbf{x}_{\text{true}}$ is the vector containing the values of the pixels of the true image in the frequency domain.

Figure 2.1 shows the speed of convergence of the different algorithms in terms of NMMSE with respect to the CPU time. It may be seen that the proposed approach outperforms all the other schemes and it is worth to be noticed that GRock for this problem does not convergence. In particular, we want to highlight that FLEXA after

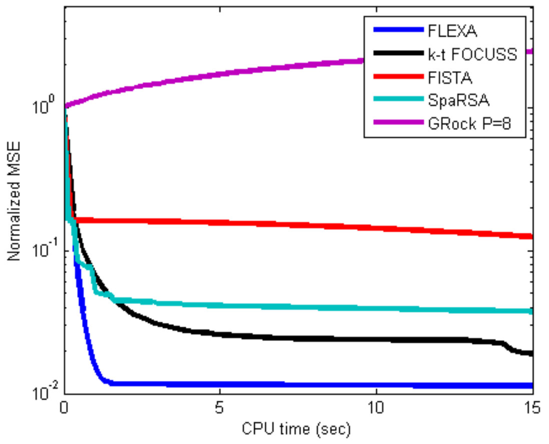

Figure 2.1. NMSE versus CPU time.

just one second reaches a value of the NMSE that other algorithms, as SpaRSA or k-t FOCUSS, reach only after tens or hundreds of seconds. Following this remark, in the next figures we show results obtained by stopping the algorithms after only one second of reconstruction process. Our aim is to show that the proposed approach, unlike the others, achieves very good results in such a short amount time that could easily lead to real-time implementation. Figure 2.2 shows the NMSE per each frame after 1 second of reconstruction process: FLEXA has the best performances in this case too.

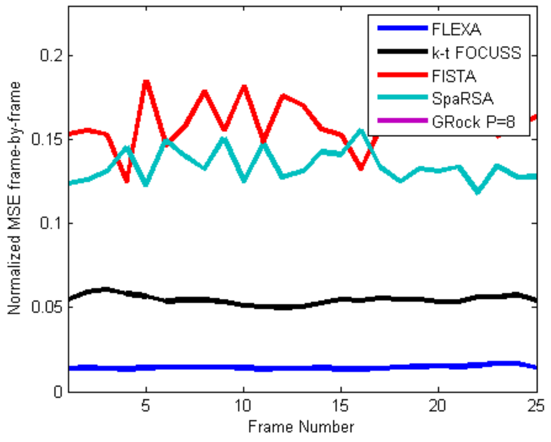

Figure 2.2. NMSE obtained on each frame after 1 second of processing.

Figure 2.7 shows the reconstruction of the 13th frame after 1 second of processing. It is very easy to see that FISTA and SpaRSA are still far from obtaining a good quality of the image, since their results appears to be blurred and foggy. k-t FOCUSS is the one that obtains a reconstruction quality which is near to the one reached by FLEXA, but the image reconstructed by FLEXA has however a higher level of detail and clearness.

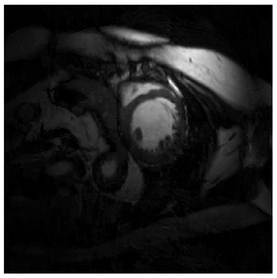
Figure 2.3. FLEXA

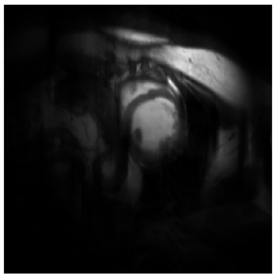
Figure 2.4. k-t FOCUSS

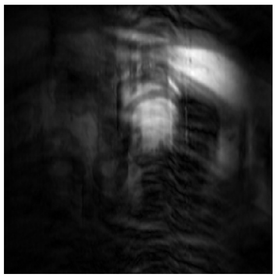
Figure 2.5. FISTA

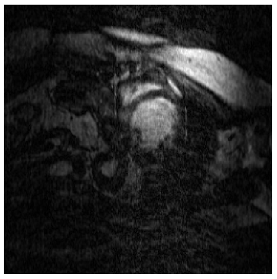
Figure 2.6. SpaRSA

Figure 2.7. Reconstruction of the 13th frame after 1 second of processing.

Looking at the entire frame sequences, they show that FLEXA captures the dynamics and the movements better than the other schemes. In order to highlight this feature, Figure 2.13 shows the so-called intensity profiles, obtained by the algorithms after

1 second of processing. This profiles show how the intensity values of one row of the images (here we chose the 100th row) varies frame-by frame. Figure 2.13 shows that FISTA and SpaRSA after 1 second are still far from having obtained a good reconstruction result; k-t FOCUSS obtains a good quality, but, unlike FLEXA, does not show any dynamic effect, which is instead present in the reference profile.

2.4 Conclusions

In this Chapter we first briefly reviewed FLEXA, a parallel algorithmic framework, based on SCA, for the solution of nonconvex and nonsmooth optimization problems. We then proved that FLEXA can achieve linear convergence rate when the optimization problem is strongly convex or, more interestingly, when it satisfies more general standard error bound conditions.

Finally, in the last part of this Chapter, we showed that the MRI reconstruction problem falls into the class of applications where FLEXA can be applied. So, we proposed a specific version of FLEXA tailored for the MRI problem and we showed that the performances of our proposed algorithm outperform those of state-of-the-art schemes.

2.5 Appendix: Error Bounds Review

As already mentioned in Section 2.2.2, many practical problems satisfy some error bound condition, and it has been proved that usually this is a sufficient requirement for first-order iterative schemes to convergence to stationary points of these problems at a linear rate. In this Section we present several error bound conditions, analogous to the KL property introduced in Section 2.2.2, we discuss equivalences and relationships among them, and we list classes of practical problems for which these error bounds hold true.

• **Notation -** For ease of presentation, we will assume $\mathcal{X} \equiv \mathbb{R}^n$, unless otherwise specified. In order to include the presence of constraints, then, we will assume that G is an extended value function, so it can express the presence of constraints by means of indicator functions.

In the following $\mathcal{X}^\star$ will denote the set of stationary points of problem (P).

Specifically, in this Dissertation we will focus on the so-called *Luo-Tseng error bound condition*, because in Chapter 4 we will be able to derive linear convergence rate of our proposed algorithmic framework for problem settings satisfying this condition.

• **Luo-Tseng error bound** - In the seminal work [24], Luo and Tseng are among the first to propose an error bound condition in order to analyze the convergence rate of first-order descent methods for nonconvex smooth optimization.

In the subsequent years the Luo-Tseng condition has been generalized [25] in order to characterize nonsmooth objective functions too.

By defining, for any convex function $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ and $\gamma > 0$, the renowned proximal operator $\text{Prox}_\phi^\gamma(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ in the following way:

$$\text{Prox}_\phi^\gamma(\mathbf{x}) \triangleq \underset{\mathbf{y}\in\mathbb{R}^n}{\arg\min} \left\{ \phi(\mathbf{y}) + \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}, \tag{2.77}$$

the Luo-Tseng error bound condition can be stated as

Definition 2.5.1 (Luo-Tseng global error bound [25]) *Under Assumption A, for every scalar η: $\inf_{\mathbf{x}\in\mathbb{R}^n} V(\mathbf{x}) \leq \eta \leq V(\mathbf{x})$, the Luo-Tseng global error bound condition holds for V if there exist scalars $\delta, \tau > 0$ such that*

$$dist(\mathbf{x}, \mathcal{X}^\star) \leq \tau \|\mathbf{x} - Prox_g^1(\mathbf{x} - \nabla_\mathbf{x} F(\mathbf{x}))\|_2,$$

whenever $V(\mathbf{x}) \leq \eta$ and $\|\mathbf{x} - Prox_G^1(\mathbf{x} - \nabla_\mathbf{x} F(\mathbf{x}))\|_2 \leq \delta$;

where, for a given subset $\mathcal{C} \subseteq \mathbb{R}^n$ and vector $\mathbf{c} \in \mathcal{C}$, we define $\text{dist}(\mathbf{c}, \mathcal{C}) \triangleq \inf\{\|\mathbf{c} - \mathbf{c}'\|_2 : \mathbf{c}' \in \mathcal{C}\}$.

In the following we will show that the Luo-Tseng error bound i) is satisfied for many problems of practical interest; ii) is equivalent (or it is strictly related) to many others well-studied error bound conditions, usually verified in practice.

Case 1: F, G convex

We start by considering Problem (P) under Assumptions A, and the additional assumption of the convexity of F.

Relationships and equivalences among definitions as Luo-Tseng error bound, Quadratic Growth (QG), KL inequality, Polyak-Łojasiewicz (PL) inequality, and metrical subregularity will be shown.

Note that all the structural conditions defined in the following have been used to prove linear convergence of iterative schemes.

• **QG and PL inequality -** In order to discuss the QG and PL properties, few definitions are in order.

Definition 2.5.2 (Restricted Secant Inequality (RSI) [26]) *A closed function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ satisfies the RSI if, for all $\mathbf{x} \in$ dom ϕ, and for a given $\kappa > 0$:*

$$dist^2(\mathbf{x}, \Phi^\star) \leq \kappa \left\langle \nabla_{\mathbf{x}} \phi(x), \mathbf{x} - \mathbf{x}^\star \right\rangle,$$

where $\mathbf{x}^\star = \underset{\mathbf{y} \in \Phi^\star}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2$, and $\Phi^\star$ is the set of minimizers of ϕ, assumed to be nonempty.

A function satisfying the RSI, is said to satisfy the Restricted Strong Convexity (RSC) property if it is convex too.

Definition 2.5.3 (Essential Strong Convexity (ESC) [26]) *A closed smooth function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ satisfies the ESC condition if, for all $\mathbf{x}, \mathbf{y} \in$ dom ϕ such that $\inf_{\mathbf{z} \in \Phi^\star} \|\mathbf{x} - \mathbf{z}\|_2 = \inf_{\mathbf{z} \in \Phi^\star} \|\mathbf{y} - \mathbf{z}\|_2$, and for a given $\kappa > 0$:*

$$\phi(\mathbf{y}) \geq \phi(\mathbf{x}) + \langle \nabla_{\mathbf{x}} \phi(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \kappa \|\mathbf{y} - \mathbf{x}\|_2^2,$$

where $\Phi^\star$ is the set of minimizers of ϕ, assumed to be nonempty.

Definition 2.5.4 (Weak Strong Convexity (WSC) [26]) *A closed smooth function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ satisfies the WSC condition if, for all $\mathbf{x} \in$ dom ϕ, and for a given $\kappa > 0$:*

$$\min_{\mathbf{x} \in \mathbb{R}^n} \phi(\mathbf{x}) \geq \phi(\mathbf{x}) + \langle \nabla_{\mathbf{x}} \phi(\mathbf{x}), \mathbf{x}^\star - \mathbf{x} \rangle + \kappa \, dist^2(\mathbf{x}, \Phi^\star),$$

where $\mathbf{x}^\star = \arg\min_{\mathbf{y} \in \phi^\star} \|\mathbf{x} - \mathbf{y}\|_2$, and $\Phi^\star$ is the set of minimizers of ϕ, assumed to be nonempty.

Definition 2.5.5 (PL inequality [26]) *A closed smooth function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ satisfies the PL inequality if, for all $\mathbf{x} \in$ dom ϕ and a given $\kappa > 0$:*

$$\phi(\mathbf{x}) - \min_{\mathbf{x} \in \mathbb{R}^n} \phi(\mathbf{x}) \leq \kappa \|\nabla_{\mathbf{x}} \phi(\mathbf{x})\|_2^2,$$

where $\mathbf{x}^\star = \arg\min_{\mathbf{y} \in \phi^\star} \|\mathbf{x} - \mathbf{y}\|_2$, and $\Phi^\star$ is the set of minimizers of ϕ, assumed to be nonempty.

Definition 2.5.6 (QG [26]) *A closed function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ satisfies the QG condition if, for all $\mathbf{x} \in$ dom ϕ, and for a given $\kappa > 0$:*

$$\kappa \, dist^2(\mathbf{x}, \Phi^\star) \leq \phi(\mathbf{x}) - \min_{\mathbf{x} \in \mathbb{R}^n} \phi,$$

where $\Phi^\star$ is the set of minimizers of ϕ, assumed to be nonempty.

Note that the above definitions are valid for a general smooth function ϕ (in fact, QG holds also for nonsmooth ϕ). The following Theorem instantiates connections among these conditions, when a convex smooth function is considered.

Proposition 2.5.1 (Equivalence under convexity and smoothness [26]) *Under Assumption A, if in addition F is convex and $G = 0$, the following hold:*

1. *Strong Convexity (SC) implies ESC;*
2. *ESC implies WSC;*
3. *WSC implies RSI;*
4. *RSI, PL inequality, QG, and the Luo-Tseng global error bound are all equivalent conditions.*

Other connections between QG and the Luo-Tseng error bound have been shown in [27] under firm convexity.

Definition 2.5.7 (Firm convexity [27]) *A closed convex function $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ is firmly convex relative to a vector $\mathbf{y} \in \mathbb{R}^n$ if the tilted function $\phi_{\mathbf{y}}(\mathbf{x}) \triangleq \phi(\mathbf{x}) - \langle \mathbf{y}, \mathbf{x} \rangle$ satisfies the following QG condition, for any compact set $\Phi \subset \mathbb{R}^n$:*

$$\kappa\, dist^2(\mathbf{x}, (\partial \phi_{\mathbf{y}})^{-1}(0)) + \inf_{\mathbf{x} \in \mathbb{R}^n} \phi_{\mathbf{y}}(\mathbf{x}) \leq \phi_{\mathbf{y}}(\mathbf{x}), \quad \forall\, \mathbf{x} \in \Phi,$$

for a suitable constant $\kappa > 0$.

A closed convex function ϕ is firmly convex if it is firmly convex for any $\mathbf{y} \in \mathbb{R}^n$, for which $(\partial \phi_{\mathbf{y}})^{-1}(0) \neq \{\emptyset\}$.

In the following, the Fenchel conjugate of a convex function $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ will be denoted as the closed convex function $\phi^\star : \mathbb{R}^n \to (-\infty; +\infty]$ defined by:

$$\phi^\star(y) = \sup_{\mathbf{x} \in \mathbb{R}^n} \left\{ \langle \mathbf{y}, \mathbf{x} \rangle - \phi(\mathbf{x}) \right\}.$$

In light of these definitions, the following statement is in order.

Proposition 2.5.2 (Firm convexity implies Luo-Tseng global error bound [27]) *Under Assumption A3-A4, assume furthermore that: F is C^1 (Lipschitzianity of $\nabla_{\mathbf{x}} F$ is not needed), $F = h(\mathbf{A}\mathbf{x})$, $h : \mathbb{R}^m \to \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and denoting as $\bar{\mathbf{y}}$ the optimal solution of the dual problem*

$$\sup_{\mathbf{y} \in \mathbb{R}^m} -\Phi(\mathbf{y}) \triangleq -F^\star(\mathbf{y}) - G^\star(-\mathbf{A}^T\mathbf{y});$$

then, the Luo-Tseng global error bound condition holds if the following conditions hold:

1. *(Compactness) $\mathcal{X}^\star$is bounded;*
2. *(a) or (b) holds:*
   - *(a) Dual nondegeneracy*
     $$\mathbf{0} \in \mathbf{A}^T ri\ (dom\ F^\star) + ri\ (dom\ G^\star),$$
     and dual strict complementarity
     $$\mathbf{0} \in ri\ (\Phi(\bar{\mathbf{y}}));$$
   - *(b) $\partial G^\star(-\mathbf{A}^T\bar{\mathbf{y}})$ and $\partial F^\star(\bar{\mathbf{y}})$ are polyhedral;*
3. *(QG of components) F is convex on dom F and firmly convex relative to $\bar{\mathbf{y}}$. G is firmly convex relative to $-\mathbf{A}^T\bar{\mathbf{y}}$.*

Assumption 1 in Proposition 2.5.2 can be dropped if 2-(b) holds and F and G are globally firmly convex.

• **KL property -** The KL property introduced in Section 2.2.2 is a specific instance of the Generalized KL property.

Definition 2.5.8 (Generalized KL property [16]) *A closed function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ has the Generalized KL property at $\bar{\mathbf{x}} \in$ dom $\partial\phi$ if there exist a neighborhood $\mathcal{C}$ of $\bar{\mathbf{x}}$, $\eta \in (0, +\infty]$, and a continuous concave function $\Psi : [0, \eta) \to \mathbb{R}_+$ with $\Psi(0) = 0$, such that:*

i) Ψ *is smooth on* $(0,\eta)$ *with* $\Psi' > 0$ *over* $(0,\eta)$*;*

ii) for all $\mathbf{x} \in \mathcal{C}$ *and* $\phi(\bar{\mathbf{x}}) < \phi(x) < \phi(\bar{\mathbf{x}}) + \eta$*:*

$$\Psi'\left(\phi(\mathbf{x}) - \phi(\bar{\mathbf{x}})\right) dist\left(0, \partial\phi(\mathbf{x})\right) \geq 1.$$

If a function ϕ satisfies the Generalized KL property at all points in dom $\partial\phi$ is called a KL function. Ψ in Definition 2.2.1 is called *desingularizing function* for ϕ. We obtain (2.24) from Definition 2.5.8 for $\Psi = c's^{1-\theta}$, with $c' > 0$ and $\theta \in [0,1)$.

The KL property is a well-studied error bound condition, because many iterative schemes have been proved to obtain linear convergence rate on optimization problems where the objective function has the KL property with an exponent between 0 and $\frac{1}{2}$. As we will see later, many practical problems of interest satisfy this requirement.

By leveraging the theory of functions with moderate behavior (defined in the following), the authors of [28] were able to prove equivalence between QG and KL property with an exponent of $\frac{1}{2}$.

Definition 2.5.9 (Function with moderate behavior [28]) *A smooth closed function* $\Psi : [0,r) \to \mathbb{R}$ *with* $\Psi(0) = 0$ *has a moderate behavior (near the origin) if, for all* $s \in (0,r)$*, and for given* $\kappa > 0$ *satisfies:*

$$s\Psi'(s) \geq \kappa\Psi(s).$$

The following proposition will be instrumental to prove the interesting connection between the QG and the KL property.

Proposition 2.5.3 (Equivalence between KL and moderate behavior [28]) *Under Assumptions A3-A4, if in addition* F *is* C^1 *and convex (Lipschitzianity of* $\nabla_{\mathbf{x}} F$ *is not needed), the following implications hold for all* $\mathbf{x} \in dom V$*:*

1. V *is a KL function with desingularizing function* $\Psi \Rightarrow dist(\mathbf{x}, \mathcal{X}^\star) \leq \Psi(V(\mathbf{x}) - \min V)$*;*

2. *Ψ has a moderate behavior, and dist$(\mathbf{x}, \mathcal{X}^\star) \leq \Psi\left(V(\mathbf{x}) - \min_{\mathbf{x}\in\mathbb{R}^n} V(\mathbf{x})\right) \Rightarrow V$ is a KL function.*

Note that $\Psi(s) = \gamma s^{\frac{1}{p}}$, for $p \geq 1, \gamma > 0$ is a function with moderate behavior. Choosing this specific function as a desingularizing function, the following corollary easily follows from Proposition 2.5.3.

Corollary 2.5.1 (Equivalence between KL and QG) *Under Assumptions A3-A4, if in addition F is C^1 and convex (Lipschitzianity of $\nabla_{\mathbf{x}} F$ is not needed), the following conditions are equivalent.*

1. *V has QG;*

2. *V has the KL property with an exponent of $\frac{1}{2}$.*

So, in light of Proposition 2.5.1, Corollary 2.5.1 shows also an equivalence between the KL property with an exponent of $\frac{1}{2}$ and the Luo-Tseng global error bound, under convexity.

• **Metrical subregularity -** By leveraging tools from variational analysis, other connections between different structural conditions of optimization problems can be shown.

Let $\Phi : \mathbb{R}^n \to \mathbb{R}^q$ be a set-valued map (multifunction), its graph is defined by graph$(\Phi) \triangleq \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^q | \mathbf{y} \in \Phi(\mathbf{x})\}$, and its inverse mapping by
$\Phi^{-1}(\mathbf{y}) \triangleq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{y} \in \Phi(\mathbf{x})\}$, for all $\mathbf{x} \in \mathbb{R}^q$.

For any given $\epsilon > 0$ and $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{B}_\epsilon(\mathbf{x}) \triangleq \{\mathbf{y} \in \mathbb{R}^n : |\mathbf{y} - \mathbf{x}| < \epsilon\}$ will denote the open ball around $\mathbf{x}$ with modulus ϵ.

Definition 2.5.10 (Metrical subregularity [29]) *$\Lambda : \mathbb{R}^q \to \mathbb{R}^n$ is metrically subregular at $(\tilde{\mathbf{y}}, \tilde{\mathbf{x}}) \in$ graph(Λ) if for some $\epsilon > 0$ there exists $\kappa \geq 0$ such that*

$$dist\left(\mathbf{y}, \Lambda^{-1}(\tilde{\mathbf{x}})\right) \leq dist\left(\tilde{\mathbf{x}}, \Lambda(\mathbf{y})\right), \qquad \forall\, \mathbf{y} \in \mathbb{B}_\epsilon(\tilde{\mathbf{y}}). \tag{2.78}$$

Proposition 2.5.4 (Equivalence between L-T and metrical subregularity [27]) *Under Assumptions A, the following conditions are equivalent.*

1. *The Luo-Tseng global error bound condition holds for V;*

2. *The subdifferential ∂V is metrically subregular at $(\mathbf{x}^\star, 0)$, for any $\mathbf{x}^\star \in \mathcal{X}^\star$.*

A condition analogous to metrical subregularity is the calmness of a set.

Definition 2.5.11 (Calmness [30]) $\Phi : \mathbb{R}^n \to \mathbb{R}^q$ *is calm (or pseudo upper-Lipschitz continuous) at* $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \in$ *graph*(Φ) *if there exist a neighborhood* $\mathbb{U}$ *of* $\tilde{\mathbf{y}}$*, a neighborhood* $\mathbb{V}$ *of* $\tilde{\mathbf{x}}$*, and* $\kappa \geq 0$ *such that*

$$\Phi(\mathbf{x}) \cap \mathbb{U} \subseteq \Phi(\tilde{\mathbf{x}}) + \kappa \|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \bar{\mathbb{B}}, \qquad \forall \mathbf{x} \in \mathbb{V}, \tag{2.79}$$

where $\bar{\mathbb{B}}$ denotes the closed unit ball centered at the origin.

Proposition 2.5.4 showed the connection between the Luo-Tseng error bound and the metric subregularity of ∂V. In fact, the following Proposition connects the calmness of a multifunction with the metrical subregularity of its inverse.

Proposition 2.5.5 (Equivalence between calmness and metrical subreg. [31]) $\Phi : \mathbb{R}^n \to \mathbb{R}^q$ *is calm at* $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \in$ *graph*(Φ) *if and only if its inverse* Φ^{-1} *is metrically subregular at* $(\tilde{\mathbf{y}}, \tilde{\mathbf{x}})$.

Leveraging this property, in [32], the calmness of a specific set has been used to establish the global Luo-Tseng error bound condition for a number of convex optimization problems, as it will be shown in Proposition 2.5.6. In order to state the Proposition, the following definition is needed.

Definition 2.5.12 (Perturbed solution set [32]) *Under Assumption A, and with* $F(\mathbf{x}) = h(\mathbf{A}\mathbf{x}) + \langle \mathbf{w}, \mathbf{x} \rangle$*,* $h : \mathbb{R}^m \to (-\infty; +\infty]$ *strongly convex smooth,* $\mathbf{A} \in \mathbb{R}^{m \times n}$*,* $\mathbf{w} \in \mathbb{R}^n$*. Assuming furthermore that* $\mathcal{X}^\star$ *is compact, the perturbed solution set* $\Gamma : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ *for any* $(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^m \times \mathbb{R}^n$ *is defined as:*

$$\Gamma(\mathbf{y}, \mathbf{z}) \triangleq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{y}, -\mathbf{z} \in \partial g(\mathbf{x})\}. \tag{2.80}$$

In order to better understand the above definition, note that, under the setting of Definition 2.5.12, it there exists a vector $\bar{\mathbf{y}} \in \mathbb{R}^m$, such that the solution set $\mathcal{X}^\star$ of (P) can also be written as [32]:

$$\mathcal{X}^\star = \Gamma\left(\bar{\mathbf{y}}, \mathbf{A}\nabla_{\mathbf{x}} h(\bar{\mathbf{y}}) + \mathbf{w}\right). \tag{2.81}$$

Proposition 2.5.6 (Calmness of Γ implies Luo-Tseng error bound [32]) *Under Assumption A, and with $f(\mathbf{x}) = h(\mathbf{A}\mathbf{x}) + \langle \mathbf{w}, \mathbf{x}\rangle$ with $h : \mathbb{R}^m \to (-\infty; +\infty]$ strongly convex smooth, $\nabla_{\mathbf{x}} h$ Lipschitz continuous, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{w} \in \mathbb{R}^n$. Assuming furthermore that $\mathcal{X}^\star$ is compact, the Luo-Tseng global error bound holds if the perturbed solution map Γ is calm at $(\bar{\mathbf{y}}, \mathbf{A}\nabla_{\mathbf{x}} h(\bar{\mathbf{y}}) + \mathbf{w}) \in \mathbb{R}^m \times \mathbb{R}^n$ for any $\mathbf{x}^\star \in \Gamma\left(\bar{\mathbf{y}}, \mathbf{A}\nabla_{\mathbf{x}} h(\bar{\mathbf{y}}) + \mathbf{w}\right) = \mathcal{X}^\star$.*

• **Relevant problems under the Luo-Tseng error bound -** Under Assumptions A3-A4, and assuming convexity of F, many interesting problems have been proved to satisfy the Luo-Tseng global error bound condition for the setting of problem (P). We list some example in the following.

- (Strongly convex case [25]): F is strongly convex and $\nabla_{\mathbf{x}} F$ is Lipschitz continuous.

- (Composite case (1) [33]): $F(x) = h(\mathbf{A}\mathbf{x}) + \langle \mathbf{w}, \mathbf{x}\rangle$ with $h : \mathbb{R}^m \to (-\infty; +\infty]$ strongly convex smooth, $\nabla_{\mathbf{x}} h$ Lipschitz continuous, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{w} \in \mathbb{R}^n$. ∂G is a polyhedral multifunction (i.e., set-valued mapping).

  Examples

  – Linear regression: $F(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, $\mathbf{b} \in \mathbb{R}^m$;

  – Logistic regression loss function:
  $F(\mathbf{x}) = \sum\limits_{i=1}^{m} \left(\log\left(1 + \exp\langle \mathbf{a}_i, \mathbf{x}\rangle\right) - b_i\langle \mathbf{a}_i, \mathbf{x}\rangle\right)$, $\mathbf{a}_i \in \mathbb{R}^n$, $b_i \in \{0; 1\}$, for all i;
  and the Poisson regression loss function:
  $F(\mathbf{x}) = \sum\limits_{i=1}^{m} \left(-b_i\mathbf{x} + \exp(\mathbf{x})\right)$, $b_i \in \mathbb{N}$, for all i. Note that these two loss functions are not strongly convex. The Luo-Tseng error bound condition

in this case holds over any compact set (e.g., for practical purposes, the linear convergence rate of an iterative scheme is preserved if it is possible to assume that the iterates remain bounded);

- G polyhedral convex function [34]
    - LASSO [5]: $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$, $\lambda \geq 0$;
    - $l_\infty-$norm regularizer: $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_\infty$, $\lambda \geq 0$;
    - fused LASSO [35]: $G(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum\limits_{i=1}^{I} |\mathbf{x}_i - \mathbf{x}_{i+1}|$, $\lambda_1, \lambda_2 \geq 0$, $n_i = 1$, for all i;
    - OSCAR [36]: $G(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum\limits_{i<j} \max\{|\mathbf{x}_i|, |\mathbf{x}_j|\}$, $\lambda_1, \lambda_2 \geq 0$, $n_i = 1$, for all i;
- G convex piecewise linear-quadratic function [37].

- (Composite case (2) [38,39]): $F(\mathbf{x}) = h(\mathbf{A}\mathbf{x})$ with $h : \mathbb{R}^m \to (-\infty; +\infty]$ strongly convex smooth, $\nabla_{\mathbf{x}} h$ Lipschitz continuous, $\mathbf{A} \in \mathbb{R}^{m\times n}$. $G(\mathbf{x}) = \sum\limits_{i=1}^{I} w_i\|\mathbf{x}_i\| + \lambda\|x\|_1$, with $w_i, \lambda \geq 0$, for all i. V coercive.

  Examples

  - Same loss functions F as in composite case (1);
  - Group LASSO [6,8]: $\lambda = 0$.

- $F(\mathbf{x}) = h(\mathbf{A}\mathbf{x})$ with $h : \mathbb{R}^m \to (-\infty; +\infty]$ strongly convex smooth, $\nabla_{\mathbf{x}} h$ Lipschitz continuous, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{w} \in \mathbb{R}^n$. $G(\mathbf{x}) = i_{\mathcal{C}}(\mathbf{x})$, with the set $\mathcal{C} \triangleq \left\{\mathbf{x} : \sum\limits_{i=1}^{m} w_i\|\mathbf{x}\|_p \leq \sigma\right\}$, where $\sigma > 0$, $p \in [1, 2]$, $w_i > 0$ for all i. Furthermore the following must hold:

$$\inf_{\mathbf{x}\in\mathbb{R}^n} V(\mathbf{x}) > \inf_{\mathbf{x}\in\mathbb{R}^n} h(\mathbf{A}\mathbf{x}). \tag{2.82}$$

  Examples

  - Indicator function of a closed ball centered in the origin, under (2.82).

- (Nuclear norm regularizer): $F(\mathbf{X}) = h(\mathbf{A}\mathbf{X}) + \langle \mathbf{W}, \mathbf{X} \rangle_F$ with $\mathbf{X} \in \mathbb{R}^{m \times n}$, $h : \mathbb{R}^l \to (-\infty; +\infty]$ strongly convex smooth, $\nabla_{\mathbf{X}} h$ Lipschitz continuous, $\mathbf{A} \in \mathbb{R}^{l \times m}$, $\mathbf{W} \in \mathbb{R}^{m \times n}$. $\mathcal{X}^\star$ compact. $G(\mathbf{X}) = \|\mathbf{X}\|_\star$. Furthermore, it must exist $\mathbf{X}^\star \in \mathcal{X}^\star$ such that the following strict complementarity condition is satisfied:

$$\mathbf{0} \in \nabla_{\mathbf{X}} F(\mathbf{X}^\star) + \mathrm{ri}\,(\partial \|\mathbf{X}^\star\|_\star)\,. \tag{2.83}$$

  Examples

  - Low rank matrix optimization, under (2.83).

- (Dual functional case [40]): $F(\mathbf{x}) = \max\limits_{\mathbf{y} \in \mathcal{Y}} \{\langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle - h(\mathbf{y})\} + \langle \mathbf{w}, \mathbf{x} \rangle$, with $h : \mathbb{R}^m \to \mathbb{R}$ strongly convex smooth, $\nabla_{\mathbf{x}} h$ Lipschitz continuous, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{w} \in \mathbb{R}^n$; and $\mathcal{Y}$ is a polyhedral set in $\mathbb{R}^n$. g is the indicator function of a polyhedral set.

  Examples

  - Elastic net penalty [41];

  - Soft-insensitive loss [42].

Case 2: F nonconvex smooth, G convex nonsmooth

We study now Problem (P) under Assumption A, dropping the requirement on the convexity of F, considered in the previous Section. It is easy to understand that in this setting, less structured with respect to the convex setting of the previous Section, connections and equivalences among the different conditions will be fewer.

• Generalized PL inequality and Luo-Tseng L_f-global error bound - The authors of [26] proved many interesting result, by defining $\mathcal{X}^\star$ as the set of minimizers of V. The following Proposition covers the smooth case.

Proposition 2.5.7 (Relationships in the smooth nonconvex setting [26]) *Under Assumption A, if in addition $G = 0$, and in all the previous definitions $\mathcal{X}^\star$ denotes the set of minimizers of V, the following hold:*

1. *SC implies ESC;*
2. *ESC implies WSC;*
3. *WSC implies RSI;*
4. *RSI implies the Luo-Tseng global error bound;*
5. *PL inequality and the Luo-Tseng global error bound are equivalent conditions;*
6. *PL inequality implies QG.*

In order to cover the nonsmooth case, the authors of [26] propose generalizations of the PL inequality and of the Luo-Tseng global error bound condition.

Definition 2.5.13 (Generalized PL inequality [26]) *A function $\Lambda(\mathbf{x}) = \phi(\mathbf{x}) + \psi(\mathbf{x})$, where $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ is closed with $L-$Lipschitz continuous gradient, and $\psi : \mathbb{R}^n \to (-\infty; \infty]$ is closed and smooth, satisfies the generalized PL inequality if it there exists a $\kappa > 0$ such that*

$$\frac{1}{2}\mathcal{D}_\psi(\mathbf{x}, L) \geq \kappa \left(\Lambda(\mathbf{x}) - \min_{\mathbf{x}\in\mathbb{R}^n} \Lambda(\mathbf{x}) \right),$$

where

$$\mathcal{D}(\mathbf{x}, \alpha) \triangleq -2\alpha \min_{\mathbf{y}\in\mathbb{R}^n} \left\{ \langle \nabla_{\mathbf{x}}\phi(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \psi(\mathbf{y}) - \psi(\mathbf{x}) \right\}.$$

Definition 2.5.14 (Luo-Tseng $L-$global error bound [26]) *A function $\Lambda(\mathbf{x}) = \phi(\mathbf{x}) + \psi(\mathbf{x})$, where $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ is closed with $L-$Lipschitz continuous gradient, and $\psi : \mathbb{R}^n \to (-\infty; \infty]$ is closed and smooth, satisfies the Luo-Tseng $L-$global error bound condition if for every scalar η: $\inf_{\mathbf{x}\in\mathbb{R}^n} \Lambda(\mathbf{x}) \leq \eta \leq \Lambda(\mathbf{x})$, and any $\gamma > 0$, there exist scalars $\delta, \tau > 0$ such that*

$$\mathit{dist}(\mathbf{x}, \Lambda^\star) \leq \tau \|\mathbf{x} - \mathit{Prox}_{\frac{1}{L}\psi}^{1/L}\left(\mathbf{x} - \frac{1}{L}\nabla_{\mathbf{x}}\phi(\mathbf{x})\right)\|_2,$$

whenever $\Lambda(\mathbf{x}) \leq \eta$ *and* $\|\mathbf{x} - Prox^{1/L}_{\frac{1}{L}\psi}\left(\mathbf{x} - \frac{1}{L}\nabla_{\mathbf{x}}\phi(\mathbf{x})\right)\|_2 \leq \delta$*.* $\Lambda^\star$ *denotes the set of minimizers of* Λ*.*

Leveraging the above definitions, the next Proposition covers the nonsmooth case.

Proposition 2.5.8 (Relationships in the composite nonconvex setting [26]) *Under Assumption A, if in all the previous definitions* $\mathcal{X}^\star$ *denotes the set of minimizers of* V*, the following conditions are equivalent.*

1. *The Luo-Tseng* $L-$*global error bound condition holds for* V*;*

2. V *satisfies the generalized PL inequality;*

3. V *has KL property with an exponent of* $\frac{1}{2}$*.*

Differently from the above results, we go back now to the definition of $\mathcal{X}^\star$ as the set of stationary points of V in problem (P).

• **Metrical subregularity and quadratic growth -** For the composite optimization problem (P) under Assumptions A, Proposition 2.5.4 holds too.

On the other hand, a relationship between QG and the Luo-Tseng error bound holds only if V is robust to tilt-perturbations.

Definition 2.5.15 (Stable strong local minimizer [27]) $\bar{\mathbf{x}}$ *is a stable strong local minimizer with constant* $\alpha > 0$ *of a function* $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ *if there exists a neighborhood* Φ *of* $\bar{\mathbf{x}}$ *so that for each vector* $\mathbf{y}$ *near the origin, there is a point* $\mathbf{x_y}$ *(necessarily unique) in* Φ*, with* $\mathbf{x_0} = \bar{\mathbf{x}}$*, so that the for the perturbed function* $\phi_{\mathbf{y}} \triangleq \phi(\cdot) - \langle \mathbf{y}, \cdot \rangle$ *the inequality*

$$\phi_{\mathbf{y}}(x) \geq \phi_{\mathbf{y}}(\mathbf{x_y}) + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{x_y}\|_2^2.$$

holds for all $\mathbf{x} \in \Phi$*.*

The following Proposition relates QG and the Luo-Tseng error bound.

Proposition 2.5.9 (QG robust to tilt-perturbations implies L-T err. bound [27]) *Given a local minimizer* $\mathbf{x}^\star$ *of problem* (P) *under Assumption A, the following conditions are equivalent*

1. *(Uniform quadratic growth)* $\mathbf{x}^\star$ *is a stable strong local minimizer of* V*;*

2. *(Tilt-stability) There exists a neighborhood* $\mathcal{X}$ *of* $\mathbf{x}^\star$ *such that the set-valued mapping*
$$\mathbf{y} \to \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{X}} \{V(\mathbf{x}) - \langle \mathbf{y}, \mathbf{x}\rangle\}$$
is single-valued and Lipschitz continuous on some neighborhood of the origin;

3. *(Subdifferential strong regularity) There exist a neighborhood* $\mathcal{X}$ *of* $\mathbf{x}^\star$ *and* $\mathcal{Y}$ *of* $\bar{\mathbf{y}} = 0$ *so that* $(\partial V)^{-1} : \mathcal{V} \to \mathcal{X}$ *is a single-valued Lipschitz continuous mapping.*

Note that statement 3 in Theorem 2.5.9 implies the metrical subregularity of ∂V at $(\mathbf{x}^\star, 0)$, which, in turn, due to Proposition 2.5.4 implies that the Luo-Tseng global error bound holds for V.

The following Corollary summarizes the results of this subsection

Corollary 2.5.2 (Metrical subregularity, QG and Luo-Tseng error bound) *Under Assumption A, the following conditions are equivalent:*

1. *The Luo-Tseng global error bound condition holds;*

2. *The subdifferential* ∂V *is metrically subregular at* $(\mathbf{x}^\star, 0)$*, for any* $\mathbf{x}^\star \in \mathcal{X}^\star$*.*

Furthermore, if any $\mathbf{x}^\star \in \mathcal{X}^\star$ *is a stable strong local minimizer, the Luo-Tseng global error bound condition holds.*

• **KL property -** In the nonconvex composite setting considered in this Section, the equivalence between the KL property with an exponent of $\frac{1}{2}$ and the Luo-Tseng error bound condition does not hold straightforwardly anymore.

The authors of [16] are able to connect the KL property and the Luo-Tseng error bound under the following additional assumption.

Assumption E (Error bound condition for V)

(E1) (proper separation of isocost surfaces [24]) For any $\mathbf{x}^\star \in \mathcal{X}^\star$, there exists $\delta > 0$ so that $V(\mathbf{y}) = V(\mathbf{x}^\star)$ whenever $\mathbf{y} \in \mathcal{X}^\star$ and $\|\mathbf{y} - \mathbf{x}^\star\|_2 \leq \delta$.

Note that under Assumptions A, E1 for problem (P) holds if, for example, V takes on only a finite number of values on $\mathcal{X}^\star$. Thus, E1 holds if V is convex or F is quadratic and G is the indicator function of a polyhedral set.

Proposition 2.5.10 (Luo-Tseng global error bound implies KL property [16]) *Under Assumptions A-E, if the Luo-Tseng global error bound holds for V, then V is a KL function with an exponent of $\frac{1}{2}$.*

• **Relevant problems under the Luo-Tseng error bound -** Under Assumptions A3-A4, the only relevant class of problems where the Luo-Tseng error bound condition holds for the nonconvex setting of problem (P), different from the problems previously listed above, is

- (Quadratic case [33, 43]) F is a smooth quadratic functions; G is the indicator function of a polyhedral set (see previous examples for valid G's).

More interestingly, we remark that the KL inequality with an exponent of $\frac{1}{2}$ it has been proved [16] to hold for nonconvex regularizers as the SCAD [12] or the MCP [13] penalties.

Case 3: F, G nonconvex and nonsmooth

In this last Section we consider problem (P) in its most general setting, i.e., removing the assumption A3 on the convexity of G too.

In the fully nonconvex setting the proximal operator, which characterizes the Luo-Tseng error bound, is not properly defined.

The authors of [44] propose the following analogous stationarity measure in order to replace the proximal operator.

Definition 2.5.16 (Generalized Luo-Tseng local error bound [44]) *Assume A4, F to be C^1, and G being a closed function; then, the generalized Luo-Tseng local error bound condition holds for V at $\mathbf{x}^\star \in \mathcal{X}^\pi$ if there exist scalars $\delta, \tau > 0$ such that*

$$dist\,(\mathbf{x}, \mathcal{X}^\pi) \leq \kappa dist\left(\mathbf{x}, \operatorname*{arg\,min}_{\mathbf{y}\in\mathbb{R}^n}\left\{G(\mathbf{y}) + \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{x}+\gamma\nabla_{\mathbf{x}}F(\mathbf{x})\|_2^2\right\}\right)$$

holds for all $\mathbf{x} \in \mathbb{B}_\epsilon(\mathbf{x}^\star)$, where $\mathcal{X}^\pi$ is the set of proximal stationary points

$$\mathcal{X}^\pi \triangleq \{\mathbf{x} : 0 \in \nabla_{\mathbf{x}}F(\mathbf{x}) + \partial^\pi G(\mathbf{x})\},$$

and $\partial^\pi G$ is the proximal subdifferential of G (see, e.g. [37]).

A relationship between the generalized Luo-Tseng error bound and the KL property with an exponent of $\frac{1}{2}$ has been proved in [44], under the semi-convexity assumption.

Definition 2.5.17 (Semi-convexity [44]) *A closed function $\phi : \mathbb{R}^n \to (-\infty; \infty]$ is semi-convex around $\bar{\mathbf{x}} \in dom\ \phi$ with modulus $\rho > 0$, if there exists $\kappa > 0$ such that the function $\phi(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x}\|_2^2$ is convex on $\mathbb{B}_\eta(\bar{\mathbf{x}})$.*

Proposition 2.5.11 (Generalized L-T local err. bound implies KL [44]) *Assume A4, F to be C^1, and G being a closed function; if furthermore G is semi-convex around $\mathbf{x}^\star \in \mathcal{X}^\pi$, Assumption E holds too, and the generalized Luo-Tseng local error bound condition holds at $\mathbf{x}^\star$. Then, V satisfies the KL property with an exponent of $\frac{1}{2}$ at $\mathbf{x}^\star$.*

On the other hand, the authors of [27] proposed the following stationarity measure, called *prox-gradient mapping* in order to replace the proximal operator.

Definition 2.5.18 (Prox-gradient mapping [27]) *Assume A4, F and G being closed functions; consider $F(\mathbf{x}) = h(c(\mathbf{x}))$, with $c : \mathbb{R}^n \to \mathbb{R}^m$ continuously differentiable, and $h : \mathbb{R}^m \to (-\infty; +\infty]$; then the prox-gradient mapping $\mathcal{G}_t$, for $t > 0$, and any $\mathbf{x} \in \mathbb{R}^n$ is defined as:*

$$\mathcal{G}_t(\mathbf{x}) \triangleq t^{-1}(\mathbf{x} - \mathcal{S}_t(\mathbf{x})),$$

where $\mathcal{S}_t$ is the stationary point map

$$\mathcal{S}_t(\mathbf{x}) \triangleq \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} \textit{ is a stationary point of } V_t(\mathbf{x}, \cdot)\},$$

and, for any $\mathbf{y} \in \mathbb{R}^n$, V_t is the quadratic perturbation

$$V_t(\mathbf{x}; \mathbf{y}) \triangleq G(\mathbf{y}) + h\left(c(\mathbf{x}) + \langle \nabla_{\mathbf{x}} c(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle\right) + \frac{1}{2t}\|\mathbf{x} - \mathbf{y}\|_2^2.$$

The following definition is also needed, and it will be instrumental to establish, as already done in the previous sections, a connection between the subregularity of ∂V and the Luo-Tseng error bound (which, as already said, will be replaced now by the prox-gradient mapping $\mathcal{G}_t$).

Definition 2.5.19 (Prox-regularity [45]) *A closed function $\phi : \mathbb{R}^n \to (-\infty; +\infty]$ is prox-regular at $\bar{\mathbf{x}}$ for $\mathbf{v} \in \partial\phi(\bar{\mathbf{x}})$ if there exist a neighborhood $\mathcal{X}$ of $\bar{\mathbf{x}}$ and $\mathcal{V}$ of $\mathbf{v}$, along with constants $\epsilon, r > 0$ so that the inequality*

$$\phi(\mathbf{y}) \geq \phi(\mathbf{x}) + \langle \mathbf{v}, \mathbf{y} - \mathbf{x} \rangle - \frac{r}{2}\|\mathbf{y} - \mathbf{x}\|_2^2,$$

holds for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ with $|\phi(\mathbf{x}) - \phi(\bar{\mathbf{x}})| < \epsilon$, and for every subgradient $\mathbf{v} \in \mathcal{V} \cap \partial\phi(\mathbf{x})$.

The following Proposition has been proved in [27], and it is an attempt to describe the structure of V through a condition similar to the Luo-Tseng error bound. This characterization could be useful in the future to prove linear convergence rate of iterative methods for problem (P) in the fully nonconvex setting.

The missing definitions in Proposition 2.5.12 can be found in [27].

Proposition 2.5.12 (Metrical subregularity and "L-T" error bound [27]) *Assume A4, and F being a closed function; consider $G(\mathbf{x}) = 0$ (for ease of notation), $F(\mathbf{x}) = h(c(\mathbf{x}))$, where $c : \mathbb{R}^n \to \mathbb{R}^m$ is continuously differentiable, $h : \mathbb{R}^m \to (-\infty; +\infty]$ is closed, and c is transverse to h at $\bar{\mathbf{x}}$, for a given $\bar{\mathbf{x}} \in \mathcal{X}^\star$. If, furthermore, the Jacobian ∇c is Lipschitz around $\bar{\mathbf{x}}$ and h is prox-regular at $c(\bar{\mathbf{x}})$ for every subgradient $w \in \partial h(c(\bar{\mathbf{x}}))$ satisfying $\langle \nabla \mathbf{c}^\star(\bar{\mathbf{x}}), \mathbf{w} \rangle = 0$, for a given $t > 0$, the following two conditions are equivalent:*

1. *It there exists a $V-$attentive localization of ∂V that is metrically subregular at $(\bar{\mathbf{x}}, 0)$.*

2. *It there exists a $V-$attentive localization of $\mathcal{G}_t$ that is metrically subregular at $(\bar{\mathbf{x}}, 0)$.*

If, additionally, h is convex, the following two conditions are equivalent:

(a) ∂V is metrically subregular at $(\bar{\mathbf{x}}, 0)$

(b) $\mathcal{G}_t$ is metrically subregular at $(\bar{\mathbf{x}}, 0)$

Note that, in the setting of the previous Sections, condition (b) is equivalent to the Luo-Tseng global error bound condition.

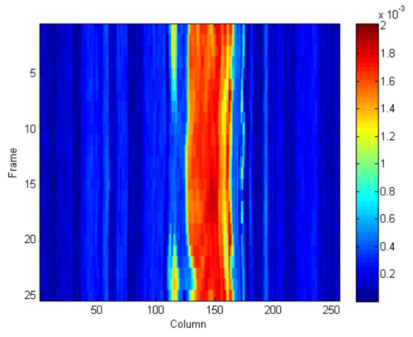

Figure 2.8. Reference

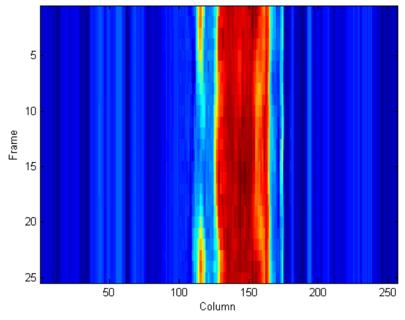

Figure 2.9. FLEXA

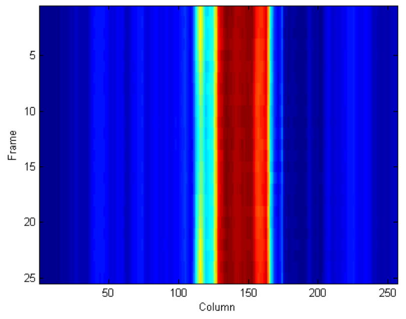

Figure 2.10. k-t FOCUSS

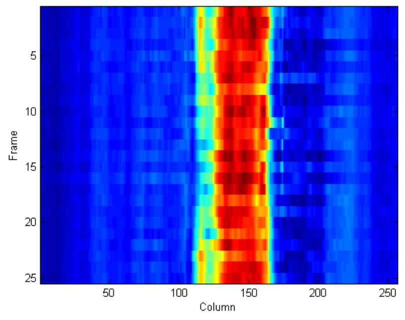

Figure 2.11. FISTA

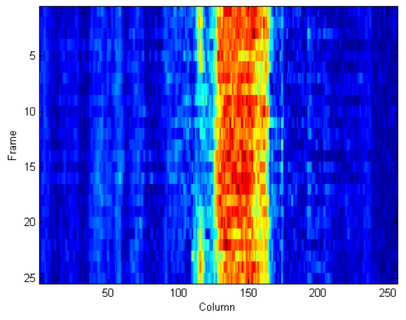

Figure 2.12. SpaRSA

Figure 2.13. Intensity profiles of the 100th row after one second of processing.

3. ASYFLEXA: ASYNCHRONOUS PARALLEL ALGORITHM FOR NONSMOOTH NONCONVEX OPTIMIZATION

We propose a new asynchronous parallel block-descent algorithmic framework for the minimization of the sum of a smooth, nonconvex function, and a nonsmooth, convex one, subject to both convex and nonconvex constraints. The proposed framework hinges on SCA techniques and a novel probabilistic model that captures key elements of modern computational architectures and asynchronous implementations in a more faithful way than current state-of-the-art models. Other key features of the framework are: i) it covers in a unified way several specific solution methods; ii) it accommodates a variety of possible parallel computing architectures; and iii) it can deal with nonconvex constraints. Almost sure convergence to stationary solutions is proved, and theoretical complexity results are provided, showing nearly ideal linear speedup when the number of workers is not too large.

Specifically, in Section 3.1 we present the optimization problem under exam, we explain the main challenges that motivated our research, we list the main contributions of our work, and we discuss the related literature. In Section 3.2 we present our proposed algorithmic framework AsyFLEXA. In Section 3.3 we discuss in detail the probabilistic model underlying the proposed method. Section 3.4 presents the main convergence results. Section 3.5 contains an extension of the results of this Chapter to the case involving the presence of nonconvex constraints. Finally, Section 3.6 shows extensive experimental results. The proofs of the Theorems are collected in the Appendix of this Chapter.

The novel results of this Chapter have been published in

- Cannelli, L. and Facchinei, F. and Kungurtsev, V. and Scutari, G., *"Asynchronous Parallel Algorithms for Nonconvex Optimization"*, Mathematical Programming, pages 1-34, 2019 [46].

- Cannelli, L. and Facchinei, F. and Kungurtsev, V. and Scutari, G., *"Asynchronous Parallel Nonconvex Large-Scale Optimization"*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4706-4710, 2017 [47].

- Cannelli, L. and Scutari, G. and Facchinei, F. and Kungurtsev, V., *"Parallel Asynchronous Lock-Free Algorithms for Nonconvex Big-Data Optimization"*, 50th IEEE Asilomar Conference on Signals, Systems and Computers, pages 1009-1013, 2016 [48]. **Best Paper Award Runner-up**

A sample code, written in C++, of AsyFLEXA for LASSO problems is available at `https://github.com/lcannell/Parallel-Asynchronous-Algorithms-For-Nonconvex-Problems`.

3.1 Introduction

It is studied an asynchronous parallel block-descent methods for the following class of nonconvex nonsmooth minimization problems:

$$\min_{\mathbf{x}\triangleq[\mathbf{x}_i]_{i=1}^N\in\mathcal{X}\subseteq\mathbb{R}^n} \quad V(\mathbf{x}) \triangleq F(\mathbf{x}) + \sum_{i=1}^N g_i(\mathbf{x}_i) \tag{P}$$

where F is a smooth, possibly nonconvex function, g_i are possibly nonsmooth, convex functions, $\mathcal{X} \triangleq \mathcal{X}_1 \times \ldots \times \mathcal{X}_N$, and each $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is a closed, possibly nonconvex set.

Instances of Problem (P) arise in many fields, including compressed sensing, machine learning, data mining, and genomics, just to name a few. Typically, in data-driven applications F might measure the misfit between the observations and the postulated model, parametrized on $\mathbf{x}$, while the regularizers g_i encode structural constraints on the solution, such as sparsity.

Many of the aforementioned applications give rise to extremely large-scale problems, which naturally call for *asynchronous, parallel* solution methods. In fact, well suited to modern computational architectures, asynchronous methods reduce the idle times of workers, mitigate communication and/or memory-access congestion, and make algorithms more fault-tolerant. In this Chapter it is introduced a general asynchronous block-descent algorithm for finding stationary solutions of Problem (P).

It is considered a generic multi-worker architecture (e.g., shared memory system, message passing-based system, cluster computer, cloud federation) wherein multiple workers, continuously and without coordination with each other, update a block-variable by solving a strongly convex block-model of Problem (P). More specifically, at iteration k, a worker updates a block-variable $\mathbf{x}_{i^k}^k$ of $\mathbf{x}^k$ to $\mathbf{x}_{i^k}^{k+1}$, with i^k in the set $\mathcal{N} \triangleq \{1, \ldots, N\}$, thus generating the vector $\mathbf{x}^{k+1}$. When updating block i^k, in general, the worker does not have access to the current vector $\mathbf{x}^k$, but it will use instead the local estimate $\mathbf{x}^{k-\mathbf{d}^k} \triangleq [x_i^{k-d_i^k}]_{i=1}^N$, where $\mathbf{d}^k \triangleq [d_i^k]_{i=1}^N$ is the "vector of delays", whose components d_i^k are nonnegative integers. Note that $\mathbf{x}^{k-\mathbf{d}^k}$ is nothing else but a combination of delayed, block-variables. The way each worker forms its own estimate $\mathbf{x}^{k-\mathbf{d}^k}$ depends on the particular architecture under consideration and it is immaterial to the analysis of the algorithm. It is only observed here that if all delays d_i^k are zeros, the model reduces to a standard synchronous one.
Given $\mathbf{x}^{k-\mathbf{d}^k}$ and i^k, block $\mathbf{x}_{i^k}^k$ is updated by solving the following *strongly convex* block-approximation of Problem (P):

$$\widehat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k}) \triangleq \underset{\mathbf{x}_{i^k} \in \tilde{\mathcal{X}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k})}{\operatorname{argmin}} \tilde{F}_{i^k}(\mathbf{x}_{i^k}; \mathbf{x}^{k-\mathbf{d}^k}) + g_{i^k}(\mathbf{x}_{i^k}), \tag{3.1}$$

and then setting

$$\mathbf{x}_{i^k}^{k+1} = \mathbf{x}_{i^k}^k + \gamma^k \left(\widehat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k}) - \mathbf{x}_{i^k}^k \right). \tag{3.2}$$

In (3.1), $\tilde{F}_{i^k}$ and $\tilde{\mathcal{X}}_{i^k}$ represent a strongly convex surrogate of F and a convex set obtained replacing the nonconvex functions defining $\mathcal{X}_{i^k}$ by suitably chosen upper convex approximations, respectively; both $\tilde{F}_{i^k}$ and $\tilde{\mathcal{X}}_{i^k}$ are built using the out-of-sync information $\mathbf{x}^{k-\mathbf{d}^k}$. If the set $\mathcal{X}_{i^k}$ is convex, then it will be always taken $\tilde{\mathcal{X}}_{i^k} = \mathcal{X}_{i^k}$. In (3.2), $\gamma^k \in (0, 1]$ is the stepsize. Note that, in the above asynchronous model, the worker that is in charge of the computation (3.1) and the consequent update (3.2) is immaterial.

3.1.1 Major contributions

The major contribution of the research work presented in this Chapter are:

1. *A new probabilistic model for asynchrony fixing some unresolved issues*: Almost all modern asynchronous algorithms for convex and nonconvex problems are modeled in a probabilistic way. Here it is put forth a novel probabilistic model describing the statistics of $(i^k, \mathbf{d}^k)$ that differs markedly from existing ones. This new model allows not only to fix some important theoretical issues that mar most of the papers in the field (see discussion below on related work), but it also permits to analyze for the first time in a sound way several practically used and effective computing settings and new asynchronous algorithms. For instance, it is widely accepted that in shared-memory systems, the best performance are obtained by first partitioning the variables among cores, and then letting each core update in an asynchronous fashion their own block-variables, according to some randomized cyclic rule. To the best of my knowledge, this is the first work proving convergence of such practically effective methods in an asynchronous setting.

2. *The ability to effectively deal with nonconvex constraints*: All the works in the literature but [49, 50] can deal only with unconstrained or convex constrained problems. On the other hand, the algorithms in [49, 50] require at each iteration the computation of the global optimal solution of nonconvex subproblems, which, except in few special cases, can be as difficult as solving the original nonconvex problem. The proposed method is the first asynchronous method that allows one to deal (un-

der adequate assumptions) with nonconvex constraints while solving only strongly convex subproblems.

3. *The possibility to leverage potentially complex, but effective subproblems* (3.1): Asynchronous methods so far are all built around a proximal linearization method, which corresponds, in the proposed framework, to setting

$$\tilde{F}_i(\mathbf{x}_{i^k};\mathbf{x}^{k-\mathbf{d}^k}) = \left\langle \nabla_{\mathbf{x}_{i^k}} F\left(\mathbf{x}^{k-\mathbf{d}^k}\right), \mathbf{x}_{i^k} - \mathbf{x}_{i^k}^{k-d^k_{i^k}} \right\rangle + \alpha \left\|\mathbf{x}_{i^k} - \mathbf{x}_{i^k}^{k-d^k_{i^k}}\right\|_2^2,$$

for some constant $\alpha > 0$. This choice often leads to efficient solution methods and, in some cases, even to subproblems that admit a solution in closed-form. For instance, it has been shown to be very efficient on composite quadratic problems, like LASSO. However, moving to more nonlinear problems, one may want to use more complex/higher order models. In fact, the more sophisticated the subproblem (3.1), the better the overall behavior of the algorithm (at least in terms of iterations) is. This happens at the price of computationally more expensive subproblems. But in asynchronous and distributed methods, the bottleneck is often given by the communication cost. In these cases, it might be desirable to reduce the communication overhead at the price of more complex subproblems to solve. Furthermore, there are many application for which one can define subproblems that, while not being proximal linearizations, still admit closed-form solutions (see. e.g., [51,52]). Overall, the ability to use more complex subproblems is an additional degree of freedom that one may want to exploit to improve the performance of the algorithm.

4. *Almost sure convergence and complexity analysis*: It is proved: i) almost sure convergence to stationary solutions of Problem (P); and ii) convergence to ϵ-stationary solutions in an $\mathcal{O}(\epsilon^{-1})$ number of iterations. It is remarked that the convergence results presented here match similar ones in the literature [49,50,53–55], which however were obtained in a simplified setting (e.g., only for unconstrained or convex constrained problems) and under unrealistic probabilistic assumptions on the pair index-delay $(i^k, \mathbf{d}^k)$ (see discussion on related work). The analysis presented here builds on an induction technique based on the proposed probabilistic model and a

novel Lyapunov function that properly combines variable dynamics and their delayed versions.

5. *Fixed and diminishing stepsize*: A *fixed stepsize* has been adopted universally in almost all papers dealing with asynchronous methods. While we will use a fixed stepsize for our complexity analysis results, it also has disadvantages. On the one hand the proper choice of the stepsize usually requires the knowledge of unknown constants and, on the other hand, and more importantly, the use of the stepsize that gives theoretical convergence usually leads to extremely slow practical convergence. In contrast to this situation we advocate the use of *diminishing stepsize* rules. This choice frees us from the necessity of a tuning of the stepsize and, in all our numerical experiments, has proven to be highly efficient.

6. *A theoretical almost linear speedup for a wide range of number of cores*: The holy grail of asynchronous methods is the ideal linear speedup (with respect to the number of workers). This theoretical limit is not achievable in practice; in fact, as the number of workers increases, the effective speedup is always limited by associated overheads (communication costs, conflicts, etc.), which make the linear growth impossible to achieve for arbitrarily large number of workers. By using the number of iterations needed to achieve an ϵ-stationary solution as a proxy for the computational time and leveraging the new proposed Lyapunov function, it is possible to show almost linear speedup in many settings of practical interest. This is the first theoretical result on speedup, based on a realistic probabilistic model for asynchrony (see discussion in contribution 1).

3.1.2 Related work

Although asynchronous block-methods have a long history (see, e.g., [9, 56–59]), their revival and probabilistic analysis have taken place only in recent years; this is mainly due to the current trend towards huge scale optimization and the availability of ever more complex computational architectures that call for efficient and resilient algorithms. Indeed, asynchronous parallelism has been applied to many state-of-the-art optimization algorithms, including stochastic gradient methods [60–66] and ADMM-like schemes [67–69]. Block-Coordinate Descent (BCD) methods are part of the folklore in optimization; more recently, they have been proven to be particularly effective in solving very large-scale problems arising, e.g., from data-intensive applications. Their asynchronous counterpart has been introduced and studied in the seminal work [54], which motivated and oriented much of subsequent research in the field, see e.g. [49, 50, 53, 55, 70]. The interested reader is referred to [71] and references therein for a detailed overview of BCD methods. There are several differences between the above methods and the framework proposed in this paper, as detailed next.

• *On the probabilistic model:* All current probabilistic models for asynchronous BCD methods are based on the (implicit or explicit) assumption that the random variables i^k and $\mathbf{d}^k$ are *independent*; this greatly simplifies the convergence analysis. However, in reality there is a strong dependence of the delays $\mathbf{d}^k$ on the updated block i^k. Consider the setting where the variables are partitioned among two workers and each worker updates only its own block-variables; let $\mathcal{N}_1$ and $\mathcal{N}_2$ be the index set of the blocks controlled by worker 1 and 2, respectively. It is clear that in the updates of worker 1 it will always be $d_i^k = 0$, for all $i \in \mathcal{N}_1$ and k, while (at least some) delays d_i^k associated with the blocks $i \in \mathcal{N}_2$ will be positive; the opposite happens to worker two. The independence assumption is unrealistic also in settings where all the workers share all the variables. Blocks that are updated less frequently than others, when updated, will have larger associated delays. This happens, for instance, in problems where i) some blocks are more expensive to update than others, because they are larger, bear more nonzero entries, or data retrieval requires longer times; or ii) the updates are carried by heterogeneous workers (e.g., some are faster or busier than others). This assumption have been tested, performing an asynchronous algorithm on two different architectures and measuring the average delay corresponding to different blocks updated. The experiments were performed on a shared-memory system with 10 cores of an Intel E5-2699Av4 processor. An asynchronous algorithm was applied to a LASSO problem [5] with 10000 variables, partitioned uniformly into 100 contiguous blocks; the Hessian matrix was generated with high sparsity on several rows. All the cores can update any block, selected uniformly at random. It has been found that blocks associated with the sparse rows of the Hessian have delays $\mathbf{d}^k$ with components between 0 and 3, while the delays of the other blocks were all bigger than 20. Even when the computing environment is homogeneous and/or the block updates have the same cost, the aforementioned dependence persists. A message-passing system on Purdue Community Cluster Snyder was simulated and two nodes of the cluster were used, each of them equipped with 10 cores of an Intel Xeon-E5 processors and its own shared memory. Every node can update every block, selected uniformly at random. An asynchronous algorithm was run on the same LASSO problem described above but now with a dense Hessian matrix. The blocks updated by node 1 have an average delay of 12 while those updated by node 2 experience an average delay of 22. This can be due to several uncontrollable factors, like operation system and memory schedulers, buses controllers, etc., which are hard to rigorously model and analyze.

Another unrealistic assumption often made in the literature [49, 53, 54, 61] is that the block-indices i^k are selected *uniformly* at random. While this assumption simplifies the convergence analysis, it limits the applicability of the model; see Examples 4 and 5 in Section 3.3.1. In a nutshell, this assumption may be satisfied only if all workers have the same computational power and have access to all variables.

Finally, this discussion on probabilistic models underlying asynchronous algorithms is concluded by mentioning the line of work dealing with stochastic gradient methods. Stochastic gradient methods are similar to block-descent approaches in that at each iteration sampling is performed to determine the nature of the update, but sampling is done among functions in an optimization problem minimizing the sum of functions, as opposed to block variables. A related, albeit different, issue of independence in the probabilistic models used in stochastic gradient methods was first noted in the technical report [64], see also [65, 66] for further developments. These papers circumvent the issue by enforcing independence (a) using a particular manner of labeling iterations as well as (b) reading the entire vector of variables regardless of the sparsity pattern among the summand functions in the objective. However, the analysis in [64–66] is (c) only performed in the context of strongly convex unconstrained problems, (d) involves uniform sampling and (e) is only applicable for the shared memory setting. Thus, while the analysis and procedures described in the references above are interesting, on the whole requirements (b)-(e) make these proposals of marginal interest in the context of block-descent methods (even assuming they can actually be adapted to our setting).

Differently from the aforementioned works, the more general and sophisticated probabilistic model presented here neither postulates the independence between i^k and $\mathbf{d}^k$ nor requires artificial changes in the algorithm [e.g., costly unnecessary readings, as in (b)] to enforce it; it handles instead the potential dependency among variables directly. By doing so, one can establish convergence without requiring any of the restrictive conditions (b)-(e), and significantly enlarge the class of computational architecture falling within the model [e.g., going beyond (d) and (e)]−see Section 3.3.1 for several examples.

• *Nonconvex constraints*: Another important feature of the presented algorithm is the ability to handle nonconvex objective functions and nonconvex constraints by an algorithm that only needs to solve, at each iteration, a strongly convex optimization subproblem. Almost all asynchronous methods cited above can handle only convex optimization problems or, in the case of fixed point problems, nonexpansive mappings. The exceptions are [62, 72] and, more relevant to our setting, [49, 50] that study unconstrained and constrained nonconvex optimization problems, respectively. However, the papers dealing with constrained problems, i.e. [49, 50], propose algorithms that require, at each iteration, the global solution of nonconvex subproblems. Except for few cases, the subproblems could be hard to solve and potentially as difficult as the original one.

• *Successive Convex Approximation*: All the asynchronous algorithms described so far use proximal linearization to define subproblems. As already pointed out, this is the first paper where subproblem models able to capture more structure of the objective functions are considered. This offers more freedom and flexibility to tailor the minimization algorithm to the problem structure, in order to obtain more efficient solution methods.

• **Notation -** Underlined symbols will denote random variables, e.g., $\underline{\mathbf{x}}^k$, $\mathbf{x}^{k-\underline{\mathbf{d}}^k}$, whereas the same symbols with no underline are the corresponding realizations.

3.2 Asynchronous Algorithmic Framework

In this section we introduce the assumptions on Problem (P) along with the formal description of the proposed algorithm. For simplicity of presentation, we begin studying (P) assuming that there are only convex constraints, i.e., all $\mathcal{X}_i$ are convex. This unnecessary assumption will be removed in Section 3.5.

Assumption A (On Problem (P)).

(A1) Each set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is nonempty, closed, and convex;

(A2) F is C^1 and its gradients $\nabla_{\mathbf{x}_i} F$ are globally $L-$Lipschitz on $\mathcal{X}$;

(A3) Each $g_i : \mathcal{X}_i \to \mathbb{R}$ is convex;

(A4) V is coercive on $\mathcal{X}$, i.e., $\lim_{\mathbf{x}\in\mathcal{X}, \|\mathbf{x}\|\to\infty} V(\mathbf{x}) = +\infty$.

As already discussed in Section 2.1, these assumptions are rather standard. For example, A2 holds trivially if $\mathcal{X}$ is bounded and $\nabla_{\mathbf{x}} F$ is locally Lipschitz; while A4 guarantees the existence of a solution.

When considering the case of case diminishing stepsize γ^k, in order to prove asymptotic convergence of our proposed method, we will also need the following additional assumption:

Assumption A* (On Problem (P)).

(A5) Each g_i is L_g-Lipschitz continuous on $\mathcal{X}_i$.

Note that by assuming all g_i's to be continuous, we are not considering the case of extended-valued functions. The use of extended-valued functions in the objective functions is sometimes used to deal with constraints by formally reducing a constrained problem to an unconstrained one. However, in our approach we will need to deal with constraints explicitly and therefore we avoid the extended-valued function formalism; consequently, in most practical cases the g_i's are norms or polyhedral functions and A3-A5 are satisfied.

We introduce now our algorithmic asynchronous framework. The asynchronous iterations performed by the workers are given in (3.1) and (3.2) [cf. Section 3.1]. However, the analysis of the algorithm based directly on (3.1)-(3.2) is not a simple task. The key idea is then to introduce a "global view" of (3.1)-(3.2) that captures through a unified, general, probabilistic model several specific computational architectures/systems and asynchronous modus operandi. The iteration $k \to k+1$ is triggered when a block-component i^k of the current $\mathbf{x}^k$ is updated by some worker using (possibly) delayed information $\mathbf{x}^{k-\mathbf{d}^k}$, thus generating the new vector $\mathbf{x}^{k+1}$. Note that, in this model, the worker that performs the update is immaterial. Given (3.1) and (3.2), it is clear that the update $\mathbf{x}^k \to \mathbf{x}^{k+1}$ is fully determined once i^k and $\mathbf{d}^k$ are specified. In several asynchronous methods, the index i^k is chosen randomly. Even when this is not the case, the values of i^k and $\mathbf{d}^k$ are difficult to preview beforehand, because they depend on several factors which are hard to model mathematically, such as the computational architecture, the specific hardware, the communication protocol employed by the workers, possible hardware failures, etc.. Therefore, we model the sequence of pairs $\{(i^k, \mathbf{d}^k)\}$ generated by the algorithmic process as a realization of a stochastic process; the probabilistic space associated to this stochastic process is formally introduced in Section 3.3. The proposed general asynchronous model is summarized in Algorithm 4, which we term Asynchronous FLexible ParallEl Algorithm (AsyFLEXA).

Algorithm 4: Asynchronous FLexible ParallEl Algorithm (AsyFLEXA)

```
Data : $\mathbf{x}^0 \in \mathcal{X}$, $\{\gamma^k\}$. Set $k = 0$.
(S.1) If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) The random variable $(\underline{i}^k, \underline{\mathbf{d}}^k)$ is realized as $(i^k, \mathbf{d}^k)$;
(S.3) $\widehat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k})$ is computed:
          $\widehat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k}) \triangleq \arg\min_{\mathbf{x}_{i^k} \in \mathcal{X}_{i^k}} \tilde{F}_{i^k}(\mathbf{x}_{i^k}; \mathbf{x}^{k-\mathbf{d}^k}) + g_{i^k}(\mathbf{x}_{i^k});$   (3.3)
(S.4) $\mathbf{x}_{i^k}^k$ is acquired;
(S.5) The block $i^k$ is updated:
          $\mathbf{x}_i^{k+1} = \begin{cases} \mathbf{x}_i^k + \gamma^k(\widehat{\mathbf{x}}_i(\mathbf{x}^{k-\mathbf{d}^k}) - \mathbf{x}_i^k), & \text{if } i = i^k \\ \mathbf{x}_i^k & \text{if } i \neq i^k; \end{cases}$   (3.4)
(S.6) Update $k \leftarrow k + 1$, and go to (S.1).
```

• **Discussion on Algorithm 4 -** Several comments are in order.

1. *On the generality of the model:* Algorithm 4 represents a gamut of asynchronous schemes and architectures, all captured in an abstract and unified way by the stochastic process modeling the specific mechanism of generation of the delay vectors $\mathbf{d}^k$ and indices i^k of the blocks to updates. For concreteness, we show next how Algorithm 4 customizes when modeling asynchrony in shared-memory and message passing-based architectures.

Example 1: Shared-memory systems. Consider a shared-memory system wherein multiple cores update in an asynchronous fashion blocks of the vector $\mathbf{x}$, stored in a shared memory. An iteration $k \to k+1$ of Algorithm 4 is triggered when a core writes the (block) update $\mathbf{x}_{i^k}^{k+1}$ in the shared memory (Step 4). Note that the cores need not know the global iteration index k. No memory lock is assumed, implying that components of the variables may be written by some cores while other components

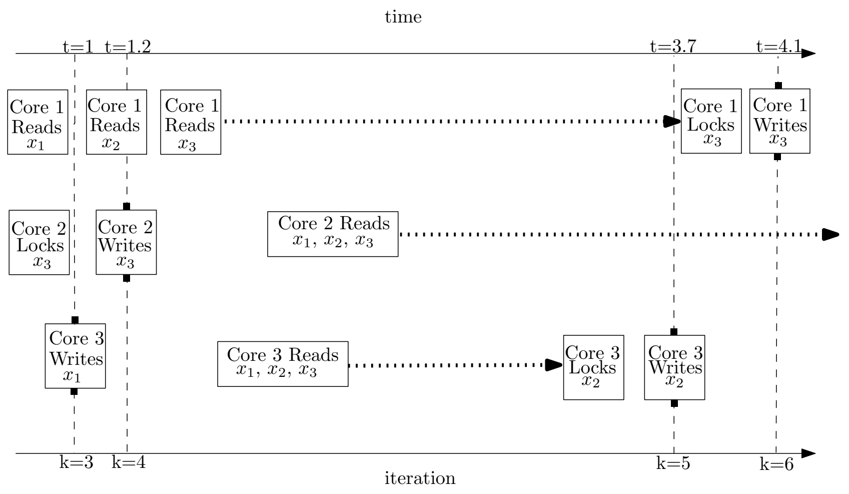

Figure 3.1. AsyFLEXA modeling block asynchronous updates in a shared-memory system: three cores, vector variables $\mathbf{x} \in \mathbb{R}^3$, scalar blocks ($n_i = 1$, for all i).

are simultaneously read by others. This *inconsistent read* produces vectors $\mathbf{x}^{k-\mathbf{d}^k} = [x_i^{k-d_i^k}]_{i=1}^N$, to be used in the computation of $\widehat{\mathbf{x}}_{i^k}$ (Step 2), whose (block) component $\mathbf{x}_i^{k-d_i^k}$ is a (possibly) delayed version of block i read by the core that is going to perform the update. Note that, while $\mathbf{x}_i^{k-d_i^k}$ existed in the shared memory at some point in time, the entire delayed vector $\mathbf{x}^{k-\mathbf{d}^k}$ might have not at any time. Also, in Step 4, it is tacitly assumed that the update of a block is *atomic* (the block is written in the shared memory as a whole) and while a core is writing that block no other core can modify the same block. This is minor requirement, which can be easily enforced in modern architectures either by a block-coordinate look or using a dual-memory writing approach, see [55, Section 1.2.1].

Figure 3.1 shows few iterations of the algorithm dynamics in the asynchronous setting described above. The (continuous) time when operations (reading, writing, computation) are performed is indicated in the top horizontal axes whereas the global (discrete) iteration counter is reported in the bottom axes. The asynchronous updates happen as follows. At iteration $k = 3$, Core 3 writes x_1^3; therefore, $\mathbf{x}^3$ differs from $\mathbf{x}^2$ in the first component. Core 2 locks x_3^2 to quickly read it and perform the linear combination with $\widehat{x}_3(\mathbf{x}^{3-\mathbf{d}^3})$ [cf. (3.4)], and updates x_3; therefore $\mathbf{x}^4$ differs from $\mathbf{x}^3$ in just the 3rd component. Note that core 2 reads x_3^2 which is equal to x_3^3, so $d_3^3 = 0$; this is because between the lock and the writing of core 2, no other cores wrote x_3 (core 3 updates x_1). At iteration $k = 5$, core 3 writes x_2^5. In this case $\mathbf{x}^{4-\mathbf{d}^4}$, used to compute $x_2^5 = (1-\gamma^k)x_2^4 + \gamma^k \widehat{x}_2(\mathbf{x}^{4-\mathbf{d}^4})$, is exactly equal to $\mathbf{x}^4$, since core 3 reads the vector entirely after the last update, so $\mathbf{d}^4 = \mathbf{0}$. A different situation happens when iteration $k = 6$ is triggered by core 1: the vector used by the core to perform its update is $\mathbf{x}^{5-\mathbf{d}^5}$, which is such that $x_1^{5-\mathbf{d}^5} = x_1^2 \neq x_1^3$, $x_2^{5-\mathbf{d}^5} = x_2^2$, and $x_3^{5-\mathbf{d}^5} = x_3^4 \neq x_3^2$; therefore, $\mathbf{x}^{5-\mathbf{d}^5}$ never existed in the shared memory at any time. It can be seen that the vector of delays at $k = 6$ reads $\mathbf{d}^5 = (3, 1, 0)^{\mathrm{T}}$. Note that the delay vector $\mathbf{d}^k$ used at a given iteration may not be unique: different values for the components d_i^k may produce the same delayed vector $\mathbf{x}^{k-\mathbf{d}^k}$. For instance, in the example above, the vector $(3, 4, 0)^{\mathrm{T}}$ could have been used in place of $\mathbf{d}^5$.

Example 2: Message-passing systems. Consider the following distributed computational architecture as studied for example in [9] and [73]: multiple workers (e.g., clouds, cluster computers) are connected through a (directed) graph, modeling the communication pattern among them. All the workers aim at cooperatively solve Problem (P), in an asynchronous fashion. Each worker i knows F, g_i and $\mathcal{X}_i$ in (P), and is in charge of the update of the subset of variables $\mathbf{x}_i$; this means that only worker i can update $\mathbf{x}_i$ (possibly a subset of them at each iteration). However, to solve its subproblem, each worker also needs to build a local estimate, possibly outdated, of the variables of the other workers. This can be done via suitably designed communication protocols, involving information exchange between immediate neighboring nodes. Note that workers can perform their local computations as well as communicate (possibly in parallel) with their neighbors at any time, without any form of coordination or centralized scheduling. In this setting, $\mathbf{x}^{k-\mathbf{d}^k}$ corresponds to the most recent information a worker has received from the others at the time of its update. Algorithm 4 can thus model also asynchronous updates and communications in such distributed systems with no shared memory.

There are of course other message-passing-based architectures where the proposed asynchronous model cannot be directly applied. For instance, this is the case of some multi-agent systems wherein the cost function V is the sum of the local objectives of the agents, and each agent does not have access to the function of the others.

2. *On the surrogate functions* $\tilde{F}_i$. A degree of freedom offered by the proposed framework is the choice of the surrogate function used in the subproblems (3.3) solved by the workers at each iteration. We consider the following general class of surrogate functions (we recall that $\nabla \tilde{F}_i$ denotes the partial gradient of $\tilde{F}_i$ with respect to the first argument).

Assumption B (On the surrogate functions $\tilde{\mathsf{F}}_i$'s). Each $\tilde{F}_i : \mathcal{X}_i \times \mathcal{X} \to \mathbb{R}$, is chosen so that:

(B1) $\tilde{F}_i(\cdot; \mathbf{y})$ is C^1 and τ-strongly convex on $\mathcal{X}_i$, for all $\mathbf{y} \in \mathcal{X}$;

(B2) $\nabla \tilde{F}_i(\mathbf{y}_i; \mathbf{y}) = \nabla_{\mathbf{y}_i} F(\mathbf{y})$, for all $\mathbf{y} \in \mathcal{X}$;

(B3) $\nabla \tilde{F}_i(\mathbf{y}; \cdot)$ is L_A-Lipschitz continuous on $\mathcal{X}$, for all $\mathbf{y} \in \mathcal{X}_i$.

Assumption B is the same set of assumptions we already considered in Section 2.2, rewrote here for ease of clarity. When we will be analyzing the iteration complexity of AsyFLEXA, we will also require the following additional assumption on the surrogate functions $\tilde{F}_i$:

Assumption B* (On the surrogate functions $\tilde{F}_i$'s). Each $\tilde{F}_i : \mathcal{X}_i \times \mathcal{X} \to \mathbb{R}$, is chosen so that:

(B4) $\nabla \tilde{F}_i(\cdot; \mathbf{y})$ is L_B-Lipschitz continuous on $\mathcal{X}_i$, for all $\mathbf{y} \in \mathcal{X}$.

As we already discussed in Section 2.2, the surrogate $\tilde{F}_i(\cdot; \mathbf{x}^k)$ should be regarded as a (simple) strongly convex local approximation of F around $\mathbf{x}^k \in \mathcal{X}$, that preserves the first order properties of F at $\mathbf{x}^k$. Indeed, the key assumption B2 simply stipulates the value of the gradient of $\tilde{F}_i(\cdot; \mathbf{x}^k)$ computed at $\mathbf{x}_i^k$ is equal to $\nabla_{\mathbf{x}_i} F(\mathbf{x}^k)$. Finding a surrogate $\tilde{F}_i$ that satisfies Assumption B-B* is in general non difficult; in any case, one can always choose $\tilde{F}_i(\mathbf{x}_i; \mathbf{x}^k) = \left\langle \nabla_{\mathbf{x}_i} F(\mathbf{x}^k), \mathbf{x}_i - \mathbf{x}_i^k \right\rangle + \alpha \|\mathbf{x}_i - \mathbf{x}_i^k\|_2^2$, where α is a positive constant, which leads to the classical proximal-gradient update. However, having the possibility to use a different $\tilde{F}_i$ may be useful to exploit some potential structure in the problem; of course, a trade-off is expected: the more complex the $\tilde{F}_i$, the more information will be retained in $\widehat{\mathbf{x}}_i(\mathbf{x}^k)$, but also the more intensive its computation is expected to be. On the other hand, the solution of more complex subproblems will in general decrease the number of information exchanges in the system, which may be a key advantage in many applications. Valid instances of $\tilde{F}_i$'s going beyond the proximal-gradient has already been discussed in Section 2.2.

3.3 AsyFLEXA: Probabilistic Model

In this section, we complete the description of AsyFLEXA, introducing the probabilistic model underlying the generation of the pairs index-delays.

Given Problem (P) and an initial point $\mathbf{x}^0$, the pair $(i^k, \mathbf{d}^k)$ in Step 1 of Algorithm 4, for each k, is a realization of a random vector $\underline{\boldsymbol{\omega}}^k \triangleq (\underline{i}^k, \underline{\mathbf{d}}^k)$, taking values on $\mathcal{N} \times \mathcal{D}$, where $\mathcal{D}$ is the set of all possible delay vectors. We anticipate that all the delays d_i^k are assumed to be bounded (Assumption C below), i.e., $d_i^k \leq D$, for all k and i. Hence, $\mathcal{D}$ is contained in the set of all possible N-length vectors whose components are integers between 0 and D. Let Ω be the sample space of all the sequences $\omega \triangleq \{(i^k, \mathbf{d}^k)\}$.[1] We will use the following shorthand notation: we set $\underline{\boldsymbol{\omega}}^{0:k} \triangleq (\underline{\boldsymbol{\omega}}^0, \underline{\boldsymbol{\omega}}^1, \ldots, \underline{\boldsymbol{\omega}}^k)$ (the first $k+1$ random variables); $\boldsymbol{\omega}^{0:k} \triangleq (\boldsymbol{\omega}^0, \boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^k)$ ($k+1$ possible values for the random variables $\underline{\boldsymbol{\omega}}^{0:k}$); and $\boldsymbol{\omega}_{0:k} \triangleq (\boldsymbol{\omega}_0, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_k)$ (the first $k+1$ elements of ω). We introduce next the probability space that will be used to build our probabilistic model.

The sample space is Ω. To define a σ-algebra on Ω, we consider, for $k \geq 0$ and every $\boldsymbol{\omega}^{0:k} \in \mathcal{N} \times \mathcal{D}$, the cylinder

$$C^k(\boldsymbol{\omega}^{0:k}) \triangleq \{\omega \in \Omega : \boldsymbol{\omega}_{0:k} = \boldsymbol{\omega}^{0:k}\},$$

[1]With a slight abuse of notation, we denote by $\boldsymbol{\omega}_k$ the k-th element of the sequence $\omega \in \Omega$, and by $\boldsymbol{\omega}^k$ the value taken by the random variable $\underline{\boldsymbol{\omega}}^k$ over ω, i.e. $\underline{\boldsymbol{\omega}}^k(\omega) = \boldsymbol{\omega}^k$.

i.e., $C^k(\boldsymbol{\omega}^{0:k})$ is the subset of Ω of all sequences ω whose first k elements are $\boldsymbol{\omega}^0, \ldots \boldsymbol{\omega}^k$. Let us denote by $\mathcal{C}^k$ the set of all possible $C^k(\boldsymbol{\omega}^{0:k})$ when $\boldsymbol{\omega}^t$, $t = 0, \ldots, k$, takes all possible values; note, for future reference, that $\mathcal{C}^k$ is a partition of Ω. Denoting by $\sigma\left(\mathcal{C}^k\right)$ the σ-algebra generated by $\mathcal{C}^k$, define for all k,

$$\mathcal{F}^k \triangleq \sigma\left(\mathcal{C}^k\right) \qquad \text{and} \qquad \mathcal{F} \triangleq \sigma\left(\cup_{t=0}^{\infty}\mathcal{C}^t\right). \tag{3.5}$$

We have $\mathcal{F}^k \subseteq \mathcal{F}^{k+1} \subseteq \mathcal{F}$ for all k. The latter inclusion is obvious, the former derives easily from the fact that any cylinder in $\mathcal{C}^{k-1}$ can be obtained as a finite union of cylinders in $\mathcal{C}^k$.

The desired probability space is fully defined once $\mathbb{P}(C^k(\boldsymbol{\omega}^{0:k}))$, the probabilities of all cylinders, are given. These probabilities should satisfy some very natural, minimal consistency properties, namely: (i) the probabilities of the union of a finite number of disjoint cylinders should be equal to the sum of the probabilities assigned to each cylinder; and (ii) suppose that a cylinder $C^k(\boldsymbol{\omega}^{0:k})$ is contained in the union U of a countably infinite number of other cylinders, then $\mathbb{P}(C^k(\boldsymbol{\omega}^{0:k})) \leq P(U)$. Suppose now that such a $\mathbb{P}$ is given. Classical results (see, e.g., [74, Theorem 1.53]) ensure that one can extend these probabilities to a probability measure P over $(\Omega, \mathcal{F})$, thus defining our working probability space $A \triangleq (\Omega, \mathcal{F}, P)$. By appropriately choosing the probabilities of the cylinders, we can model in a unified way many cases of practical interest; several examples are given in Section 3.3.1.

Given A, we can finally define the discrete-time, discrete-value stochastic process $\underline{\boldsymbol{\omega}}$, where $\{\underline{\boldsymbol{\omega}}^k(\omega)\}$ is a sample path of the process. The k-th entry $\underline{\boldsymbol{\omega}}^k(\omega)$ of $\underline{\boldsymbol{\omega}}(\omega)$−the k-th element of the sequence ω−is a realization of the random vector $\underline{\boldsymbol{\omega}}^k = (\underline{i}^k, \underline{\mathbf{d}}^k) : \Omega \mapsto \mathcal{N} \times \mathcal{D}$. This process fully describes the evolution of Algorithm 1. Indeed, given an instance of Problem (P) and a starting point, the trajectories of the variables $\mathbf{x}^k$ and $\mathbf{x}^{k-\mathbf{d}^k}$ are completely determined once a sample path $\{(i^k, \mathbf{d}^k)\}$ is drawn from $\underline{\boldsymbol{\omega}}$.

Note that the joint probability

$$p_{\underline{\boldsymbol{\omega}}^{0:k}}(\boldsymbol{\omega}^{0:k}) \triangleq \mathbb{P}(\underline{\boldsymbol{\omega}}^{0:k} = \boldsymbol{\omega}^{0:k})$$

is simply the probability of the corresponding cylinder: $C^k(\boldsymbol{\omega}^{0:k})$. We will often need to consider the conditional probabilities $p((i,\mathbf{d})\,|\,\boldsymbol{\omega}^{0:k}) \triangleq \mathbb{P}(\underline{\boldsymbol{\omega}}^{k+1} = (i,\mathbf{d})|\underline{\boldsymbol{\omega}}^{0:k} = \boldsymbol{\omega}^{0:k})$. Note that we have

$$p((i,\mathbf{d})\,|\,\boldsymbol{\omega}^{0:k}) = \frac{\mathbb{P}(C^{k+1}(\boldsymbol{\omega}^{0:k+1}))}{\mathbb{P}(C^k(\boldsymbol{\omega}^{0:k}))}, \tag{3.6}$$

where we tacitly assume $p((i,\mathbf{d})\,|\,\boldsymbol{\omega}^{0:k}) = 0$, if $\mathbb{P}(C^k(\boldsymbol{\omega}^{0:k})) = 0$. We remark that these probabilities need not be known in practice to implement the algorithm. They are instead determined based on the particular system (hardware architecture, software implementation, asynchrony, etc.) one is interested to model. Here, we make only some minimal assumptions on these probabilities and stochastic model, as stated next.

Assumption C (On the probabilistic model). Given AsyFLEXA and the stochastic process $\underline{\boldsymbol{\omega}}$, suppose that

(C1) There exists a $D \geq 0$, such that $d_i^k \leq D$, for all i and k;

(C2) For all $i \in \mathcal{N}$ and $\omega \in \Omega$, there exists at least one $t \in [0,\ldots,B]$, with $B > 0$, such that

$$\sum_{\mathbf{d}\in\mathcal{D}} p((i,\mathbf{d})\,|\,\boldsymbol{\omega}^{0:k+t-1}) \geq p_{\min}, \quad \text{if} \quad p_{\underline{\boldsymbol{\omega}}^{0:k+t-1}}(\boldsymbol{\omega}^{0:k+t-1}) > 0,$$

for some $p_{\min} > 0$;

(C3) $d_{i^k}^k = 0$, for any $k \geq 0$.

These are quite reasonable assumptions, with very intuitive interpretations. C1 just limits the age of the old information used in the updates. Condition C2 guarantees that every B iterations each block-index i has a non negligible positive probability to be updated. These are minimal requirements that are satisfied in practically all computational environments. The condition $d_{i^k}^k = 0$ means that when a worker updates the i^k-th block, it uses the most recent value of that block-variable. This assumption is automatically satisfied, e.g., in a message passing-based system or in a shared memory-based architecture if the variables are partitioned and assigned to different cores (see Example 6 in Section 3.3.1). If instead all the cores can update all variables, $d_{i^k}^k = 0$ can be simply enforced by a software lock on the i^k-th block of the shared memory: once a core c has read a block-variable $\mathbf{x}_{i^k}$, no other core can change it, until c has performed its update. Note that in practice it is very unlikely that this lock affects the performance of the algorithm, since usually the number of cores is much smaller than the number of block-variables. Actually, in some systems, this lock can bring in some benefits. For instance, consider two cores sharing all variables, with one core much faster than the other. A lock on $\mathbf{x}_{i^k}^k$ will prevent potentially much older information to overwrite most recent updates of the faster core. Note also that conditions similar to C3 are required by all block asynchronous methods in the literature but [55]: they take the form of locking the variable to update before performing a prox operation [54].

We will use Assumption C when considering AsyFLEXA equipped with a fixed stepsize γ. When we are interested in proving convergence for diminishing stepsizes rules we will replace Assumption C with the following set of similar assumptions.

Assumption C' (On the probabilistic model). Given AsyFLEXA and the stochastic process $\underline{\boldsymbol{\omega}}$, suppose that

(C1') There exists a $D \geq 0$, such that $d_i^k \leq D$, for all i and k;

(C2') For all $i = 1, \ldots, N$ and $\boldsymbol{\omega}^{0:k-1}$ such that $p_{\underline{\boldsymbol{\omega}}^{0:k-1}}(\boldsymbol{\omega}^{0:k-1}) > 0$, it holds

$$\sum_{\mathbf{d} \in \mathcal{D}} p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k-1}) \geq p_{\min},$$

for some $p_{\min} > 0$;

(C3') It holds that

$$\mathbb{P}\left(\left\{\omega \in \Omega \,:\, \liminf_{k \to \infty} \, p(\boldsymbol{\omega} \,|\, \boldsymbol{\omega}^{0:k-1}) > 0\right\}\right) = 1.$$

C1' is a duplicate of C1, repeated here for ease of reference. C2 guarantees that at each iteration every block-index i has a non negligible positive probability to be updated. C3 simply says, roughly speaking, that the probability of the event that comprises all ω such that for an infinite number of iterations the algorithm picks-up a pair index-delay whose probability decreases to zero, must have zero probability. Very loosely speaking, C3 requires that the overall probability of increasingly unlikely possible evolutions of the algorithm be zero. A very simple case in which C2 and C3 are automatically satisfied is when the probabilities $p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k})$ are independent of history $\boldsymbol{\omega}^{0:k}$, for all k.

Remark The knowledge of the probability space A is by no means required from the workers to perform the updates. One need not even specify explicitly the probability distribution; it is sufficient to show that a probability space A satisfying Assumption C exists for the specific system (e.g., computational architecture, asynchronous protocol, etc.) under consideration. We show next how to do so for several schemes of practical interest.

3.3.1 Examples and special cases

The proposed model encompasses a gamut of practical schemes, some of which are discussed next. It is of note that our framework allows us to analyze in a unified way not only randomized methods, but also deterministic algorithms.

1. Deterministic sequential cyclic BCD: In a deterministic, cyclic method there is only one core that cyclically updates all block-variables; for simplicity we assume the natural order, from 1 to N. Since there is only one core, the reading is always consistent and there are no delays: $\mathcal{D} = \{\mathbf{0}\}$. To represent the cyclic choice it is now enough to assign probability 1 too all cylinders of the type

$$C^k = \{\omega : \omega_0 = (1, \mathbf{0}), \omega_1 = (2, \mathbf{0}), \ldots, \omega_k = ((k \mod N) + 1, \mathbf{0})\}$$

and probability zero to all others. It is easy to see that Assumption C (or Assumption C') is satisfied. This can be seen as a probabilistic model of the deterministic algorithm in [51]. The consequence however is that, by Theorems 3.4.1-3.4.2, convergence can be claimed only in a probabilistic sense (a.s.). This is not surprising, as we are describing a deterministic algorithm as limiting case of a probabilistic model.

2. Randomized sequential BCD: Suppose now that there is only one core selecting at each iteration randomly an index i, with a positive probability. Therefore, at each iteration, $\mathbf{x}^{k-\mathbf{d}^k} = \mathbf{x}^k$ or, equivalently, $\mathcal{D} = \{\mathbf{0}\}$. This scheme can be described by a stochastic process, where the cylinders are assigned arbitrary probabilities but satisfying all the conditions given in previous subsection.

3. Randomized parallel BCD: Suppose that there are C cores and the block-variables are partitioned in C groups $I_1, I_2, \ldots, I_C$; each set I_c is assigned to one core only, say c. Hence, if core c performs the update at iteration k, all variables $i \in I_c$ satisfy $d_i^k = 0$. Denote by $\mathbf{0}(c)$, $\mathbf{1}(c)$, $\ldots$, and $(\boldsymbol{C-1})(c)$, $c = 1, \ldots, C$, the N-length vectors whose components are zeros in the positions of the block-variables in the set I_c and all $0, 1, \ldots, C-1$ in the other positions, respectively. Set $\mathcal{D} = \{\mathbf{0}(c), \mathbf{1}(c), \ldots, (\boldsymbol{C-1})(c),\ c = 1\ldots, C\}$, and denote by c^k the core performing the update at iteration k. Assign to the cylinders the following probabilities: $\forall i^0, i^1, \ldots, i^{2C-1}, \ldots \in \mathcal{N}$,

$$\mathbb{P}(C^0((i^0, \mathbf{0}(c^0)))) = 1/N,$$

$$\mathbb{P}(C^1((i^0, \mathbf{0}(c^0)), (i^1, \mathbf{1}(c^1)))) = 1/N^2,$$

$$\ldots$$

$$\mathbb{P}(C^{C-1}((i^0, \mathbf{0}(c^0)), (i^1, \mathbf{1}(c^1)), \ldots, (i^{C-1}, \boldsymbol{C-1}(c^{C-1})))) = 1/N^C,$$

$$\mathbb{P}(C^C((i^0, \mathbf{0}(c^0)), (i^1, \mathbf{1}(c^1)), \ldots, (i^{C-1}, \boldsymbol{C-1}(c^{C-1})), (i^C, \mathbf{0}(c^C)))) = 1/N^{C+1},$$

$$\mathbb{P}(C^{C+1}((i^0, \mathbf{0}(c^0)), (i^1, \mathbf{1}(c^1)), \ldots, (i^C, \mathbf{0}(c^C)), (i^{C+1}, \mathbf{1}(c^{C+1})))) = 1/N^{C+2},$$

$$\ldots$$

$$\mathbb{P}(C^{2C-1}((i^0, \mathbf{0}(c^0)), (i^1, \mathbf{1}(c^1)), \ldots, (i^{2C-1}, \boldsymbol{C-1}(c^{2C-1})))) = 1/N^{2C},$$

$$\ldots$$

In words, in the first C iterations (from $k = 0$ to $k = C - 1$), all updates are performed using the same vector $\mathbf{x}^{k-\mathbf{d}^k} = \mathbf{x}^0$; and at each iteration any index has uniform probability to be selected. This situation is then repeated for the next C iterations, this time using $\mathbf{x}^{k-\mathbf{d}^k} = \mathbf{x}^C$, and so on. This model clearly corresponds to a randomized parallel block-coordinate descent method wherein C cores update C block-variables chosen uniformly at random. Note that Assumption C (or Assumption C') is trivially satisfied.

The example above clearly shows that defining probabilities by using the cylinders can be quite tedious even in simple cases. Using (3.6) we can equivalently define the probabilistic model by specifying the conditional probabilities $p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k})$, which is particularly convenient when at every iteration k the probability that $\underline{\boldsymbol{\omega}}^k$ takes value $(i, \mathbf{d})$ is independent of $\boldsymbol{\omega}^{0:k-1}$. We exemplify this alternative approach in the following examples.

4. Asynchronous BCD in shared memory systems: Consider a generic shared memory system, under Assumption C3. Then, the set $\mathcal{D}$ is given by all the N-length vectors whose components are non negative integers between 0 and D. Suppose that, at every k, all cores select an index uniformly at random, but the probabilities associated with the delays can be different. Then, for every $k \geq 0$, given $\boldsymbol{\omega}^{0:k}$, and $i \in \mathcal{N}$, we have

$$\sum_{\mathbf{d} \in \mathcal{D}} p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k}) = \frac{1}{N}.$$

This setting is consistent with the one studied in [49, 50, 54, 62].

Our probabilistic model however is more general than that of [49, 50, 54, 62]. For instance, differently from [49, 50, 54, 62], we can easily model scenarios wherein $\sum_{\mathbf{d} \in \mathcal{D}} p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k})$ are not uniform and/or depend on the iteration and/or on the history of the algorithm. This possibility has important ramifications, since the assumption that the indices are selected uniformly at random is extremely strong and unrealistic. In fact, it is satisfied only if all cores have the same computational power and have access to all variables. This is not the case, in most of practical settings.

For instance, consider a computational architecture composed of two CPUs sharing all the variables, with one CPU much faster than the other. If the recent history exhibits iterations with a small value of $\|\mathbf{d}^k\|_\infty$, then it is more likely that the slower core will perform the next update, and vice versa. Similar situations are expected also in other common settings, such as shared memory systems with variable partitioning (see Example 5 below) and message passing-based architectures. This clearly shows that our model captures realistic architectures more faithfully.

5. Asynchronous BCD in shared memory systems with variable partitioning: Consider the setting as in Example 4, but now partition the variables across cores, as described in Example 3. This is the configuration most often used in numerical experiments, since it has proven to be most effective in practice; it also models a message passing architecture. In order to satisfy C3, it is enough to set, for all $\boldsymbol{\omega}^{0:k}$ and $i \in I_c$,

$$p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k}) = 0, \text{ if some } \mathbf{d}_j \neq \mathbf{0},\, j \in I_c.$$

A variant of this setting is the *without replacement* updating scheme considered in the numerical experiments of [54]: the block-variables are partitioned among the cores and, at each "epoch", variables in each partition are first randomly shuffled and then updated cyclically by the core. This choice of the updates was shown to be numerically very effective. While [54] cannot provide any theoretical analysis of such a scheme, we can easily cover this case by just merging this example with Example 2.

Other examples: Several other examples can be considered, which we omit because of space limitation. Here we only mention that it is quite straightforward to analyze by our model also "hybrid" systems, which combine somehow two or more examples described above. For instance, consider a cluster computer system wherein the optimization variables are partitioned across the machines; let I_m be the set of variables controlled by machine m and stored in its internal shared memory. The update of the variables in I_m is performed by the processors/cores of machine m according to some shared memory-based asynchronous scheme (e.g., subject to inconsistent read). The information on the variables not in I_m is instead updated through communication with the other processors (message passing)

3.4 AsyFLEXA: Convergence Results

We present now our main convergence theorems, in the setting described so far. The extension to the case of nonconvex constraints is addressed in Sec. 3.5.

As already discussed in the Introduction of this Chapter, we consider for AsyFLEXA both fixed and diminishing stepsizes.

3.4.1 AsyFLEXA with fixed stepsize γ

In the rest of this Section, we will use $\|M_V(\mathbf{x})\|_2$ as a measure of optimality, with

$$M_V(\mathbf{x}) \triangleq \mathbf{x} - \underset{\mathbf{y}\in\mathcal{X}}{\arg\min} \left\{ \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x} \rangle + \sum_{i=1}^{N} g_i(\mathbf{y}_i) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}. \tag{3.7}$$

This is a valid measure of stationarity because $M_V(\mathbf{x})$ is continuous and $\|M_V(\mathbf{x})\|_2 = 0$ if and only if $\mathbf{x}$ is a stationary solution of Problem (P).

To state our major convergence result, we need to introduce first the following intermediate definitions. Recalling the definition of B as in Assumption C2, let $\underline{\mathcal{K}}_i^k$ be the (random) set of iterations between $k-D$ and $k+B-1$ at which the block-variable i has been updated, $\mathcal{K}_i^k \triangleq \{t \in [k-D; k+B-1] \,|\, \underline{i}^t = i\}$, while $\bar{\underline{\mathcal{K}}}_i^k$ is the subset of $\underline{\mathcal{K}}_i^k$ containing only the elements of $\underline{\mathcal{K}}_i^k$ (iterations) between $k-D$ and $k-1$. Our convergence results leverage a Lyapunov function $\tilde{V}$ that suitably combines present and past iterates, and it is defined as

$$\tilde{V}(\mathbf{x}^k, \ldots, \mathbf{x}^{k-D}) \triangleq V(\mathbf{x}^k) + D\frac{L}{2}\left(\sum_{l=k-D}^{k-1} (l-(k-1)+D)\,\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2^2\right), \tag{3.8}$$

where it is understood that $\mathbf{x}^l = \mathbf{x}^0$, if $l < 0$; therefore, $\tilde{V}$ is well defined for any $k \geq 0$. Note that, by this convention, $\tilde{V}(\mathbf{x}^0, \ldots, \mathbf{x}^{0-D}) = V(\mathbf{x}^0)$. Furthermore, we also have $V^\star \triangleq \min_{\mathbf{x}\in\mathcal{X}} V(\mathbf{x}) \le \min_{[\mathbf{x}^k,\ldots,\mathbf{x}^{k-D}]\in\mathcal{X}^{D+1}} \tilde{V}(\mathbf{x}^k,\ldots,\mathbf{x}^{k-D})$. We are now ready to state our major convergence result.

Theorem 3.4.1 *Given Problem* (P) *under Assumptions A-B-B***-C, along with the stochastic process* $\underline{\boldsymbol{\omega}}$*. Let* $\{\underline{\mathbf{x}}^k\}$ *be the sequence generated by AsyFLEXA. Suppose that the stepsize* γ^k *is fixed* $\gamma^k = \gamma \in (0;1]$ *for any* $k \ge 0$ *and such that*

$$\gamma < \frac{\tau}{L + \frac{D^2 L}{2}}. \tag{3.9}$$

Define T_ϵ *to be the first iteration such that* $\mathbb{E}\left(\|M_V(\underline{\mathbf{x}}^k)\|_2^2\right) \le \epsilon$*. Then, the following hold:*

(a) Every limit point of $\{\underline{\mathbf{x}}^k\}$ *is a stationary solution of* (P) *a.s.;*

(b) The sequence of objective function values $\{V(\underline{\mathbf{x}}^k)\}$ *converges a.s.;*

$$\begin{aligned}(c)\quad T_\epsilon \le\ & \frac{C_1(\gamma,D)(B+1)(V(\mathbf{x}^0)-V^\star)}{\epsilon} \\ &+ \frac{C_2(\gamma,D)\gamma^2}{\epsilon}\underbrace{\sum_{k=0}^{T_\epsilon}\mathbb{E}\left(\sum_{i=1}^{N}\underline{M}_i^k\sum_{t\in\underline{\mathcal{K}}_i^k}\left(\tilde{V}(\underline{\mathbf{x}}^t,\ldots,\underline{\mathbf{x}}^{t-D}) - \tilde{V}(\underline{\mathbf{x}}^{t+1},\ldots,\underline{\mathbf{x}}^{t+1-D})\right)\right)}_{B},\end{aligned} \tag{3.10}$$

where:

$$C_1(\gamma,D) \triangleq \frac{2\left(1+(1+L_B)(1+L_A+L_B)+\gamma^2 N p_{min}\alpha^{-1}(1+(L+1)^2)\right)}{\gamma\left(\tau-\gamma\left(L+\frac{D^2L}{2}\right)\right)(p_{min}-p_{min}\alpha)}, \tag{3.11}$$

$$C_2(\gamma,D) \triangleq \frac{2BL_A(1+L_A+L_B)}{\gamma\left(\tau-\gamma\left(L+\frac{D^2L}{2}\right)\right)(p_{min}-p_{min}\alpha)}, \tag{3.12}$$

α *is an arbitrary fixed value in* $(0;1)$*,* $\underline{M}_i^k \triangleq \max_{l=k,\ldots,k+B}|\underline{\bar{\mathcal{K}}}_i^l|$.

Proof See the Appendix of this Chapter. ■

The theorem states that convergence to stationary points occurs a.s. (the objective function values converge too); it also gives an estimate of the number of iterations T_ϵ necessary to enforce $\mathbb{E}\left(\|M_V(\mathbf{x}^k)\|_2^2\right) \le \epsilon$. Convergence is guaranteed if, in particular, the stepsize is sufficiently small; the bound (3.9) makes this precise. Note that if the method is synchronous, $D = 0$, the bound in (3.9), going like the inverse of the Lipschitz constant, becomes the renowned conditions used in many synchronous (proximal-gradient-like) schemes. The term $D^2/2$ in the denominator of (3.9) should then be seen as the price to pay for asynchrony: the larger the possible delay D, the smaller γ should be to tolerate such delays. Roughly speaking, this means that the more chaotic the computational environment, the more conservative the step should be, and consequently the smaller the steps of the algorithm are.

The interpretation of the bound (3.10) is not immediate, because of the presence of the term B; we now elaborate on it. If there exists a (deterministic) bound C on $\underline{M}_i^k$, i.e., $\underline{M}_i^k \leq C$ for all k and i, then one can write

$$\begin{aligned} B &\leq C\mathbb{E}\left(\sum_{k=0}^{T_\epsilon}\sum_{i=1}^{N}\sum_{t\in\underline{\mathcal{K}}_i^k}\left(\tilde{V}(\underline{\mathbf{x}}^t,\ldots,\underline{\mathbf{x}}^{t-D}) - \tilde{V}(\underline{\mathbf{x}}^{t+1},\ldots,\underline{\mathbf{x}}^{t+1-D})\right)\right) \\ &\leq C(B+D)(V(\mathbf{x}^0) - V^\star). \end{aligned}$$

Therefore, (3.10) can be upper bounded as

$$T_\epsilon \leq \Big[C_1(\gamma, D)\cdot(B+1) + C_2(\gamma, D)\cdot\gamma^2\cdot C\cdot(B+D)\Big]\frac{V(\mathbf{x}^0) - V^\star}{\epsilon}. \tag{3.13}$$

Recalling the definition of $\underline{M}_i^k$ and that $|\bar{\underline{\mathcal{K}}}_i^k|$ is a random variable counting the number of times the index i has been updated in the iteration window $[k-D, k-1]$, $\underline{M}_i^k \leq D$ always holds; therefore, one can always take $C = D$. Of course this is a very rough approximation: it is hard to expect that in a given time window always the same variable, $\bar{i}$, is updated and, even if this were the case, all other $\underline{M}_i^k$, $i \neq \bar{i}$, would be 0 and not D. Consider for example the commonly analyzed "uniform case" where the processing of every block-variable requires the same time. In this case one can reasonably take $C = 1$ in (3.13) *independently of the number of workers.*

This intuition is corroborated by our experiments, which are summarized in Table 3.1. AsyFLEXA was ran on two different architectures, namely: a shared-memory system with 10 cores, and a message passing architecture composed of two nodes, with 10 cores each. Two LASSO problems with 10,000 variables each were considered,

Table 3.1.
Average delay and maximum delay D for AsyFLEXA, ran on a multi-core machine and on a message passing system.

	Average Delay	D	# of cores
Multi-core Machine: Balanced Workload	1.11	3	10
Multi-core Machine: Unbalanced Workload	2.58	28	10
Message Passing System: Balanced Workload	1.87	30	10 per node
Message Passing System: Unbalanced Workload	3.01	36	10 per node

and the variables were equally partitioned across the workers. In the first LASSO instance, the Hessian matrix was a dense matrix, which models situations where the workload is equally distributed across the workers. In the second LASSO problem, the Hessian matrix had many sparse rows, to create some unbalancedness in the workers' workload. Table 3.1 shows the empirical average delay (the average is taken over the components of the delay vector and time) and the maximum delay D, estimated in 500 epochs (one epoch is triggered when all blocks have been updated once). As expected, D is much larger than the experienced average delay, confirming that (3.13) with $C = D$ is a very conservative bound. While C can always be pessimistically upper bounded by D, a tighter value can be found by tailoring the analysis to the specific problem and architecture under consideration.

We remark the importance of the use, in the complexity analysis, of the $\underline{M}_k^i$, counting the number of times the index i has been updated in a certain iteration window. The use of these variables seems to be a new feature of our analysis. While getting a sharp estimate for the upper bound C may be difficult in practice, the bound (3.10) gives a good insight into the elements that really influence the algorithm, showing that what really matters, in some expressions appearing in (3.10), is not D, but the usually much smaller number of times the blocks are actually updated. The use of these variables allows us to get a sharper bound with respect to the case in which one sets $C = D$. From this point of view, we believe that typical upper bounds, as those obtained in [49,50,53–55], where D is the only considered "delay", do not give an accurate description of the actual worst-case scenario. Finally, we note that from our numerical tests it seems that the theoretical upper bound on the stepsize as in (3.9) is in general rather conservative. This is however not surprising and common to this type of analysis, as several bounds are derived considering worst-case scenarios. On the other hand, to the time of writing, this is the only available convergence result of an asynchronous algorithm in the considered problem setting and under a realistic probabilistic model.

Almost linear speedup: To study the speedup achievable by the proposed method, we make two simplifying assumptions, consistent with those made in the literature, namely: (a) D is proportional to the number of workers, which is reasonable in "symmetric" situations; and (b) T_ϵ is a good proxy for the number of iterations performed by the algorithm to reach the desired accuracy. Choose the stepsize γ to be small enough so that (3.9) is always satisfied in the range of values of D under consideration; then $C_1(\gamma, D)$ and $C_2(\gamma, D)$ can be taken to be constants. Consider now the two summands in square brackets in (3.13). Without the second term, one would have ideal linear speedup. However, since one can expect the second term to be much smaller than the first (at least when D is not large), an almost linear speedup can be anticipated. In fact, by (3.11) and (3.12), the second term is smaller than the first one, if γ is sufficiently small. It is also interesting to consider the behavior of the speedup when the number N of variables increases. To this end, note that, given a fixed number of workers and the problem Lipschitz constants, $C_2(\gamma, D)$ is constant, while $C_1(\gamma, D)$ is an increasing function of N, going to infinite as N goes to infinity. Therefore, we can conclude that i) the larger the values of N, the smaller the negative influence of the number of workers on the speedup; and ii) as N increases, almost linear speedups should be expected for larger and larger ranges of number of workers.

To support the above statements, we measured the speedup achieved by AsyFLEXA on the LASSO problem considered in the first row of Table 3.1. We ran AsyFLEXA

Table 3.2.
Comparison between ideal and measured speedup of AsyFLEXA running on a multi-core machine with 1-40 cores; LASSO problem from 5000 to 40000 variables; balanced workload; $\tau = 50, L = 10$, and $\gamma = 0.1$; no failures observed.

# of workers	1	2	4	8	10	20	30	40
Ideal speedup	1	2	4	8	10	20	30	40
Observed speedup $5k$ variables	1	1.6	3.7	6.7	8.9	19.1	24.6	30.4
Observed speedup $10k$ variables	1	1.8	4.1	7.9	8.5	18.7	26.3	32.3
Observed speedup $20k$ variables	1	1.9	3.9	7.8	8.5	19.3	28.9	35.9
Observed speedup $40k$ variables	1	1.9	3.9	7.9	9.3	18.9	29.4	37.7

using up to 40 cores and a stepsize γ satisfying the theoretical upper bound (3.9). Table 3.2 shows the results of these experiments. Consistently with our previous observations, AsyFLEXA experiences a practical speedup which is reasonably close to the ideal one, but, in some cases, deteriorates somewhat when the number of workers increases, see in particular the results for 30 and 40 workers. Even more interestingly, the results in Table 3.2 show that the speedup achieved by AsyFLEXA becomes closer and closer to the ideal one as the number of variables N increases, supporting the theoretical behavior expected from (3.13) and the considerations made above. Note that the sometimes slightly more erratic behavior observed for a smaller number of workers is explained by the fact that, for this case, almost ideal speedup is expected for all values of N and so statistical fluctuations can play a significant role. Extensive numerical simulations to investigate this trend will be the subject of future research; the interested reader can find some preliminary experiments in Section 3.6.

3.4.2 AsyFLEXA with diminishing stepsize γ^k

Algorithms with diminishing stepsizes usually work better in practical experiments, because they do not need to be as small as usually required by the convergence theory for fixed stepsizes. Furthermore, the choice of a fixed proper stepsize would require the knowledge of unkwnown problem parameters, whose (generally loose) estimates might result in a very small value of the stepsize. In the rest of this Section, we consider the following assumptions on the stepsize sequence $\{\gamma^k\}$.

Assumption D (On the stepsize). The sequence $\{\gamma^k\}$ is chosen so that

(D1) $\gamma^k \downarrow 0$; $\sum_{k=0}^{\infty} \gamma^k = +\infty$; and $\sum_{k=0}^{\infty} (\gamma^k)^2 < +\infty$;

(D2) $\frac{\gamma^{k+1}}{\gamma^k} \geq \eta \in (0, 1)$.

Conditions in D1 are standard and satisfied by most practical diminishing stepsize rules; see, e.g., Section 2.2. D2 dictates that $\sum_{k=0}^{\kappa} \gamma^k \geq \gamma^0(\eta^0 + \eta^1 + \cdots + \eta^\kappa)$, that is the partial sums $\sum_{k=0}^{\kappa} \gamma^k$ must be minorized be a *convergent* geometric series. Given that $\sum_{k=0}^{\infty} \gamma^k = +\infty$, D2 is clearly very mild and indeed it is also satisfied by most classical diminishing stepsize rules. For example, (2.12) satisfies Assumption D and has been found very effective in our experiments.

We can now provide the following convergence result of AsyFLEXA.

Theorem 3.4.2 *Given Problem* (P) *under Assumptions A-A*-B-C'-D, along with the stochastic process* $\underline{\boldsymbol{\omega}}$. *Let* $\{\underline{\mathbf{x}}^k\}$ *be the sequence generated by AsyFLEXA. Suppose that* $\{\underline{\mathbf{x}}^k\}$ *is bounded almost surely (a.s.). Then, the following hold:*

(a) Every limit point of $\{\underline{\mathbf{x}}^k\}$ *is a stationary solution of Problem* (P) *a.s.;*

(b) The sequence of objective function values $\{V(\underline{\mathbf{x}}^k)\}$ *converges a.s..*

Proof See the Appendix of this Chapter. ■

In addition to Assumptions A-A*-B-C'-D, Theorem 3.4.2 requires the sequence $\{\mathbf{x}^k\}$ to be bounded a.s.. The following corollary provides some concrete conditions for this boundedness to hold.

Corollary 3.4.3 *The same conclusions of Theorem 3.4.2 hold if the assumption on the a.s. boundedness of $\{\underline{\mathbf{x}}^k\}$ is replaced by either of the following two:*

i) $\{\gamma^k\}$ is such that $\gamma^{k+1} \leq \gamma^k$, for sufficiently large k; and $\|\nabla_{\mathbf{x}_i} F\| \leq L_{\nabla_i}$ on $\mathcal{X}_i$, for any $i \in \mathcal{N}$;

ii) C3 holds.

Proof See the Appendix of this Chapter. ■

3.5 Nonconvex Constraints

In this Section, we remove the assumption that all constraints are convex, and study the following more general nonconvex constrained optimization problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & V(\mathbf{x}) = F(\mathbf{x}) + \sum\limits_{i=1}^{N} g_i(\mathbf{x}_i) \\ & \left.\begin{array}{l} \mathbf{x}_i \in \mathcal{X}_i, \qquad i = 1, \ldots, N, \\ c_1(\mathbf{x}_1) \leq 0, \ldots, c_N(\mathbf{x}_N) \leq 0, \end{array}\right\} \triangleq \mathcal{K} \end{array} \tag{P$'$}$$

where $c_i(\mathbf{x}_i) \leq 0$ are nonconvex private constraints, with $c_i : \mathcal{X}_i \to \mathbb{R}^{m_i}$; let also define $\mathcal{K}_i \triangleq \{\mathbf{x}_i \in \mathcal{X}_i : c_i(\mathbf{x}_i) \leq 0, \}$. Note that $c_i(\mathbf{x}_i)$ is a vector function, whose individual component is denoted by $c_{i,j}$, with $j = 1, \ldots, m_i$. Problem (P$'$) is motivated by several applications in signal processing, machine learning, and networking; see [75] and references therein for some concrete examples.

To deal with nonconvex constraints, we need some *regularity* of the constraint functions. Anticipating that all c_i are assumed to be C^1 on $\mathcal{X}_i$, we will use the Mangasarian-Fromovitz Constraint Qualification (MFCQ).

Definition 3.5.1 *A point $\bar{\mathbf{x}} \in \mathcal{K}$ satisfies the MFCQ if the following implication is satisfied:*

$$\left.\begin{array}{l} \mathbf{0} \in \sum_{i=1}^{N} \sum_{j \in \bar{J}_i} \mu_{i,j} \nabla_{\mathbf{x}} c_{i,j}(\bar{\mathbf{x}}_i) + N_{\mathcal{X}}(\bar{\mathbf{x}}) \\ \mu_{i,j} \geq 0, \forall j \in \bar{J}_i, \forall i \in \mathcal{N} \end{array}\right\} \Rightarrow \mu_{i,j} = 0, \forall j \in \bar{J}_i, \forall i \in \mathcal{N}, \tag{3.14}$$

where $N_{\mathcal{X}}(\bar{\mathbf{x}}) \triangleq \{\mathbf{z} \in \mathcal{X} : \mathbf{z}^T(\mathbf{y} - \bar{\mathbf{x}}) \leq 0, \forall \mathbf{y} \in \mathcal{X}\}$ is the normal cone to $\mathcal{X}$ at $\bar{\mathbf{x}}$, and $\bar{J}_i \triangleq \{j : c_{i,j}(\bar{\mathbf{x}}_i) = 0\}$ is the index set of nonconvex constraints that are active at $\bar{\mathbf{x}}_i$.

We study Problem (P′) under the following assumptions.

Assumption A′ (On the problem model).

(A1′) Each set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is nonempty, closed, and convex;

(A2′) F is C^1 and its gradients $\nabla_{\mathbf{x}_i} F$ are globally L-Lipschitz continuous on $\mathcal{K}$;

(A3′) Each $g_i : \mathcal{X}_i \to \mathbb{R}$ is convex;

(A4′) $\mathcal{K}$ is a compact set;

(A5′) Each $c_{i,j}$ is C^1;

(A6′) All feasible points of problem (P′) satisfy the MFCQ.

Assumptions A1′-A3′ are a duplication of A1-A3, repeated here for ease of reference; A4′ is stronger than A4, and made here for the sake of simplicity (one could relax it with A4); and A5′ is a standard differentiability assumption on the non convex constraints $c_{i,j}$.

• **AsyFLEXA-NCC -** We are now ready to introduce our asynchronous algorithmic framework for (P′), termed AsyFLEXA-NCC (where NCC stands for Non Convex Constraints). The method is still given by Algorithm 4, with the only difference that now also the nonconvex constraints are replaced by suitably chosen convex approximations; the probabilistic model concerning the choice of the the pair index-delays is the same as the one we used in the case of convex constraints, see Section 3.3.

More specifically, AsyFLEXA-NCC is given by Algorithm 4 (with fixed stepsize $\gamma^k = \gamma$, as highlighted later) wherein the subproblem (3.3) in Step 2 is replaced by

$$\widehat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k}) \triangleq \underset{\mathbf{x}_{i^k} \in \mathcal{K}_{i^k}(\mathbf{x}_{i^k}^k)}{\arg\min} \left\{ \tilde{F}_{i^k}(\mathbf{x}_i; \mathbf{x}^{k-\mathbf{d}^k}) + g_{i^k}(\mathbf{x}_i) \right\}, \tag{3.15}$$

where $\tilde{F}_{i^k}$ is defined as in (3.3); $\mathcal{K}_{i^k}(\mathbf{x}_{i^k}^k)$ is a convex approximation of $\mathcal{K}_{i^k}$ at $\mathbf{x}^{k-\mathbf{d}^k}$, defined as

$$\mathcal{K}_{i^k}(\mathbf{x}_{i^k}^k) \triangleq \{\mathbf{x}_i \in \mathcal{X}_{i^k} : \tilde{c}_{i^k,j}(\mathbf{x}_i; \mathbf{x}_{i^k}^k) \leq 0,\ j = 1, \ldots, m_{i^k}\};$$

and $\tilde{c}_{i^k,j} : \mathcal{X}_{i^k} \times \mathcal{K}_{i^k} \to \mathbb{R}$ is a suitably chosen surrogate of $c_{i^k,j}$. Note that K_{i^k} depends on $\mathbf{x}_{i^k}^k$ and not on $\mathbf{x}_{i^k}^{k-d_{i^k}^k}$, because of Assumption C3 ($d_{i^k}^k = 0$).

The surrogate functions $\tilde{c}_{i^k,j}$ can be chosen according to the following assumptions.

Assumption E (On the surrogate functions $\tilde{c}_{i,j}$'s).

(E1) Each $\tilde{c}_{i,j}(\cdot; \mathbf{y})$ is C^1 on an open set containing $\mathcal{X}_i$, and convex on $\mathcal{X}_i$ for all $\mathbf{y} \in \mathcal{K}_i$;

(E2) $\tilde{c}_{i,j}(\mathbf{y}; \mathbf{y}) = c_i(\mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}_i$;

(E3) $c_{i,j}(\mathbf{z}) \leq \tilde{c}_{i,j}(\mathbf{z}; \mathbf{y})$ for all $\mathbf{z} \in \mathcal{X}_i$ and $\mathbf{y} \in \mathcal{K}_i$;

(E4) $\tilde{c}_{i,j}(\cdot; \cdot)$ is continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

(E5) $\nabla_{\mathbf{y}_i} c_{i,j}(\mathbf{y}) = \nabla \tilde{c}_{i,j}(\mathbf{y}; \mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}_i$;

(E6) $\nabla \tilde{c}_{i,j}(\cdot; \cdot)$ is continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

(E7) Each $\tilde{c}_{i,j}(\cdot; \cdot)$ is Lipschitz continuous on $\mathcal{X}_i \times \mathcal{K}_i$.

Roughly speaking, Assumption E requires $\tilde{c}_{i,j}$ to be an upper convex approximation of $c_{i,j}$ having the same gradient of $c_{i,j}$ at the base point $\mathbf{y}$. Finding such approximations is less difficult than it might seem at a first sight. Two examples are given below, while we refer the reader to [14, 75] for a richer list.

• Suppose $c_{i,j}$ has a $L_{\nabla c_{i,j}}$-Lipschitz continuous gradient on the (compact) set $\mathcal{K}_i$. By the Descent Lemma [9, Proposition A32], the following convex approximation satisfies Assumption E:

$$\tilde{c}_{i,j}(\mathbf{x};\mathbf{y}) \triangleq c_{i,j}(\mathbf{y}) + \nabla_{\mathbf{x}} c_{i,j}(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{L_{\nabla c_{i,j}}}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 \geq c_{i,j}(\mathbf{x}).$$

• Suppose that $c_{i,j}$ has a DC structure, that is, $c_{i,j}(\mathbf{x}) = c^+_{i,j}(\mathbf{x}) - c^-_{i,j}(\mathbf{x})$, $c^+_{i,j}$ and $c^-_{i,j}$ are two convex and continuously differentiable functions. By linearizing the concave part $-c^-_{i,j}$ and keeping the convex part $c^+_{i,j}$ unchanged, we obtain the following convex upper approximation of $c_{i,j}$ that satisfies Assumption E:

$$\tilde{c}_{i,j}(\mathbf{x};\mathbf{y}) \triangleq c^+_{i,j}(\mathbf{x}) - c^-_{i,j}(\mathbf{y}) - \nabla_{\mathbf{x}} c^-_{i,j}(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \geq c_{i,j}(\mathbf{x}).$$

Note that the former example is quite general and in principle can be applied to practically all constraints, even if it could be numerically undesirable if $L_{\nabla c_{i,j}}$ is too large; the latter example covers, in a possibly more suitable way, the case of concave constraints.

• **AsyFLEXA-NCC: Convergence -** In order to gauge convergence, we redefine the stationarity measure M_V, to account for the presence of nonconvex constraints. We use $\|M^c_V(\mathbf{x})\|_2$, with

$$M^c_V(\mathbf{x}) = \mathbf{x} - \underset{\mathbf{y}\in\mathcal{K}_1(\mathbf{x}_1)\times\ldots\times\mathcal{K}_N(\mathbf{x}_N)}{\arg\min} \left\{ \langle \nabla_{\mathbf{x}} f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}.$$

This is is a valid merit function: $\|M^c_V(\mathbf{x})\|_2$ is continuous and it is zero only at stationary solutions of (P′) [76].

Theorem 3.5.1 *Given Problem* (P′) *under Assumptions A'-B-B***-C-E, along with the stochastic process* $\underline{\boldsymbol{\omega}}$*. Let* $\{\underline{\mathbf{x}}^k\}$ *be the sequence generated by AsyFLEXA-NCC. Suppose that the stepsize sequence* $\{\gamma^k\}$ *is fixed* $\gamma^k = \gamma \in (0;1]$ *and chosen as in (3.9). Define* T_ϵ *to be the first iteration such that* $\mathbb{E}\left(\|M^c_V(\underline{\mathbf{x}}^k)\|_2^2\right) \leq \epsilon$*. Then, the following hold: i)* $\underline{\mathbf{x}}^k \in \mathcal{K}_1(\underline{\mathbf{x}}_1^k)\times\ldots\times\mathcal{K}_N(\underline{\mathbf{x}}^N) \subseteq \mathcal{K}$ *for all* $k \geq 0$ *(iterate feasibility); and ii) all results in Theorem 3.4.1 hold with* M_V *replaced by* M^c_V*.*

Proof See the Appendix of this Chapter. ■

We are aware of only one other BCD-asynchronous method [49, 50] able to deal with nonconvex constraints. This method requires the ability to find global minima of *nonconvex* subproblems while our scheme does not suffer from this drawbacks, as it only calls for the solution of strongly convex subproblems. On the other hand, it needs a feasible starting point and the ability to build approximations $\tilde{c}_{i,j}$ satisfying Assumption E. While our requirements are easier to be met in practice (and our analysis is based on a grounded probabilistic model), we think that the two approaches complement each other and may cover different applications.

3.6 Numerical Results

In this section we test the diminshing-stepsize version of AsyFLEXA on (convex) LASSO problems and on a nonconvex sparse learning problem, and compare it to state-of-art methods. Results for the fixed-stepsize version of the method whose complexity has been studied in Section 3.4.1 are not reported. Indeed, as usual for these methods, if the theoretical stepsize (3.9) is used, the algorithm simply does not make any practical progress towards optimality in a reasonable number of iterations. If, on the other hand, we disregard the bound (3.9) that, we recall, is an upper bound, and instead we use a "large" stepsize (in particular we found the value of 0.95 practically effective), the numerical results are essentially the same as those obtained with the diminishing-stepsize version, for which we report the results. Since this latter version, as shown in Theorem 3.4.2, has theoretical convergence guarantee for the chosen stepsize rule, we present the experimental results only for this version of the method.

3.6.1 Implementation

All tested codes have been written in C++ using Open-MP. The algorithms were tested on the Purdue's Community Cluster Snyder on a machine with two 10-Core Intel Xeon-E5 processors (20 cores in total) and 256 GB of RAM.

• textbfAsyFLEXA - We tested AsyFLEXA setting $N = n$, i.e. we considered scalar subproblems. These subproblems, for the choice of $\tilde{F}_i$ that we used and that we specify below for each class of problems, can be solved in closed form using the soft-thresholding operator [20]. For the stepsize sequence $\{\gamma^k\}$ we used the rule $\gamma^{k+1} = \gamma^k(1 - \mu\gamma^k)$ with $\gamma^0 = 1$ and $\mu = 10^{-6}$.

We compared AsyFLEXA with the following state-of-the-art asynchronous schemes: AsySPCD and ARock. We underline that the theoretical stepsize rules required for the convergence of AsySPCD and ARock lead to practical non-convergence, since they prescribe extremely small stepsizes. For both algorithms we made several attempts to identify practical rules that bring the best numerical behavior, as detailed below.

• **AsySPCD -** This is the asynchronous parallel stochastic proximal gradient coordinate descent algorithm for the minimization of convex and nonsmooth objective functions presented in [54]. Since the algorithm was observed to perform poorly if the stepsize γ is chosen according to [54, Th. 4.1]−the value guaranteeing theoretical convergence−in our experiments we used $\gamma = 1$, which violates [54, Th. 4.1] but was effective on our test problems. Note also that this is the value actually used in the numerical tests reported in [54]. We underline that in order to implement the algorithm it is also required to estimate some Lipschitz-like constants.

• **ARock -** ARock [55] is an asynchronous parallel algorithm proposed to compute fixed-points of a nonexpansive operator. Thus it can be used to solve convex optimization problems. The algorithm requires a stepsize and the knowledge of the Lipschitz constant of ∇f. Here, again, the use of stepsizes that guarantee theoretical convergence leads to very poor practical performance. In this case we found that the most practical, effective version of the method could be obtained by using for the stepsize the same rule employed in our AsyFLEXA, with a safeguard that guarantees that the stepsize never becomes smaller than 0.1.

We also tested the stochastic version of Asynchronous PALM [49], an asynchronous version of the Proximal Alternating Linearized Minimization (PALM) algorithm for the minimization of nonsmooth and nonconvex objective functions. We did not report the results, because its practical implementation (using unitary stepsize rather than the one guaranteeing convergence, for the reasons explained above) basically coincides with the one of AsySPCD.

Before reporting the numerical results it is of interest to briefly contrast these algorithms in order to better highlight some interesting properties of AsyFLEXA.

1. In all tests we partitioned the variables among the cores, with only one core per partition. Therefore, each core is in charge of updating its assigned variables and no other core will update those variables. There is consensus in the literature that this configuration brings better results experimentally in shared memory architectures, see e.g. [54, 61]. Furthermore, if one is considering a message passing architecture, this is actually the only possible choice. It is then interesting to note that this choice is fully covered by our theory, while the theory of ARock [55] requires that each core has access to all variables and therefore can not handle it. The theory supporting AsySPCD [54] requires that at each iteration a variable is updated, with all variables selected with *equal* probability. The equal probability requirement in the partitioned configuration can not be theoretically excluded, but is totally unlikely: it would require that all cpu are identical and that all subproblems require exactly the same amount of time to be processed. In any case, even not considering the partitioned variables issue, the practical choice for the stepsizes adopted are not covered by the theory in [54] and [55]. We believe this clearly shows our analysis to be robust and well matched to practical computational architectures.

2. It is of interest to note that we run experiments also on *nonconvex* problems. Our algorithm has theoretical convergence guarantees for nonconvex problems while neither ARock nor AsySPCD has any such guarantees.

3. Both AsySPCD and ARock (and also PALM) theoretically require, for their implementation, the estimation of some Lipschitz constants of the problem functions. Although in our test problems the estimations could be reasonably done, this requirement in general can be extremely time consuming if at all possible on other problems.

4. One last difference worth mentioning is that contrary to AsyFLEXA and ARock, AsySPCD and PALM require a memory lock of the variable(s) being updated while the solution of the corresponding subproblems are computed. In the partitioned configuration we adopted this is not an issue, because nobody else can update the variables being updated by a given core. However, when other configurations are chosen this can become a source of delays, especially if the subproblems are complex and take some time to be solved. This could easily happen, for example, if the subproblems involve more than one variable or if they are nonconvex, as those that could appear in PALM

We are now ready to illustrate the numerical results.

3.6.2 LASSO

Consider the LASSO problem, i.e., Problem (P), with $F(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, $g_i(\mathbf{x}_i) = \lambda\|\mathbf{x}_i\|_1$, $i \in \mathcal{N}$, $\mathcal{X} = \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. In all implementations of the algorithms, we precomputed the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and the vector $\mathbf{A}^{\mathrm{T}}\mathbf{b}$ offline. In all the experiments, the starting point of all the algorithms was set to the zero vector. For problems in which the optimal value $V^\star$ is not known, in order to compute the relative error we estimated $V^\star$ by running a synchronous version of AsyFLEXA until variations in the objective value were undetectable.

For AsyFLEXA we set $\tilde{F}_i(x_i; \mathbf{x}^{k-\mathbf{d}^k}) = F(x_i; \mathbf{x}_{-i}^{k-\mathbf{d}^k}) + \frac{\tau^k}{2}(x_i - x_i^{k-d_i^k})^2$, where $\mathbf{x}_{-i}$ denotes the vector obtained from $\mathbf{x}$ by deleting the block $\mathbf{x}_i$. The sequence $\{\tau^k\}$, shared among all the cores, was updated every n iterations, according to the heuristic proposed for FLEXA [1].

We run tests on LASSO problems generated according to three different methods; all curves reported are averaged over five independent random problem realizations.

Gaussian problems [54]: In this setting we generated the LASSO problem according to the procedure used in [54]; where the matrix $\mathbf{A}$ has samples taken from a Gaussian $\mathcal{N}(0, 1)$ distribution, $\bar{\mathbf{x}} \in \mathbb{R}^n$ is a sparse vector with s nonzeros and $\mathbf{b} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{e}$, with $\mathbf{e} \in \mathbb{R}^m \sim \mathcal{N}(0, \sigma^2)$. In particular we chose $m = 20000$, $n = 40000$, $s = 40$ and $\sigma = 0.01$. For the regularization parameter we used the value suggested in [54]: $\lambda = 20\sqrt{m \log(n)}\sigma$. In Figure 3.2 we plot the relative error on the objective function versus the CPU time, using 2 and 20 cores ((c) = cores). The figure shows that when increasing the number of cores, all the algorithms converge quickly to the solution with comparable performances. In this particular setting, and contrary to what happens in all other problems, AsySPCD has a slightly better behavior than AsyFLEXA but it does not have convergence guarantees.

In order to quantify the scalability of the algorithms, in Figure 3.3 we plot the speedup achieved by each of the algorithms versus the number of cores (of course we run all algorithms also for values between $c = 1$ and $c = 20$, more precisely for $c = 1, 2, 4, 8, 10, 20$). We defined the speedup as the ratio between the runtime on a single core and the runtime on multiple cores. The runtimes we used are the CPU times needed to reach a relative error strictly less than 10^{-4}. The figure shows that all the algorithms obtain a good gain in the performances by increasing the number of cores which is not far from the ideal speedup.

Nesterov's problems [77]: Here we generated a LASSO problem using the random generator proposed by Nesterov in [77], which permits us to control the sparsity of the

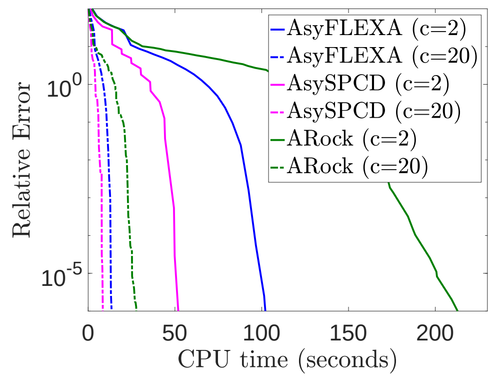

Figure 3.2. LASSO: comparison in terms of relative error versus CPU time (in seconds) for Liu and Wright's problems [54].

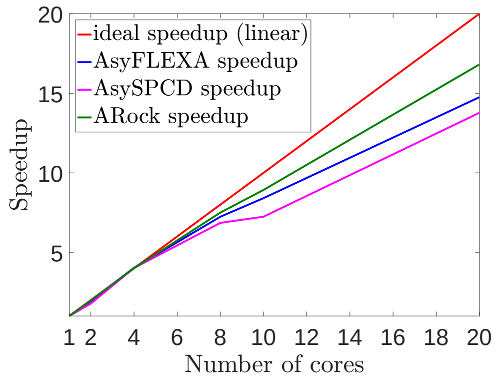

Figure 3.3. LASSO: speedup of the tested algorithms for Liu and Wright's problems [54].

solution. We considered a problem with 40000 variables and matrix $\mathbf{A}$ having 20000 rows, and set $\lambda = 1$; the percentage of nonzero in the solution is 1%. In Figure 3.5 we plot the relative error on the objective function (note that for Nesterov's model the optimal solution is known) versus the CPU time, using 2 and 20 cores. The figure clearly shows that AsyFLEXA significantly outperforms all the other algorithms on these problems. Moreover, the empirical convergence speed significantly increases with the number of cores, which instead is not observed for the other algorithms, see Figure 3.5, where we only report data for AsyFLEXA, given that the other algorithms do not reach the prefixed threshold error value of 10^{-4} in one hour of computation time.

Gondzio's problem: We generated these LASSO problems using the generator proposed by Gondzio in [78]. The key feature of this generator is the possibility of choosing the condition number of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and we set it to 10^4. We generated a matrix $\mathbf{A}$ with 2^{14} rows and $\lceil 1.01n \rceil$ columns, where the ceiling function $\lceil \cdot \rceil$ returns the smallest integer greater than or equal to its argument. The sparsity in the solution is 0.1% and we set $\lambda = 1$. Figure 3.6 shows the relative error with respect to the CPU

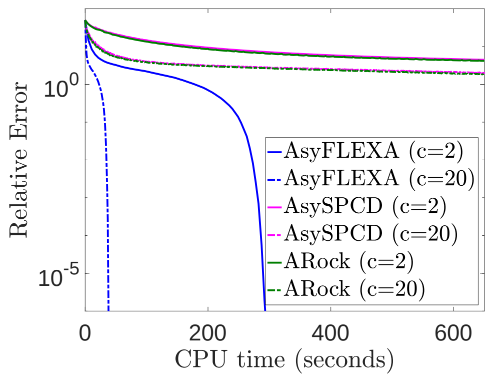

Figure 3.4. LASSO: comparison in terms of relative error versus CPU time (in seconds) for Nesterov's problems [77].

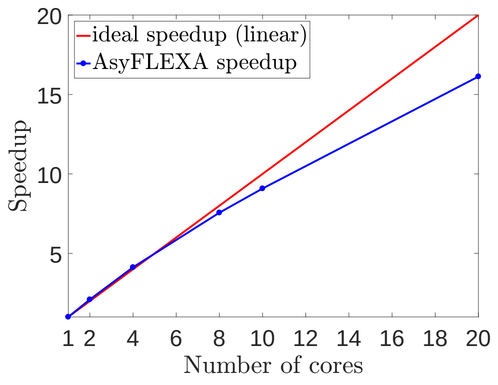

Figure 3.5. LASSO: speedup of AsyFLEXA for Nesterov's problems [77].

time for the different algorithms we tested, when using 2 and 16 cores. Figure 3.7 that shows the speedup achieved by our algorithm on these problems (in this case we run the algorithm for $c = 1, 2, 4, 8, 16$). We see that the behavior of the algorithm is qualitatively very similar to the one obtained for the previous set of problems.

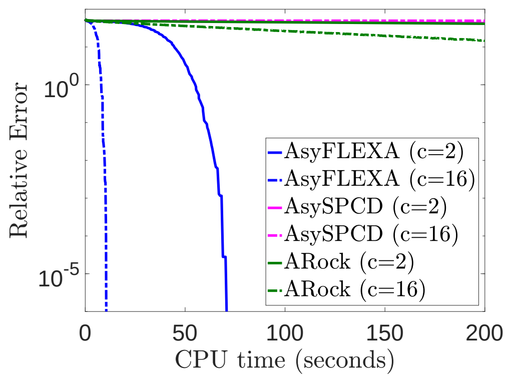

Figure 3.6. LASSO: comparison in terms of relative error versus CPU time (in seconds) for Gondzio's problems [78]

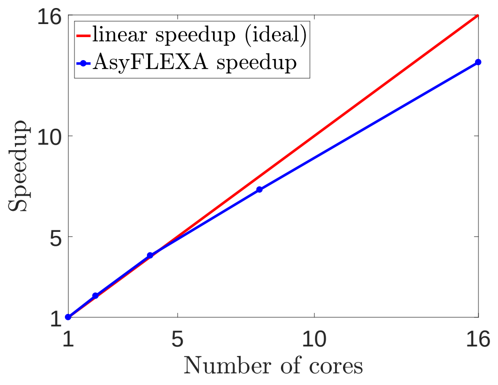

Figure 3.7. LASSO: speedup of AsyFLEXA for Gondzio's problems [78]

3.6.3 Nonconvex Sparse Learning

We consider now the following nonconvex instance of problem (P):

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad V(\mathbf{x}) \triangleq \underbrace{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2}_{=l(\mathbf{x})} + \lambda R(\mathbf{x}), \tag{3.16}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}$, $\mathbf{b}\in\mathbb{R}^m$, $\lambda>0$ and $R(\mathbf{x})$ is a regularizer used to balance the amount of sparsity in the solution. (3.16) is a standard formulation used for doing regression in order to recover a sparse signal from a noisy observation vector $\mathbf{b}$ [79]. Common choices for the regularizer R are surrogates of the l_0 norm, as the frequently used l_1 norm (in this case we recover the LASSO problem of the previous section). However, it is known that nonconvex surrogates of the l_0 norm can provide better solutions than the l_1 norm, in terms of balance between compression and reconstruction error, see e.g. [80]. For this reason we tested here two nonconvex regularizer functions: the exponential one $R(\mathbf{x}) = R_{\exp}(\mathbf{x}) = \sum_{i=1}^{n} r_{\exp}(x_i)$, with $r_{\exp}(x_i) = 1-e^{-\theta_{\exp}|x_i|}$ and $\theta_{\exp}>0$, and the logarithmic one $R(\mathbf{x}) = R_{\log}(\mathbf{x}) = \sum_{i=1}^{n} r_{\log}(x_i)$, with $r_{\log}(x_i) = \frac{\log(1+\theta_{\log}|x_i|)}{\log(1+\theta_{\log})}$ and $\theta_{\log}>0$. Note that, given the fact that R is nonconvex and nonsmooth, Problem (3.16) does not immediately appear to be in the format needed by our algorithm, i.e. the sum of a smooth term and of a possibly nonsmooth convex one. Nevertheless, we can put the problem in this form, as explained next. In both cases ($r_{\exp}$ and $r_{\log}$) the function r possesses a DC structure (see Example 3) in Section 2.2) that allows us to rewrite it as

$$r(x) = \underbrace{\eta(\theta)|x|}_{\triangleq r^+(x)} - \underbrace{\eta(\theta)|x| - r(x)}_{\triangleq r^-(x)}, \tag{3.17}$$

with $\eta_{\exp}(\theta_{\exp}) = \theta_{\exp}$ and $\eta_{\log}(\theta_{\log}) = \frac{\theta_{\log}}{\log(1+\theta_{\log})}$. It is easily verified that r^+ is convex and that, slightly more surprisingly, r^- is continuously differentiable with a Lipschitz gradient [80]. Given these facts, it is now easily seen that problem (3.16) can be put in the setting of Problem (P), by setting $F(\mathbf{x}) = l(\mathbf{x}) - \lambda\sum_{i=1}^{n} r^-(x_i)$ and $g_i(x) = \lambda r^+(x)$. Following the idea of see Example 3) in Section 2.2, we can implement

AsyFLEXA to solve problem (3.16) by considering scalar blocks ($N = n$) and the following formulation for the best-response $\widehat{x}_i$, $i \in \mathcal{N}$:

$$\widehat{x}_i(\mathbf{x}^{k-\mathbf{d}^k}) = \\ \underset{x_i \in \mathbb{R}}{\arg\min} \left\{ l(x_i, \mathbf{x}_{-i}^{k-\mathbf{d}^k}) - \left\langle \nabla r^-(x_i^{k-d_i^k}), x - x_i^{k-d_i^k} \right\rangle + r^+(x_i) + \tfrac{\tau^k}{2}(x_i - x_i^{k-d_i^k})^2 \right\}. \tag{3.18}$$

As it is possible to see, in (3.18) we are preserving the convexity in $l(\mathbf{x})$ and $r^+(x)$, while linearizing the nonconvex term $-r^-(x)$. Note that, once again, the solution of (3.18) can be computed in closed-form through the soft-thresholding operator [20]. For the positive constant τ^k we use again the same heuristic rule used for the LASSO problems.

In order to generate the 5 random instances of the problem we must specify how we generate the quadratic function $l(\cdot)$. In order to favour AsySPCD and ARock we use the Liu and Wright's generator used for the first set of LASSO problems discussed above. In particular, we generated the underlying sparse linear model according to: $\mathbf{b} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{e}$ where $\mathbf{A}$ has 20000 rows (with values normalized to one) and 40000 columns. $\mathbf{A}$, $\bar{\mathbf{x}}$ and $\mathbf{e}$ have i.i.d. elements coming from a Gaussian $\mathcal{N}(0, \sigma^2)$ distribution, with $\sigma = 1$ for $\mathbf{A}$ and $\bar{\mathbf{x}}$, and $\sigma = 0.1$ for the noise vector $\mathbf{e}$. To impose sparsity on $\bar{\mathbf{x}}$, we randomly set to zero 95% of its component.

Since (3.16) is nonconvex, we compared the performance of the algorithms using the following distance to stationarity: $\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_\infty$. Note that this is a valid stationarity measure: it is continuous and $\|\widehat{\mathbf{x}}(\mathbf{x}^\star) - \mathbf{x}^\star\|_\infty = 0$ if and only if $\mathbf{x}^\star$ is a stationary solution of (3.16). In Figure 3.8 we plot the stationarity measure versus the CPU time, for all the algorithms using 2 and 20 cores, for problem (3.16) where $R(\mathbf{x}) = R_{\log}(\mathbf{x})$ with $\theta_{\log} = 20$; the curves are averaged over 5 independent realizations. All the algorithms were observed to converge to the same stationary solution of (3.16) even if, we recall, AsySPCD and ARock have no formal proof of convergence in this nonconvex setting. The figure shows that, even in the nonconvex case, AsyFLEXA has good performances and actually behaves better that all the other algorithms while being also guaranteed to converge. As a side issue it is inter-

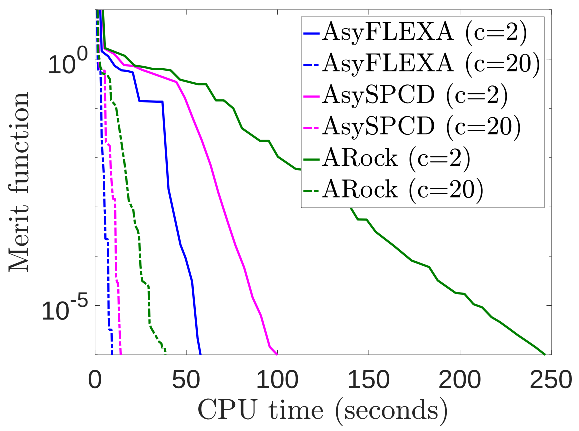

Figure 3.8. Nonconvex problem: comparison in terms of relative error versus CPU time (in seconds).

esting to illustrate the utility of our nonconvex regularizers with respect to the more standard l_1 norm regularizer. In Figure 3.9 and Figure 3.10 we show respectively the Normalized Mean Square Error (NMSE) (defined as: NMSE$\triangleq \|\mathbf{x}-\mathbf{x}^\star\|_2^2/\|\mathbf{x}^\star\|_2^2$) and the percentage of nonzeros obtained by solving the aforementioned problem with AsyFLEXA, for different values of the regularization parameter λ and for different surrogates of the l_0 norm: the l_1 norm, the exponential function ($\theta_{\text{exp}} = 20$) and the logarithmic function ($\theta_{\text{log}} = 20$). AsyFLEXA is terminated when the merit function goes below 10^{-4} or after $100n$ iterations. These two figures interestingly show that the two nonconvex regularizers obtain their lowest NMSE for a value of λ which corresponds to a good compression result, close to the number of nonzeros of the original signal. On the other side, the l_1 norm attains its lowest NMSE by reconstructing the signal with more of 15% of nonzeros, much more than the original 5%. In order to get close to the desired number of nonzeros in the reconstruction with the l_1 norm, it is necessary to increase the value of λ, but this leads to a worsening in terms of NMSE of at least two times with respect to the minimum NMSE attainable.

A full understanding of the numerical behavior of our algorithm and its comparison with existing alternatives certainly needs further numerical tests, on both larger and

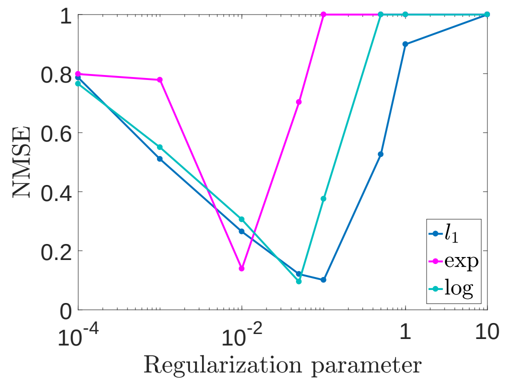

Figure 3.9. Nonconvex problem: NMSE for different values of the regularization parameter λ and for different surrogates of the l_0 norm.

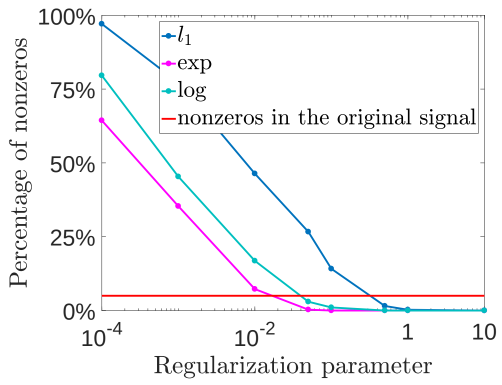

Figure 3.10. Nonconvex problem: percentage of nonzeros in the solution for different values of the regularization parameter λ and for different surrogates of the l_0 norm.

more complex problems. But the substantial results already reported here seem to clearly indicate that our method is reliable and robust and capable to efficiently solve problems that other methods cannot tackle. The improved performance of AsyFLEXA with respect to AsySPCD and ARock on some classes of problems seems due, in our experience, to the much wider flexibility we have in the choice of the surrogates $\tilde{F}_i$'s, which are not linked in any way to the Lipschitz constants of the problem functions.

3.7 Conclusions

In this Chapter we proposed a novel model for the parallel block-descent asynchronous minimization of the sum of a nonconvex smooth function and a convex nonsmooth one, subject to nonconvex constraints. Our model captures the essential features of modern multi-core architectures by providing a more realistic probabilistic description of asynchrony that that offered by the state of the art. Building on our new probabilistic model, we proved sublinear convergence rate of our algorithm and a near linear speedup when the number of workers is not too large. Experimental results on convex and nonconvex problems show that our algorithm compares favorably to existing asynchronous schemes.

3.8 Appendix: Proofs of Theorems

In the rest of this Section, we simplify the notation with respect to the one used in Chapter 3, by using $\tilde{\mathbf{x}}^k \triangleq \mathbf{x}^{k-\mathbf{d}^k}$.

3.8.1 Preliminaries

1. On conditional probabilities. In our developments, we will consider the conditional expectation of random variables $\mathbf{Z}$ on Ω of the type $\mathbb{E}(\mathbf{Z}|\mathcal{F}^k)$. The following simple fact holds.

Proposition 3.8.1 *Let* $\mathbf{Z}$ *be a random variable defined on* Ω, *and let* $\mathcal{F}^k$ *be defined in* (3.5). *Then*

$$\mathbb{E}\left(\mathbf{Z}|\mathcal{F}^k\right) = \sum_{(i,\mathbf{d})\in\mathcal{N}\times\mathcal{D}} p\left((i,\mathbf{d})\ |\boldsymbol{\omega}^{0:k}\right)\mathbf{Z}\left((i,\mathbf{d}),\boldsymbol{\omega}^{0:k}\right). \tag{3.19}$$

Proof Recall that $\mathcal{F}^k$ is the σ-algebra generated by $\mathcal{C}^k$, which is a finite partition of Ω. Therefore, one can write [81, Example 5.1.3]

$$\mathbb{E}\left(\mathbf{Z}|\mathcal{F}^k\right) = \frac{\mathbb{E}\left(\mathbf{Z}; C^k(\boldsymbol{\omega}^{0:k})\right)}{\mathbb{P}\left(C^k(\boldsymbol{\omega}^{0:k})\right)}.$$

The thesis follows readily from (3.6) and the fact that $\mathbf{Z}$ depends only on $\boldsymbol{\omega}^{0:k+1}$ and takes a finite number of values. ■

2. Properties of the best response $\widehat{\mathbf{x}}(\cdot)$. We introduce next some basic properties of the best-response maps defined in (3.3) and (3.15).

Proposition 3.8.2 ([1]) *Given the best-response map* $\widehat{\mathbf{x}}(\cdot) \triangleq [\widehat{\mathbf{x}}_i(\cdot)]_{i=1}^N$, *with* $\widehat{\mathbf{x}}_i(\cdot)$ *defined in* (3.3). *Under Assumptions A-B, the following hold.*

(a) [Optimality]: *For any $i \in \mathcal{N}$ and $\mathbf{y} \in \mathcal{X}$,*

$$\langle \widehat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i, \nabla_{\mathbf{y}_i} f(\mathbf{y}) \rangle + g_i(\widehat{\mathbf{x}}_i(\mathbf{y})) - g_i(\mathbf{y}_i) \leq -\tau \|\widehat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i\|_2^2 \,; \tag{3.20}$$

(b) [Lipschitz continuity]: *For any $i \in \mathcal{N}$ and $\mathbf{y}, \mathbf{z} \in \mathcal{X}$,*

$$\|\widehat{\mathbf{x}}_i(\mathbf{y}) - \widehat{\mathbf{x}}_i(\mathbf{z})\|_2 \leq \tilde{L} \|\mathbf{y} - \mathbf{z}\|_2, \tag{3.21}$$

with $\tilde{L} = L_A/\tau$;

(c) [Fixed-point characterization]: *The set of fixed-points of $\widehat{\mathbf{x}}(\cdot)$ coincides with the set of stationary solutions of Problem* (P)*. Therefore $\widehat{\mathbf{x}}(\cdot)$ has at least one fixed point.*

(d) [Boundedness]: *If $\nabla_{\mathbf{x}_i} f$ is bounded on $\mathcal{X}_i$ for some $L_{\nabla_i} \in (0, +\infty)$, then so is:*

$$\|\widehat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i\|_2^2 \leq M_i \tag{3.22}$$

for any $i \in \mathcal{N}$ and $\mathbf{y} \in \mathcal{X}$, with $M_i = \frac{L_{\nabla_i} + L_g}{\tau}$

Proposition 3.8.3 ([76]) *Given the best-response map $\widehat{\mathbf{x}}(\cdot) \triangleq [\widehat{\mathbf{x}}_i(\cdot)]_{i=1}^N$, with $\widehat{\mathbf{x}}_i(\cdot)$ defined in* (3.15)*. Under Assumptions A′-B-E, the following hold.*

(a) [Optimality]: *For any $i \in \mathcal{N}$ and $\mathbf{y} \in \mathcal{K}$,*

$$\langle \widehat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i, \nabla_{\mathbf{y}_i} f(\mathbf{y}) \rangle + g_i(\widehat{\mathbf{x}}_i(\mathbf{y})) - g_i(\mathbf{y}_i) \leq -\tau \|\widehat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i\|_2^2 \,; \tag{3.23}$$

(b) [Lipschitz continuity]: *For any $i \in \mathcal{N}$ and $\mathbf{y}, \mathbf{z} \in \mathcal{K}$,*

$$\|\widehat{\mathbf{x}}_i(\mathbf{y}) - \widehat{\mathbf{x}}_i(\mathbf{z})\|_2 \leq \tilde{\tilde{L}} \|\mathbf{y} - \mathbf{z}\|_2^{1/2}, \tag{3.24}$$

with $\tilde{\tilde{L}} > 0$.

3. Representation of $\tilde{\mathbf{x}}^k$. Since at each iteration only one block of variables is updated, $\tilde{\mathbf{x}}_i^k$, $i \in \mathcal{N}$, can be written as

$$\tilde{\mathbf{x}}_i^k = \mathbf{x}_i^k + \sum_{l \in \mathcal{K}_i^k} \left(\mathbf{x}_i^l - \mathbf{x}_i^{l+1} \right), \tag{3.25}$$

where $\mathcal{K}_i^k$ is defined in Section 3.4.

4. Martingale Theorem. To prove our main convergence results, we leverage the following standard convergence theorem for non-negative almost supermartingales.

Theorem 3.8.1 ([82]) *Let $(\Omega, \mathcal{T}, \mathbb{P})$ be a probability space. Let $\mathscr{T} = \{\mathcal{T}^k\}$ be a sequence of sub-sigma algebras of $\mathcal{T}$ such that $\forall k \geq 0$, $\mathcal{T}^k \subset \mathcal{T}^{k+1}$. Define $l_+(\mathscr{T})$ as the set of sequences of $[0, +\infty)$-valued random variables $\{\underline{\xi}^k\}$, where $\underline{\xi}^k$ is $\mathcal{T}^k$-measurable, and $l_+^1(\mathscr{T}) := \{\{\underline{\xi}^k\} \in l_+(\mathscr{T}) | \sum_{k=1}^{+\infty} \underline{\xi}^k < +\infty \quad a.s.\}$. Let $\{\underline{\alpha}^k\}$, $\{\underline{\nu}^k\} \in l_+(\mathscr{T})$, and $\{\underline{\eta}^k\} \in l_+^1(\mathscr{T})$ be such that*

$$\mathbb{E}(\underline{\alpha}^{k+1}|\mathcal{T}^k) + \underline{\nu}^k \leq \underline{\alpha}^k + \underline{\eta}^k. \tag{3.26}$$

Then $\{\underline{\nu}^k\} \in l_+^1(\mathscr{T})$, and $\{\underline{\alpha}^k\}$ converges to a $[0, +\infty)$-valued random variable almost surely. □

5. Further definitions. In order to prove the statements of Theorem 3.4.2 we introduce the following two sets. Given
$\omega = \{(i^k, \mathbf{d}^k)\} \in \Omega$ and some $D(\omega) > 0$, let

$$\mathcal{V}^k(\omega) = \left\{(i, \mathbf{d}) \in \mathcal{N} \times \mathcal{D} \, : \, p\left((i, \mathbf{d})|\boldsymbol{\omega}^{0:k-1}\right) \geq D(\omega)\right\}, \tag{3.27}$$

and

$$\tilde{\mathcal{V}}^k(\omega) \triangleq \left\{((j, \mathbf{d}_j))_{j=1}^N \in \mathcal{V}^k(\omega) \, : (i, \mathbf{d}_i) = (i^k, \mathbf{d}^k) \text{ for some } i\right\}. \tag{3.28}$$

Note that $\tilde{\mathcal{V}}^k(\omega)$ is not uniquely defined when there exist multiple vectors $\mathbf{d}_j$ such that $(j, \mathbf{d}_j) \in \mathcal{V}^k(\omega)$, with $j \neq i$. In such a case, the specific choice of $\mathbf{d}_j$ is irrelevant for our developments; however, to eliminate any ambiguity in the definition (3.28) we tacitly assume that the aforementioned $\mathbf{d}_j$ chosen in (3.28) is the one that maximizes $p((j, \cdot)|\boldsymbol{\omega}^{0:k-1})$.

Definition 3.8.1 *The set $\mathcal{V}^k(\omega)$ is said to cover $\mathcal{N}$ if $(i, \mathbf{d}) \in \mathcal{V}^k(\omega)$, for some $\mathbf{d} \in \mathcal{D}$ and all $i \in \mathcal{N}$.*

Note that under C3', there exists a $\bar{\Omega} \subseteq \Omega$, with $\mathbb{P}(\bar{\Omega}) = 1$, such that the following holds: for every $\omega \in \bar{\Omega}$, one can find a $D(\omega) > 0$, such that the resulting $\mathcal{V}^k(\omega)$ covers $\mathcal{N}$ and consequently $\tilde{\mathcal{V}}^k(\omega)$ has cardinality N.

3.8.2 Proof of Theorem 3.4.1

In this section, the best-response map $\widehat{\mathbf{x}}(\cdot)$ is the one defined in (3.3). For any given realization $\omega \in \Omega$ and $k \geq 0$, the following holds:

$$\begin{aligned}
V(\mathbf{x}^{k+1}) &= f(\mathbf{x}^{k+1}) + g(\mathbf{x}^{k+1}) \\
&\overset{(a)}{=} f(\mathbf{x}^{k+1}) + \sum_{i \neq i^k} g_i(\mathbf{x}_i^k) + g_{i^k}(\mathbf{x}_{i^k}^{k+1}) \\
&\overset{(b)}{\leq} f(\mathbf{x}^k) + \gamma \left\langle \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k \right\rangle + \sum_{i \neq i^k} g_i(\mathbf{x}_i^k) + g_{i^k}(\mathbf{x}_{i^k}^{k+1}) \\
&\quad + (\nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) - \left\langle \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k)), \gamma(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k) \right\rangle + \frac{\gamma^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&\overset{(c)}{\leq} f(\mathbf{x}^k) + \gamma \left\langle \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k \right\rangle \\
&\quad + \left\langle \nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \gamma(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k) \right\rangle + \frac{\gamma^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k\|_2^2 \\
&\quad + \sum_{i \neq i^k} g_i(\mathbf{x}_i^k) + \gamma g_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)) + g_{i^k}(\mathbf{x}_{i^k}^k) - \gamma g_{i^k}(\tilde{\mathbf{x}}_{i^k}^k) \\
&\overset{(d)}{\leq} V(\mathbf{x}^k) - \gamma \left(\tau - \frac{\gamma L}{2}\right) \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k\|_2^2 \\
&\quad + L \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \|\gamma(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k)\|_2 \\
&\overset{(e)}{\leq} V(\mathbf{x}^k) - \gamma\ (\tau - \gamma L)\ \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k\|_2^2 + \frac{L}{2} \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 \\
&\overset{(f)}{=} V(\mathbf{x}^k) - \gamma\ (\tau - \gamma L)\ \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 + \frac{L}{2} \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2, \qquad (3.29)
\end{aligned}$$

where (a) follows from the updating rule of AsyFLEXA; in (b) we used the Descent Lemma [9, Proposition A32] on F; (c) comes from A3 and C3; in (d) we used Proposition 3.8.2 and A2; (e) is due to the Young's inequality; and (f) is due to C3.

We bound $\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2$ as follows:

$$\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 \overset{\text{(a)}}{\leq} \left(\sum_{l=k-D}^{k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\right)^2 \overset{\text{(b)}}{\leq} D \sum_{l=k-D}^{k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2$$

$$= D\left(\sum_{l=k-D}^{k-1} (l-(k-1)+D)\, \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 - \sum_{l=k+1-D}^{k} (l-k+D)\, \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right) \tag{3.30}$$

$$+ D^2\gamma^2 \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2,$$

where (a) comes from (3.25); and (b) is due to the Jensen's inequality.

Using (3.30) in (3.29), the Lyapunov function (3.8), and rearranging the terms, the following holds: for all $k \geq 0$,

$$\tilde{V}(\mathbf{x}^{k+1} \ldots, \mathbf{x}^{k+1-D})$$
$$\leq \tilde{V}(\mathbf{x}^k, \ldots, \mathbf{x}^{k-D}) - \gamma\left(\tau - \gamma\left(L + \frac{D^2L}{2}\right)\right) \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2; \tag{3.31}$$

and

$$\tilde{V}(\mathbf{x}^{k+B} \ldots, \mathbf{x}^{k+B-D}) \leq \tilde{V}(\mathbf{x}^{k+B-1}, \ldots, \mathbf{x}^{k+B-1-D})$$
$$- \gamma\left(\tau - \gamma\left(L + \frac{D^2L}{2}\right)\right) \|\widehat{\mathbf{x}}_{i^{k+B-1}}(\tilde{\mathbf{x}}^{k+B-1}) - \mathbf{x}_{i^{k+B-1}}^{k+B-1}\|_2^2$$
$$\leq \tilde{V}(\mathbf{x}^k, \ldots, \mathbf{x}^{k-D}) - \gamma\left(\tau - \gamma\left(L + \frac{D^2L}{2}\right)\right) \sum_{t=k}^{k+B-1} \|\widehat{\mathbf{x}}_{i^t}(\tilde{\mathbf{x}}^t) - \mathbf{x}_{i^t}^t\|_2^2. \tag{3.32}$$

Taking conditional expectation both sides we have that the following holds a.s.:

$$\mathbb{E}\left(\tilde{V}(\underline{\mathbf{x}}^{k+B} \ldots, \underline{\mathbf{x}}^{k+B-D})|\mathcal{F}^{k-1}\right) \leq \tilde{V}(\underline{\mathbf{x}}^k, \ldots, \underline{\mathbf{x}}^{k-D})$$
$$- \gamma\left(\tau - \gamma\left(L + \frac{D^2L}{2}\right)\right) \sum_{t=k}^{k+B-1} \mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\tilde{\underline{\mathbf{x}}}^t) - \mathbf{x}_{\underline{i}^t}^t\|_2^2|\mathcal{F}^{t-1}\right). \tag{3.33}$$

Using (3.9), (3.33), A4, and the Martingale's theorem [82], we deduce that i) $\{\tilde{V}(\underline{\mathbf{x}}^k, \ldots, \underline{\mathbf{x}}^{k-D})\}$, and thus $\{V(\underline{\mathbf{x}}^k, \ldots, \underline{\mathbf{x}}^{k-D})\}$ converge a.s., ii) $\{\underline{\mathbf{x}}^k\}$ is bounded on $\mathcal{X}$ a.s., and iii)

$$\lim_{k\to+\infty} \sum_{t=k}^{k+B-1} \mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\tilde{\underline{\mathbf{x}}}^t) - \underline{\mathbf{x}}_{\underline{i}^t}^t\|_2|\mathcal{F}^{t-1}\right) = 0, \quad \text{a.s..} \tag{3.34}$$

From (3.34), it follows that there exists a set $\bar{\Omega} \subseteq \Omega$, with $\mathbb{P}(\bar{\Omega}) = 1$, such that for any $\omega \in \bar{\Omega}$,

$$\begin{aligned}
&\sum_{t=k}^{k+B-1} \mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t) - \mathbf{x}_{\underline{i}^t}^t\|_2^2 | \mathcal{F}^{t-1}\right) \\
&\overset{(a)}{=} \sum_{t=k}^{k+B-1} \sum_{(i,\mathbf{d})\in\mathcal{N}\times\mathcal{D}} p\left((i,\mathbf{d})|\boldsymbol{\omega}^{0:t-1}\right) \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^t) - \mathbf{x}_i^t\|_2 \\
&\overset{(b)}{\geq} p_{\min} \sum_{i=1}^{N} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)}) - \mathbf{x}_i^{k+t_k(i)}\|_2, \qquad (3.35)
\end{aligned}$$

where in (a) we used (3.19); and in (b) we used C2 and defined $t_k(i) \triangleq \min\{t \in [0; B] | p(i|\omega^{0:t+k-1}) \geq p_{\min}\}$. We also have:

$$\begin{aligned}
&\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_2 \leq \sum_{i=1}^{N} \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|_2 \\
&\overset{(a)}{\leq} \sum_{i=1}^{N} \left(\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)}) - \mathbf{x}_i^{k+t_k(i)}\|_2 + (1+L_{\widehat{\mathbf{x}}}) \|\tilde{\mathbf{x}}^{k+t_k(i)} - \mathbf{x}^k\|_2\right) \\
&\overset{(b)}{\leq} \sum_{i=1}^{N} \Bigg(\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)}) - \mathbf{x}_i^{k+t_k(i)}\|_2 + (1+L_{\widehat{\mathbf{x}}})\Bigg(\|\mathbf{x}^{k+t_k(i)} - \mathbf{x}^k\|_2 \\
&\quad + \sum_{l=k+t_k(i)-D}^{k+t_k(i)-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\Bigg)\Bigg) \\
&\overset{(c)}{\leq} \sum_{i=1}^{N} \left(\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)}) - \mathbf{x}_i^{k+t_k(i)}\|_2 + 2\gamma(1+L_{\widehat{\mathbf{x}}}) \sum_{l=k-D}^{k+B-1} \|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l) - \mathbf{x}_{i^l}^l\|_2\right), \qquad (3.36)
\end{aligned}$$

where in (a) we used Proposition 3.8.2; (b) comes from (3.25); and (c) from the updating rule of AsyFLEXA. We deduce from (3.34), (3.35), and (3.36), that

$$\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_2 = 0. \qquad (3.37)$$

Since the sequence $\{\mathbf{x}^k\}$ is bounded, it has at least one limit point $\bar{\mathbf{x}}$ that belongs to $\mathcal{X}$. By the continuity of $\widehat{\mathbf{x}}(\cdot)$ (see Proposition 3.8.2) and (3.37), it must be $\widehat{\mathbf{x}}(\bar{\mathbf{x}}) = \bar{\mathbf{x}}$, and thus by Proposition 3.8.2 $\bar{\mathbf{x}}$ is a stationary solution of Problem (P). Since (3.37) holds for any $\omega \in \bar{\Omega}$, the previous results hold a.s..

Let us now define:

$$\widehat{\mathbf{y}}_i(\mathbf{x}^k) = \underset{\mathbf{y}_i\in\mathcal{X}_i}{\operatorname{argmin}} \left\{\nabla_{\mathbf{x}_i} f(\mathbf{x}^k)^{\mathrm{T}}(\mathbf{y}_i - \mathbf{x}_i^k) + g_i(\mathbf{y}_i) + \frac{1}{2}\|\mathbf{y}_i - \mathbf{x}_i^k\|_2^2\right\}, \qquad (3.38)$$

and note that $M_V(\mathbf{x}) = [\mathbf{x}_1^k - \widehat{\mathbf{y}}_1(\mathbf{x}^k), \ldots, \mathbf{x}_N^k - \widehat{\mathbf{y}}_N(\mathbf{x}^k)]^{\mathrm{T}}$. It is easy to check that $\widehat{\mathbf{y}}(\cdot)$ is $L_{\widehat{\mathbf{y}}}$-Lipschitz continuous on $\mathcal{X}$, with $L_{\widehat{\mathbf{y}}} \triangleq L + 1$. Fix a realization $\omega \in \Omega$. The optimality of $\widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)$ along with the convexity of g_{i^k}, leads

$$\left\langle \nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) + \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k) - \mathbf{x}_{i^k}^k, \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k) \right\rangle \tag{3.39}$$

$$+ g_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)) - g_{i^k}(\widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)) \geq 0. \tag{3.40}$$

Similarly, one can write for $\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)$:

$$\left\langle \nabla \tilde{f}_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k), \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k) - \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) \right\rangle + g_{i^k}(\widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)) - g_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k))) \geq 0. \tag{3.41}$$

Summing (3.40) and (3.41), adding and subtracting $\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)$, and using the gradient consistency assumption B2, yields

$$\begin{aligned}&\left\langle \nabla \tilde{f}_{i^k}(\mathbf{x}_{i^k}^k; \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k) + \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k, \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k) \right\rangle \\ &\geq \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2. \end{aligned} \tag{3.42}$$

Summing and subtracting $\nabla \tilde{f}_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \mathbf{x}^k)$ and using the triangular inequality, the LHS of (3.42) can be upper bounded as

$$\begin{aligned}&\|\nabla \tilde{f}_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k)\|_2 \\ &\quad + \|\nabla \tilde{f}_{i^k}(\mathbf{x}_{i^k}^k; \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \mathbf{x}^k)\|_2 + \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2 \\ &\geq \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2. \end{aligned} \tag{3.43}$$

We can further upper-bound the left hand side invoking B3 and B4, and write:

$$\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2 \leq (1 + L_B) \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2 + L_A \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 . \tag{3.44}$$

Finally, squaring both sides, we get

$$\begin{aligned}\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2 \ &\leq\ (1 + L_B)^2 \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 + L_A^2 \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 \\ &+ 2L_A(1 + L_B) \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2 \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2. \end{aligned} \tag{3.45}$$

We bound next the term $\|\mathbf{x}_{i^k}^k - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2$. We write

$$\begin{aligned}
&\|\mathbf{x}_{i^k}^k - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2 = \|\mathbf{x}_{i^k}^k - \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) + \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2 \\
&\leq\ 2\left(\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 + \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2\right) \\
&\overset{(a)}{\leq} \left(2 + 2(1+L_B)^2\right)\ \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 + 2L_A^2\,\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 \\
&\quad + 4L_A(1+L_B)\,\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2\,\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \\
&\overset{(b)}{\leq} 2\left(1 + (1+L_B)(1+L_A+L_B)\right)\ \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&\quad + 2L_A(1+L_A+L_B)\,\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2,
\end{aligned} \tag{3.46}$$

where (a) comes from (3.45); and (b) follows from the Young's inequality. Note that

$$\begin{aligned}
&\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2\ =\ \sum_{i=1}^N \|\mathbf{x}_i^k - \tilde{\mathbf{x}}_i^k\|_2^2\ \overset{(a)}{\leq}\ \sum_{i=1}^N \left(\sum_{l\in\bar{\mathcal{K}}_i^k} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\right)^2 \\
&\overset{(b)}{\leq} \sum_{i=1}^N \bar{M}_i^k \sum_{l\in\bar{\mathcal{K}}_i^k} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\ =\ \gamma^2 \sum_{i=1}^N \bar{M}_i^k \sum_{l\in\bar{\mathcal{K}}_i^k} \|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l) - \mathbf{x}_{i^l}^l\|_2^2,
\end{aligned} \tag{3.47}$$

where (a) comes from (3.25); and in (b) we used the Jensen's inequality and defined $\bar{M}_i^k \triangleq |\bar{\mathcal{K}}_i^k|$. Combining (3.46) and (3.47), we get:

$$\begin{aligned}
\|\mathbf{x}_{i^k}^k - \widehat{\mathbf{y}}_{i^k}(\mathbf{x}^k)\|_2^2\ &\leq\ 2\left(1 + (1+L_B)(1+L_A+L_B)\right)\ \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&+ 2\gamma^2 L_A(1+L_A+L_B) \sum_{i=1}^N \bar{M}_i^k \sum_{l\in\bar{\mathcal{K}}_i^k} \|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l) - \mathbf{x}_{i^l}^l\|_2^2.
\end{aligned} \tag{3.48}$$

We take now the conditional expectation of the term on the LHS of (3.48), and obtain

$$\begin{aligned}
&\sum_{t=k}^{k+B} \mathbb{E}\left(\|\mathbf{x}_{\underline{i}^t}^t - \widehat{\mathbf{y}}_{\underline{i}^t}(\mathbf{x}^t)\|_2^2 | \mathcal{F}^{t-1}\right)(\omega)\ \overset{(a)}{=}\ \sum_{t=k}^{k+B} \sum_{i=1}^N p(i|\omega^{0:t-1})\,\|\mathbf{x}_i^t - \widehat{\mathbf{y}}_i(\mathbf{x}^t)\|_2^2 \\
&\overset{(b)}{\geq} \sum_{i=1}^N p_{\min}\,\|\mathbf{x}_i^{k+t_k(i)} - \widehat{\mathbf{y}}_i(\mathbf{x}^{k+t_k(i)})\|_2^2 \qquad (3.49) \\
&\overset{(c)}{\geq} p_{\min} \sum_{i=1}^N \left(\|\mathbf{x}_i^k - \widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2 - \|\mathbf{x}_i^{k+t_k(i)} - \widehat{\mathbf{y}}_i(\mathbf{x}^{k+t_k(i)}) - \mathbf{x}_i^k + \widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2\right)^2 \\
&\geq p_{\min} \sum_{i=1}^N \Big(\|\mathbf{x}_i^k - \widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2^2 \\
&\quad - 2\|\mathbf{x}_i^k - \widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2\,\|\mathbf{x}_i^{k+t_k(i)} - \widehat{\mathbf{y}}_i(\mathbf{x}^{k+t_k(i)}) - \mathbf{x}_i^k + \widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2\Big),
\end{aligned}$$

where in (a) we used (3.19); (b) follows from C2; and in (c) we used the reverse triangle inequality. By (3.49) and (3.48), we obtain:

$$
\begin{aligned}
&p_{\min}\sum_{i=1}^{N}\|\mathbf{x}_i^k-\widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2^2 \;=\; p_{\min}\,\|M_V(\mathbf{x}^k)\|_2^2\\
&\leq \sum_{t=k}^{k+B}\Bigg(2\left(1+(1+L_B)(1+L_A+L_B)\right)\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2|\mathcal{F}^{t-1}\right)(\omega)\\
&\quad+2\gamma^2L_A(1+L_A+L_B)\sum_{i=1}^{N}\bar{M}_i^t\sum_{l\in\bar{\mathcal{K}}_i^t}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2\Bigg)\\
&\quad+2p_{\min}\sum_{i=1}^{N}\|\mathbf{x}_i^k-\widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2\,\|\mathbf{x}_i^{k+t_k(i)}-\widehat{\mathbf{y}}_i(\mathbf{x}^{k+t_k(i)})-\mathbf{x}_i^k+\widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2\\
&\overset{(a)}{\leq} 2\left(1+(1+L_B)(1+L_A+L_B)\right)\sum_{t=k}^{k+B}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2|\mathcal{F}^{t-1}\right)(\omega)\\
&\quad+2B\gamma^2L_A(1+L_A+L_B)\sum_{i=1}^{N}M_i^k\sum_{l\in\mathcal{K}_i^k}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2+p_{\min}\alpha\,\|M_V(\mathbf{x}^k)\|_2^2\\
&\quad+p_{\min}\alpha^{-1}\sum_{i=1}^{N}\|\mathbf{x}_i^{k+t_k(i)}-\widehat{\mathbf{y}}_i(\mathbf{x}^{k+t_k(i)})-\mathbf{x}_i^k+\widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2^2\\
&\overset{(b)}{\leq} 2\left(1+(1+L_B)(1+L_A+L_B)\right)\sum_{t=k}^{k+B}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2|\mathcal{F}^{t-1}\right)(\omega)\\
&\quad+2B\gamma^2L_A(1+L_A+L_B)\sum_{i=1}^{N}M_i^k\sum_{l\in\mathcal{K}_i^k}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2+p_{\min}\alpha\,\|M_V(\mathbf{x}^k)\|_2^2\\
&\quad+2p_{\min}\alpha^{-1}\sum_{i=1}^{N}\left(\|\mathbf{x}_i^{k+t_k(i)}-\mathbf{x}_i^k\|_2^2+\|\widehat{\mathbf{y}}_i(\mathbf{x}^{k+t_k(i)})-\widehat{\mathbf{y}}_i(\mathbf{x}^k)\|_2^2\right)
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(c)}{\leq} 2\left(1+(1+L_B)(1+L_A+L_B)\right)\sum_{t=k}^{k+B}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2|\mathcal{F}^{t-1}\right)(\omega)\\
&\quad+2B\gamma^2L_A(1+L_A+L_B)\sum_{i=1}^{N}M_i^k\sum_{l\in\mathcal{K}_i^k}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2+p_{\min}\alpha\,\|M_V(\mathbf{x}^k)\|_2^2\\
&\quad+2\gamma^2p_{\min}\alpha^{-1}(1+L^2_{\widehat{\mathbf{y}}})\sum_{i=1}^{N}\sum_{l=k}^{k+t_k(i)-1}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2\\
&\leq 2\left(1+(1+L_B)(1+L_A+L_B)\right)\sum_{t=k}^{k+B}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2|\mathcal{F}^{t-1}\right)(\omega)\\
&\quad+2B\gamma^2L_A(1+L_A+L_B)\sum_{i=1}^{N}M_i^k\sum_{l\in\mathcal{K}_i^k}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2+p_{\min}\alpha\,\|M_V(\mathbf{x}^k)\|_2^2\\
&\quad+2\gamma^2p_{\min}\alpha^{-1}(1+L^2_{\widehat{\mathbf{y}}})N\sum_{l=k}^{k+B-1}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^2, \qquad (3.50)
\end{aligned}
$$

where in (a) we used the Young's inequality and the definition of M_i^k (cf. Section 3.4); in (b) we used the triangle and Jensen's inequalities; and (c) comes from the updating rule of the algorithm. Rearranging the terms and taking expectation of both sides, we get:

$$
\begin{aligned}
&\mathbb{E}\left(\|M_V(\underline{\mathbf{x}}^k)\|_2^2\right)\\
&\quad\leq\frac{2\left(1+(1+L_B)(1+L_A+L_B)+\gamma^2Np_{\min}\alpha^{-1}(1+L^2_{\widehat{\mathbf{y}}})\right)}{p_{\min}-\alpha p_{\min}}\\
&\qquad\sum_{t=k}^{k+B}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2\right)\\
&\qquad+\frac{2B\gamma^2L_A(1+L_A+L_B)}{p_{\min}-\alpha p_{\min}}\,\mathbb{E}\left(\sum_{i=1}^{N}\underline{M}_i^k\sum_{t\in\underline{\mathcal{K}}_i^k}\|\widehat{\mathbf{x}}_{\underline{i}^t}(\underline{\tilde{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2\right). \qquad (3.51)
\end{aligned}
$$

Invoking (3.9) and (3.31), we can write

$$
\begin{aligned}
&\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}^k_{i^k}\|_2^2\\
&\quad\leq\frac{1}{\gamma\left(\tau-\gamma\left(L+\frac{D^2L}{2}\right)\right)}\left(\tilde{V}(\mathbf{x}^k,\ldots,\mathbf{x}^{k-D})-\tilde{V}(\mathbf{x}^{k+1}\ldots,\mathbf{x}^{k+1-D})\right). \qquad (3.52)
\end{aligned}
$$

Using this bound in (3.51), we get

$$\mathbb{E}\left(\|M_V(\underline{\mathbf{x}}^k)\|_2^2\right) \leq C_1 \sum_{t=k}^{k+B} \mathbb{E}\left(\tilde{V}(\underline{\mathbf{x}}^t,\ldots,\underline{\mathbf{x}}^{t-D}) - \tilde{V}(\underline{\mathbf{x}}^{t+1},\ldots,\underline{\mathbf{x}}^{t+1-D})\right)$$
$$+\gamma^2 C_2\,\mathbb{E}\left(\sum_{i=1}^N \underline{M}_i^k \sum_{t\in\underline{\mathcal{K}}_i^k}\left(\tilde{V}(\underline{\mathbf{x}}^t,\ldots,\underline{\mathbf{x}}^{t-D}) - \tilde{V}(\underline{\mathbf{x}}^{t+1},\ldots,\underline{\mathbf{x}}^{t+1-D})\right)\right). \qquad (3.53)$$

Finally,

$$\begin{aligned} T_\epsilon \epsilon &\leq \sum_{k=0}^{T_\epsilon} \mathbb{E}\left(\|M_V(\underline{\mathbf{x}}^k)\|_2^2\right) \\ &\leq C_1 \sum_{k=0}^{T_\epsilon} \mathbb{E}\left(\tilde{V}(\underline{\mathbf{x}}^k,\ldots,\underline{\mathbf{x}}^{k-D}) - \tilde{V}(\underline{\mathbf{x}}^{k+B+1},\ldots,\underline{\mathbf{x}}^{k+B+1-D})\right) \\ &\quad + \gamma^2 C_2\,\mathbb{E}\left(\sum_{i=1}^N \underline{M}_i^k \sum_{t\in\underline{\mathcal{K}}_i^k}\left(\tilde{V}(\underline{\mathbf{x}}^t,\ldots,\underline{\mathbf{x}}^{t-D}) - \tilde{V}(\underline{\mathbf{x}}^{t+1},\ldots,\underline{\mathbf{x}}^{t+1-D})\right)\right) \\ &\leq C_1(B+1)(V(\mathbf{x}^0) - V^\star) \\ &\quad + C_2\gamma^2 \sum_{k=0}^{T_\epsilon} \mathbb{E}\left(\sum_{i=1}^N \underline{M}_i^k \sum_{t\in\underline{\mathcal{K}}_i^k}\left(\tilde{V}(\underline{\mathbf{x}}^t,\ldots,\underline{\mathbf{x}}^{t-D}) - \tilde{V}(\underline{\mathbf{x}}^{t+1},\ldots,\underline{\mathbf{x}}^{t+1-D})\right)\right). \end{aligned} \qquad (3.54)$$

This completes the proof.

3.8.3 Proof of Theorem 3.4.2

We prove the Theorem only under condition i) of Corollary 3.4.3. The proof of Theorem 3.4.2 under condition ii) of Corollary 3.4.3 can be easily obtained combining the following proof and the proof of Theorem 3.4.1 in the previous Section.

In this section, the best-response map $\widehat{\mathbf{x}}(\cdot)$ is the one defined in (3.3).

For any given realization $\omega \in \Omega$ and $k \geq D$, the following holds:

$$\begin{aligned}
&V(\mathbf{x}^{k+1}) = f(\mathbf{x}^{k+1}) + g(\mathbf{x}^{k+1}) \overset{\text{(a)}}{=} f(\mathbf{x}^{k+1}) + \sum_{i \neq i^k} g_i(\mathbf{x}_i^{k+1}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1}) \\
&\overset{\text{(b)}}{=} f(\mathbf{x}^{k+1}) + \sum_{i \neq i^k} g_i(\mathbf{x}_i^{k}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1}) \\
&\overset{\text{(c)}}{\leq} f(\mathbf{x}^{k}) + \gamma^k \left\langle \nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k), \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k \right\rangle + \frac{(\gamma^k)^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&+ \sum_{i \neq i^k} g_i(\mathbf{x}_i^{k}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1}) = f(\mathbf{x}^{k}) + \gamma^k \left\langle \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k \right\rangle \qquad (3.55) \\
&+ \left\langle \nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \mathbf{x}_{i^k}^{k+1} - \mathbf{x}_{i^k}^k \right\rangle + \frac{(\gamma^k)^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&+ \sum_{i \neq i^k} g_i(\mathbf{x}_i^{k}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1}) \overset{\text{(d)}}{\leq} f(\mathbf{x}^{k}) + \gamma^k \left\langle \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k \right\rangle \\
&+ \left\langle \nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) - \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \mathbf{x}_{i^k}^{k+1} - \mathbf{x}_{i^k}^k \right\rangle + \gamma^k \left\langle \nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k), \tilde{\mathbf{x}}_{i^k}^k - \mathbf{x}_{i^k}^k \right\rangle \\
&+ \frac{(\gamma^k)^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 + \sum_{i \neq i^k} g_i(\mathbf{x}_i^{k}) + \gamma^k g_{i^k}(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)) + (1-\gamma^k) g_{i^k}(\mathbf{x}_{i^k}^k) \\
&+ \gamma^k (g_{i^k}(\tilde{\mathbf{x}}_{i^k}^k) - g_{i^k}(\tilde{\mathbf{x}}_{i^k}^k)) \\
&\overset{\text{(e)}}{\leq} V(\mathbf{x}^k) - \gamma^k \tau \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k\|_2^2 + L \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \|\mathbf{x}_{i^k}^{k+1} - \mathbf{x}_{i^k}^k\|_2 \\
&+ \gamma^k (L_\nabla + L_g) \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 + \frac{(\gamma^k)^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&\overset{\text{(f)}}{\leq} V(\mathbf{x}^k) - \gamma^k \tau \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k\|_2^2 + \frac{L}{2} \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 + \frac{L}{2} \|\mathbf{x}_{i^k}^{k+1} - \mathbf{x}_{i^k}^k\|_2^2 \\
&+ \gamma^k (L_\nabla + L_g) \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 + \frac{(\gamma^k)^2 L}{2} \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 \\
&= V(\mathbf{x}^k) - \gamma^k \tau \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}_{i^k}^k\|_2^2 + \frac{L}{2} \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 \\
&+ (\gamma^k)^2 L \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2^2 + \gamma^k (L_\nabla + L_g) \|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2
\end{aligned}$$

where in (a) we used the separability of g; (b) follows from the updating rule of AsyFLEXA; in (c) we applied the Descent Lemma [9, Proposition A32] on F; (d) comes from A3; in (e) we used Proposition 3.8.2, A2-A5, and condition i) of Corollary

3.4.3, with $L_\nabla = \max_{i\in\mathcal{N}} L_{\nabla_i}$; and (f) is due to the Young's inequality.
We bound now the term $-\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}^k_{i^k}\|_2^2$ in (3.55). We have:

$$\begin{aligned}
-\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}^k_{i^k}\|_2^2 &= -\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k} + \mathbf{x}^k_{i^k} - \tilde{\mathbf{x}}^k_{i^k}\|_2^2 \\
&\le -\left(\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + \|\mathbf{x}^k_{i^k} - \tilde{\mathbf{x}}^k_{i^k}\|_2^2\right) + 2\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2\|\mathbf{x}^k_{i^k} - \tilde{\mathbf{x}}^k_{i^k}\|_2 \\
&\le -\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + \alpha\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + \alpha^{-1}\|\mathbf{x}^k_{i^k} - \tilde{\mathbf{x}}^k_{i^k}\|_2^2,
\end{aligned} \tag{3.56}$$

where α is any given positive constant. Substituting (3.56) in (3.55) we have:

$$\begin{aligned}
&V(\mathbf{x}^{k+1}) \le V(\mathbf{x}^k) - \gamma^k\tau(1-\alpha)\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + \gamma^k\tau\alpha^{-1}\|\mathbf{x}^k_{i^k} - \tilde{\mathbf{x}}^k_{i^k}\|_2^2 \\
&+ \frac{L}{2}\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 + (\gamma^k)^2 L\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + \gamma^k(L_\nabla + L_g)\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \\
&\le V(\mathbf{x}^k) - \gamma^k(\tau(1-\alpha) - \gamma^k L)\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 \\
&+ \left(\frac{L}{2} + \gamma^k\tau\alpha^{-1}\right)\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 + \gamma^k(L_\nabla + L_g)\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2
\end{aligned} \tag{3.57}$$

We bound $\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2$ as follows:

$$\begin{aligned}
&\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 \overset{\text{(a)}}{\le} \left(\sum_{l=k-D}^{k-1}\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\right)^2 \overset{\text{(b)}}{\le} D\sum_{l=k-D}^{k-1}\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 \\
&= D\left(\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 - \sum_{l=k+1-D}^{k}(l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right) \\
&+ D^2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2 \\
&= D\left(\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 - \sum_{l=k+1-D}^{k}(l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right) \\
&+ D^2(\gamma^k)^2\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2
\end{aligned} \tag{3.58}$$

where (a) comes from (3.25) and (b) is due to the Jensen's inequality. Similarly we have:

$$\begin{aligned}
&\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \leq \sum_{l=k-D}^{k-1} \|\mathbf{x}^l - \mathbf{x}^{l+1}\|_2 \\
&= \sum_{l=k-D}^{k-1} (l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 \\
&+ D\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 \\
&= \sum_{l=k-D}^{k-1} (l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 \qquad (3.59) \\
&+ D\gamma^k \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2 \\
&\leq \sum_{l=k-D}^{k-1} (l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 \\
&+ D\gamma^k \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \tilde{\mathbf{x}}^k_{i^k}\|_2 + D\gamma^k\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \\
&\overset{(a)}{\leq} \sum_{l=k-D}^{k-1} (l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + D\gamma^k M \\
&+ D\gamma^k\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2,
\end{aligned}$$

where (a) comes from Proposition 3.8.2 with $M = \max_{i \in \mathcal{N}} M_i$. We can rearrange the terms in (3.59) to get:

$$\begin{aligned}
&(1 - D\gamma^k)\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \\
&\leq \sum_{l=k-D}^{k-1} (l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + D\gamma^k M
\end{aligned} \qquad (3.60)$$

Invoking D1, there exists a sufficiently large k, say $\bar{k}$, such that:$(1 - D\gamma^{\bar{k}-1}) > 0$. For any $k \geq \bar{k}$ we also get: $(1 - D\gamma^{k}) > (1 - D\gamma^{\bar{k}-1}) > 0$. From these considerations and (3.60) we have:

$$\begin{aligned}\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2 \;\leq\; & (1 - D\gamma^k)^{-1}\Big(\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 \\ & - \sum_{l=k+1-D}^{k}(l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + D\gamma^k M\Big) \\ & \leq (1 - D\gamma^{k-1})^{-1}\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 \\ & - (1 - D\gamma^k)^{-1}\sum_{l=k+1-D}^{k}(l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + (1 - D\gamma^k)^{-1}D\gamma^k M\end{aligned} \tag{3.61}$$

Let us now define the following Lyapunov function. Let $k \geq \bar{k}$:

$$\begin{aligned}\tilde{V}(\mathbf{x}^k,\ldots,\mathbf{x}^{k-D}) = & V(\mathbf{x}^k) + D\left(\frac{L}{2} + \tau\alpha^{-1}\right)\left(\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right) \\ & + (L_\nabla + L_g)(1 - D\gamma^{k-1})^{-1}\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\end{aligned}$$

Using (3.58) and (3.61) in (3.57) and rearranging the terms, the following holds almost surely: for all $k \geq \bar{k}$

$$\begin{aligned}& \mathbb{E}(\tilde{V}(\underline{\mathbf{x}}^{k+1}\ldots,\mathbf{x}^{k+1-D})|\mathcal{F}^{k-1}) \leq \tilde{V}(\underline{\mathbf{x}}^k,\ldots,\underline{\mathbf{x}}^{k-D}) \\ & - \gamma^k\left(\tau(1-\alpha) - \gamma^k\left(L + \frac{D^2L}{2} + \gamma^k D^2\tau\alpha^{-1}\right)\right)\mathbb{E}(\|\widehat{\mathbf{x}}_{\underline{i}^k}(\tilde{\underline{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2|\mathcal{F}^{k-1}) + Z^k,\end{aligned} \tag{3.62}$$

where $Z^k \triangleq \frac{(\gamma^k)^2(L_\nabla + L_g)DM}{1-D\gamma^k}$. Since: $\lim_{k\to+\infty}\frac{Z^k}{(\gamma^k)^2} = (L_\nabla + L_g)DM$ and $\sum_{k=0}^{+\infty}(\gamma^k)^2 < +\infty$, we note that: $\sum_{k=0}^{+\infty} Z^k < +\infty$.

Invoking D1 and choosing $\alpha < 1$, we have that for sufficiently large k there exists some positive constant β_1 such that the following holds almost surely:

$$\begin{aligned}& \mathbb{E}(\tilde{V}(\underline{\mathbf{x}}^{k+1},\ldots,\mathbf{x}^{k+1-D})|\mathcal{F}^{k-1}) \\ & \leq \tilde{V}(\underline{\mathbf{x}}^k,\ldots,\underline{\mathbf{x}}^{k-D}) - \beta_1\gamma^k\mathbb{E}(\|\widehat{\mathbf{x}}_{\underline{i}^k}(\tilde{\underline{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2|\mathcal{F}^{k-1}) + Z^k.\end{aligned} \tag{3.63}$$

We can now apply Theorem 3.8.1 with the following identifications:
$\{\underline{\alpha}^k\} = \{\tilde{V}(\underline{\mathbf{x}}^k, \ldots, \underline{\mathbf{x}}^{k-D})\}$, $\{\underline{\eta}^k\} = \{Z^k\}$ and
$\{\underline{\nu}^k\} = \{\gamma^k \mathbb{E}(\|\widehat{\mathbf{x}}_{\underline{i}^k}(\tilde{\underline{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2 | \mathcal{F}^{k-1})\}$ (note that we assumed without loss of generality that $V(\underline{\mathbf{x}}^k) \geq 0$; indeed, since is uniformly bounded from below, one can always find a positive constant c and write instead $V(\underline{\mathbf{x}}^k) + c \geq 0$). We deduce that i) $\{\tilde{V}(\underline{\mathbf{x}}^k, \ldots, \underline{\mathbf{x}}^{k-D})\}$ converges almost surely; and ii).

$$\lim_{k\to+\infty} \sum_{t=\bar{k}}^{k} \left(\gamma^t \, \mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\tilde{\underline{\mathbf{x}}}^t) - \mathbf{x}^t_{\underline{i}^t}\|_2^2 | \mathcal{F}^{t-1}\right)\right) < +\infty \quad \text{a.s..} \tag{3.64}$$

Since $\tilde{V}(\underline{\mathbf{x}}^k, \ldots, \underline{\mathbf{x}}^{k-D})$ is coercive from A4, this implies that the sequence $\{\mathbf{x}^k\}$ is bounded almost surely.
Since $\sum_{k=0}^{+\infty} \gamma^k = +\infty$, it follows from (3.64) that there exists a $\bar{\Omega} \subseteq \Omega$, with $\mathbb{P}(\bar{\Omega}) = 1$, such that for any $\omega \in \bar{\Omega}$:

$$\liminf_{k\to+\infty} \mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^k}(\tilde{\underline{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2 | \mathcal{F}^{k-1}\right)(\omega) = 0. \tag{3.65}$$

Since we have from Proposition 3.8.1:

$$\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^k}(\tilde{\underline{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2 | \mathcal{F}^{k-1}\right)(\omega) = \sum_{(i,\mathbf{d})\in\mathcal{N}\times\mathcal{D}} p\left((i,\mathbf{d})|\boldsymbol{\omega}^{0:k-1}\right) \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2^2 \tag{3.66}$$

(recall that we are using the shorthand notation:
$\mathbf{x}^k = \underline{\mathbf{x}}^k(\omega)$ and $\tilde{\mathbf{x}}^k(\mathbf{d}) = (\tilde{\underline{\mathbf{x}}}^k(\mathbf{d}))(\omega)$), it follows from (3.65) that

$$\liminf_{k\to+\infty} \sum_{(i,\mathbf{d})\in\mathcal{N}\times\mathcal{D}} p\left((i,\mathbf{d})|\omega^{0:k-1}\right) \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2^2 = 0. \tag{3.67}$$

Under C3', there exists a $D(\omega) > 0$ such that [cf. (3.28)]

$$\begin{aligned} &\sum_{(i,\mathbf{d})\in\mathcal{N}\times\mathcal{D}} p\left((i,\mathbf{d})|\omega^{0:k-1}\right) \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2^2 \\ &\geq D(\omega) \sum_{(i,\mathbf{d})\in\mathcal{V}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2^2 \geq D(\omega) \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2^2. \end{aligned} \tag{3.68}$$

Using (3.67) and (3.68), we obtain $\liminf_{k\to+\infty} \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2 = 0$. We prove next that $\lim_{k\to+\infty} \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2 = 0$. To do so, it is sufficient to show that the limsup of the same quantity goes to zero. Let us set for notational simplicity $z^k \triangleq \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2$. We show next that $\lim_{k\to+\infty} z^k = 0$. Suppose by contradiction that $\limsup_{k\to+\infty} z^k > 0$. Since it is also $\liminf_{k\to+\infty} z^k = 0$, there exists a $\upsilon > 0$ such that $z^k > 2\upsilon$ for infinitely many k and also $z^k < \upsilon$ for infinitely many k. Therefore, one can always find an infinite set of indexes, say $\mathcal{S}$, having the following properties: for any $k \in \mathcal{S}$, there exists an integer $j_k > k$ such that

$$z^k < \upsilon, \qquad z^{j_k} > 2\upsilon \tag{3.69}$$

$$\upsilon \le z^j \le 2\upsilon \qquad k < j < j_k. \tag{3.70}$$

Given the above bounds, the following holds: for all $k \in \mathcal{S}$,

$$
\begin{aligned}
\upsilon < z^{j_k} - z^k = & \sum_{(i,\mathbf{d}) \in \tilde{\mathcal{V}}^{j_k}(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{j_k}(\mathbf{d})) - \mathbf{x}_i^{j_k}\|_2 - \sum_{(i,\mathbf{d}) \in \tilde{\mathcal{V}}^{k}(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k}(\mathbf{d})) - \mathbf{x}_i^{k}\|_2 \\
\overset{\text{(a)}}{\leq} & \sum_{i=1}^N \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{j_k}(\mathbf{d}_i')) - \mathbf{x}_i^{j_k} - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d}_i)) + \mathbf{x}_i^k\|_2 \\
\leq & \sum_{i=1}^N \|\mathbf{x}_i^{j_k} - \mathbf{x}_i^k\|_2 + \sum_{i=1}^N \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{j_k}(\mathbf{d}_i')) - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d}_i))\|_2 \\
\overset{\text{(b)}}{\leq} & N\|\mathbf{x}^{j_k} - \mathbf{x}^k\|_2 + \tilde{L}\sum_{i=1}^N \|\tilde{\mathbf{x}}^{j_k}(\mathbf{d}_i') - \tilde{\mathbf{x}}^k(\mathbf{d}_i)\|_2 \overset{\text{(c)}}{=} N\sum_{t=k}^{j_k-1} \gamma^t \|\widehat{\mathbf{x}}_{i^t}(\tilde{\mathbf{x}}^t(\mathbf{d}^t)) - \mathbf{x}_{i^t}^t\|_2 \\
& + \tilde{L}\sum_{i=1}^N \|\mathbf{x}^{j_k} + \sum_{l \in K^{j_k}(\mathbf{d}_i')} (\mathbf{x}^l - \mathbf{x}^{l+1}) - \mathbf{x}^k + \sum_{h \in K^k(\mathbf{d}_i)} (\mathbf{x}^{h+1} - \mathbf{x}^h)\|_2 \\
\leq & N\sum_{t=k}^{j_k-1} \gamma^t \|\widehat{\mathbf{x}}_{i^t}(\tilde{\mathbf{x}}^t(\mathbf{d}^t)) - \mathbf{x}_{i^t}^t\|_2 + \tilde{L}N\|\mathbf{x}^{j_k} - \mathbf{x}^k\|_2 \\
& + \tilde{L}\sum_{i=1}^N \left(\sum_{l \in K^{j_k}(\mathbf{d}_i')} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + \sum_{h \in K^k(\mathbf{d}_i)} \|\mathbf{x}^{h+1} - \mathbf{x}^h\|_2 \right) \\
\leq & (1+\tilde{L})N\sum_{t=k}^{j_k-1} \gamma^t \|\widehat{\mathbf{x}}_{i^t}(\tilde{\mathbf{x}}^t(\mathbf{d}^t)) - \mathbf{x}_{i^t}^t\|_2 \\
& + \tilde{L}N\left(\sum_{l=j_k-D}^{j_k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + \sum_{h=k-D}^{k-1} \|\mathbf{x}^{h+1} - \mathbf{x}^h\|_2 \right) \overset{\text{(d)}}{\leq} 2(1+\tilde{L})N\upsilon \sum_{t=k}^{j_k-1} \gamma^t \\
& + \tilde{L}N\left(\sum_{l=j_k-D}^{j_k-1} \gamma^l \|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l(\mathbf{d}^l)) - \mathbf{x}_{i^l}^l\|_2 + \sum_{h=k-D}^{k-1} \gamma^h \|\widehat{\mathbf{x}}_{i^h}(\tilde{\mathbf{x}}^h(\mathbf{d}^h)) - \mathbf{x}_{i^h}^h\|_2 \right) \\
\overset{\text{(e)}}{\leq} & 2(1+\tilde{L})N\upsilon \sum_{t=k-D}^{j_k-1} \gamma^t + 2\tilde{L}N\bar{M}\sum_{t=k-D}^{j_k-1} \gamma^t \\
\leq & 2N(\upsilon(1+\tilde{L}) + \tilde{L}\bar{M}) \sum_{t=k-D}^{j_k-1} \gamma^t, \qquad (3.71)
\end{aligned}
$$

where in (a) we used the definition of the sets $\tilde{\mathcal{V}}^{j_k}(\omega)$ and $\tilde{\mathcal{V}}^k(\omega)$ and we denoted by $\mathbf{d}_i'$ and $\mathbf{d}_i$ the d-component of the pairs $(i, \mathbf{d}') \in \tilde{\mathcal{V}}^{j_k}(\omega)$ and $(i, \mathbf{d}) \in \tilde{\mathcal{V}}^k(\omega)$, respectively; in (b) we used Proposition 3.8.2; (c) follows from (3.25) and Step 4 of AsyFLEXA; (d) is due to (3.69) and (3.70); and in (e) we used the the boundedness of the iterates implying $\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2 < \bar{M}$, for some finite constant $\bar{M} > 0$, and any k. It follows from (3.71) that

$$\liminf_{k\to+\infty} \sum_{t=k-D}^{j_k-1} \gamma^t \geq \frac{\upsilon}{2N(\upsilon(1+\tilde{L}) + \tilde{L}\bar{M})} > 0. \tag{3.72}$$

Consider now the following term: for $k \in \mathcal{S}$, let

$$\begin{aligned}
z^{k+1} - z^k \;&\leq\; N\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 \\
&+\tilde{L}N\left(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 + \sum_{l=k-D+1}^{k} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2 + \sum_{h=k-D}^{k-1} \|\mathbf{x}^{h+1} - \mathbf{x}^h\|_2\right) \\
&\leq N(1+2\tilde{L})\gamma^k \|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k(\mathbf{d}^k)) - \mathbf{x}_{i^k}^k\|_2 + 2\bar{M}\tilde{L}N \sum_{t=k-D}^{k-1} \gamma^t \\
&\overset{\text{(a)}}{\leq} \gamma^k N(1+2\tilde{L})\left(\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k(\mathbf{d}^k)) - \mathbf{x}_{i^k}^k\|_2 + \tfrac{2\bar{M}\tilde{L}}{1+2\tilde{L}} \sum_{t=k-D}^{k-1} \left(\tfrac{1}{\eta}\right)^{k-t}\right) \\
&\overset{\text{(b)}}{=} \gamma^k N(1+2\tilde{L})\left(\|\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k(\mathbf{d}^k)) - \mathbf{x}_{i^k}^k\|_2 + \tfrac{2\bar{M}\tilde{L}}{1+2\tilde{L}}\theta\right) \\
&\leq \gamma^k N(1+2\tilde{L})\left(z^k + \tfrac{2\bar{M}\tilde{L}}{1+2\tilde{L}}\theta\right);
\end{aligned} \tag{3.73}$$

where in (a) we used D2 and in (b) we defined $\theta \triangleq \sum_{t=k-D}^{k-1} \left(\frac{1}{\eta}\right)^{k-t} = \sum_{l=1}^{D} \left(\frac{1}{\eta}\right)^{l}$. It turns out that for sufficiently large $k \in \mathcal{S}$ so that $\gamma^k N(1+2\tilde{L}) < \frac{\upsilon}{\upsilon + 2\frac{2\bar{M}\tilde{L}}{1+2\tilde{L}}\theta}$, it must be $z^k \geq \upsilon/2$; otherwise the condition $z^{k+1} \geq \upsilon$ would be violated [cf.(3.70)]. In the same way, using $z^k \geq \upsilon/2$, one can now show that it must be $z^{k-1} \geq \upsilon/4$, otherwise $z^k \geq \upsilon/2$ would be violated. By applying iteratively this same reasoning one can show that for each $j \in [k-D, k]$: $z^j \geq \upsilon/2^{k+1-j}$. For the ω we are considering, (3.64) holds true; therefore, $\lim_{k\to+\infty} \sum_{t=k-D}^{j_k-1} \gamma^t z^t = 0$, that together with (3.70) and the relation $z^j \geq \upsilon/2^{k+1-j}$. just proved, implies $\lim_{k\to+\infty} \sum_{t=k-D}^{j_k-1} \gamma^t = 0$. But this equality contradicts (3.72). Hence $\limsup_{k\to+\infty} z^k = 0$ and consequently $\lim_{k\to+\infty} z^k = 0$.

Using C2' and $\lim_{k\to+\infty} z^k = 0$, let us prove now that $\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_2 = 0$. To begin, notice that $\lim_{k\to+\infty} z^k = 0$ implies $\lim_{k\to+\infty} \sum_{l=k-D}^{k} z^l = 0$. Let us now analyze the following term

$$\begin{aligned}
&\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_2 \;\leq\; \sum_{i=1}^{N} \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|_2 \\
&= \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) + \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2 \\
&\leq \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2 + \tilde{L}\sum_{i=1}^{N} \|\mathbf{x}^k - \tilde{\mathbf{x}}^k(\mathbf{d}_i)\|_2 \\
&\overset{\text{(a)}}{\leq} \sum_{(i,\mathbf{d})\in\tilde{\mathcal{V}}^k(\omega)} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2 + N\tilde{L}\sum_{l=k-D}^{k-1} \|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l(\mathbf{d}^l)) - \mathbf{x}_{i^l}^l\|_2 \\
&\leq z^k + N\tilde{L}\sum_{l=k-D}^{k-1} z^l \;\leq\; \beta_2 \sum_{l=k-D}^{k} z^l,
\end{aligned} \tag{3.74}$$

for some $\beta_2 > 0$, where (a) follows from (3.25) and Step 4 of AsyFLEXA. The bound (3.74) together with $\lim_{k\to+\infty} \sum_{l=k-D}^{k} z^l = 0$. leads to $\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_2 = 0$. Finally, since the sequence $\{\mathbf{x}^k\}$ is bounded, it has a limit point in $\mathcal{X}$, denoted by $\bar{\mathbf{x}}$. By the continuity of $\widehat{\mathbf{x}}(\cdot)$ (cf. Proposition 3.8.2) and the equality $\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_2 = 0$. we just proved, it must be $\widehat{\mathbf{x}}(\bar{\mathbf{x}}) = \bar{\mathbf{x}}$. By Proposition 3.8.2, $\bar{\mathbf{x}}$ is also a stationary solution of Problem (P).
The above result holds for any $\omega \in \bar{\Omega}$, which completes the proof.

3.8.4 Proof of Theorem 3.5.1

In this section, the best-response map $\widehat{\mathbf{x}}(\cdot)$ is the one defined in (3.15).

Statement (ii) of the theorem follow readily from the feasibility of $\mathbf{x}^0 \in \mathcal{K}$ and the fact that $\mathbf{x}_{i^k}^{k+1} = \mathbf{x}_{i^k}^k + \gamma(\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k)$ is a convex combinations of points in $\mathcal{K}_{i^k}(\mathbf{x}_{i^k}^k)$.

To prove statement (ii), let us fix a realization $\omega \in \Omega$. Following the steps from (3.29) to (3.33), one can prove that the following holds a.s.:

$$\mathbb{E}\left(\tilde{V}\left(\underline{\mathbf{x}}^{k+B}\dots,\underline{\mathbf{x}}^{k+B-D}\right)|\mathcal{F}^{k-1}\right) \le \tilde{V}(\underline{\mathbf{x}}^{k},\dots,\underline{\mathbf{x}}^{k-D})$$
$$-\gamma\left(\tau-\gamma\left(L+\frac{D^2L}{2}\right)\right)\sum_{t=k}^{k+B-1}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\tilde{\underline{\mathbf{x}}}^t)-\mathbf{x}^t_{\underline{i}^t}\|_2^2|\mathcal{F}^{t-1}\right). \tag{3.75}$$

Using (3.9), (3.75) and A4', we deduce that i) $\left\{V(\underline{\mathbf{x}}^k,\dots,\underline{\mathbf{x}}^{k-D})\right\}$ converges a.s., and ii)

$$\lim_{k\to+\infty}\sum_{t=k}^{k+B-1}\mathbb{E}\left(\|\widehat{\mathbf{x}}_{\underline{i}^t}(\tilde{\underline{\mathbf{x}}}^t)-\underline{\mathbf{x}}^t_{\underline{i}^t}\|_2|\mathcal{F}^{t-1}\right)=0 \quad \text{a.s.} \tag{3.76}$$

It follows from (3.76) and C2 that

$$\lim_{k\to+\infty}\sum_{i=1}^{N}\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)})-\mathbf{x}_i^{k+t_k(i)}\|_2=0 \quad \text{a.s.} \tag{3.77}$$

Therefore, there exists a set $\bar{\Omega}\subseteq\Omega$, with $\mathbb{P}(\bar{\Omega})=1$, such that, for any $\omega\in\bar{\Omega}$,

$$\begin{aligned}
\|\widehat{\mathbf{x}}(\mathbf{x}^k)-\mathbf{x}^k\|_2 &\le \sum_{i=1}^{N}\|\widehat{\mathbf{x}}_i(\mathbf{x}^k)-\mathbf{x}_i^k\|_2 \overset{(a)}{\le} \sum_{i=1}^{N}\Bigg(\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)})-\mathbf{x}_i^{k+t_k(i)}\|_2\\
&\quad+\|\tilde{\mathbf{x}}^{k+t_k(i)}-\mathbf{x}^k\|_2^{1/2}\left(\|\tilde{\mathbf{x}}^{k+t_k(i)}-\mathbf{x}^k\|_2^{1/2}+\tilde{\tilde{L}}\right)\Bigg)\\
&\overset{(b)}{\le}\sum_{i=1}^{N}\Bigg(\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)})-\mathbf{x}_i^{k+t_k(i)}\|_2+\Bigg(\|\mathbf{x}^{k+t_k(i)}-\mathbf{x}^k\|_2^{1/2}\\
&\quad+\sum_{l=k+t_k(i)-D}^{k+t_k(i)-1}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2^{1/2}\Bigg)\\
&\quad\left(\|\mathbf{x}^{k+t_k(i)}-\mathbf{x}^k\|_2^{1/2}+\sum_{l=k+t_k(i)-D}^{k+t_k(i)-1}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2^{1/2}+\tilde{\tilde{L}}\right)\Bigg)\\
&\overset{(c)}{\le}\sum_{i=1}^{N}\Bigg(\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^{k+t_k(i)})-\mathbf{x}_i^{k+t_k(i)}\|_2\\
&\quad+2\sqrt{\gamma}\sum_{l=k-D}^{k+B-1}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^{1/2}\left(2\sqrt{\gamma}\sum_{l=k-D}^{k+B-1}\|\widehat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l)-\mathbf{x}^l_{i^l}\|_2^{1/2}+\tilde{\tilde{L}}\right)\Bigg),
\end{aligned} \tag{3.78}$$

where in (a) we used Proposition 3.8.3; (b) comes from (3.25); and in (c) we used the updating rule of AsyFLEXA-NCC. Using (3.76), (3.77) and (3.78), we conclude that

$$\lim_{k\to+\infty}\|\widehat{\mathbf{x}}(\mathbf{x}^k)-\mathbf{x}^k\|_2=0. \tag{3.79}$$

A straightforward generalization of [76, Theorem 14] together with (3.79) proves that every limit point of $\{\mathbf{x}^k\}$ is a stationary solution of Problem (P$'$). Since (3.79) holds for any given realization $\omega \in \bar{\Omega}$, the above results hold a.s..

Iteration complexity can be proved following the steps (3.38)-(3.54) and using the convexification of the nonconvex constraint sets where needed; details are omitted.

4. DASYFLEXA: DISTRIBUTED ASYNCHRONOUS ALGORITHM FOR NONSMOOTH NONCONVEX OPTIMIZATION

We consider convex and nonconvex constrained optimization with a partially separable objective function: agents minimize the sum of local objective functions, each of which is known only by the associated agent and depends on the variables of that agent and those of a few others. This partitioned setting arises in several applications of practical interest. We propose the first distributed, asynchronous algorithm with rate guarantees for this class of problems. When the objective function is nonconvex, the algorithm provably converges to a stationary solution at a sublinear rate whereas linear rate is achieved under the renowned Luo-Tseng error bound condition (which is less stringent than strong convexity). Numerical results on matrix completion and LASSO problems show the effectiveness of our method.

The rest of the Chapter is organized as follows. Section 4.1 presents the class of problems under exam, together with the major contributions of this Chapter, and a discussion of related works. Section 4.2 show some motivating application that falls into the partitioned problem setting considered in this Chapter. The proposed algorithm is introduced and analyzed in Section 4.3. Section 4.4 presents the main convergence results. Section 3.5 contains an extension of the results of this Chapter to the case involving the presence of nonconvex constraints. Finally, Section 3.6 shows experimental results on convex and nonconvex problems. The proofs of the Theorems are collected in the Appendix of this Chapter.

The novel results of this Chapter have been published in

- Cannelli, L. and Facchinei, F. and Scutari, G., and Kungurtsev, V., *"Asynchronous Optimization over Graphs: Linear Convergence under Error Bound Conditions"*, submitted to IEEE Transactions on Automatic Control, 2019.

- Cannelli, L. and Facchinei, F. and Scutari, G., *"Multi-agent asynchronous nonconvex large-scale optimization"*, IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pages 1-5, 2017 [83].

- Cannelli, L. and Facchinei, F. and Kungurtsev, V. and Scutari, G., *"Essentially Cyclic Asynchronous Nonconvex Large-Scale Optimization"*, IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) (Sapporo), pages 1-5, 2017 [84]. **Best Paper Award Runner-up**

4.1 Introduction

In this Chapter we study distributed, nonsmooth, nonconvex optimization with a partially separable sum-cost function. Specifically, consider a set of N agents, each of them controlling/updating a subset of the n variables $\mathbf{x} \in \mathbb{R}^n$. Partitioning $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)^T$, $\mathbf{x}_i \in \mathbb{R}^{n_i}$ is the block of variables owned by agent $i \in \mathcal{N} \triangleq \{1, \ldots, N\}$, with $\sum_i n_i = n$. All agents cooperatively aim at solving the following problem:

$$\min_{\mathbf{x}_i \in \mathcal{X}_i, i \in \mathcal{N}} \quad V(\mathbf{x}) \triangleq \underbrace{\sum_{i=1}^{N} f_i(\mathbf{x}_{\mathcal{N}_i})}_{\triangleq F(\mathbf{x})} + \underbrace{\sum_{i=1}^{N} g_i(\mathbf{x}_i)}_{\triangleq G(\mathbf{x})}, \tag{P}$$

where $\mathcal{N}_i$ denotes a small subset of $\mathcal{N}$ including the index i and $\mathbf{x}_{\mathcal{N}_i} \triangleq [\mathbf{x}_j]_{j \in \mathcal{N}_i}$ denotes the column vector containing the blocks of $\mathbf{x}$ indexed by $\mathcal{N}_i$; $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is a closed convex set; f_i is a smooth (nonconvex) function that depends only on $\mathbf{x}_{\mathcal{N}_i}$; and g_i is a convex (nonsmooth) function, instrumental to encode structural constraints on the solution, such as sparsity. Both f_i and g_i are assumed to be known only by agent i.

The above formulation is motivated by a variety of applications of practical interest in different fields. For instance, natural loss functions arising from many machine learning problems have the "sparse" pattern of V in (P): n and N are both very large but each f_i depends only on a very small number of components of $\mathbf{x}$; i.e., each subvector $\mathbf{x}_{\mathcal{N}_i}$ contains just a few components of $\mathbf{x}$. The same partitioned structure in (P) is suitable also to model networked systems wherein agents are connected through a physical communication network and can communicate only with their immediate neighbors. In this setting, often $\mathcal{N}_i$ represents the set of neighbors of agent i (including agent i itself). Examples of such applications include resource allocation problems and network utility maximization [85], state estimation in power networks [86], cooperative localization in wireless networks [87], and map building in robotic networks. Some concrete instances of Problem (P) are discussed in Section 4.2.

4.1.1 Major Contributions

We focus on the design of distributed, asynchronous algorithms for (P), in the following sense: i) Agents can update their block-variables at any time, without any coordination; and ii) when updating their own variables, agents can use a delayed out-of-sync information from the others. In (ii) no constraint is imposed on the delay profiles: delays can be arbitrary, possibly time-varying (but bounded). 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, unknown, time-varying delays.

While several forms of asynchrony have been studied in the literature–see Section 4.1.2 for an overview of most relevant results–we are not aware of any distributed scheme that is compliant to the asynchronous model (i)-(ii) and tailored to the *partitioned* (nonconvex) formulation (P). This Chapter fills this gap and proposes a general distributed, asynchronous algorithmic framework for convex and nonconvex instances of (P). The algorithm builds on SCA techniques: agents solve asynchronously [in the sense (i) and (ii) above] strongly convex approximations of the original problem (P) by using (possibly) outdated information on the variables and the gradients of the other agents. 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. For nonconvex instances of V, we prove that i) every limit point of the sequence generated by the proposed asynchronous algorithm is a stationary solution of (P); and ii) a suitable measure of stationarity vanishes at a sublinear rate. When V further satisfies the Luo-Tseng error bound condition [24, 33], both the sequence and the objective value converge at an R-linear rate. This error bound condition is weaker than strong convexity and it is satisfied by a variety of problems of interest, such as LASSO, Group LASSO, and Logistic Regression, just to name a few (cf. Section 4.3.1). While linear convergence under error bounds has been proved in the literature for many *centralized* algorithms [24, 25, 27, 39, 59], we are not aware of any such a result in the distributed setting; current works require strong convexity to establish linear rate of synchronous and asynchronous *distributed* algorithms (see, e.g., [88–91] and references therein). This Chapter represents the first effort proving linear convergence of a *distributed asynchronous* algorithm without invoking strong convexity. As a byproduct, our results provide a positive answer to the open question whether linear convergence could be proved for *distributed* asynchronous algorithms solving highly dimensional empirical risk minimization problems, such as LASSO and Logistic Regression, a fact that was empirically observed but, to our knowledge, never proved.

4.1.2 Related Works

Since the seminal work [73], asynchronous parallelism has been applied to several centralized solution methods, including (randomized) block-coordinate descent schemes [9, 46, 47, 49, 50, 53, 54, 59, 73], and stochastic gradient algorithms [61, 62, 92]. However, those methods are not applicable to Problem (P), since they would require each agent to know the entire objective function V.

Distributed schemes exploring (some form of) asynchrony have been studied in [55, 67–69, 93–109]; next, we group them based upon the asynchrony features (i) and (ii).

(a) *Random activation and no delays* [68, 69, 93–98]: While substantially different in the form of the updates performed by the agents, these schemes are all asynchronous in the sense of feature (i) only–agents (or edge-connected agents) are randomly activated but, when performing their computations/updates, they must use the current information from their neighbors. This means that no form of delay is allowed. Furthermore, between two activations, agents are assumed to be in idle mode (i.e., able to continuously receive information). Some form of coordination is thus needed to enforce the above conditions. With the exception of [97], all the schemes in this group can deal only with convex objectives; and none of the above works provide a convergence rate or complexity analysis.

(b) *Synchronous activation and delays* [99–104]: These schemes consider synchronous activation/updates of the agents, which can tolerate fixed computation delays (e.g., outdated gradient information) [99, 100] or fixed [101, 104] or time-varying [102, 103] communication delays. However delays cannot be arbitrary, but must be such that no loss can ever occur in the network: every agent's message must reach its intended destination within a finite time interval. Finally, all these algorithms are applicable only to convex problems.

(c) *Random/cyclic activations and some form of delay* [55,67,105–109]: These schemes allow for random [55, 105, 106, 108] or deterministic uncoordinated [67, 107, 109] activation of the (edge-based) agents, together with the presence of some form of delay in the updates/computations. Specifically, [105, 106, 109] can handle link failures–the information sent by an agent to its neighbors either gets lost or received with *no delay*–but cannot deal with other forms of delay (e.g., communication delays). In [55, 108] a probabilistic model is assumed whereby agents are randomly activated and update their local variables using possibly delayed information. The model requires that the random variables modeling the activation of the agents are i.i.d and independent of the delay vector used by the agent to perform its update. While this assumption makes the convergence analysis possible, in reality, there is a strong dependence of the delays on the activation index, as also noted by the same authors [55,108]; see [46,47] for a detailed discussion on this issue and several counter examples. Closer to our setting are the asynchronous methods in [55,67,88,107,108]. These models however assume that each function f_i depends on the *entire* vector $\mathbf{x}$. As a consequence, a consensus mechanism on all the optimization variables is employed among the agents at each iteration. Because of that, a direct application of these consensus-based algorithms to the partitioned formulation (P) would lead to very inefficient schemes calling for unnecessary computation and communication overheads. Finally, notice that, with the exception of [67,88,106] (resp. [109]), all these schemes are applicable to convex problems (resp. undirected graphs) only, with [105] further assuming that all the functions f_i have the same minimizer.

4.2 Motivating Examples

We discuss next two instances of Problem (P), which will be also used in our numerical experiments to test our algorithms (cf. Section 4.5). The first case study is the matrix completion problem–an example of large-scale nonconvex empirical risk minimization. We show how to exploit the sparsity pattern in the data to rewrite the problem in the form (P), so that efficient asynchronous algorithms levering multi-core architectures can be developed. The second example deals with learning problems from networked data sets; in this setting data are distributed across multiple nodes, whose communication network is modeled as a (directed) graph. The goal here is to design asynchronous algorithms that run on network, without requiring the presence of a sink node collecting the information (or data) from all the agents.

Example #1 –Matrix completion: The matrix completion problem consists of estimating a low-rank matrix $\mathbf{Z} \in \mathbb{R}^{M\times N}$ from a subset $\Omega \subseteq \{1,\ldots,M\} \times \{1,\ldots,N\}$ of its entries. Postulating the low-rank factorization $\mathbf{Z} = \mathbf{X}^T\mathbf{Y}$, with $\mathbf{X} \in \mathbb{R}^{r\times M}$ and $\mathbf{Y} \in \mathbb{R}^{r\times N}$, the optimization problem reads [110]:

$$\min_{\substack{\mathbf{X}\in\mathbb{R}^{r\times M}\\ \mathbf{Y}\in\mathbb{R}^{r\times N}}} V(\mathbf{X},\mathbf{Y}) \triangleq \frac{1}{2}\left\|(\mathbf{X}^T\mathbf{Y}-\mathbf{Z})_\Omega\right\|_F^2 + \frac{\lambda}{2}\left\|\mathbf{X}\right\|_F^2 + \frac{\xi}{2}\left\|\mathbf{Y}\right\|_F^2, \tag{4.1}$$

where $\|\cdot\|_F$ is the Frobenius norm; $(\cdot)_\Omega$ is the projection operator, defined as $[(\mathbf{X})_\Omega]_{(i,j)} = \mathbf{X}_{(i,j)}$, if $(i,j) \in \Omega$; and $[(\mathbf{X})_\Omega]_{(i,j)} = 0$ otherwise; and $\lambda, \xi > 0$ are regularization parameters. In many applications, the amount of data is so large that storage and processing from a single agent (e.g., core, machine) is not efficient or even feasible. The proposed approach is then to leverage multi-core machines by first casting (4.1) in the form (P), and then employing the parallel asynchronous framework developed in this paper.

Consider a distributed environment composed of N agents, and assume that the known entries z_{mn}, $(m,n) \in \Omega$, are partitioned among the agents. This partition along with the sparsity pattern of $(\mathbf{Z})_\Omega$ induce naturally the following splitting of the optimization variables $\mathbf{X}$ and $\mathbf{Y}$ across the agents. Let $\mathbf{x}_m$ and $\mathbf{y}_n$ denote the m-th and the n-th column of $\mathbf{X}$ and $\mathbf{Y}$, respectively; the agent owning z_{mn} will control/update the variables $\mathbf{x}_m$ (or $\mathbf{y}_n$), and it is connected to the agent that optimizes the column $\mathbf{y}_n$ (or $\mathbf{x}_m$). By doing so, we minimize the overlapping across the block-variables and, consequently, the communications among the agents. Problem (4.1) can be then rewritten in the multi-agent form (P), setting

$$f_i((\mathbf{X},\mathbf{Y})_{\mathcal{N}_i}) = \frac{1}{2}\sum_{(m,n)\in\Omega_i} (\mathbf{x}_m^T\mathbf{y}_n - z_{mn})^2 \tag{4.2}$$

and

$$g_i\left(\{\mathbf{x}_m\}_{m\in X_i}, \{\mathbf{y}_n\}_{n\in Y_i}\right) = \frac{\lambda}{2}\sum_{m\in X_i}\|\mathbf{x}_m\|_2^2 + \frac{\xi}{2}\sum_{n\in Y_i}\|\mathbf{y}_n\|_2^2, \tag{4.3}$$

where $\Omega_i \subseteq \Omega$ contains the indices associated to the components of $(\mathbf{Z})_\Omega$ owned by agent i, and X_i (resp. Y_i) is the set of the column indexes of $\mathbf{X}$ (resp. $\mathbf{Y}$) controlled by agent i.

Example #2 – Empirical risk minimization over networks: Consider now a network setting where data are distributed across N geographically separated nodes. As concrete example, let us pick the renowned LASSO problem [5]:

$$\min_{\mathbf{x}=(\mathbf{x}_1^T,\ldots,\mathbf{x}_N^T)^T\in\mathbb{R}^n} \quad \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2+\lambda\|\mathbf{x}\|_1, \tag{4.4}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}, \mathbf{b}\in\mathbb{R}^m$, and $\lambda>0$ is a regularization parameter. Note that (4.4) easily falls into Problem (P); for each $i\in\mathcal{N}$, it is sufficient to set $f_i(\mathbf{x})=\|\mathbf{A}_i\mathbf{x}+\mathbf{b}_i\|_2^2$, with $\mathbf{A}_i\in\mathbb{R}^{m\times n}$ and $\mathbf{b}_i\in\mathbb{R}^m$ such that $\mathbf{A}=\sum_{i=1}^N\mathbf{A}_i$ and $\mathbf{b}=\sum_{i=1}^N\mathbf{b}_i$; and $g_i=\|\mathbf{x}_i\|_1$. $\mathbf{A}_i$ and $\mathbf{b}_i$ represent in fact the data stored at agent i's side. Under specific sparsity patterns in the data, the local matrices $\mathbf{A}_i$ may be (or constructed to be) such that each local function f_i depends only on some of the block variables $\mathbf{x}_i$. These dependencies will define the sets $\mathcal{N}_i$ associated to each agent i. Note that $\mathcal{N}_i$ need not coincide with the neighbors of agent i in the communication network (graph). That is, the graph modeling the dependence across the block-variables–the one with node set $\mathcal{N}$ and edge set $\mathcal{E}=\{(i,j):j\in\mathcal{N}_i,\text{for some}\, i\in\mathcal{N}\}$–might not coincide with the communication graph. This can be desirable, e.g., when the communication graph is populated by inefficient communication links, which one wants to avoid to use.

4.3 Distributed Asynchronous Algorithm

In the proposed asynchronous model, agents update their block-variables without any coordination. Let k be the iteration counter: the iteration $k\to k+1$ is triggered when an agent, say i, updates its own block $\mathbf{x}_i$ from $\mathbf{x}_i^k$ to $\mathbf{x}_i^{k+1}$. Hence, $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ only differ in the ith block $\mathbf{x}_i$.

To perform its update, agent i minimizes a strongly convex approximation of $\sum_{j\in\mathcal{N}_i}f_j$–the part of V that depends on $\mathbf{x}_i$–using possibly outdated information collected from the other agents $j\in\mathcal{N}_i$. To represent this situation, let $\mathbf{x}_j^{k-d_j^k(i,i)}$, $j\in\mathcal{N}_i\backslash\{i\}$, denote the estimate held by agent i of agent j's variable $\mathbf{x}_j^k$, where $d_j^k(i,i)$ is a nonnegative (integer) delay (the reason for the double index (i,i) in d_j^k

will become clear shortly). If $d_j^k(i,i) = 0$, agent i owns the most recent information on the variable of agent j, otherwise $\mathbf{x}_j^{k-d_j^k(i,i)}$ is some delayed version of $\mathbf{x}_j^k$. We define as $\mathbf{d}^k(i,i) \triangleq [d_l^k(i,i)]_{l\in\mathcal{N}_i}$ the *delay vector* collecting these delays; for ease of notation $\mathbf{d}^k(i,i)$ contains also the value $d_i^k(i,i)$, set to zero, as each agent has always access to current values of its own variables. Using the above notation and recalling that f_i depends on $\mathbf{x}_{\mathcal{N}_i}$, agent i at iteration k solves the following strongly convex subproblem:

$$\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i)) \triangleq \underset{\mathbf{x}_i\in\mathcal{X}_i}{\arg\min} \Bigg\{ \tilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right) + \sum_{j\in\mathcal{N}_i\setminus\{i\}} \left\langle \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}\right), \mathbf{x}_i - \mathbf{x}_i^k \right\rangle + g_i(\mathbf{x}_i) \Bigg\}, \tag{4.5}$$

where we defined $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)} \triangleq [\mathbf{x}_l^{k-d_l^k(i,j)}]_{l\in\mathcal{N}_j}$, $j \in \mathcal{N}_i$, and

$$\tilde{\mathbf{x}}^k(i) \triangleq \left[\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}\right]_{j\in\mathcal{N}_i}. \tag{4.6}$$

The term $\tilde{f}_i$ in (4.5) is a strongly convex surrogate that replaces the nonconvex function f_i known by agent i; an outdated value of the variables of the other agents is used, $\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}$, to build this function. Examples of valid surrogates are discussed in Sec. 4.3.1. The second term in (4.5) approximates $\sum_{j\in\mathcal{N}_i\setminus\{i\}} f_j$ by replacing each f_j by its first order approximation at (possibly outdated) $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$ (with $\nabla_{\mathbf{x}_i} f_j$ denoting the gradient of f_j with respect to the block $\mathbf{x}_i$), where $\mathbf{d}^k(i,j) \triangleq \left[d_l^k(i,j)\right]_{l\in\mathcal{N}_j}$, with $d_l^k(i,j) \geq 0$ representing the delay of the information that i knows about the gradient $\nabla_{\mathbf{x}_i} f_j$. This source of delay is due to two facts, namely: i) agents $j \in \mathcal{N}_i \setminus \{i\}$ may communicate to i its gradient $\nabla_{\mathbf{x}_i} f_j$ occasionally; and ii) $\nabla_{\mathbf{x}_i} f_j$ is generally computed at some outdated point, as agent j itself may not have access of the last information of the variables of the agents in $\mathcal{N}_j \setminus \{j\}$.

Once $\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i))$ has been computed, agent i sets

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \gamma\left(\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i)) - \mathbf{x}_i^k\right), \tag{4.7}$$

where $\gamma \in (0; 1]$ is suitably chosen stepsize (cf. Sec. 4.3.1).

The proposed distributed asynchronous algorithm, termed Distributed Asynchronous FLexible ParallEl Algorithm (DAsyFLEXA), is formally described in Algorithm 1. We set $\mathbf{x}_i^t = \mathbf{x}_i^0$, for all $t < 0$ and $i \in \mathcal{N}$, without loss of generality.

Algorithm 1 Distributed Asynchronous FLexible ParallEl Algorithm (DAsyFLEXA)

```
**Initialization:** $k$=0; $\mathbf{x}^0 \in \mathcal{X} \triangleq \prod_i \mathcal{X}_i$; $\mathbf{x}^t = \mathbf{x}^0$, $t<0$; $\gamma \in (0;1]$.
**while** a termination criterion is not met **do**
    (S.1): Pick agent $i^k$ and delays $\{\mathbf{d}^k(i^k, j)\}_{j \in \mathcal{N}_{i^k}}$;
    (S.2): Compute $\widehat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k(i^k))$ according to (4.5);
    (S.3): Update $\mathbf{x}_{i^k}^k$ according to (3.2);
    (S.4): Update the global iteration counter $k \leftarrow k+1$;
**end while**
```

We stress that agents need know neither the iteration counter k nor the vector of delays. No one "picks agent i^k and the delays $\{\mathbf{d}^k(i^k, j)\}_{j \in \mathcal{N}_{i^k}}$" in (S.1). This is just an *a posteriori* view of the algorithm dynamics: all agents asynchronously and continuously collect information from their neighbors and use it to update $\mathbf{x}_i$; when one agent has completed an updated the iteration index k is increased and i^k is defined.

4.3.1 Assumptions

Before studying convergence of Algorithm 1, we state the main assumptions on Problem (P) and the algorithmic choices.

On Problem (P)**.** Below, we will use the following conventions: When a function is said to be differentiable on a certain domain, it is understood that the function is differentiable on an open set containing the domain. We say that f_i is block-LC^1 on a set if it is continuously differentiable on that set and $\nabla_{\mathbf{x}_j} f_i$ are locally Lipschitz.

We say V is *coercive* on $\mathcal{X} = \prod_i \mathcal{X}_i$, if $\lim_{\|\mathbf{x}\| \to +\infty, \mathbf{x} \in \mathcal{X}} V(\mathbf{x}) = +\infty$; this is equivalent to requiring that all level sets of V in $\mathcal{X}$ are compact.

Assumption A (On Problem (P)**):**

(A1) Each set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is nonempty, closed, and convex;

(A2) At least one of the following conditions is satisfied

(a) $\mathcal{L}^0 \triangleq \{\mathbf{x} \in \mathcal{X} : V(\mathbf{x}) \leq V(\mathbf{x}^0)\}$ is compact and all f_i are block-LC^1 on $\mathcal{X}_{\mathcal{N}_i} \triangleq \prod_{j \in \mathcal{N}_i} \mathcal{X}_j$;

(b) All f_i are C^1 and their gradients $\nabla_{\mathbf{x}_j} f_i$, $j \in \mathcal{N}_i$, are *globally* Lipschitz on $\mathcal{X}_{\mathcal{N}_i}$;

(A3) Each $g_i : \mathcal{X}_i \to \mathbb{R}$ is convex;

(A4) Problem (P) has a solution;

(A5) The communication graph $\mathcal{G}$ is undirected and connected.

The above assumptions are standard, and similar to those already considered in Chapter 3 and Chapter 2. For instance, A2(a) holds if V is coercive on $\mathcal{X}$ and all f_i are block-LC^1 on $\mathcal{X}_{\mathcal{N}_i}$. Note that Ex. 2 satisfies A2(b); A2(a) is motivated by applications such as Ex. 1, which do not satisfy A2(b). A3 is a common assumption in the literature of parallel and distributed methods for the class of problems (P); two renowned examples are $g_i(\mathbf{x}_i) = \|\mathbf{x}_i\|_1$ and $g_i(\mathbf{x}_i) = \|\mathbf{x}_i\|_2$. Finally, A4 is satisfied if, for example, V is coercive or if $\mathcal{X}$ is bounded.

Remark 4.3.1 *Extensions to the case of directed graphs or the case where each agent updates multiple block-variables are easy, but not discussed here for the sake of simplicity.*

The aim of Algorithm 1 is to find *stationary solutions* of (P), i.e. points $\mathbf{x} \in \mathcal{X}$ such that for some $\boldsymbol{\xi} \in \partial G(\mathbf{x})$ it holds

$$\langle \nabla F(\mathbf{x}) + \boldsymbol{\xi}, \mathbf{y} - \mathbf{x} \rangle \geq 0, \qquad \forall \mathbf{y} \in \mathcal{X}. \tag{4.8}$$

Let $\mathcal{X}^\star \subseteq \mathbb{R}^n$ denote the set of such stationary solutions.

On an error bound condition. We prove *linear convergence* of Algorithm 1 under the Luo-Tseng error bound condition, which is stated next. Recall the definition of the proximal operator: given $\alpha > 0$, define $\texttt{prox}_{\alpha G} : \mathbb{R}^n \to \mathbb{R}^n$ as

$$\texttt{prox}_{\alpha G}(\mathbf{z}) \triangleq \underset{\mathbf{y}\in\mathcal{X}}{\arg\min} \left\{\alpha G(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2\right\}. \tag{4.9}$$

Furthermore, given $\mathbf{x} \in \mathbb{R}^n$, let

$$d(\mathbf{x}, \mathcal{X}^\star) \triangleq \min_{\mathbf{x}^\star\in\mathcal{X}^\star} \|\mathbf{x}-\mathbf{x}^\star\|_2, \qquad P_{\mathcal{X}^\star}(\mathbf{x}) \triangleq \underset{\mathbf{x}^\star\in\mathcal{X}^\star}{\text{argmin}}\|\mathbf{x}-\mathbf{x}^\star\|_2. \tag{4.10}$$

Note that $P_{\mathcal{X}^\star}(\mathbf{x}) \neq \emptyset$, as $\mathcal{X}^\star$ is closed.

Assumption B (Luo-Tseng error bound):

(B1) For any $\eta > \min_{\mathbf{x}\in\mathcal{X}} V(\mathbf{x})$, there exist $\epsilon, \kappa > 0$ such that:

$$\left.\begin{array}{l} V(\mathbf{x}) \leq \eta, \\ \|\mathbf{x} - \texttt{prox}_G\left(\nabla F(\mathbf{x}) - \mathbf{x}\right)\|_2 \leq \epsilon \end{array}\right\} \Rightarrow$$
$$d(\mathbf{x}, \mathcal{X}^\star) \leq \kappa \left\|\mathbf{x} - \texttt{prox}_G\left(\nabla F(\mathbf{x}) - \mathbf{x}\right)\right\|_2;$$

(B2) There exists $\delta > 0$ such that

$$\left.\begin{array}{l} \mathbf{x}, \mathbf{y} \in \mathcal{X}^\star, \\ V(\mathbf{x}) \neq V(\mathbf{y}) \end{array}\right\} \Rightarrow \|\mathbf{x}-\mathbf{y}\|_2 \geq \delta.$$

B1 is the Luo-Tseng error bound condition that we discussed in detail in Section 2.5. B1 is a local Lipschitzian error bound: the distance of $\mathbf{x}$ from $\mathcal{X}^\star$ is of the same order of the norm of the residual $\mathbf{x} - \texttt{prox}_G\left(\nabla F(\mathbf{x}) - \mathbf{x}\right)$ at $\mathbf{x}$. It is not difficult to check, that $\mathbf{x} \in \mathcal{X}^\star$ if and only if $\mathbf{x} - \texttt{prox}_G\left(\nabla F(\mathbf{x}) - \mathbf{x}\right) = 0$. Error bounds of this kind have been extensively studied in the literature; see [24, 25] and references therein. Examples of problems satisfying Assumption B include: LASSO, Group LASSO, Logistic Regression, unconstrained optimization with smooth nonconvex quadratic objective or $F(\mathbf{A}\mathbf{x})$, with F being strongly convex and $\mathbf{A}$ being an arbitrary operator (not necessarily full rank). B2 states that the level curves of V restricted to $\mathcal{X}^\star$ are "properly separated". B2 is trivially satisfied, e.g., if V is convex, if $\mathcal{X}$ is bounded, or if (P) has a finite number of stationary solutions. To read more details about B2 we again refer the reader to Section 2.5, where we discussed B2 about Assumption E.

We state next the main assumptions concerning the free parameters in the algorithm design.

On the subproblems (4.5)**.** The surrogate functions $\tilde{f}_i$ satisfy the following fairly standard conditions ($\nabla \tilde{f}_i$ denotes the partial gradient of $\tilde{f}_i$ w.r.t. the first argument).

Assumption C Each $\tilde{f}_i : \mathcal{X}_i \times \mathcal{X}_{\mathcal{N}_i} \to \mathbb{R}$ is chosen so that

(C1) $\tilde{f}_i(\cdot; \mathbf{y})$ is C^1 and τ-strongly convex on $\mathcal{X}_i$, for all $\mathbf{y} \in \mathcal{X}_{\mathcal{N}_i}$;

(C2) $\nabla \tilde{f}_i(\mathbf{y}_i; \mathbf{y}_{\mathcal{N}_i}) = \nabla_{\mathbf{y}_i} f_i(\mathbf{y}_{\mathcal{N}_i})$, for all $\mathbf{y} \in \mathcal{X}$;

(C3) $\nabla \tilde{f}_i(\mathbf{y}; \cdot)$ is L_i-Lipschitz continuous on $\mathcal{X}_{\mathcal{N}_i}$, for all $\mathbf{y} \in \mathcal{X}_i$.

A wide array of possible surrogate functions $\tilde{f}_i$ satisfying Assumption C can be found in Section 2.2.

On the asynchronous/communication model. The way agent i builds its own estimates $\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}$ and $\nabla_{\mathbf{x}_i} f_j(\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)})$, $j \in \mathcal{N}_i \backslash \{i\}$, depends on the particular asynchronous model and communication protocol under consideration and it is immaterial to the convergence of Algorithm 1. This is a major departure from previous works, such as [69,93,97,107], which instead enforce specific asynchrony and communication protocols. We only require the following mild conditions.

Assumption D (On the asynchronous model):

(D1) Every block variable of $\mathbf{x}$ is updated at most every $B \geq N$ iterations, i.e., $\cup_{t=k}^{k+B-1} i^t = \mathcal{N}$, for all k;

(D2) $\exists\, D \in [0, B]$, such that every component of $\mathbf{d}^k(i,j)$, $i \in \mathcal{N}$, $j \in \mathcal{N}_i$, is not greater than D, for any $k \geq 0$.[1]

[1]While `(S.2)` in Algorithm 1 is defined once $\mathbf{d}^k(i^k, j)$, $j \in \mathcal{N}_{i^k}$ is given, here we extend the definition of the delay vectors $\mathbf{d}^k(i,j)$ to all $i, j \in \mathcal{N}$, whose values are set to the delays of the information known by the associated agent on the variables and gradients of the others, at the time agent i^k performs its update. This will simplify the notation in some of the technical derivations.

Assumption D is satisfied virtually in all practical scenarios. D1 controls the frequency of the updates and is satisfied, for example, by any *essentially cyclic rules*, while D2 limits the age of the old information used in the updates. D1 is very easy to implement in practice: for instance it is automatically satisfied if each agent wakes up and performs an update whenever some internal clock ticks, without any need of centralized coordination. D2 imposes a mild condition on the communication protocol employed by the agents: information used in the agents' updates can not become infinitely old. While this implies agents communicate sufficiently often, it does not enforce any specific protocol on the activation/idle time/communication. For instance, differently from several asynchronous schemes in the literature [68, 69, 93, 94, 97, 98, 105], agents need not be always in "idle mode" to continuously receive messages from their neighbors. Notice that time varying delays satisfying D2 model also packet losses.

4.4 DAsyFLEXA: Convergence Results

In this Section we state the main convergence results for DAsyFLEXA. As already done in Chapter 3, we will consider both fixed and diminishing stepsizes for DAsyFLEXA. The former instrumental to obtain iteration complexity results of our scheme -in particular linear convergence rate on a specific class of problems-, the latter more efficient for practical implementations.

4.4.1 DAsyFLEXA with fixed stepsize γ

For nonconvex instances of (P), an appropriate measure of optimality is needed to evaluate the progress of the algorithm towards stationarity. In order to define such a measure, we first introduce the following quantities: for any $k \geq 0$,

$$\bar{\mathbf{x}}^k(i) \triangleq [\mathbf{x}^k_{\mathcal{N}_j}]_{j\in\mathcal{N}_i}, \quad \bar{\mathbf{x}}^k \triangleq [\bar{\mathbf{x}}^k(i)]_{i\in\mathcal{N}}, \tag{4.11}$$

where the former vector can be seen as the "synchronous" instance of $\tilde{\mathbf{x}}^k(i)$ wherein all the delays $\mathbf{d}^k(i,j)$ are $\mathbf{0}$. Convergence to stationarity is monitored by the following merit function:

$$M_V(\mathbf{x}^k) = \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2^2, \quad \text{with} \quad \widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) \triangleq [\widehat{\mathbf{x}}_l(\bar{\mathbf{x}}^k(l))]_{l\in\mathcal{N}}. \tag{4.12}$$

Note that M_V is a valid measure of stationarity, as M_V is continuous and $M_V(\mathbf{x}^k) = 0$ if and only if $\mathbf{x}^k \in \mathcal{X}^\star$.

The following theorem shows that, when the agents use a sufficiently small stepsize, the sequence of the iterates produced by DAsyFLEXA converges to a stationary solution of (P), driving $M_V(\mathbf{x}^k)$ to zero at a sublinear rate. In the theorem we use two positive constants, L and C_1, whose definition is given in the Appendix of this Chapter [cf. (4.34)]. Suffices to say, here, that L is essentially a Lipschitz constant for the partial gradients $\nabla_{\mathbf{x}_i} f_i$ whose definition varies according to whether A2(a) or A2(b) holds. In the latter case, L is simply the largest global Lipschitz constant for all $\nabla_{\mathbf{x}_j} f_i$'s. In the former case, the sequences $\{\mathbf{x}^k\}$, $\{\tilde{\mathbf{x}}^k\}$, $\tilde{\mathbf{x}}^k \triangleq [\tilde{\mathbf{x}}^k(i)]_{i\in\mathcal{N}}$, and $\{\widehat{\mathbf{x}}(\tilde{\mathbf{x}}^k)\}$ are proved to be bounded [cf. Theorem 4.4.1(c)]; L is then the Lipschitz constant of all $\nabla_{\mathbf{x}_j} f_i$'s over the compact set confining these sequences.

Theorem 4.4.1 *Given Problem* (P) *under Assumption A; let* $\{\mathbf{x}^k\}$ *be the sequence generated by DAsyFLEXA, under Assumptions C, and D. Choose* $\gamma \in (0,1]$ *such that* $\gamma < \frac{2\tau}{L(2+\rho^2 D^2)}$, *with* $\rho \triangleq \max_{i\in\mathcal{N}} |\mathcal{N}_i|$. *Then, there hold:*

(a) Any limit point of $\{\mathbf{x}^k\}$ *is a stationary solution of* (P);

(b) In at most T_ϵ *iterations, DAsyFLEXA drives the stationarity measure* $M_V(\mathbf{x}^k)$ *below* ϵ, $\epsilon > 0$, *where*

$$T_\epsilon = \left\lceil C_1 \left(V(\mathbf{x}^0) - \min_{\mathbf{x}\in\mathcal{X}} V(\mathbf{x}) \right) \cdot \frac{1}{\epsilon} \right\rceil,$$

where $C_1 > 0$ *is a constant defined in Appendix 4.7.3 [cf.* (4.34)*], which depends on* $\rho, L_i, i \in \mathcal{N}, L, \tau, \gamma, N, B,$ *and* D.

(c) If, in particular A2(a) is satisfied, $\{\mathbf{x}^k\}$ *is bounded.*

Proof See the Appendix of this Chapter. ■

Theorem 4.4.1 provides a unified set of convergence conditions for a gamut of algorithms, asynchronous models and communication protocols. Note that if the method is synchronous, i.e. $D = 0$, the condition on γ, reduces to the renowned condition used in the (proximal)-gradient algorithms. The term D^2 in the denominator of the upper-bound on γ should then be seen as the price to pay for asynchrony: the larger the possible delay D, the smaller γ should be to make the algorithm robust to asynchrony/delays.

Theorem 4.4.2 improves on the convergence of DAsyFLEXA, when V satisfies the error bound condition in Assumption B. Specifically, convergence of the whole sequence $\{\mathbf{x}^k\}$ to a stationary solution $\mathbf{x}^\star$ is established (in contrast with subsequence convergence in Theorem 4.4.1 (b)], and suitable subsequences that converge *linearly* are identified.

Theorem 4.4.2 *Given Problem* (P) *under Assumptions A and B, let* $\{\mathbf{x}^k\}$ *be the sequence generated by DAsyFLEXA, under Assumptions C and D. Suppose that* $\gamma/\tau > 0$ *is sufficiently small. Then,* $\{\mathbf{x}^{t+kB}\}$ *and* $\{V(\mathbf{x}^{t+kB})\}$, $t \in \{0, \ldots, B-1\}$, *converge at least R-linearly to* $\mathbf{x}^\star$ *and* $V^\star = V(\mathbf{x}^\star)$, *respectively, with* $\mathbf{x}^\star \in \mathcal{X}^\star$*:*

$$V(\mathbf{x}^{t+kB}) - V^\star = \mathcal{O}\left(\lambda^{t+kB}\right),$$

$$\|\mathbf{x}^{t+kB} - \mathbf{x}^\star\| = \mathcal{O}\left(\sqrt{\lambda^{t+kB}}\right),$$

where $\lambda \in (0;1)$ *is a constant defined in Appendix 4.7.4 [cf.* (4.44)*], which depends on* $\rho, L_i, i \in \mathcal{N}, L, \tau, \gamma, N, B,$ *and* D.

Proof See the Appendix of this Chapter. ■

In essence, the theorem proves a B-steps linear convergence rate. To the best of our knowledge, this is the first (linear) convergence rate result in the literature for an asynchronous algorithm in the setting considered in this paper.

4.4.2 DAsyFLEXA with diminishing stepsize γ^k

Note that since we strive for a fully distributed analysis it cannot be simply assumed that γ^k is a classical diminishing stepsize sequence. Indeed, this would entail that each agent should know the iteration counter k, which is physically impossible, unless we introduce strong coordination requirements among the agents. Instead we assume that each agent independently draws the stepsize from a *local* stepsize sequence $\{\alpha^t\}$ going through it as agent's updates progress. We only require that all the agents use the same sequence. Figure 4.1 shows an example of how the choice of stepsizes could evolve with three agents when $\{\alpha^t\} = \{1/2, 1/3, \ldots\}$ Here it is possible to see that the first update performed by the algorithm, for $k = 0$, is executed by agent 1 which uses the first element α^0 of its local sequence $\{\alpha^t\}$; after this, agent 2 updates two times its variables, using the first two elements α^0 and α^1 of its own stepsize sequence; for $k = 3$, agent 3 performs the update and, since this is the first update executed by agent 3, it uses the first element α^0 of its local sequence. So, the sequence of stepsizes used in the first four iterations of the algorithm is: $\{\alpha^0, \alpha^0, \alpha^1, \alpha^0\}$, which coincides with the first four elements $\{\gamma^0, \gamma^1, \gamma^2, \gamma^3\}$ of the global stepsize sequence $\{\gamma^k\}$ that appears in the updating rule (3.2).
The local common stepsize sequence must satisfy the following assumption.

Assumption E (On the stepsize). For any $t \geq 0$

(E1) $\alpha^t \downarrow 0$; and $\sum_{t=0}^{\infty} \alpha^t = +\infty$;

(E2) $\alpha^{t+1} \leq \alpha^t$ for sufficiently large t;

E1 is quite standard and satisfied by most of practical diminishing stepsize rules [9]. E2 is very mild and satisfied by classical choices of α^t in diminishing stepsize methods. For example, the following rule satisfies Assumption E and has been found very effective in our experiments [1,51]: $\alpha^{t+1} = \alpha^t \left(1 - \mu\alpha^t\right), \alpha^0 \in (0, 1]$ and $\mu \in (0, 1)$.

We can now provide the convergence results for DAsyFLEXA when equipped with diminshing stepsize.

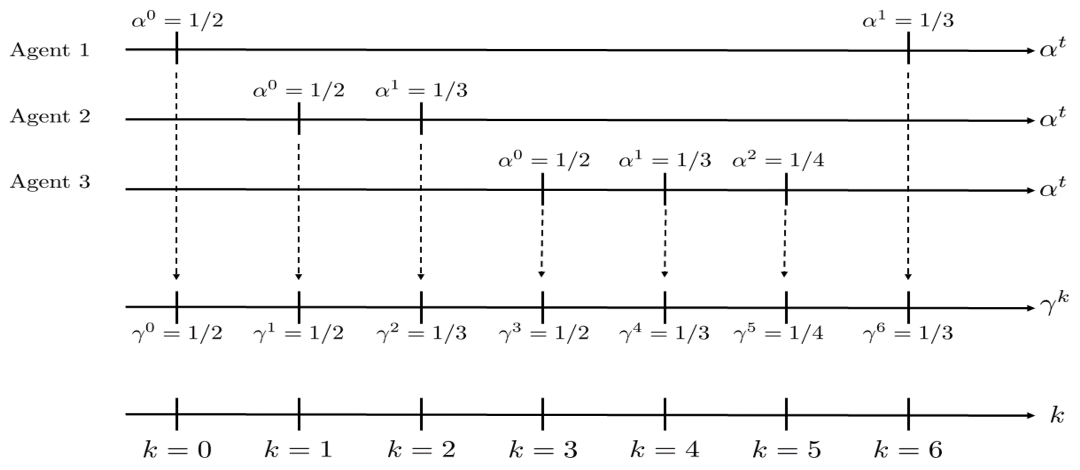

Figure 4.1. The global stepsize sequence $\{\gamma^k\}$ is a composition of elements coming from the local stepsize sequences $\{\alpha^t\}$.

Theorem 4.4.3 *Given Problem* (P) *under Assumptions A-C-D-E, let* $\{\mathbf{x}^k\}$ *be the sequence generated by DAsyFLEXA. For any* $\epsilon > 0$, *define* T_ϵ *to be the first iteration such that* $M_V(\mathbf{x}^k) \leq \epsilon$. *Then, the following hold:*

(a) Any limit point of $\{\mathbf{x}^k\}$ *is a stationary point of Problem* (P)*;*

(b) $T_\epsilon \leq \inf \left\{ k \geq 0 | \sum_{t=0}^{k} \tilde{\gamma}^t \geq \frac{c}{\epsilon} \right\}$, *where c is a positive constant, and* $\tilde{\gamma}^t \triangleq \min_{k \in [t-D; t+B-1]} \gamma^k$*;*

(c) If A2(a) is satisfied, the sequence $\{\mathbf{x}^k\}$ *is bounded.*

Proof See the Appendix of this Chapter. ■

In analogy to Theorem 4.4.1, Theorem 4.4.3 proves a sublinear convergence rate for the proposed algorithmic framework.

4.5 Nonconvex Constraints

In this section, we remove the assumption that all constraints are convex. For ease of presentation, the same notation already used in the previous sections will be kept for the quantities which are analogous to those already defined.

Specifically, we study the following more general nonconvex constrained optimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}_{\mathcal{N}_i}) + \sum_{i=1}^{N} g_i(\mathbf{x}_i), \\ & \left.\begin{array}{l} \mathbf{x}_i \in \mathcal{X}_i, \quad i = 1, \ldots, N \\ c_1(\mathbf{x}_1) \leq 0, \ldots, c_N(\mathbf{x}_N) \leq 0 \end{array}\right\} \triangleq \mathcal{K} \end{aligned} \tag{4.13}$$

where $c_i(\mathbf{x}_i) \leq 0$ are nonconvex private constraints, with $c_i : \mathcal{X}_i \to \mathbb{R}^{m_i}$; let also define $\mathcal{K}_i \triangleq \{\mathbf{x}_i \in \mathcal{X}_i : c_i(\mathbf{x}_i) \leq 0\}$. Note that $c_i(\mathbf{x}_i)$ is a vector function, whose individual component is denoted by $c_{i,j}$, with $j = 1, \ldots, m_i$. Problem (4.13) is motivated by several applications in signal processing, machine learning, and networking; see, [111] and references therein for some concrete examples.
We require on Problem (4.13) the following Assumption, similar to that already stated for Problem (P).

Assumption A' (On the problem model). Suppose that

(A1′) Each set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is nonempty, closed, and convex;

(A2′) At least one of the following conditions is satisfied

(a) $\mathcal{L}^0 \triangleq \{\mathbf{x} \in \mathcal{X} : V(\mathbf{x}) \leq V(\mathbf{x}^0)\}$ is compact and all f_i are block-LC^1 on $\mathcal{X}$

(b) All f_i are C^1 and their gradients $\nabla_{\mathbf{x}_j} f_i$ are *globally* Lipschitz on $\mathcal{X}$ for all $j \in \mathcal{N}_i$;

(A3′) Each $g_i : \mathcal{X}_i \to \mathbb{R}$ is convex;

(A4′) $\mathcal{K}$ is a compact set;

(A5′) $\mathcal{G}$ is undirected and connected.

(A6′) Each $c_{i,j} : \mathcal{X}_i \to \mathbb{R}$ is C^1;

(A7′) All feasible points of problem (P′) satisfy the MFCQ (see Definition 3.5.1).

A1′-A2′-A3′-A5′ are duplicate of A1-A2-A3-A5 and listed again for ease of reference. A4′ guarantees the existence of a solution and it is now stronger than A4, and made here for the sake of simplicity (one could relax it with A4, having a more involved convergence proof). A6′ is a standard differentiability requirement on the nonconvex constraints $c_{i,j}$. Finally, we remark that one could relax A7′ and require regularity only at specific points, but at the cost of more convoluted statements; we leave this task to the reader.

The algorithm proposed for solving Problem (4.13) is a generalization of DAsyFLEXA: SCA techniques are leveraged again and at each iteration, a suitably chosen "convex approximation" of the original problem is solved. The difference with the method introduced in Section 4.3 is that now also the nonconvex constraints will be approximated. More specifically, the subproblem solved to update block i now becomes: given $\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}_i^k}$, and $s_{i,j}^k$ for $j \in \mathcal{N}_i \backslash \{i\}$

$$\begin{aligned}&\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i))\\ &\triangleq \underset{\mathbf{x}_i \in \mathcal{K}_i(\mathbf{x}_i^k)}{\arg\min} \left\{ \tilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right) + \sum_{j \in \mathcal{N}_i \backslash \{i\}} \langle \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_i^{k-\mathbf{d}^k(i,j)}} \right), \mathbf{x}_i - \mathbf{x}_i^k \rangle + g_i(\mathbf{x}_i) \right\}.\end{aligned} \tag{4.14}$$

where $\tilde{f}_i : \mathcal{X}_i \times \prod_{j \in \mathcal{N}_i} \mathcal{K}_j \to \mathbb{R}$ is a strongly convex surrogate of f_i, and $\mathcal{K}_i(\mathbf{x}_i^k)$ is a convex approximation of $\mathcal{K}_i$ at $\mathbf{x}_i^k$, defined as

$$\mathcal{K}_i(\mathbf{x}_i^k) \triangleq \{\mathbf{x}_i \in \mathcal{X}_i : \tilde{c}_{i,j}(\mathbf{x}_i; \mathbf{x}_i^k) \leq 0,\ j = 1, \ldots, m_i\},$$

and $\tilde{c}_{i,j} : \mathcal{X}_i \times \mathcal{K}_i \to \mathbb{R}$ is a suitably chosen surrogate of $c_{i,j}$. Note that $\mathcal{K}_i$ does not depend on a delayed version of $\mathbf{x}_i^k$, because agent i always owns the most up-to-date version of its block-variable, i.e. $d_i^k(i,i) = 0$, for all i, k.

The assumptions required on the surrogate functions $\tilde{f}_i$ are the same already stated in Assumption 3 in Section 4.3.

The functions $\tilde{c}_{i,j}$ are chosen to satisfy the following assumptions ($\nabla\tilde{c}_{i,j}$ below denotes the partial gradient of $\tilde{c}_{i,j}$ with respect to the first argument).

Assumption F (On the surrogate functions $\tilde{c}_{i,j}$). Each $\tilde{c}_{i,j} : \mathcal{X}_i \times \mathcal{K}_i \to \mathbb{R}$ is chosen so that:

(F1) Each $\tilde{c}_{i,j}(\cdot;\mathbf{y})$ is C^1 and convex on $\mathcal{X}_i$, for all $\mathbf{y} \in \mathcal{K}_i$;

(F2) $\tilde{c}_{i,j}(\mathbf{y};\mathbf{y}) = c_i(\mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}_i$;

(F3) $c_{i,j}(\mathbf{z}) \leq \tilde{c}_{i,j}(\mathbf{z};\mathbf{y})$ for all $\mathbf{z} \in \mathcal{X}_i$ and $\mathbf{y} \in \mathcal{K}_i$;

(F4) $\tilde{c}_{i,j}(\cdot;\cdot)$ is continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

(F5) $\nabla_{\mathbf{y}_i} c_{i,j}(\mathbf{y}) = \nabla\tilde{c}_{i,j}(\mathbf{y};\mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}_i$;

(F6) $\nabla\tilde{c}_{i,j}(\cdot;\cdot)$ is continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

(F7) Each $\tilde{c}_{i,j}(\cdot;\cdot)$ is Lipschitz continuous on $\mathcal{X}_i \times \mathcal{K}_i$.

Roughly speaking, Assumption F requires $\tilde{c}_{i,j}$ to be an upper convex approximation of $c_{i,j}$ having the same gradient of $c_{i,j}$ in the point in which the approximation is made. We want to remark that it is quite easy to find suitable convex approximations $\tilde{c}_{i,j}$ in most practical problems; we refer the reader to Chapter 3 and [112] for several examples and applications.

DAsyFLEXA-NCC: The algorithm we propose to solve Problem (4.13) is DAsyFLEXA wherein the best-response map in `(S.2)` is replaced by (4.14); we call this new algorithmic framework DAsyFLEXA-NCC (where NCC stands for Non Convex Constraints). Convergence results analogous to those obtained for DAsyFLEXA are stated in the following theorems.

Theorem 4.5.1 *Given Problem* (4.13) *under Assumption A'-C-D-F, let* $\{\mathbf{x}^k\}$ *be the sequence generated by DAsyFLEXA-NCC. Suppose that the stepsize* γ^k *is fixed* $\gamma^k = \gamma \in (0;1]$ *and chosen as in Theorem 4.4.1. Then, the following hold:*

(a) The sequence of the iterates is feasible: $\mathbf{x}^k \in \mathcal{K}_1(\mathbf{x}_1^k) \times \ldots \times \mathcal{K}_N(\mathbf{x}_N^k) \subseteq \mathcal{K}$ *for all* $k \geq 0$*;*

(b) The same results of Theorem 4.4.2 hold;

Proof Proof of part (a) follows from Lemma 9 in [112]. We omit the proof of part (b) because is a straightforward generalization of previous proofs presented in this Dissertation. Specifically, to prove statement (b) it is sufficient to follow the steps of the proof of Theorem 3.4.1, together with the results derived in the proof of Theorem 4.4.1. ∎

In this Section we report some numerical results on the two problems described in Section 4.2. We compare DAsyFLEXA with the PrimalDual asynchronous algorithm in [108], which seems to be the closest distributed asynchronous scheme to DAsyFLEXA. However, there are some important differences between these two algorithms. First, the PrimalDual algorithm [108] does not exploit the sparsity pattern of the objective function V; every agent instead controls and updates a local copy of the *entire* vector $\mathbf{x}$, which requires employing a consensus mechanism to enforce an agreement on such local copies. This leads to an unnecessary communication overhead among the agents. Second, no explicit estimate of the gradients of the other agents is employed; the lack of this knowledge is overcome by introducing additional communication variables, which lead to contribute to increase the communication cost. Third, the PrimalDual algorithm does not have convergence guarantees in the nonconvex case. We tested the algorithms on the Archimedes1 cluster computer at Purdue University, equipped with two 22-cores Intel E5-2699Av4 processors (44 cores in total) and 512GB of RAM.

4.5.1 Distributed LASSO

We simulate the (convex) LASSO problem as stated in (4.4). The underlying sparse linear model is generated as follows: $\mathbf{b} = \mathbf{A}\mathbf{x}^\star + \mathbf{e}$, where $\mathbf{A}$ has 15000 rows and 30000 columns. $\mathbf{A}$, $\mathbf{x}^\star$ and $\mathbf{e}$ have i.i.d. elements drawn from a Gaussian $\mathcal{N}(0,\sigma^2)$ distribution, with $\sigma = 1$ for $\mathbf{A}$ and $\mathbf{x}^\star$, and $\sigma = 0.1$ for the noise vector $\mathbf{e}$. Entries of $\mathbf{A}$ are normalized by the spectral norm of $\mathbf{A}$. To impose sparsity on $\mathbf{x}^\star$ and $\mathbf{A}$, we randomly set to zero 95% of their components. Finally, in (4.4), we set $\lambda = 1$.

We consider a fixed, undirected network composed of 50 agents; $\mathbf{x} \in \mathbb{R}^{30000}$ is partitioned in 50 block-variables $\mathbf{x}_i \in \mathbb{R}^{600}$, $i \in \{1,\ldots,50\}$, each of them controlled by one agent. We define the local functions f_i and g_i as described in Sec. 4.2 (cf. Ex. #2); each $\mathbf{A}_i$ (resp. $\mathbf{b}_i$) is all zeros but its ith row (resp. component), which coincides with that of $\mathbf{A}$ (resp. $\mathbf{b}$). This induces the following communication pattern among the agents: each agent i is connected only to the agents js owning the $\mathbf{x}_j$s corresponding to the nonzero column-entries of $\mathbf{A}_i$.

We simulate DAsyFLEXA using the following surrogate functions (satisfying Assumption C):

$$\begin{aligned}&\tilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right) \\ &= \left\langle \nabla_{\mathbf{x}_i} f_i\left(\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right), \mathbf{x}_i - \mathbf{x}_i^k \right\rangle + \frac{\tau_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_2^2,\end{aligned} \tag{4.15}$$

where $\tau_i > 0$ is a tunable parameter, which is updated following the same heuristic used in [47]. The stepsize γ is set to 0.9. Note that, using (4.15), problem (4.5) has a closed-form solution via the renowned soft-thresholding operator.

We simulate the following asynchronous model. Each agent is activated periodically, every time a local clock triggers. The agents' local clocks have the same frequency but different phase shift, which are selected uniformly at random within $[5, 50]$. Based upon its activation, each agent: i) performs its update and then broadcasts its gradient vector $\nabla_{\mathbf{x}_i} f_i$ together with its own block-variable $\mathbf{x}_i$ to the agents in $\mathcal{N}_i \backslash \{i\}$; and ii) modifies the phase shift of its local clock by selecting uniformly at random a new value in $[5, 50]$.

The step sizes of the PrimalDual scheme [108] are tuned by hand in order to obtain the best performances; specifically we set $\alpha = 0.9$, and $\eta_i = 1.5$ for $i = 1,\ldots,50$

(see [108] for details on these parameters). Both algorithms are initialized from the same randomly chosen point, drawn from $\mathcal{N}(0, 1)$.

Figure 4.2 plots relative error $(V(\mathbf{x}^k) - V^\star)/V^\star$ of the two algorithms versus the number of iterations. Figure 4.3 shows the same function versus the number of message exchanges, where each scalar variable sent from an agent to one of its neighbor is counted as one message exchanged. All the curves are averaged over 10 independent realizations.

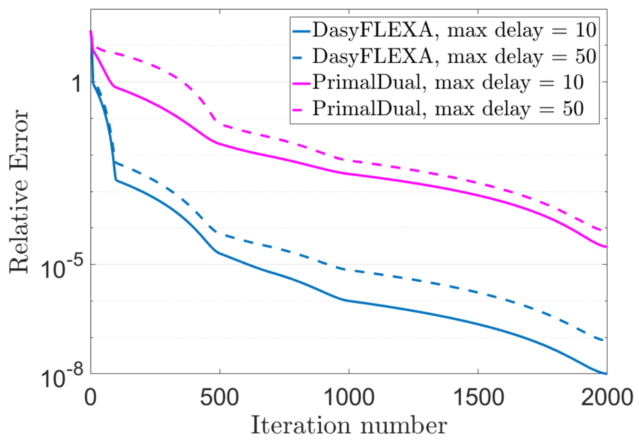

Figure 4.2. LASSO problem: Relative error vs. # of iterations.

DAsyFLEXA outperforms the PrimalDual scheme [108]. Also, as anticipated, PrimalDual requires much more communications than DAsyFLEXA.

4.5.2 Distributed Matrix Completion

In this section we consider the (nonconvex) Distributed Matrix Completion problem (4.1). We generate a 2500×2500 matrix $\mathbf{Z}$ with samples drawn from $\mathcal{N}(0, 1)$; and we set $\lambda = \xi = 1$ and $r = 4$. We consider a fixed undirected network composed of 50 agents, and we partitioned the columns of $\mathbf{X}$ uniformly among 25 agents, and those of $\mathbf{Y}$ uniformly among the other 25 agents. We sampled uniformly at random

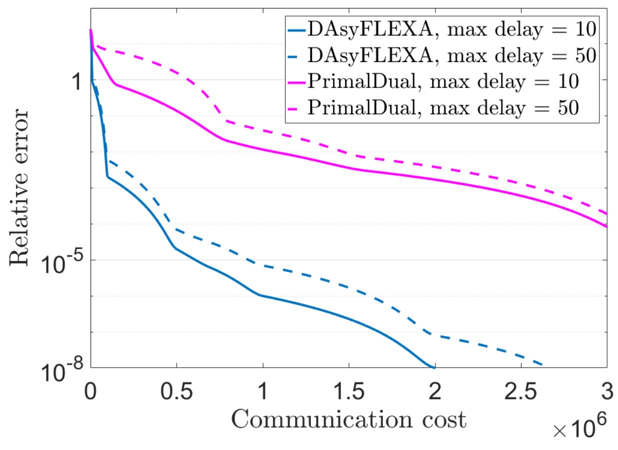

Figure 4.3. LASSO problem: Relative error vs. # of message exchanges.

10% of the entries of $\mathbf{Z}$, and distributed these samples z_{mn} to the agents owing the corresponding column $\mathbf{x}_m$ of $\mathbf{X}$ or $\mathbf{y}_n$ of $\mathbf{Y}$, choosing randomly between the two.

We applied the following instance of DAsyFLEXA to (4.1). Consider one of the agents that optimizes some columns of $\mathbf{X}$, say agent i. Since each f_i is biconvex in $\mathbf{X}$ and $\mathbf{Y}$, the following surrogate function satisfies Assumption C:

$$\begin{aligned}&\tilde{f}_i\left(\{\mathbf{x}_m\}_{m\in X_i};(\mathbf{X},\mathbf{Y})_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right) \\ &=\frac{1}{2}\sum_{(m,n)\in\Omega_i}\left(\mathbf{x}_m^T\mathbf{y}_n^{k-d^k_{j(n)}(i,i)}-z_{mn}\right)^2+\frac{\tau_i}{2}\sum_{(m,n)\in\Omega_i}\|\mathbf{x}_m-\mathbf{x}_m^k\|_2^2;\end{aligned} \tag{4.16}$$

where $j(n)$ is the index $j\in\mathcal{N}_i$ of the agent that controls $\mathbf{y}_n$, and $\tau_i>0$ is updated following the same heuristic used in [47] (the surrogate function for the agents that update columns of $\mathbf{Y}$ is the same as (4.16), with the obvious change of notation). Note that (4.16) preserves the block-wise convexity present in the original function f_i, which contrasts with the common approach in the literature based on the linearization of f_i. We remark that, problem (4.5) with the surrogate (4.16) has a closed-form solution (up to a matrix inversion).

The rest of the setup is the same as that described for the LASSO problem. Figure 4.4 and Figure 4.5 plot $\|\widehat{\mathbf{x}}^k-\mathbf{x}^k\|_\infty$ (a valid measure of stationarity), with $\widehat{\mathbf{x}}_i$ defined as in (4.5), obtained by DAsyFLEXA and the PrimalDual algorithm [108] versus the number of iterations and message exchanges, respectively. On our tests, we observed that all the algorithms converged to the same stationary solution. The results confirm the behavior observed in the previous section for convex problems: DAsyFLEXA has better performances than PrimalDual, and the difference is mostly significant is terms of communication cost.

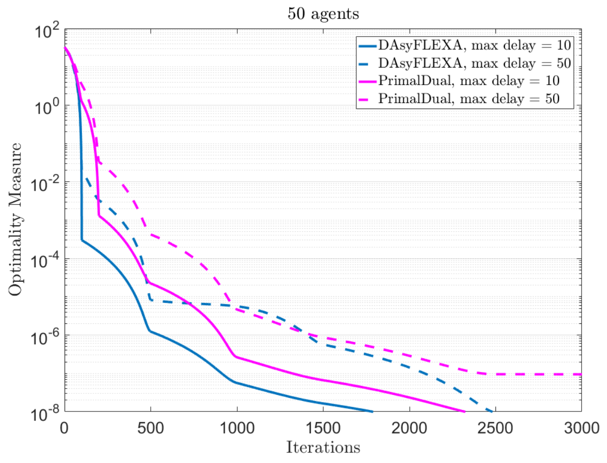

Figure 4.4. Matrix completion: stationarity distance vs. # of iterations.

4.6 Conclusions

We proposed a novel general asynchronous algorithmic framework for the constrained minimization of a nonconvex and nonsmooth sum-cost function in multi-agent graph-partitioned networks. Sublinear convergence rate to stationary solutions is proved. Linear convergence rate is established under the additional Luo-Tseng error bound. Simulation results on distributed LASSO and matrix completion problems show that the proposed algorithm compares favorably on state-of-the-art asynchronous schemes.

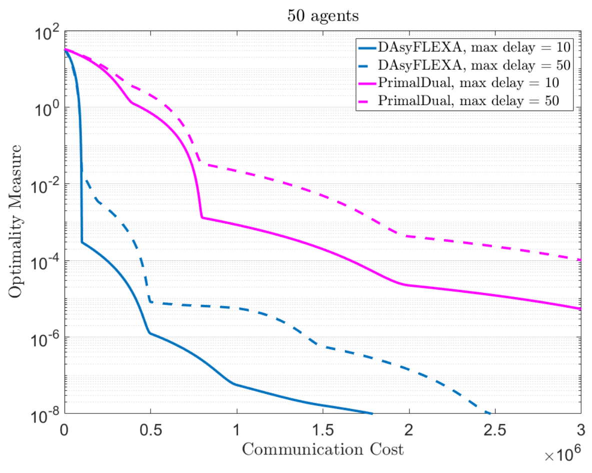

Figure 4.5. Matrix completion: stationarity distance vs. # of message exchanges.

4.7 Appendix: Proofs of Theorems

In this section we prove Theorems 4.4.1 and 4.4.2. We begin introducing some notation and definitions instrumental for the proofs (cf. Section 4.7.1), followed by some preliminary results (cf. Section 4.7.2). Theorems 4.4.1 and 4.4.2 are proved in Section 4.7.3 and Section 4.7.4, respectively. Some miscellanea results supporting the main proofs are given in Section 4.7.5.

4.7.1 Notation

Vectors $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$ have different length. It is convenient to replace them with equal-length vectors retaining of course the same information. This is done introducing the following (column) vectors $\mathbf{x}^k(i,j) \triangleq (\mathbf{x}_l^k(i,j))_{l=1}^N \in \mathcal{X}$, defined as:

$$[\mathbf{x}_l^k(i,j)]_{l\in\mathcal{N}_j} \triangleq \mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}, \tag{4.17a}$$

$$\mathbf{x}_l^k(i,j) = \mathbf{x}_l^k, \quad l \notin \mathcal{N}_j. \tag{4.17b}$$

In words, the blocks of $\mathbf{x}^k(i,j)$ indexed by $\mathcal{N}_j$ coincide with $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$ whereas the other block-components, irrelevant to the proofs, are conveniently set to their most up-to-date values. We will use the shorthand $\mathbf{x}_{\mathcal{N}_j}^k(i,j) \triangleq [\mathbf{x}_l^k(i,j)]_{l\in\mathcal{N}_j}$.

Since at each iteration $k \geq 0$ only one block of $\mathbf{x}^k$ is updated, and because of Assumption D2, it is not difficult to check that the delayed vector $\mathbf{x}^k(i,j)$ can be written as

$$\mathbf{x}^k(i,j) = \mathbf{x}^k + \sum_{l\in\mathcal{K}^k(i,j)} (\mathbf{x}^l - \mathbf{x}^{l+1}), \tag{4.18}$$

where $\mathcal{K}^k(i,j)$ is a subset of $\{k-D,\ldots,k-1\}$ whose elements depend on which block variables have been updated in the window $[\max\{0,k-D\},\max\{0,k-1\}]$. Recall that it is assumed $\mathbf{x}^t = \mathbf{x}^0$, for $t<0$.

Finally, let us introduce the following shorthands for the best-response map $\widehat{\mathbf{x}}_i(\cdot)$ defined in (4.5) [recall also (4.6)]:

$$\widehat{\mathbf{x}}^k \triangleq \left[\widehat{\mathbf{x}}_i^k\right]_{i\in\mathcal{N}}, \quad \widehat{\mathbf{x}}_i^k \triangleq \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i)); \tag{4.19a}$$

$$\Delta\widehat{\mathbf{x}}^k \triangleq \left[\Delta\widehat{\mathbf{x}}_i^k\right]_{i\in\mathcal{N}}, \quad \Delta\widehat{\mathbf{x}}_i^k \triangleq \widehat{\mathbf{x}}_i^k - \mathbf{x}_i^k. \tag{4.19b}$$

On the constant L. The proofs rely on some Lipschitz properties of $\nabla_{\mathbf{x}_j} f_i$'s. To provide a unified proof under either A2(a) or A2(b), we introduce a constant $L>0$ whose value depends on whether A2(a) or A2(b) hold. Specifically:

• **A2(a) holds**: the gradients $\nabla_{\mathbf{x}_j} f_i$'s are not globally Lipschitz on the sets $\mathcal{X}_{\mathcal{N}_i}$'s; our approach to study convergence is to ensure that they are Lipschitz continuous on suitably defined sets containing the sequences generated by Algorithm 1. We define these sets as follows. Define first the set $\texttt{Cube} \triangleq \{\mathbf{w}\in\mathcal{X} : \|\mathbf{w}\|_\infty \leq U\}$, where U is positive constant that ensures $\mathcal{L}^0 \subseteq \texttt{Cube}$ (note that $U < +\infty$ because $\mathcal{L}^0$ is bounded). Then, we define a proper widening $\bar{\mathcal{L}}^0$ of $\mathcal{L}^0$: $\bar{\mathcal{L}}^0 \triangleq (\mathcal{L}^0 + \psi\mathcal{B}) \cap \mathcal{X}$, where $\mathcal{B}$ is the unitary ball centered in the origin, and $\psi>0$ is a finite positive constant defined as

$$\psi \triangleq \max_{i\in\mathcal{N}} \max_{\substack{\tilde{\mathbf{w}}(i)\triangleq[\mathbf{w}_{\mathcal{N}_j}(j)]_{j\in\mathcal{N}_i} \\ \mathbf{w}(j)\in\texttt{Cube}}} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{w}}(i)) - \mathbf{w}_i(i)\|_2. \tag{4.20}$$

Note that $\bar{\mathcal{L}}^0$ is compact, because $\mathcal{L}^0$ is bounded and $\psi < +\infty$ [given that $\texttt{Cube}$ is bounded and $\widehat{\mathbf{x}}(\cdot)$ is continuous, due to (4.5), A2, A3, and C3]. Consider now any vector $\mathbf{x} \in \bar{\mathcal{L}}^0$. A2(a) and compactness of $\bar{\mathcal{L}}^0$ imply that the gradients $\nabla_{\mathbf{x}_j} f_i$'s are globally Lipschitz over the sets containing the subvectors $\mathbf{x}_{\mathcal{N}_i}$'s, with L being the maximum value of the Lipschitz constant of all the gradients over these sets.

• **A2(b) holds:** In this case, L is simply the global Lipschitz constant $\nabla_{\mathbf{x}_i} f_i$ over the whole space.

Remark 4.7.1 *To make sense of the complicated definition of L under A2(a), we anticipate how this constant will be used. Our proof leverages the decent lemma to majorize $V(\mathbf{x}^{k+1})$. To do so, each $\nabla_{\mathbf{x}_j} f_i$ needs to be globally Lipschitz on a convex set containing $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$. This is what the convex set $\bar{\mathcal{L}}^0$ is meant for: $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ belong to $\bar{\mathcal{L}}^0$ and thus $\nabla_{\mathbf{x}_j} f_i$ is L-Lipschitz continuous.*

4.7.2 Preliminaries

We summarize next some properties of the map $\widehat{\mathbf{x}}_i^k$ in (4.5).

Proposition 4.7.1 *Given Problem* (P) *under Assumption A, let $\{\mathbf{x}^k\}$ be the sequence generated by DAsyFLEXA, under Assumptions B and C. Suppose also that $\mathbf{x}^k \in \bar{\mathcal{L}}^0$ for all k. There hold:*

(a) [Optimality] For any $i \in \mathcal{N}$ and $k \geq 0$,

$$\sum_{j\in\mathcal{N}_i} \left\langle \nabla_{\mathbf{x}_i} f_j(\mathbf{x}^k_{\mathcal{N}_j}(i,j)), \Delta\widehat{\mathbf{x}}_i^k \right\rangle + g_i(\widehat{\mathbf{x}}_i^k) - g_i(\mathbf{x}_i^k) \leq -\tau \|\Delta\widehat{\mathbf{x}}_i^k\|_2^2; \tag{4.21}$$

(b) [Lipschitz continuity] For any $i \in \mathcal{N}$ and $k, h \geq 0$,

$$\|\widehat{\mathbf{x}}_i^k - \widehat{\mathbf{x}}_i^h\|_2 \leq \frac{L_m}{\tau}\|\mathbf{x}^k(i,i) - \mathbf{x}^h(i,i)\|_2 + \frac{L}{\tau}\sum_{j\in\mathcal{N}_i\setminus\{i\}} \|\mathbf{x}^k(i,j) - \mathbf{x}^k(i,j)\|_2, \tag{4.22}$$

where $L_m \triangleq \max_{i\in\mathcal{N}} L_i$;

(c) [Fixed-points] $\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) = \mathbf{x}^k$ if and only if $\mathbf{x}^k$ is a stationary solutions of Problem (P) *(recall the definition* (4.11) *of $\bar{\mathbf{x}}^k$);*

(d) [Error bound] For any $k \geq 0$,

$$\|\mathbf{x}^k - \boldsymbol{prox}_G\left(\mathbf{x}^k - \nabla F(\mathbf{x}^k)\right)\|_2 \leq (1 + L + NL_m)\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2. \tag{4.23}$$

Proof We prove only (d); the proof of (a)-(c) follows similar steps of that in [1, Proposition 8], and thus is omitted. Invoking the optimality of $\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k)$, we have

$$\begin{aligned}&\left\langle \nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k(i)); \mathbf{x}^k_{\mathcal{N}_i}\right) + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}^k_{\mathcal{N}_j}\right), \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k(i)) - \mathbf{z}_i\right\rangle \\ &+ g_i(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k(i))) - g_i(\mathbf{z}_i) \leq 0,\end{aligned}$$

for all $\mathbf{z} \in \mathcal{X}$ and $i \in \mathcal{N}$. Setting $\check{\mathbf{x}}^k \triangleq \mathtt{prox}_G\left(\mathbf{x}^k - \nabla F(\mathbf{x}^k)\right)$, and invoking the variational characterization of the proximal operator, we have

$$\left\langle \nabla_{\mathbf{x}} F(\mathbf{x}^k) + \check{\mathbf{x}}^k - \mathbf{x}^k, \check{\mathbf{x}}^k - \mathbf{w}\right\rangle + G(\check{\mathbf{x}}^k) - G(\mathbf{w}) \leq 0,$$

for all $\mathbf{w} \in \mathcal{X}$. Summing the two inequalities above, with $\mathbf{z} = \check{\mathbf{x}}^k$, $\mathbf{w} = \widehat{\mathbf{x}}(\bar{\mathbf{x}}^k)$, and using C1 and C2, yields

$$\begin{aligned}&\tau\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2^2 + \|\check{\mathbf{x}}^k - \mathbf{x}^k\|_2^2 \leq \|\check{\mathbf{x}}^k - \mathbf{x}^k\|_2\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k(i)) - \mathbf{x}^k\|_2 \\ &+ \sum_{i=1}^N \left\langle \nabla\tilde{f}_i(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k(i)); \mathbf{x}^k_{\mathcal{N}_i}) - \nabla\tilde{f}_i(\mathbf{x}^k_i; \mathbf{x}^k_{\mathcal{N}_i}), \check{\mathbf{x}}^k_i - \mathbf{x}^k_i\right\rangle \\ &\overset{A2, C2-C3}{\leq} \|\mathbf{x}^k - \check{\mathbf{x}}^k\|_2\left((1 + L + NL_m)\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2\right).\end{aligned}$$

■

4.7.3 Proof of Theorem 4.4.1

The proof is organized in the following steps:

Step 1–Lyapunov function & its descent: We define an appropriate Lyapunov function $\tilde{V}$ and prove that it is monotonically nonincreasing along the iterations. This also proves Theorem 4.4.1(c);

Step 2–Vanishing x-stationarity: Building on the descent properties of the Lyapunov function, we prove $\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2 = 0$ [Theorem 4.4.1(a)];

Step 3–Convergence rate: We prove the sublinear convergence rate of $\{M_V(\mathbf{x}^k)\}$ as stated in Theorem 4.4.1(c).

The above steps are proved under Assumptions A, C, and D.

Step 1–Lyapunov function & its descent

Introduce the following Lyapunov-like function:

$$\tilde{V}(\mathbf{x}^k, \ldots, \mathbf{x}^{k-D}) \triangleq V(\mathbf{x}^k) + \frac{DL\rho^2}{2}\left(\sum_{l=k-D}^{k-1} (l-(k-1)+D) \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right), \quad (4.24)$$

where L is defined in Sec.4.7.1. Note that

$$\tilde{V}^\star \triangleq \min_{[\mathbf{y}^i \in \mathcal{X}]_{i=1}^{D+1}} \tilde{V}(\mathbf{y}^1, \ldots, \mathbf{y}^{D+1}) = \min_{\mathbf{x}\in\mathcal{X}} V(\mathbf{x}).$$

The following lemma establishes the descent properties of $\tilde{V}$ and also proves Theorem 4.4.1(c).

Lemma 4.7.2 *Given $\tilde{V}$ defined in* (4.24)*, the following hold:*
(a) For any $k \geq 0$:

$$\tilde{V}(\mathbf{x}^{k+1} \ldots, \mathbf{x}^{k+1-D}) \leq \tilde{V}(\mathbf{x}^k, \ldots, \mathbf{x}^{k-D}) - \gamma\left(\tau - \gamma\frac{L\,(2+D^2\rho^2)}{2}\right) \|\Delta\widehat{\mathbf{x}}_{i^k}^k\|_2^2. \quad (4.25)$$

(b) If, in particular, A2(a) is satisfied: $\mathbf{x}^k \in \mathcal{L}^0$, for all $k \geq 0$.

Proof We prove the two statements by induction. For $k = 0$,

$$\begin{aligned}
V(\mathbf{x}^1) &= \sum_{i=1}^{N} f_i(\mathbf{x}^1_{\mathcal{N}_i}) + g_{i^0}(\mathbf{x}^1_{i^0}) + \sum_{i \neq i^0} g_i(\mathbf{x}^1_i) \\
&\overset{(4.7)}{=} \sum_{i=1}^{N} f_i(\mathbf{x}^1_{\mathcal{N}_i}) + g_{i^0}(\mathbf{x}^1_{i^0}) + \sum_{i \neq i^0} g_i(\mathbf{x}^0_i) \\
&\overset{(a)}{\leq} \sum_{i=1}^{N} f_i(\mathbf{x}^0_{\mathcal{N}_i}) + \gamma \sum_{j \in \mathcal{N}_{i^0}} \Big\langle \nabla_{\mathbf{x}_{i^0}} f_j(\mathbf{x}^0_{\mathcal{N}_j}) \\
&\quad + \nabla_{\mathbf{x}_{i^0}} f_j\left(\mathbf{x}^0_{\mathcal{N}_j}(i^0, j)\right) - \nabla_{\mathbf{x}_{i^0}} f_j\left(\mathbf{x}^0_{\mathcal{N}_j}(i^0, j)\right), \Delta\widehat{\mathbf{x}}^0_{i^0} \Big\rangle \\
&\quad + \frac{\gamma^2 L}{2} \|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2 + g_{i^0}(\mathbf{x}^1_{i^0}) + \sum_{i \neq i^0} g_i(\mathbf{x}^0_i) \\
&\overset{A3}{\leq} \sum_{i=1}^{N} f_i(\mathbf{x}^0_{\mathcal{N}_i}) + \gamma \Big\langle \sum_{j \in \mathcal{N}_{i^0}} \nabla_{\mathbf{x}_{i^0}} f_j\left(\mathbf{x}^0_{\mathcal{N}_j}(i^0, j)\right), \Delta\widehat{\mathbf{x}}^0_{i^0} \Big\rangle \\
&\quad + \gamma \Big\langle \sum_{j \in \mathcal{N}_{i^0}} \left(\nabla_{\mathbf{x}_{i^0}} f_j(\mathbf{x}^0_{\mathcal{N}_j}) - \nabla_{\mathbf{x}_{i^0}} f_j\left(\mathbf{x}^0_{\mathcal{N}_j}(i^0, j)\right)\right), \Delta\widehat{\mathbf{x}}^0_{i^0} \Big\rangle + \frac{\gamma^2 L}{2} \|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2 \\
&\quad + \sum_{i=1}^{N} g_i(\mathbf{x}^0_i) + \gamma g_{i^0}(\widehat{\mathbf{x}}^0_{i^0}) - \gamma g_{i^0}(\mathbf{x}^0_{i^0}) \overset{(4.21), A2}{\leq} V(\mathbf{x}^0) \\
&\quad - \gamma\left(\tau - \frac{\gamma L}{2}\right) \|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2 + \gamma L \|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2 \sum_{j \in \mathcal{N}_{i^0}} \|\mathbf{x}^0 - \mathbf{x}^0(i^0, j)\|_2 \\
&\overset{(b)}{\leq} V(\mathbf{x}^0) - \gamma(\tau - \gamma L) \|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2 + \frac{L\rho}{2} \sum_{j \in \mathcal{N}_{i^0}} \underbrace{\|\mathbf{x}^0 - \mathbf{x}^0(i^0, j)\|_2^2}_{\texttt{term I}},
\end{aligned} \tag{4.26}$$

where (a) follows from the descent lemma and the definition of L; and in (b) we used Young's inequality. Note that in (a) we used the fact that $\mathbf{x}^0$ and $\mathbf{x}^1$ belong to $\bar{\mathcal{L}}^0$ (cf. Remark 4.7.1).

We now bound `term I` in (4.26). It is convenient to study the more general term $\|\mathbf{x}^k - \mathbf{x}^k(i^k, j)\|_2^2$, $j \in \mathcal{N}_{i^k}$. There holds:

$$\begin{aligned}\|\mathbf{x}^k - \mathbf{x}^k(i^k, j)\|_2^2 &\overset{(4.18)}{\leq} \left(\sum_{l=k-D}^{k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\right)^2 \leq D \sum_{l=k-D}^{k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 \\ &= D\Bigg(\sum_{l=k-D}^{k-1} (l-(k-1)+D)\,\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 \\ &\quad - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\Bigg) + D^2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2. \end{aligned} \tag{4.27}$$

Combining (4.26) and (4.27) one can check that statements (a) and (b) of the lemma hold at $k = 0$, that is, $\tilde{V}(\mathbf{x}^1, \ldots, \mathbf{x}^0) \leq \tilde{V}(\mathbf{x}^0, \ldots, \mathbf{x}^0)$, and $V(\mathbf{x}^1) \leq \tilde{V}(\mathbf{x}^1, \ldots, \mathbf{x}^0) \leq \tilde{V}(\mathbf{x}^0, \ldots, \mathbf{x}^0) = V(\mathbf{x}^0)$, respectively.
Assume now that the two statements hold at iteration k. It is easy to check that the analogous of (4.26) also holds at iteration $k+1$ with the term $\sum_{j \in \mathcal{N}_{i^k}} \|\mathbf{x}^k - \mathbf{x}^k(i^k, j)\|_2$ in the analogous of `term I` at iteration k, majorized using (4.27). Combining (4.26) at $k+1$ with (4.27) one can check that statement (a) of the lemma holds at $k+1$. We also get: $V(\mathbf{x}^{k+1}) \leq \tilde{V}\left(\mathbf{x}^{k+1}, \ldots, \mathbf{x}^{k+1-D}\right) \overset{(4.26)}{\leq} \tilde{V}\left(\mathbf{x}^k, \ldots, \mathbf{x}^{k-D}\right) \leq \tilde{V}(\mathbf{x}^0, \ldots, \mathbf{x}^0) = V(\mathbf{x}^0)$, which proves statement (b) of the lemma at $k+1$. This completes the proof.
■

Step 2 – Vanishing x-stationarity

It follows from A4 and Lemma 4.7.2 that, if $\gamma < \frac{2\tau}{L(2+\rho^2 D^2)}$, $\{\tilde{V}(\mathbf{x}^{k-D}, \ldots, \mathbf{x}^k)\}$ and thus $\{V(\mathbf{x}^k)\}$ converge. Therefore,

$$\lim_{k \to +\infty} \|\Delta\widehat{\mathbf{x}}^k_{i^k}\|_2 = 0. \tag{4.28}$$

The next lemma extends the vanishing properties of a single block $\Delta\widehat{\mathbf{x}}^k_{i^k}$ to the entire vector $\Delta\widehat{\mathbf{x}}^k$.

Lemma 4.7.3 *For any $i \in \mathcal{N}, k \geq 0$, and $h, t \in [k, k+B-1]$, there hold:*

$$\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^t(i)) - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^h(i))\|_2^2 \leq C_2 \sum_{l=k-D}^{k+B-2} \|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \tag{4.29}$$

$$\|\Delta\widehat{\mathbf{x}}^h\|_2^2 \leq 2\Big(NC_2 + 1\Big) \sum_{l=k-D}^{k+B-1} \|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2. \tag{4.30}$$

with

$$C_2 \triangleq \frac{3\gamma^2(B + 2D - N + 1)\rho\,(L_m^2 + (\rho - 1)L^2)}{\tau^2}.$$

Proof See Section 4.7.5. ■

Using (4.30) and (4.28) yields

$$\lim_{k\to+\infty} \|\Delta\widehat{\mathbf{x}}^k\|_2 = 0. \tag{4.31}$$

Furthermore, invoking (4.28), (4.29), and (4.31) together with $\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2 \leq \|\Delta\widehat{\mathbf{x}}^k\|_2 + \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}^k\|_2$, leads to

$$\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2 = 0, \tag{4.32}$$

which, together with Proposition 4.7.1(c), proves Theorem 4.4.1(a).

Step 3 – Convergence rate

We use the Lyapunov function $\tilde{V}$ to study the vanishing rate of $\{M_V(\mathbf{x}^k)\}$. Due to (4.32) and the definition of M_V, we know that M_V is converging to 0. Therefore T_ϵ is finite. Using $M_V(\mathbf{x}^k) > \epsilon$, for all $k \in \{0, \ldots, T_\epsilon - 1\}$, we have

$$\begin{aligned}
T_\epsilon \epsilon &\leq \sum_{k=0}^{T_\epsilon - 1} M_V(\mathbf{x}^k) \leq 2 \sum_{k=0}^{T_\epsilon - 1} \left(\|\Delta \widehat{\mathbf{x}}^k\|_2^2 + \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}^k\|_2^2 \right) \\
&\overset{(4.22),(4.30)}{\leq} 2 \sum_{k=0}^{T_\epsilon - 1} \left(2\left(NC_2 + 1\right) \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \right. \\
&\quad + \sum_{i=1}^{N} \left(\frac{L_m^2 \rho}{\tau^2} \|\mathbf{x}^k(i,i) - \mathbf{x}^k\|_2^2 \right. \\
&\quad \left. \left. + \frac{L^2 \rho}{\tau^2} \sum_{j \in \mathcal{N}_i \setminus \{i\}} \|\mathbf{x}^k(i,j) - \mathbf{x}^k\|_2^2 \right) \right) \\
&\overset{(4.18)}{\leq} 2 \left(2\left(NC_2 + 1\right) + \frac{DC_2}{3(B + 2D - N + 1)} \right) \\
&\quad \cdot \sum_{k=0}^{T_\epsilon - 1} \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \\
&\overset{(a)}{\leq} C_3 \sum_{k=0}^{T_\epsilon - 1} \sum_{l=k-D}^{k+B-1} \left(\tilde{V}\left(\mathbf{x}^l, \ldots, \mathbf{x}^{l-D}\right) - \tilde{V}\left(\mathbf{x}^{l+1}, \ldots, \mathbf{x}^{l+1-D}\right) \right) \\
&= C_3 \sum_{k=0}^{T_\epsilon - 1} \left(\tilde{V}\left(\mathbf{x}^{k-D}, \ldots, \mathbf{x}^{k-2D}\right) - \tilde{V}\left(\mathbf{x}^{k+B}, \ldots, \mathbf{x}^{k+B-D}\right) \right) \\
&\leq C_3(B + D - 1) \left(V(\mathbf{x}^0) - \min_{\mathbf{x} \in \mathcal{X}} V(\mathbf{x}) \right),
\end{aligned} \tag{4.33}$$

where in (a) we used (4.25) and defined C_3 as

$$C_3 \triangleq \frac{4 \left(2\left(NC_2 + 1\right) + \frac{DC_2}{3(B+2D-N+1)} \right)}{\gamma \left(2\tau - \gamma L \left(2 + D^2 \rho^2 \right) \right)}.$$

Statement (b) of the theorem follows readily by defining

$$C_1 \triangleq C_3(B + D - 1). \tag{4.34}$$

4.7.4 Proof of Theorem 4.4.2

We study now convergence of Algorithm 1 under the additional Assumption B.

First of all, note that one can always find $\eta, \epsilon, \kappa > 0$ such that B1 holds. In fact, i) by Lemma 4.7.2, there exist some η and sufficiently small γ/τ such that $V(\mathbf{x}^k) \leq \eta$, for all $k \geq 0$; and ii) since $\|\mathbf{x}^k - \mathtt{prox}_G\left(\nabla F(\mathbf{x}^k) - \mathbf{x}^k\right)\|_2$ is asymptotically vanishing [Proposition 4.7.1(d) and (4.32)], one can always find some $\epsilon > 0$ such that $\|\mathbf{x}^k - \mathtt{prox}_G\left(\nabla F(\mathbf{x}^k) - \mathbf{x}^k\right)\|_2 \leq \epsilon$, for all $k \geq 0$.

The proof proceeds along the following steps. **Step 1:** We first show that the liminf of $\left\{V(\mathbf{x}^k)\right\}$ is a stationary point $V^\star$, see (4.39). **Step 2** shows that $\left\{V(\mathbf{x}^k)\right\}$ approaches $V^\star$ linearly, up to an error of the order $\mathcal{O}\left(\sum_{l=k-D}^{k+B-1} \|\Delta\widehat{\mathbf{x}}_{i_l}^l\|_2^2\right)$, see (4.43). Finally, in **Step 3** we show that the term $\sum_{l=k-D}^{k+B-1} \|\Delta\widehat{\mathbf{x}}_{i_l}^l\|_2^2$ is overall vanishing at a geometric rate, implying the convergence of $\left\{V(\mathbf{x}^k)\right\}$ to $V^\star$ at a geometric rate.

Step 1

Pick any vector $\mathbf{x}^\star(\mathbf{x}^k) \in P_{\mathcal{X}^\star}(\mathbf{x}^k)$, where $P_{\mathcal{X}^\star}(\mathbf{x}) \triangleq \arg\min_{\mathbf{x}^\star \in \mathcal{X}^\star} \|\mathbf{x} - \mathbf{x}^\star\|_2$, $\mathbf{x} \in \mathbb{R}^n$. Note that:

$$d(\mathbf{x}^k, \mathcal{X}^\star) = \|\mathbf{x}^\star(\mathbf{x}^k) - \mathbf{x}^k\|_2 \overset{B1}{\leq} \kappa \|\mathbf{x}^k - \mathtt{prox}_G\big(\nabla_{\mathbf{x}} F(\mathbf{x}^k) - \mathbf{x}^k\big)\|_2. \tag{4.35}$$

Using (4.35), (4.32), and (4.23), yields

$$\lim_{k \to +\infty} \|\mathbf{x}^\star(\mathbf{x}^k) - \mathbf{x}^\star(\mathbf{x}^{k+1})\| = 0. \tag{4.36}$$

This, together with B2, imply that there exists an index $\bar{k} \geq 0$ and a scalar $V^\star$ such that

$$V(\mathbf{x}^\star(\mathbf{x}^k)) = V^\star, \quad \forall\, k \geq \bar{k}. \tag{4.37}$$

By the Mean Value Theorem, there exists a vector $\boldsymbol{\xi}^k = \beta^k \mathbf{x}^\star(\mathbf{x}^k) + (1-\beta^k)\mathbf{x}^k$, for some $\beta^k \in (0;1)$, such that, for any $k \geq \bar{k}$,

$$\begin{aligned}
&V^\star - V(\mathbf{x}^k) = \left\langle \nabla_{\mathbf{x}} F(\boldsymbol{\xi}^k), \mathbf{x}^\star(\mathbf{x}^k) - \mathbf{x}^k \right\rangle + G(\mathbf{x}^\star(\mathbf{x}^k)) \\
&- G(\mathbf{x}^k) \leq \left\langle \nabla_{\mathbf{x}} F(\boldsymbol{\xi}^k) - \nabla_{\mathbf{x}} F(\mathbf{x}^\star(\mathbf{x}^k)), \mathbf{x}^\star(\mathbf{x}^k) - \mathbf{x}^k \right\rangle \\
&\overset{(a)}{\leq} \frac{N(\rho^2 L^2 + 1)}{2} \|\mathbf{x}^\star(\mathbf{x}^k) - \mathbf{x}^k\|_2^2 \\
&\overset{(4.23),(4.35)}{\leq} \frac{N\kappa(\rho^2 L^2 + 1)(1 + L + N L_m)}{2} \|\widehat{\mathbf{x}}(\bar{\mathbf{x}})^k - \mathbf{x}^k\|_2,
\end{aligned} \tag{4.38}$$

where (a) follows from A2 and $\|\boldsymbol{\xi}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 = \|\beta^k \mathbf{x}^\star(\mathbf{x}^k) + (1-\beta^k)\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 \leq \|\mathbf{x}^\star(\mathbf{x}^k) - \mathbf{x}^k\|_2^2$.

By invoking (4.38), together with (4.32), we obtain

$$\liminf_{k \to +\infty} V(\mathbf{x}^k) \geq V^\star. \tag{4.39}$$

Step 2

We next show that $V(\mathbf{x}^k)$ approaches $V^\star$ at a linear rate.

To this end, consider (4.26) with 0 and 1 replaced by k and $k+1$ respectively; we have the following:

$$\begin{aligned}
&V(\mathbf{x}^{k+1}) \leq V(\mathbf{x}^k) - \gamma\,(\tau - \gamma L)\,\|\Delta \widehat{\mathbf{x}}_{i^k}\|_2^2 \\
&+ \frac{L\rho}{2} \sum_{j \in \mathcal{N}_{i^k}} \|\mathbf{x}^k - \mathbf{x}^k(i^k, j)\|_2^2 \overset{(4.18)}{\leq} V(\mathbf{x}^k) \\
&- \gamma\,(\tau - \gamma L)\,\|\Delta \widehat{\mathbf{x}}_{i^k}\|_2^2 + \frac{\gamma^2 D L \rho^2}{2} \sum_{l=k-D}^{k-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2.
\end{aligned} \tag{4.40}$$

Is easy to see that, for any $k \geq \bar{k}$, (4.40) implies:

$$\begin{aligned} V(\mathbf{x}^{k+B}) - V^\star \leq\, & V(\mathbf{x}^k) - V^\star \\ & - \gamma\left(\tau - \frac{\gamma L(2+BD\rho^2)}{2}\right)\sum_{l=k}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 \\ & + \frac{B\gamma^2 DL\rho^2}{2}\sum_{l=k-D}^{k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2. \end{aligned} \tag{4.41}$$

To prove the desired result we will combine next (4.41) with the following lemma.

Lemma 4.7.4 *For any $k \geq 0$, there holds:*

$$\begin{aligned} V(\mathbf{x}^{k+B}) - V(\mathbf{x}^\star(\mathbf{x}^k)) \leq\, & (1-\gamma)\big(V(\mathbf{x}^k) - V(\mathbf{x}^\star(\mathbf{x}^k))\big) \\ & + \gamma\,(N\alpha_1 + (B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \end{aligned} \tag{4.42}$$

where α_1 and α_2 are two positive constants defined in Appendix 4.7.5 [see (4.67) *and* (4.69)*, respectively].*

Proof See Section 4.7.5. ■

Multiplying the two sides of (4.41) and (4.42) by $(N\alpha_1 + (B-N)\alpha_2)$ and $\tau - \gamma L(2+BD\rho^2)/2$ respectively, and adding the two inequalities together, yields

$$V(\mathbf{x}^{k+B}) - V^\star \leq \theta\left(V(\mathbf{x}^k) - V^\star\right) + \zeta\sum_{l=k-D}^{k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \tag{4.43}$$

for all $k \geq \bar{k}$, where

$$\theta \triangleq \frac{(1-\gamma)(2\tau - \gamma L(BD\mathcal{N}_m^2+2)) + 2N\alpha_1 + 2(B-N)\alpha_2}{2\tau - \gamma L(BD\mathcal{N}_m^2+2) + 2N\alpha_1 + 2(B-N)\alpha_2},$$

and

$$\zeta \triangleq \frac{(N\alpha_1 + (B-N)\alpha_2)(2\tau + \gamma L(BD\rho^2(\gamma-1)+2))}{2\tau - \gamma L(BD\mathcal{N}_m^2+2) + 2N\alpha_1 + 2(B-N)\alpha_2}.$$

Step 3

We can now apply Lemma 4.5 in [59] by noticing that (4.41), (4.39), and (4.43) correspond, respectively, to (4.21), (4.22), and to the first inequality after (4.23) in [59]. Theorem 4.4.2 readily follows, setting

$$\lambda \triangleq 1 - \frac{\gamma^2}{2}\frac{2\tau - \gamma L(BD\rho^2+2)}{2N\alpha_1 + 2(B-N)\alpha_2 + \gamma(2-\gamma)(2\tau - \gamma L(BD\rho^2+2))}. \tag{4.44}$$

4.7.5 Miscellanea results

This section contains the proofs of Lemma 4.7.3 and Lemma 4.7.4.

Proof of Lemma 4.7.3: (i) Assume without loss of generality that $t \leq h$. We have

$$\begin{aligned}
&\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^t(i)) - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^h(i))\|_2^2 \overset{(4.22)}{\leq} \frac{\rho L_m^2}{\tau^2}\|\mathbf{x}^t(i,i) - \mathbf{x}^h(i,i)\|_2^2 \\
&+ \frac{\rho L^2}{\tau^2}\sum_{j\in\mathcal{N}_i\setminus\{i\}}\|\mathbf{x}^t(i,j) - \mathbf{x}^h(i,j)\|_2^2 \\
&\overset{(4.18),(4.7)}{\leq} \left(\frac{3\rho\left(L_m^2 + (\rho-1)L^2\right)}{\tau^2}\right)\Bigg(\gamma^2(B-N+1) \\
&\sum_{l=t}^{h-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 + D\gamma^2\left(\sum_{l=t-D}^{t-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 + \sum_{l=h-D}^{h-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\right)\Bigg).
\end{aligned}$$

(ii) Define $r_i^{h,k} \triangleq \underset{t\in[k;k+B-1]:i^t=i}{\arg\min}|t-h|$. We have:

$$\begin{aligned}
\|\Delta\widehat{\mathbf{x}}^h\|_2^2 &\leq 2\sum_{i=1}^N\left(\|\widehat{\mathbf{x}}_i^h - \widehat{\mathbf{x}}_i^{r_i^{h,k}}\|_2^2 + \|\Delta\widehat{\mathbf{x}}_i^{r_i^{h,k}}\|_2^2\right) \qquad (4.45)\\
&\overset{(4.29)}{\leq} 2\sum_{i=1}^N\left(C_2\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 + \|\Delta\widehat{\mathbf{x}}_i^{r_i^{h,k}}\|_2^2\right).
\end{aligned}$$

Proof of Lemma 4.7.4: Define T_i^k+1 as the number of times agent i performs its update within $[k, k+B-1]$; let $l_{i,0}^k, \ldots, l_{i,T_i^k}^k$,be the iteration indexes of such updates.

By the Mean Value Theorem, there exists a vector $\boldsymbol{\xi}^k = \beta^k \mathbf{x}^\star(\mathbf{x}^k) + (1-\beta^k)\mathbf{x}^k$, for some $\beta^k \in (0,1)$, such that

$$\begin{aligned}
&V(\mathbf{x}^{k+B}) - V(\mathbf{x}^\star(\mathbf{x}^k)) = \left\langle \nabla_{\mathbf{x}} F(\boldsymbol{\xi}^k), \mathbf{x}^{k+B} - \mathbf{x}^\star(\mathbf{x}^k) \right\rangle \\
&+ G(\mathbf{x}^{k+B}) - G(\mathbf{x}^\star(\mathbf{x}^k)) \\
&= \sum_{i=1}^{N} \Bigg(\underbrace{\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k) \right\rangle}_{\texttt{term II}} \\
&\quad + \sum_{t=1}^{T_i^k - 1} \underbrace{\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,t+1}^k} - \mathbf{x}_i^{l_{i,t}^k} \right\rangle}_{\texttt{term III}} \\
&\quad + \underbrace{\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{k+B} - \mathbf{x}_i^{l_{i,T_i^k}^k} \right\rangle}_{\texttt{term IV}} \Bigg) + G(\mathbf{x}^{k+B}) - G(\mathbf{x}^\star(\mathbf{x}^k)).
\end{aligned} \tag{4.46}$$

To prove (4.42), it is then sufficient show that `term II`, `term III`, and `term IV` in (4.46) converge at a geometric rate up to an error of the order $\mathcal{O}\left(\sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i_l}^l\|_2^2\right)$. To do this, we first show that `term II`, `term III`, and `term IV` converges at a geometric rate up to the error terms $a_{i,4}^k$, $b_{i,t,4}^k$, and $c_{i,4}^k$, respectively [see (4.47), (4.50), and (4.53)]. Then, we prove that each of these errors is of the order $\mathcal{O}\left(\sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i_l}^l\|_2^2\right)$, as desired [see (4.66), and (4.68)].

`Term II` can be upper bounded as

$$
\begin{aligned}
&\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k) \right\rangle \overset{A2}{\leq} \left\langle \nabla_{\mathbf{x}_i} F\left(\widehat{\mathbf{x}}^{l_{i,0}^k}\right), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k) \right\rangle \\
&+ \underbrace{\rho L \left\| \widehat{\mathbf{x}}^{l_{i,0}^k} - \boldsymbol{\xi}^k \right\|_2 \left\| \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k) \right\|_2}_{\triangleq a_{i,1}^k} \overset{A2,C2,C3}{\leq} \Bigg\langle \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i) \right) \\
&+ \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j) \right), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k) \Bigg\rangle \\
&+ \left\| \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k) \right\|_2 \Bigg(L_i \left\| \widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,0}^k} - \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i) \right\|_2 \\
&\quad + L \sum_{j \in \mathcal{N}_i \setminus \{i\}} \left\| \widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,0}^k} - \mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j) \right\|_2 \Bigg) + a_{i,1}^k \\
&\overset{(a)}{\leq} (\gamma - 1) \Bigg\langle \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i) \right) + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j) \right), \Delta \widehat{\mathbf{x}}_i^{l_{i,0}^k} \Bigg\rangle \\
&\quad + g_i\left(\mathbf{x}_i^\star(\mathbf{x}^k)\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) + a_{i,2}^k \overset{C2}{\leq} g_i(\mathbf{x}_i^\star(\mathbf{x}^k)) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) \\
&+ (\gamma - 1) \left\langle \sum_{j \in \mathcal{N}_i} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j) \right), \Delta \widehat{\mathbf{x}}_i^{l_{i,0}^k} \right\rangle \\
&+ (1 - \gamma) \left\| \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i) \right) - \nabla \tilde{f}_i \left(\mathbf{x}_i^{l_{i,0}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i) \right) \right\|_2 \\
&\cdot \left\| \Delta \widehat{\mathbf{x}}_i^{l_{i,0}^k} \right\|_2 + a_{i,2}^k \overset{(b)}{\leq} g_i\left(\mathbf{x}_i^\star(\mathbf{x}^k)\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) + \frac{1-\gamma}{\gamma} \left(V\left(\mathbf{x}^{l_{i,0}^k}\right) - V\left(\mathbf{x}^{l_{i,0}^k + 1}\right) \right) \\
&+ (1 - \gamma) \left\| \sum_{j \in \mathcal{N}_i} \left(\nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k} \right) - \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j) \right) \right) \right\|_2 \\
&\cdot \left\| \Delta \widehat{\mathbf{x}}_i^{l_{i,0}^k} \right\|_2 + \frac{L\gamma(1-\gamma)}{2} \left\| \Delta \widehat{\mathbf{x}}_i^{l_{i,0}^k} \right\|_2^2 + a_{i,3}^k + (1-\gamma)\left(g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) - g_i\left(\mathbf{x}_i^{l_{i,0}^k}\right) \right) \\
&\overset{(c)}{=} \frac{1-\gamma}{\gamma} \left(V\left(\mathbf{x}^{l_{i,0}^k}\right) - V\left(\mathbf{x}^{l_{i,0}^k + 1}\right) \right) + g_i\left(\mathbf{x}_i^\star(\mathbf{x}^k)\right) + (\gamma - 1) g_i\left(\mathbf{x}_i^{l_{i,0}^k}\right) \\
&\quad - \gamma g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) + a_{i,4}^k;
\end{aligned} \tag{4.47}
$$

where the quantities $a_{i,2}^k$ in (a), and $a_{i,3}^k$ in (b) are defined in (4.48) and (4.49) at the bottom of the next page, respectively; furthermore in (b) we used the descent lemma, and in (c) we defined

$$\begin{aligned} a_{i,4}^k \triangleq\ & a_{i,3}^k + \frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2^2 + (1-\gamma) \\ & \cdot \underbrace{\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}\right) - \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right)\right)\right\|_2 \left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2}_{\texttt{term VII}}. \end{aligned}$$

$$a_{i,2}^k \triangleq a_{i,1}^k + \underbrace{\left\|\mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\|_2 \left(L_i\left\|\widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,0}^k} - \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right\|_2 + L\sum_{j\in\mathcal{N}_i\setminus\{i\}}\left\|\widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,0}^k} - \mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right\|_2\right)}_{\texttt{term V}} \tag{4.48}$$

$$a_{i,3}^k \triangleq a_{i,2}^k + (1-\gamma)\underbrace{\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right) - \nabla\tilde{f}_i\left(\mathbf{x}_i^{l_{i,0}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)\right\|_2 \left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2}_{\texttt{term VI}} \tag{4.49}$$

`Term III` can be upper bounded as: for any i and $t \in [1, T_i^k - 1]$,

$$\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,t+1}^k} - \mathbf{x}_i^{l_{i,t}^k} \right\rangle \overset{A2}{\leq} \left\langle \nabla_{\mathbf{x}_i} F\left(\widehat{\mathbf{x}}^{l_{i,t}^k}\right), \mathbf{x}_i^{l_{i,t+1}^k} - \mathbf{x}_i^{l_{i,t}^k} \right\rangle$$
$$+ \underbrace{\rho L \left\| \widehat{\mathbf{x}}^{l_{i,t}^k} - \boldsymbol{\xi}^k \right\|_2 \left\| \mathbf{x}_i^{l_{i,t}^k} - \mathbf{x}_i^{l_{i,t+1}^k} \right\|_2}_{\triangleq b_{i,t,1}^k} \overset{A2,C2,C3}{\leq} \left\langle \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i) \right) \right.$$
$$\left. + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j) \right), \mathbf{x}_i^{l_{i,t+1}^k} - \mathbf{x}_i^{l_{i,t}^k} \right\rangle$$
$$+ \left\| \mathbf{x}_i^{l_{i,t}^k} - \mathbf{x}_i^{l_{i,t+1}^k} \right\|_2 \left(L_i \left\| \widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,t}^k} - \mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i) \right\|_2 \right.$$
$$\left. + L \sum_{j \in \mathcal{N}_i \setminus \{i\}} \left\| \widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,t}^k} - \mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j) \right\|_2 \right) + b_{i,t,1}^k$$
$$\overset{(a)}{\leq} (\gamma - 1) \left\langle \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i) \right) + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j) \right), \Delta \widehat{\mathbf{x}}_i^{l_{i,t}^k} \right\rangle$$
$$+ g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right) + b_{i,t,2}^k \overset{C2}{\leq} g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)$$
$$+ (\gamma - 1) \left\langle \sum_{j \in \mathcal{N}_i} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j) \right), \Delta \widehat{\mathbf{x}}_i^{l_{i,t}^k} \right\rangle$$
$$+ (1 - \gamma) \left\| \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i) \right) - \nabla \tilde{f}_i \left(\mathbf{x}_i^{l_{i,t}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i) \right) \right\|_2$$
$$\cdot \left\| \Delta \widehat{\mathbf{x}}_i^{l_{i,t}^k} \right\|_2 + b_{i,t,2}^k \overset{(b)}{\leq} g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)$$
$$+ \frac{1-\gamma}{\gamma} \left(V\left(\mathbf{x}^{l_{i,t}^k}\right) - V\left(\mathbf{x}^{l_{i,t}^k + 1}\right) \right)$$
$$+ (1 - \gamma) \left\| \sum_{j \in \mathcal{N}_i} \left(\nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k} \right) - \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j) \right) \right) \right\|_2$$
$$\left\| \Delta \widehat{\mathbf{x}}_i^{l_{i,t}^k} \right\|_2 + \frac{L\gamma(1-\gamma)}{2} \left\| \Delta \widehat{\mathbf{x}}_i^{l_{i,t}^k} \right\|_2^2 + b_{i,t,3}^k + (1-\gamma) \left(g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right) - g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) \right)$$
$$= \frac{1-\gamma}{\gamma} \left(V\left(\mathbf{x}^{l_{i,t}^k}\right) - V\left(\mathbf{x}^{l_{i,t}^k + 1}\right) \right) \quad + \gamma \left(g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right) \right) + b_{i,t,4}^k; \tag{4.50}$$

where the quantities $b_{i,t,2}^{k}$ in (a), and $b_{i,t,3}^{k}$ in (b) are defined in (4.51) and (4.52) at the bottom of the next page, respectively; furthermore in (b) we used the descent lemma, and in (c) we defined

$$b_{i,t,4}^{k} \triangleq b_{i,t,3}^{k} + \frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_{i}^{l_{i,t}^{k}}\right\|_{2}^{2} + (1-\gamma)$$
$$\cdot \underbrace{\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^{k}}\right) - \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^{k}}(i,j)\right)\right)\right\|_{2}\left\|\Delta\widehat{\mathbf{x}}_{i}^{l_{i,t}^{k}}\right\|_{2}}_{\texttt{termVII}}.$$

$$b_{i,t,2}^{k} \triangleq b_{i,t,1}^{k} + \underbrace{\left\|\mathbf{x}_{i}^{l_{i,t}^{k}} - \mathbf{x}_{i}^{l_{i,t+1}^{k}}\right\|_{2}\left(L_i\left\|\widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,t}^{k}} - \mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^{k}}(i,i)\right\|_{2} + L\sum_{j\in\mathcal{N}_i\setminus\{i\}}\left\|\widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,t}^{k}} - \mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^{k}}(i,j\right\|_{2}\right)}_{\texttt{term VIII}} \tag{4.51}$$

$$b_{i,t,3}^{k} \triangleq b_{i,t,2}^{k} + (1-\gamma)\underbrace{\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_{i}^{l_{i,t}^{k}};\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^{k}}(i,i)\right) - \nabla\tilde{f}_i\left(\mathbf{x}_{i}^{l_{i,t}^{k}};\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^{k}}(i,i)\right)\right\|_{2}\left\|\Delta\widehat{\mathbf{x}}_{i}^{l_{i,t}^{k}}\right\|_{2}}_{\texttt{term VI}} \tag{4.52}$$

Following similar steps, we can bound `term IV`, as

$$
\begin{aligned}
&\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{k+B} - \mathbf{x}_i^{l^k_{i,T_i^k}} \right\rangle \overset{A2}{\leq} \left\langle \nabla_{\mathbf{x}_i} F\left(\widehat{\mathbf{x}}^{l^k_{i,T_i^k}}\right), \mathbf{x}_i^{k+B} - \mathbf{x}_i^{l^k_{i,T_i^k}} \right\rangle \\
&+ \underbrace{\rho L \left\| \widehat{\mathbf{x}}^{l^k_{i,T_i^k}} - \boldsymbol{\xi}^k \right\|_2 \left\| \mathbf{x}_i^{l^k_{i,T_i^k}} - \mathbf{x}_i^{k+B} \right\|_2}_{c^k_{i,1}} \overset{A2,C2,C3}{\leq} \left\langle \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}}; \mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i) \right) \right. \\
&\left. + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j) \right), \mathbf{x}_i^{k+B} - \mathbf{x}_i^{l^k_{i,T_i^k}} \right\rangle \\
&+ \left\| \mathbf{x}_i^{l^k_{i,T_i^k}} - \mathbf{x}_i^{k+B} \right\|_2 \left(L_i \left\| \widehat{\mathbf{x}}_{\mathcal{N}_i}^{l^k_{i,T_i^k}} - \mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i) \right\|_2 \right. \\
&\left. + L \sum_{j \in \mathcal{N}_i \setminus \{i\}} \left\| \widehat{\mathbf{x}}_{\mathcal{N}_j}^{l^k_{i,T_i^k}} - \mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j) \right\|_2 \right) + c^k_{i,1} \\
&\overset{(a)}{\leq} (\gamma - 1) \left\langle \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}}; \mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i) \right) + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j) \right), \Delta \widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right\rangle \\
&+ g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}} \right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right) + c^k_{i,2} \overset{C2}{\leq} g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}} \right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right) \\
&+ (\gamma - 1) \left\langle \sum_{j \in \mathcal{N}_i} \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j) \right), \Delta \widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right\rangle \\
&+ (1 - \gamma) \left\| \nabla \tilde{f}_i \left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}}; \mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i) \right) - \right. \\
&\left. \nabla \tilde{f}_i \left(\mathbf{x}_i^{l^k_{i,T_i^k}}; \mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i) \right) \right\|_2 \left\| \Delta \widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right\|_2 + c^k_{i,2} \overset{(b)}{\leq} g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}} \right) \\
&- g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right) + \frac{1-\gamma}{\gamma} \left(V\left(\mathbf{x}^{l^k_{i,T_i^k}} \right) - V\left(\mathbf{x}^{l^k_{i,T_i^k}+1} \right) \right) \\
&+ (1 - \gamma) \left\| \sum_{j \in \mathcal{N}_i} \left(\nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}} \right) - \nabla_{\mathbf{x}_i} f_j \left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j) \right) \right) \right\|_2 \\
&\left\| \Delta \widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right\|_2 + \frac{L\gamma(1-\gamma)}{2} \left\| \Delta \widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right\|_2^2 + c^k_{i,3} + (1 - \gamma) \left(g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right) - g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}} \right) \right) \\
&= \frac{1-\gamma}{\gamma} \left(V\left(\mathbf{x}^{l^k_{i,T_i^k}} \right) - V\left(\mathbf{x}^{l^k_{i,T_i^k}+1} \right) \right) \quad + \gamma \left(g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}} \right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}} \right) \right) + c^k_{i,4};
\end{aligned}
\tag{4.53}
$$

where the quantities $c_{i,2}^k$ in (a), and $c_{i,3}^k$ in (b) are defined in (4.54) and (4.55) at the bottom of the next page, respectively; furthermore in (b) we used the descent lemma, and in (c) we defined

$$c_{i,4}^k \triangleq c_{i,3}^k + \frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}}\right\|_2^2 + (1-\gamma)$$
$$\cdot \underbrace{\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}\right) - \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j)\right)\right)\right\|_2 \left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}}\right\|_2}_{\texttt{term VII}}.$$

We now show that the error terms $a_{i,4}^k$, $b_{i,t,4}^k$, and $c_{i,4}^k$, are of the order $\mathcal{O}\left(\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i_l}^l\|_2^2\right)$. To do so, in the following we properly upper bound each term inside $a_{i,4}^k$, $b_{i,t,4}^k$, and $c_{i,4}^k$.

We begin noticing that, by the definition of $\boldsymbol{\xi}^k$, it follows

$$\begin{aligned}
\|\widehat{\mathbf{x}}^h - \boldsymbol{\xi}^k\|_2 &= \|(1-\beta^k)\mathbf{x}^k + \beta^k\mathbf{x}^\star(\mathbf{x}^k) - \widehat{\mathbf{x}}^h\|_2 \\
&\leq \|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2 + \|\widehat{\mathbf{x}}^h - \mathbf{x}^k\|_2 \\
&\leq \|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2 + \|\Delta\widehat{\mathbf{x}}^h\| + \|\mathbf{x}^h - \mathbf{x}^k\|, \qquad (4.56)
\end{aligned}$$

for all $h \in [k; k+B-1]$.

$$c_{i,2}^k \triangleq c_{i,1}^k + \underbrace{\left\|\mathbf{x}_i^{l^k_{i,T_i^k}} - \mathbf{x}_i^{k+B}\right\|_2 \left(L_i\left\|\widehat{\mathbf{x}}_{\mathcal{N}_i}^{l^k_{i,T_i^k}} - \mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i)\right\|_2 + L\sum_{j\in\mathcal{N}_i\setminus\{i\}}\left\|\widehat{\mathbf{x}}_{\mathcal{N}_j}^{l^k_{i,T_i^k}} - \mathbf{x}_{\mathcal{N}_j}^{l^k_{i,T_i^k}}(i,j)\right\|_2\right)}_{\texttt{term IX}} \tag{4.54}$$

$$c_{i,3}^k \triangleq c_{i,2}^k + (1-\gamma)\underbrace{\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}};\mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i)\right) - \nabla\tilde{f}_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}};\mathbf{x}_{\mathcal{N}_i}^{l^k_{i,T_i^k}}(i,i)\right)\right\|_2\left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k}}\right\|_2}_{\texttt{term VI}} \tag{4.55}$$

1) Bounding $a_{i,1}^k$: there holds

$$a_{i,1}^k \overset{(a)}{\leq} \frac{3\rho L}{2}\Bigg(2\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 + (1+\gamma^2)\|\Delta\widehat{\mathbf{x}}^{l_{i,0}^k}\|_2^2$$
$$+ 2\|\mathbf{x}^{l_{i,0}^k} - \mathbf{x}^k\|_2^2\Bigg) \overset{(b)}{\leq} 3\rho L\Bigg(\kappa^2(1+L+NL_m)^2\Big(\|\Delta\widehat{\mathbf{x}}^k\|_2^2$$
$$+ C_2\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\Big) + (NC_2+1)(1+\gamma^2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2$$
$$+ \gamma^2(B-N+1)\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\Bigg) \overset{(c)}{\leq} \rho L\beta_1\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \tag{4.57}$$

where in (a) we used (4.56) and the Young's inequality; (b) follows from (4.29), (4.30), and the fact that, for any $k \geq 0$,

$$\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2 \overset{B1}{\leq} \kappa\|\mathbf{x}^k - \mathtt{prox}_G\left(\nabla_{\mathbf{x}}F(\mathbf{x}^k) - \mathbf{x}^k\right)\|_2$$
$$\overset{(4.23)}{\leq} \kappa(1+L+NL_m)\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2$$
$$\leq \kappa(1+L+NL_m)\left(\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}^k\|_2 + \|\Delta\widehat{\mathbf{x}}^k\|_2\right); \tag{4.58}$$

and in (c) we used (4.30) and defined

$$\beta_1 \triangleq C_2\Bigg(\kappa^2(1+L+NL_m)^2(2N+1) + N(1+\gamma^2)\Bigg)$$
$$+ \kappa^2(1+L+NL_m)^2 + 1 + \gamma^2(B-N+2).$$

2) Bounding $b_{i,t,1}^k$ and $c_{i,1}^k$: for $t \in [1; T_i^k - 1]$,

$$b_{i,t,1}^k \overset{(a)}{\leq} \frac{\rho L}{2}\Big(3\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 + (3+\gamma^2)\|\Delta\widehat{\mathbf{x}}^{l_{i,t}^k}\|_2^2$$
$$+ 3\|\mathbf{x}^{l_{i,t}^k} - \mathbf{x}^k\|_2^2\Big) \overset{(b)}{\leq} \frac{\rho L}{2}\Bigg(6\kappa^2(1+L+NL_m)^2\Bigg(\|\Delta\widehat{\mathbf{x}}^k\|_2^2$$
$$+ C_2\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\Bigg) + 2(NC_2+1)(3+\gamma^2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2$$
$$+ 3\gamma^2(B-N+1)\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\Bigg) \overset{(c)}{\leq} \rho L\beta_2\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \tag{4.59}$$

where in (a) we used (4.56) and the Young's inequality; (b) follows from (4.29), (4.30), (4.58); and in (c) we used (4.30) and defined

$$\begin{aligned}\beta_2 \triangleq & C_2\left(3\kappa^2(1+L+NL_m)^2(2N+1)+N(3+\gamma^2)\right)\\ & +6\kappa^2(1+L+NL_m)^2+3+\frac{\gamma^2}{2}(3B-3N+5).\end{aligned}$$

Following the same steps as in (4.59), it is not difficult to prove:

$$c_{i,1}^k \leq \rho L \beta_2 \sum_{l=k-D}^{k+B-1} \|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2. \tag{4.60}$$

3) Bounding `term V` : there holds,

$$\begin{aligned}\texttt{term III} & \overset{(a)}{\leq} 2\|\mathbf{x}^k-\mathbf{x}^\star(\mathbf{x}^k)\|_2^2+2\gamma^2\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2^2\\ & +(L_i^2+L^2(\rho-1))\left(\|\Delta\widehat{\mathbf{x}}^{l_{i,0}^k}\|_2^2+D\gamma^2\sum_{l=l_{i,0}^k-D}^{l_{i,0}^k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\right)\Bigg)\\ & \overset{(b)}{\leq} \beta_4\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2,\end{aligned} \tag{4.61}$$

where in (a) we used (4.18) and the Young's inequality; and in (b) we used (4.29), (4.30), (4.58), and defined

$$\begin{aligned}\beta_4 \triangleq & 2C_2\Bigg(2\kappa^2(1+L+NL_m)^2(2N+1)\\ & +N\left(L_m^2+L^2(\rho-1)\right)\Bigg)+2\kappa^2(1+L+NL_m)^2\\ & +\left(L_m^2+L^2(\rho-1)\right)(1+D\gamma^2)+2\gamma^2.\end{aligned}$$

4) Bounding `term VI` : for $t\in[0,T_i^k]$,

$$\begin{aligned}\texttt{term I} & \overset{(a)}{\leq} (L^2+L_i^2)\|\mathbf{x}^{l_{i,t}^k}(i,i)-\widehat{\mathbf{x}}^{l_{i,t}^k}\|_2^2+\frac{1}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2^2\\ & \overset{(4.18)}{\leq} (L^2+L_i^2)\left(2\left\|\Delta\widehat{\mathbf{x}}^{l_{i,t}^k}\right\|_2^2+2D\gamma^2\sum_{h=l_{i,t}^k-D}^{l_{i,t}^k-1}\|\Delta\widehat{\mathbf{x}}_{i^h}^h\|_2^2\right)\\ & +\frac{1}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2^2 \overset{(b)}{\leq} \beta_3\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2,\end{aligned} \tag{4.62}$$

where in (a) we used A2, B2, B3, and the Young's inequality; and in (b) we used (4.30) and defined

$$\beta_3 \triangleq 2(L^2+L_m^2)\left(2NC_2+D\gamma^2+1\right)+\frac{1}{2}.$$

5) Bounding `term VII` : for $t \in [0, T_i^k]$,

$$\begin{aligned}
\texttt{term II} &\overset{(a)}{\leq} \frac{1}{2}\left(\rho L^2 \sum_{j\in\mathcal{N}_i}\left\|\mathbf{x}^{l_{i,t}^k}-\mathbf{x}^{l_{i,t}^k}(i,j)\right\|_2^2+\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2^2\right)\\
&\overset{(4.18)}{\leq} \frac{1}{2}\left(\rho^2L^2D^2\gamma^2\sum_{l=l_{i,t}^k-D}^{l_{i,t}^k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2+\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2^2\right)\\
&\leq \frac{\rho^2L^2D^2\gamma^2+1}{2}\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \qquad (4.63)
\end{aligned}$$

where in (a) we used A2 and the Young's inequality.

6) Bounding `term VIII` **and** `term IX` : for $t \in [1, T_i^k - 1]$

$$\begin{aligned}
\texttt{term IV} &\overset{(a)}{\leq} \gamma^2\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2^2+(L_i^2+L^2(\rho-1))\left(\left\|\Delta\mathbf{x}^{l_{i,t}^k}\right\|_2^2\right.\\
&\left.+D\gamma^2\sum_{l=l_{i,t}^k-D}^{l_{i,t}^k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\right)\Bigg) \overset{(b)}{\leq} \beta_5\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \qquad (4.64)
\end{aligned}$$

where in (a) we used (4.18) and the Young's inequality; and in (b) we used (4.30), and defined

$$\beta_5 \triangleq \left(L_m^2+L^2(\rho-1)\right)\left(2NC_2+D\gamma^2+2\right)+\gamma^2.$$

As done in (4.64), it is easy to prove that

$$\texttt{term V} \leq \beta_5\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2. \qquad (4.65)$$

Using the above results, we can bound $a_{i,4}^k$, $b_{i,t,4}^k$, and $c_{i,4}^k$. According to definition of $a_{i,4}^k$, we have

$$a_{i,4}^k \overset{(4.57),(4.62)-(4.61)}{\leq} \alpha_1\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \qquad (4.66)$$

where

$$\alpha_1 \triangleq \left((1-\gamma)\left(\beta_3 + \frac{L\gamma(\rho^2 LD^2\gamma + 1) + 1}{2}\right) + \rho L\beta_1 + \beta_4\right). \tag{4.67}$$

For $b_{i,t,4}^k$ and $c_{i,4}^k$, we have: $t \in [1, T_i^k - 1]$,

$$b_{i,i,4}^k; c_{i,4}^k \overset{(4.59)-(4.63),(4.64),(4.65)}{\leq} \alpha_2 \sum_{l=k-D}^{k+B-1} \|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2, \tag{4.68}$$

where

$$\alpha_2 \triangleq \left((1-\gamma)\left(\beta_3 + \frac{L\gamma(\rho^2 LD^2\gamma + 1) + 1}{2}\right) + \rho L\beta_2 + \beta_5\right). \tag{4.69}$$

Combining (4.46), (4.47), (4.50), (4.53), (4.66), and (4.68) yields:

$$\begin{aligned}
&V(\mathbf{x}^{k+B}) - V(\mathbf{x}^\star(\mathbf{x}^k)) \leq \frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k) - V(\mathbf{x}^{k+B})\right) \\
&+ \sum_{i=1}^N \left(\gamma\left(g_i\left(\mathbf{x}_i^{l_{i,T_i^k}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,T_i^k}^k}\right)\right)\right. \\
&+ \gamma\sum_{t=1}^{T_i^k-1}\left(g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)\right) + (\gamma-1)g_i\left(\mathbf{x}_i^{l_{i,0}^k}\right) \\
&\left.- \gamma g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) + g_i\left(\mathbf{x}_i^{k+B}\right)\right) + (N\alpha_1 \\
&+ (B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 \overset{A3}{\leq} \frac{1-\gamma}{\gamma}\big(V(\mathbf{x}^k) \\
&- V(\mathbf{x}^{k+B})\big) + \sum_{i=1}^N\left(\gamma\left(g_i\left(\mathbf{x}_i^{l_{i,T_i^k}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,T_i^k}^k}\right)\right)\right. \\
&+ \gamma\sum_{t=1}^{T_i^k-1}\left(g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)\right) + (\gamma-1)g_i\left(\mathbf{x}_i^{l_{i,0}^k}\right) \\
&\left.- \gamma g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) + (1-\gamma)g_i\left(\mathbf{x}_i^{l_{i,T_i^k}^k}\right) + \gamma g_i\left(\widehat{\mathbf{x}}_i^{l_{i,T_i^k}^k}\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&+ (N\alpha_1 + (B-N)\alpha_2)) \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \\
&= \frac{1-\gamma}{\gamma}\big(V(\mathbf{x}^k) - V(\mathbf{x}^{k+B})\big) + \sum_{i=1}^{N}\left(g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k}}\right)\right. \\
&\quad + \gamma \sum_{t=1}^{T_i^k-1}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right) \\
&\quad + (\gamma-1) g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right) - \gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right) + (N\alpha_1 \\
&\quad + (B-N)\alpha_2) \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \overset{A3}{\leq} \frac{1-\gamma}{\gamma}\big(V(\mathbf{x}^k) \\
&- V(\mathbf{x}^{k+B})\big) + \sum_{i=1}^{N}\left((1-\gamma) g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k-1}}\right) + \gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T_i^k-1}}\right)\right. \\
&+ \gamma \sum_{t=1}^{T_i^k-1}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right) + (\gamma-1) g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right) - \gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right) \\
&+ (N\alpha_1 + (B-N)\alpha_2) \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \\
&= \frac{1-\gamma}{\gamma}\big(V(\mathbf{x}^k) - V(\mathbf{x}^{k+B})\big) + \sum_{i=1}^{N}\left(g_i\left(\mathbf{x}_i^{l^k_{i,T_i^k-1}}\right)\right. \\
&\quad + \gamma \sum_{t=1}^{T_i^k-2}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right) + (\gamma-1) g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right) - \gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right) \\
&\quad + (N\alpha_1 + (B-N)\alpha_2) \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \leq \frac{1-\gamma}{\gamma}\big(V(\mathbf{x}^k) \\
&- V(\mathbf{x}^{k+B})\big) + (N\alpha_1 + (B-N)\alpha_2) \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2.
\end{aligned}
\tag{4.70}
$$

4.7.6 Proof of Theorem 4.4.3

The proof of this Theorem will follow a path similar to that of the proof of Theorem 4.4.1.

First of all, we notice that the two statements of Lemma 4.7.2 holds in this setting just by replacing γ with γ^k. The proof remains the same.

This implies that: i) statement c) of Theorem 4.4.3 holds true; ii) the following holds: for any $k \geq 0$

$$\tilde{V}(\mathbf{x}^{k+1-D} \ldots, \mathbf{x}^{k+1}) \leq \tilde{V}(\mathbf{x}^{k-D}, \ldots, \mathbf{x}^{k}) - \gamma^k \left(\tau - \gamma^k \frac{L\,(2 + D^2 \mathcal{N}_m^2)}{2} \right) \|\Delta \widehat{\mathbf{x}}_{i^k}^k\|_2^2. \tag{4.71}$$

Invoking E1, we have that for sufficiently large k there exists some positive constant β_1 such that the following holds:

$$\begin{aligned} \tilde{V}(\mathbf{x}^{k+B-D} \ldots, \mathbf{x}^{k+B}) &\leq \tilde{V}\left(\mathbf{x}^{k+B-D-1}, \ldots, \mathbf{x}^{k+B-1}\right) - \beta_1 \gamma^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^k}^k\|_2^2 \\ &\leq \tilde{V}\left(\mathbf{x}^{k-D}, \ldots, \mathbf{x}^{k}\right) - \beta_1 \sum_{t=k}^{k+B-1} \gamma^t \|\Delta \widehat{\mathbf{x}}_{i^t}^t\|_2^2. \end{aligned} \tag{4.72}$$

This result implies that i) $\left\{\tilde{V}\left(\mathbf{x}^{k-D}, \ldots, \mathbf{x}^{k}\right)\right\}$, and thus $\left\{V(\mathbf{x}^k)\right\}$, converge; and ii)

$$\lim_{k \to +\infty} \sum_{l=0}^{+\infty} \left(\sum_{t=l}^{l+B-1} \gamma^t \|\Delta \widehat{\mathbf{x}}_{i^t}^t\|_2^2 \right). \tag{4.73}$$

From the above result and E2, we also have

$$\lim_{k \to +\infty} \sum_{l=0}^{+\infty} \left(\tilde{\gamma}^l \sum_{t=l-D}^{l+B-1} \|\Delta \widehat{\mathbf{x}}_{i^t}^t\|_2^2 \right). \tag{4.74}$$

By replacing γ with γ^k, it is easy to see that Lemma 4.7.3 holds. Following the same steps done in (4.33) we can also prove that, for any $k \geq 0$

$$\sum_{k=0}^{+\infty} \tilde{\gamma}^k M_V(\mathbf{x}^k) \overset{(a)}{\leq} C_4 \sum_{k=0}^{+\infty} \tilde{\gamma}^k \sum_{l=k-D}^{k+B-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2 \overset{(b)}{\leq} c < +\infty, \tag{4.75}$$

where in (a) we defined

$$C_4 \triangleq 2 \left(2(NC_2 + 1) + \frac{DC_2}{2(B + 2D - N + 1)} \right); \tag{4.76}$$

while (b) comes from (4.74). In order to finish the proof, let us now define the quantity $N_\epsilon \triangleq \inf\left\{K \geq 0 | \sum_{k=0}^{K} \tilde{\gamma}^k \geq \frac{c}{\epsilon}\right\}$ and let us assume by contradiction that $M_V(\mathbf{x}^k) > \epsilon$ for $k \in [0; N_\epsilon]$. This would imply:

$$\sum_{k=0}^{N_\epsilon} \tilde{\gamma}^k M_V(\mathbf{x}^k) > \epsilon \sum_{k=0}^{N_\epsilon} \tilde{\gamma}^k \geq \epsilon \frac{c}{\epsilon} = c. \tag{4.77}$$

This result contradicts (4.74), so it there exists $T_\epsilon \in [0; N_\epsilon]$ such that $M_V(\mathbf{x}^k) \leq \epsilon$ and this proves statement (b) of the Theorem. Statement (b) of Theorem 4.4.3 and Proposition 4.7.1(c) prove that statement (c) of Theorem 4.4.3 holds true too.

5. SUMMARY

The Dissertation provides an efficient, asynchronous, algorithmic framework for constrained, nonconvex, optimization problems, with nonsmooth regularizers. Both centralized and distributed settings are considered, and convergence to stationary points is proved, enjoying a sublinear convergence rate in the general nonconvex case, and a linear convergence rate when standard error bound conditions are satisfied. Extensive simulation results show that the proposed framework is faster than state-of-the-art schemes on problems of practical interests.

One research question that remains open is how to extend the algorithmic framework proposed in Chapter 4 to a problem setting where each local function depends on all the block-variables. Distributed asynchronous schemes that can deal with this scenario exist in the literature, see, e.g., [108, 113], but it is not clear whether, under standard error bound conditions, as the KL property, they can achieve linear convergence rate. The iteration complexity results derived in Chapter 2 suggest that the theoretical approach that has been considered might get the desired convergence results in a fully distributed setting too.

Another open question, always in the distributed setting, concerns the presence of local coupling constraints. Indeed, in the problem setting considered in Chapter 4, the constraint sets have a separable Cartesian structure. It is not clear what happens in a multi-agent scenario, where each agent knows a local constraint set, which depends on the variables of the other agents too. Optimization methods have been proposed to deal with this setting [114, 115], but, even in the strongly convex case, linear convergence rate of these schemes has not been demonstrated.

REFERENCES

REFERENCES

[1] F. Facchinei, G. Scutari, and S. Sagratella. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(7):1874–1889, 2015.

[2] F. H. Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

[3] F. Facchinei, G. Scutari, and S. Sagratella. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(7):1874–1889, 2015.

[4] L. Cannelli, P. Scarponi, G. Scutari, and L. Ying. A parallel algorithm for compressed sensing dynamic mri reconstruction. *Proceedings of the International Society for Magnetic Resonance in Medicine (ISMRM) Scientific Meeting Exhibition (Toronto)*, 2015.

[5] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[6] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[7] S. K. Shevade and S. S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

[8] L. Meier, S. Van De Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[9] D. P. Bertsekas and J. N. Tsitsiklis. Parallel and distributed computation: numerical methods. *Prentice-Hall Englewood Cliffs, NJ*, 23, 1989.

[10] J. De Leeuw. Block-relaxation algorithms in statistics. *Information systems and data analysis*, pages 308–324, 1994.

[11] Y. Sun, P. Babu, and D. P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2016.

[12] J. Fan. Comments on ¡¡wavelets in statistics: A review¿¿ by a. antoniadis. *Journal of the Italian Statistical Society*, 6(2):131, 1997.

[13] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

[14] F. Facchinei, L. Lampariello, and G. Scutari. Feasible methods for nonconvex nonsmooth problems with applications in green communications. *Mathematical Programming*, 164(1):1–36, 2016.

[15] G. Scutari and Y. Sun. Parallel and distributed successive convex approximation methods for big-data optimization. *Multi-agent Optimization*, pages 141–308, 2018.

[16] G. Li and T. K. Pong. Calculus of the exponent of kurdyka–λojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of computational mathematics*, 18(5):1199–1232, 2018.

[17] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009.

[18] J. A. Fessler. Model-based image reconstruction for mri. *IEEE Signal Processing Magazine*, 27(4):81–89, 2010.

[19] G. A. Wright. Magnetic resonance imaging. *IEEE Signal Processing Magazine*, 14(1):56–66, 1997.

[20] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[21] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye. k-t focuss: a general compressed sensing framework for high resolution dynamic mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 61(1):103–116, 2009.

[22] S. J. Wright, R. D. Nowak, and M.A.T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[23] Z. Peng, M. Yan, and W. Yin. Parallel and distributed sparse optimization. In *47th Asilomar Conference on Signals, Systems and Computers*, pages 659–646. IEEE, 2013.

[24] Z.-Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

[25] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

[26] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-λojasiewicz condition. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811, 2016.

[27] D. Drusvyatskiy and A. S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

[28] 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. *Mathematical Programming*, 165(2):471–507, 2017.

[29] A.L. Dontchev and R.T. Rockafellar. Regularity and conditioning of solution mappings in variational analysis. *Set-Valued Analysis*, 12(1-2):79–109, 2004.

[30] J.J. Ye and X.Y. Ye. Necessary optimality conditions for optimization problems with variational inequality constraints. *Mathematics of Operations Research*, 22(4):977–997, 1997.

[31] A. L. Dontchev and R. T. Rockafellar. Implicit functions and solution mappings. *Springer Monogr. Math.*, 2009.

[32] Z. Zhou and A. M.-C. So. A unified approach to error bounds for structured convex optimization problems. *Mathematical Programming*, 165(2):689–728, 2017.

[33] Z.-Q. Luo and P. Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992.

[34] R. T. Rockafellar. *Convex analysis*. Princeton university press, 2015.

[35] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[36] H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

[37] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[38] P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, 2010.

[39] 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*, 1(2):163–186, 2013.

[40] Z.-Q. Luo and P. Tseng. On the convergence rate of dual ascent methods for linearly constrained convex minimization. *Mathematics of Operations Research*, 18(4):846–867, 1993.

[41] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[42] W. Chu, S. S. Keerthi, and C. J. Ong. Bayesian support vector regression using a unified loss function. *IEEE transactions on neural networks*, 15(1):29–44, 2004.

[43] Z.-Q. Luo and P. Tseng. Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. *SIAM Journal on Optimization*, 2(1):43–54, 1992.

[44] X. Wang, J. Ye, X. Yuan, S. Zeng, and J. Zhang. Perturbation techniques for convergence analysis of proximal gradient method and other first-order algorithms via variational analysis. *arXiv preprint arXiv:1810.10051*, 2018.

[45] R. Poliquin and R.T. Rockafellar. Prox-regular functions in variational analysis. *Transactions of the American Mathematical Society*, 348(5):1805–1838, 1996.

[46] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari. Asynchronous parallel algorithms for nonconvex optimization. *Math. Program. https://doi.org/10.1007/s10107-019-01408-w*, pages 1–34, 2019.

[47] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari. Asynchronous parallel nonconvex large-scale optimization. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4706–4710, 2017.

[48] L. Cannelli, G. Scutari, F. Facchinei, and V. Kungurtsev. Parallel asynchronous lock-free algorithms for nonconvex big-data optimization. *50th Asilomar Conference on Signals, Systems and Computers*, pages 1009–1013, 2016.

[49] D. Davis. The asynchronous palm algorithm for nonsmooth nonconvex problems. *arXiv preprint arXiv:1604.00526*, 2016.

[50] D. Davis, B. Edmunds, and M. Udell. The sound of apalm clapping: Faster nonsmooth nonconvex optimization with stochastic asynchronous palm. *Advances in Neural Information Processing Systems 29*, pages 226–234, 2016.

[51] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang. Decomposition by partial linearization: Parallel optimization of multi-agent systems. *IEEE Transactions on Signal Processing*, 62(3):641–656, 2014.

[52] A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari. Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(15):3914–3929, 2015.

[53] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *The Journal of Machine Learning Research*, 16(1):285–322, 2015.

[54] J. Liu and S. J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

[55] Z. Peng, Y. Xu, M. Yan, and W. Yin. Arock: an algorithmic framework for asynchronous parallel coordinate updates. *SIAM Journal on Scientific Computing*, 38(5):A2851–A2879, 2016.

[56] G. M. Baudet. Asynchronous iterative methods for multiprocessors. *Journal of the ACM (JACM)*, 25(2):226–244, 1978.

[57] D. Chazan and W. Miranker. Chaotic relaxation. *Linear algebra and its applications*, 2(2):199–222, 1969.

[58] A. Frommer and D. B. Szyld. On asynchronous iterations. *Journal of computational and applied mathematics*, 123(1):201–216, 2000.

[59] P. Tseng. On the rate of convergence of a partially asynchronous gradient projection algorithm. *SIAM Journal on Optimization*, 1(4):603–619, 1991.

[60] A. Nedić, D. P. Bertsekas, and V. S. Borkar. Distributed asynchronous incremental subgradient methods. *Studies in Computational Mathematics*, 8(C):381–407, 2001.

[61] F. Niu, B. Recht, C. Ré, and S. J. Wright. Hogwild: a lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems 24*, pages 693–701, 2011.

[62] X. Lian, Y. Huang, Y. Li, and J. Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems 28*, pages 2719–2727, 2015.

[63] Z. Huo and H. Huang. Asynchronous stochastic gradient descent with variance reduction for non-convex optimization. *arXiv preprint arXiv:1604.03584*, 2016.

[64] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017.

[65] R. Leblond, F. Pedregosa, and S. Lacoste-Julien. ASAGA: Asynchronous parallel SAGA. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 46–54, 2017.

[66] F. Pedregosa, R. Leblond, and S. Lacoste-Julien. Breaking the nonsmooth barrier: A scalable parallel method for composite optimization. *Advances in Neural Information Processing Systems 30*, pages 56–65, 2017.

[67] M. Hong. A distributed, asynchronous and incremental algorithm for nonconvex optimization: An admm approach. *IEEE Transactions on Control of Network System*, PP(99), 2017.

[68] E. Wei and A. Ozdaglar. On the o (1= k) convergence of asynchronous distributed alternating direction method of multipliers. *Global Conference on Signal and Information Processing (GlobalSIP)*, pages 551–554, 2013.

[69] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. *52nd IEEE Conference on Decision and Control*, pages 3671–3676, 2013.

[70] Z. Peng, Y. Xu, M. Yan, and W. Yin. On the convergence of asynchronous parallel iteration with unbounded delays. *Journal of the Operations Research Society of China*, 7(1):5–42, 2019.

[71] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[72] H. Yun, H.-F. Yu, C.-J. Hsieh, S.V.N. Vishwanathan, and I. Dhillon. Nomad: Non-locking, stochastic multi-machine algorithm for asynchronous and decentralized matrix completion. *Proceedings of the VLDB Endowment*, 7(11):975–986, 2014.

[73] J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[74] A. Klenke. Probability theory: a comprehensive course. *Springer Science & Business Media*, 2013.

[75] 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 Transactions on Signal Processing*, 65(8):1945–1960, 2017.

[76] G. Scutari, F. Facchinei, and L. Lampariello. Parallel and distributed methods for constrained nonconvex optimization-part i: Theory. *IEEE Transactions on Signal Processing*, 65(8):1929–1944, 2017.

[77] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[78] K. Fountoulakis and J. Gondzio. A second-order method for strongly convex\ ell _1-regularization problems. *Mathematical Programming*, 156(1-2):189–219, 2016.

[79] S. Zhang and J. Xin. Minimization of transformed l_1 penalty: theory, difference of convex function algorithm, and robust application in compressed sensing. *Mathematical Programming*, 169(1):307–336, 2018.

[80] H. A. Le Thi, T. P. Dinh, H. M. Le, and X. T. Vo. Dc approximation approaches for sparse optimization. *European Journal of Operational Research*, 244(1):26–46, 2015.

[81] R. Durrett. Probability: theory and examples. *Cambridge university press*, 2010.

[82] H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. *Herbert Robbins Selected Papers*, pages 111–135, 1985.

[83] L. Cannelli, F. Facchinei, and G. Scutari. Multi-agent asynchronous nonconvex large-scale optimization. *IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5, 2017.

[84] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari. Essentially cyclic asynchronous nonconvex large-scale optimization. *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2017.

[85] R. Carli and G. Notarstefano. Distributed partition-based optimization via dual decomposition. *52nd IEEE Conference on Decision and Control*, pages 2979–2984, 2013.

[86] V. Kekatos and G. B. Giannakis. Distributed robust power system state estimation. *IEEE Transactions on Power Systems*, 28(2):1617–1626, 2013.

[87] T. Erseghe. A distributed and scalable processing method based upon admm. *IEEE Signal Processing Letters*, 19(9):563–566, 2012.

[88] Ye Tian, Ying Sun, and Gesualdo Scutari. Achieving linear convergence in distributed asynchronous multi-agent optimization. *arXiv preprint arXiv:1803.10359*, 2018.

[89] Ying Sun, Amir Daneshmand, and Gesualdo Scutari. Convergence rate of distributed optimization algorithms based on gradient tracking. *arXiv preprint arXiv:1905.02637*, 2019.

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

[91] Angelia Nedić, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optimiz.*, 27(4):2597–2633, 2017.

[92] A. Mokhtari, A. Koppel, and A. Ribeiro. A class of parallel doubly stochastic algorithms for large-scale learning. *arXiv preprint arXiv:1606.04991*, 2016.

[93] P. Bianchi, W. Hachem, and F. Iutzeler. A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization. *IEEE Transactions on Automatic Control*, 61(10):2947–2957, 2016.

[94] K. Srivastava and A. Nedić. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):772–790, 2011.

[95] I. Notarnicola and G. Notarstefano. Asynchronous distributed optimization via randomized dual proximal gradient. *IEEE Transactions on Automatic Control*, 62(5):2095–2106, 2017.

[96] J. Xu, S. Zhu, Y. C. Soh, and L. Xie. Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Transactions on Automatic Control*, 63(2):434–448, 2018.

[97] I. Notarnicola and G. Notarstefano. A randomized primal distributed algorithm for partitioned and big-data non-convex optimization. *55th IEEE Conference on Decision and Control*, pages 153–158, 2016.

[98] A. Nedić. Asynchronous broadcast-based convex optimization over a network. *IEEE Transactions on Automatic Control*, 56(6):1337–1351, 2011.

[99] H. Wang, X. Liao, T. Huang, and C. Li. Cooperative distributed optimization in multiagent networks with delays. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2):363–369, 2015.

[100] J. Li, G. Chen, Z. Dong, and Z. Wu. Distributed mirror descent method for multi-agent optimization with delay. *Neurocomputing*, 177:643–650, 2016.

[101] K. I. Tsianos and M. G. Rabbat. Distributed dual averaging for convex optimization under communication delays. *IEEE American Control Conference (ACC)*, pages 1067–1072, 2012.

[102] K. I. Tsianos and M. G. Rabbat. Distributed consensus and optimization under communication delays. *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 974–982, 2011.

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

[104] T. T. Doan, C. L. Beck, and R. Srikant. Impact of communication delays on the convergence rate of distributed optimization algorithms. *arXiv preprint arXiv:1708.03277*, 2017.

[105] 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 Transactions on Signal Processing*, 63(4):811–858, 2015.

[106] S. Kumar, R. Jain, and K. Rajawat. Asynchronous optimization over heterogeneous networks via consensus admm. *IEEE Transactions on Signal and Information Processing over Networks*, 3(1):114–129, 2017.

[107] M. Eisen, A. Mokhtari, and A. Ribeiro. Decentralized quasi-newton methods. *IEEE Transactions on Signal Processing*, 65(10):2613–2628, 2017.

[108] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed. Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks*, 4(2):293–307, 2018.

[109] 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 preprint arXiv:1707.09178*, 2017.

[110] N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. *Advances in neural information processing systems 17*, pages 1329–1336, 2005.

[111] 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 Transactions on Signal Processing*, 65(8):1945–1960, 2017.

[112] G. Scutari, F. Facchinei, and L. Lampariello. Parallel and distributed methods for constrained nonconvex optimization-part i: Theory. *IEEE Transactions on Signal Processing*, 65(8):1929–1944, 2017.

[113] Y. Tian, Y. Sun, and G. Scutari. Asy-sonata: Achieving linear convergence in distributed asynchronous multiagent optimization. *56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 543–551, 2018.

[114] W. Shi, Q. Ling, G. Wu, and W. Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22):6013–6023, 2015.

[115] Shu Liang, Le Yi Wang, and George Yin. Distributed smooth convex optimization with coupled constraints. *IEEE Transactions on Automatic Control*, 2019.

VITA

VITA

[Loris Cannelli received his B.S. and M.S. in electrical and telecommunication engineering from the University of Perugia, Italy, his M.S. in electrical engineering from State University of New York at Buffalo, NY, and his Ph.D. in industrial engineering from the Purdue University, West Lafayette, IN, USA. His research interests include optimization algorithms, machine learning, and big-data analytics.]