Lecture Notes in Mathematics 2224
CIME Foundation Subseries

Angelia Nedić · Jong-Shi Pang
Gesualdo Scutari · Ying Sun

Multi-agent Optimization

Cetraro, Italy 2014

Francisco Facchinei · Jong-Shi Pang
Editors

FONDAZIONE
CIME

Springer

Lecture Notes in Mathematics 2224

More information about this series at http://www.springer.com/series/304

Fondazione C.I.M.E., Firenze

FONDAZIONE
CIME
ROBERTO CONTI
CENTRO INTERNAZIONALE MATEMATICO ESTIVO
INTERNATIONAL MATHEMATICAL SUMMER CENTER

C.I.M.E. stands for *Centro Internazionale Matematico Estivo*, that is, International Mathematical Summer Centre. Conceived in the early fifties, it was born in 1954 in Florence, Italy, and welcomed by the world mathematical community: it continues successfully, year for year, to this day.

Many mathematicians from all over the world have been involved in a way or another in C.I.M.E.'s activities over the years. The main purpose and mode of functioning of the Centre may be summarised as follows: every year, during the summer, sessions on different themes from pure and applied mathematics are offered by application to mathematicians from all countries. A Session is generally based on three or four main courses given by specialists of international renown, plus a certain number of seminars, and is held in an attractive rural location in Italy.

The aim of a C.I.M.E. session is to bring to the attention of younger researchers the origins, development, and perspectives of some very active branch of mathematical research. The topics of the courses are generally of international resonance. The full immersion atmosphere of the courses and the daily exchange among participants are thus an initiation to international collaboration in mathematical research.

C.I.M.E. Director (2002 – 2014)
Pietro Zecca
Dipartimento di Energetica "S. Stecco"
Università di Firenze
Via S. Marta, 3
50139 Florence
Italy
e-mail: zecca@unifi.it

C.I.M.E. Director (2015 –)
Elvira Mascolo
Dipartimento di Matematica "U. Dini"
Università di Firenze
viale G.B. Morgagni 67/A
50134 Florence
Italy
e-mail: mascolo@math.unifi.it

C.I.M.E. Secretary
Paolo Salani
Dipartimento di Matematica "U. Dini"
Università di Firenze
viale G.B. Morgagni 67/A
50134 Florence
Italy
e-mail: salani@math.unifi.it

CIME activity is carried out with the collaboration and financial support of INdAM (Istituto Nazionale di Alta Matematica)

For more information see CIME's homepage: **http://www.cime.unifi.it**

Angelia Nedić • Jong-Shi Pang •
Gesualdo Scutari • Ying Sun

Multi-agent Optimization

Cetraro, Italy 2014

Francisco Facchinei • Jong-Shi Pang
Editors

Springer

FONDAZIONE
CIME
ROBERTO CONTI
CENTRO INTERNAZIONALE MATEMATICO ESTIVO
INTERNATIONAL MATHEMATICAL SUMMER CENTER

Authors

Angelia Nedić
School of Electrical, Computer and Energy Engineering
Arizona State University
Tempe, AZ, USA

Jong-Shi Pang
The Daniel J. Epstein Department
of Industrial and Systems Engineering
University of Southern California
Los Angeles, CA, USA

Gesualdo Scutari
School of Industrial Engineering
Purdue University
West Lafayette, IN, USA

Ying Sun
School of Industrial Engineering
Purdue University
West Lafayette, IN, USA

Editors

Francisco Facchinei
DIAG
Università di Roma La Sapienza
Rome, Italy

Jong-Shi Pang
The Daniel J. Epstein Department
of Industrial and Systems Engineering
University of Southern California
Los Angeles, CA, USA

ISSN 0075-8434 ISSN 1617-9692 (electronic)
Lecture Notes in Mathematics
C.I.M.E. Foundation Subseries
ISBN 978-3-319-97141-4 ISBN 978-3-319-97142-1 (eBook)
https://doi.org/10.1007/978-3-319-97142-1

Library of Congress Control Number: 2018960254

Mathematics Subject Classification (2010): Primary 2010 - 90; Secondary 2010 - 49

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume collects lecture notes that stem from courses taught at CIME Summer School in Applied Mathematics "Centralized and Distributed Multi-agent Optimization: Models and Algorithms," held at the Hotel San Michele, Cetraro, Italy, from June 23 to June 28, 2014.

Multi-agent optimization is at the forefront of modern optimization theory and has recently undergone a dramatic development, stimulated by new applications in a host of diverse disciplines. Multi-agent optimization, including new modeling paradigms and both centralized and distributed solution algorithms, lies at the intersection of classical optimization, game theory, and variational inequality theory. As the area has undergone such an explosive growth in recent years, it seemed timely and appropriate to provide an overview that described ongoing research and important trends in detail, with an emphasis on mathematical problems arising in telecommunications, a field that provides many challenging and stimulating problems.

The lectures were delivered by world-leading experts, and they all were real models of clarity, capable of getting students to the heart of current research and of generating genuine interest. There were 61 students, with the majority coming from European countries, but with a substantial number of students arriving from non-European countries as diverse as Japan, India, Indonesia, Pakistan, Congo, Brazil, Iran, Lebanon, and Turkey. Since the school covered both theoretical and applicative topics, it attracted both mathematicians and (mathematically oriented) engineers, thus creating a vibrant and stimulating environment. It is interesting to remark that the school saw the participation not only of PhD students and young researchers, but also of a few more senior and well-established researchers. The lectures were well organized and integrated, so that participants in the school could gather a clear, multifaceted view of cutting-edge topics in multi-agent optimization. Particular attention was devoted to illustrate both theoretical issues and practical applications. There was time for extra discussions, and students took advantage of this opportunity and spent time with both the lecturers and other fellow students, establishing contacts that led to fruitful collaborations.

The plan of the lectures was as follows:

- **Differential Variational Inequalities**: Jong-Shi Pang, Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, California, USA.
- **Distributed Optimization over Networks**: Angelia Nedić, School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, Arizona, USA.
- **Advanced Decomposition Algorithms for Multi-agent Systems**: Gesualdo Scutari, School of Industrial Engineering, Purdue University, West Lafayette, Indiana, USA.
- **Optimization Methods for Resource Management: Complexity, Duality and Approximation**: Zhi-Quan Luo, School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China.

This volume contains chapters relative to the first three series of lectures listed above, and hopefully, it will help bring the themes discussed at the school, along with the lecturers' authoritative viewpoint on the subject, to the attention of a wider audience. These lecture notes are precious since they provide a systematic introduction to exciting research topics that, for the most part, have not yet found their way into books or surveys. Therefore, they provide the reader with a unique opportunity to get a concise and clear overview of new research areas. Indeed, some of the lectures really amount to small treatises that, we are sure, will be widely used for teaching and self-study for many years to come.

Such a successful school would not have been possible without the help of many people. We would like to express here our warmest gratitude and sincere appreciation to the CIME Foundation, and in particular to the Director, Professor Pietro Zecca; to the Scientific Secretary, Professor Elvira Mascolo; to the Board Secretary, Professor Paolo Salani; to Professor Fabio Schoen (roles at the time of the school); and to all the CIME staff for their invaluable help, support, and patience. We are also delighted to acknowledge the prestigious sponsorship of the European Mathematical Society that gave financial support for the participation of selected students. Finally, our greatest thanks go to the four lecturers, who found time in their busy schedules to come and teach such effective courses at the school, and to all students, whose enthusiasm and lively participation made the school really memorable.

Rome, Italy
Los Angeles, CA, USA

Francisco Facchinei
Jong-Shi Pang

Contents

Chapter 1
Distributed Optimization Over Networks

Angelia Nedić

Abstract The advances in wired and wireless technology necessitated the development of theory, models and tools to cope with new challenges posed by large-scale optimization problems over networks. The classical optimization methodology works under the premise that all problem data is available to some central entity (computing agent/node). This premise does not apply to large networked systems where typically each agent (node) in the network has access only to its private local information and has a local view of the network structure. This chapter will cover the development of such distributed computational models for time-varying networks, both deterministic and stochastic, which arise due to the use of different synchronous and asynchronous communication protocols in ad-hoc wireless networks. For each of these network dynamics, distributed algorithms for convex constrained minimization will be considered. In order to emphasize the role of the network structure in these approaches, our main focus will be on direct primal (sub)-gradient methods. The development of these methods combines optimization techniques with graph theory and the non-negative matrix theory, which model the network aspect. The lectures will provide some basic background theory on graphs, graph Laplacians and their properties, and the convergence results for related stochastic matrix sequences. Using the graph models and optimization techniques, the convergence and convergence rate analysis of the methods will be presented. The convergence rate results will demonstrate the dependence of the methods' performance on the problem and the network properties, such as the network capability to diffuse the information.

1.1 Introduction

Recent advances in wired and wireless technology have lead to the emergence of large-scale networks such as Internet, mobile ad-hoc networks, and wireless sensor networks. Their emergence gave rise to new network application domains ranging

A. Nedić (✉)
Arizona State University, Tempe, AZ, USA
e-mail: Angelia.Nedich@asu.edu

F. Facchinei, J.-S. Pang (eds.), *Multi-agent Optimization*, Lecture Notes
in Mathematics 2224, https://doi.org/10.1007/978-3-319-97142-1_1

from data-base networks, social and economic networks to decentralized in-network operations including resource allocation, coordination, learning, and estimation.

As a result, there is a necessity to develop new models and tools for the design and performance analysis of such large complex networked systems. The problems arising in such networks stem mainly from two aspects, namely, *a lack of central authority or a coordinator* (a master node), and an inherent *dynamic of the network connectivity structure*. The lack of central authority in a network system naturally requires decentralized architecture for operations over the network (such as in the case of Internet). In some applications, the decentralized architecture is often preferred over a centralized architecture due to several reasons: (1) the size of the network (the number of agents) and the resources needed to coordinate (i.e., communicate) with a large number of agents; (2) a centralized network architecture is not desirable since it is not robust to the failure of the central entity; and (3) the privacy of agent information often cannot be preserved in a centralized systems. Furthermore, additional challenges in decentralized operations over such networks are encountered from the network connectivity structure that can vary over time due to unreliable communication links or mobility of the network agents.

The challenge is to control, coordinate, and analyze the performance of such networks. As a particular goal, one would like to develop distributed optimization algorithms that can be deployed in such networks that do not have a central coordinator, but exploit the network connectivity to achieve a global network performance objective. Thus, it is desirable that such algorithms are:

- Locally distributed in the sense that they rely on local information and observations only, i.e., the agents can exchange some limited information with their one-hop neighbors only;
- Robust against changes in the network topology (since the topology is not necessarily static as the communication links may not function perfectly);
- Easily implementable in the sense that the local computations performed by the agents are not expensive.

We next provide some examples of large scale networks and applications that arise within such networks.

Example 1 (Sensor Networks) A new computing concept based on a system of small sensors also referred to as motes or smart dust sensors, see Fig. 1.1. The sensors are of small size and have some computational, sensing and communication capabilities. They can be used in many different ways, such as for example, they may be mixed into concrete in order to monitor the structural health of buildings and bridges (smart structures), or be placed on power grids to monitor the power load (smart grids).

A specific problem of interest that supports a number of applications in sensor networks, such as building a piece-wise approximation of the coverage area, multi-sensor target localization and tracking problems, is the determination of Voronoi cells. A Voronoi cell of a sensor in a network is the locus of points in a sensor field that are the closest to a given sensor among all other sensors [6]. Upon determining

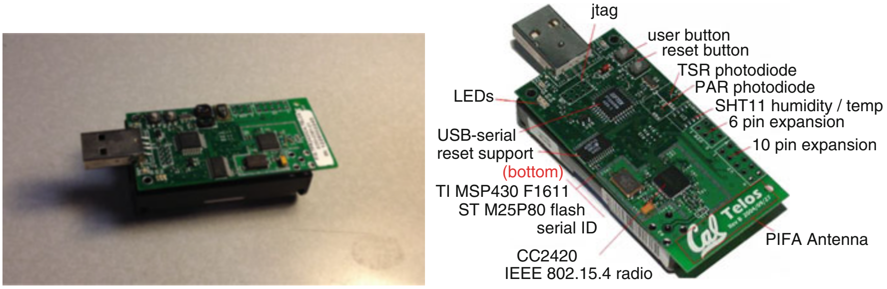


Fig. 1.1 A mote and its functionalities

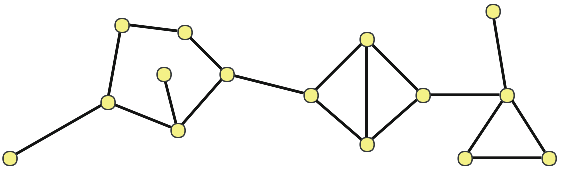


Fig. 1.2 A peer-to-peer network

such a partition in a distributed fashion, each sensor acts as a representative for the points in its cell. □

Example 2 (Computing Aggregates in Peer-to-Peer (P2P) Networks) In a P2P network consisting of m nodes, each node i has its local data/files stored with average size θ_i, which is known to node i only. The nodes are connected over a static undirected network (see Fig. 1.2), and they want to jointly compute the average file size $\frac{1}{m}\sum_{i=1}^{m}\theta_i$ without a central coordinator. In control theory and game theory literature, the problem is known as the *agreement or consensus problem* [15, 34, 60, 67, 171].

The problem has an optimization formulation, as follows:

$$\min_{x\in\mathbb{R}} \sum_{i=1}^{m}(x-\theta_i)^2.$$

It is a convex unconstrained problem with a strongly convex objective. Its unique solution θ^* is the average of the values θ_i, i.e., $\theta^* = \frac{1}{m}\sum_{i=1}^{m}\theta_i$. The solution cannot easily be computed when the agents have to calculate it in a distributed fashion by communicating only locally. In this case, the agents need to agree on the average of the values they hold.

In a more general variant of the consensus problem, the agents want to agree on some common value, which need not be the average of the values they initially have. For example, in a problem of leaderless heading alignment, autonomous agents move in a two-dimensional plane region with the same speed but different headings (they are refracted from the boundary to prevent them from leaving the area) [60, 174]. The objective is to design a local protocol that will ensure the alignment of the agent headings, while the agents communications are constrained by a given maximal distance. □

Another motivating example for distributed optimization over networks is a special machine learning problem, known as Support Vector Machine or maximum margin classifier. We discuss this problem in a centralized setting and then, we will see how it naturally fits in a distributed setting in the situations when the privacy of the data is of concern or when the data is too large to be shared.

Example 3 (Support Vector Machine (SVM)) We are given a data set $\{z_j, y_j\}_{j=1}^d$ consisting of d points, where $z_j \in \mathbb{R}^n$ is a measurement vector and $y_j \in \{+1, -1\}$ is its label. Assuming that the data can be perfectly separated, the problem consists of determining a hyperplane that separates data the best, i.e., solving the following convex problem:

$$\min_{x\in\mathbb{R}^n} F(x), \quad \text{where} \quad F(x) = \frac{\rho}{2}\|x\|^2 + \sum_{j=1}^{d} \max\left\{0, 1 - y_j\langle x, z_j\rangle\right\},$$

where $\rho > 0$ is a regularizing parameter that indicates the importance of having a small-norm solution. Given that the objective function is strongly convex, a solution exists and it is unique (see Fig. 1.3 for an illustration). The problem can be solved by using a subgradient method. If the data is distributed across several data centers, say m centers, then the joint problem can be written as:

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{m} \left(\frac{\rho}{2m}\|x\|^2 + \sum_{j\in D_i} \max\{0, 1 - y_j\langle x, z_j\rangle\} \right),$$

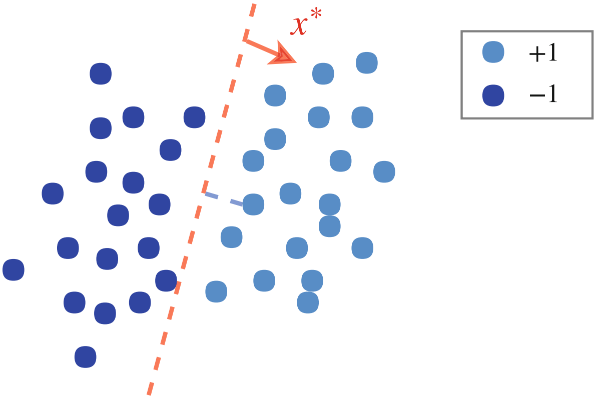


Fig. 1.3 A maximum margin separating hyperplane for a single data center

where D_i is the collection of data points at the center i. Letting

$$f_i(x) = \frac{\rho}{2m}\|x\|^2 + \sum_{j\in D_i} \max\{0, 1 - y_j\langle x, z_j\rangle\},$$

we see that the distributed variant of the problem assumes the following form:

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

where the function f_i is known to center i. In this setting, sharing the function f_i with any other center amounts to sharing the entire data collection D_i available at center i. When the data is private or the data sets are too large, sharing the data is not an option and the problem has to be solved in a distributed manner. □

Many more examples of distributed problems and the use of consensus can be found in the domains of bio-inspired systems, self-organized systems, social networks, opinion dynamics, and autonomous (robotic) systems. For such examples, a reader may refer to some recent books and monographs on robotic networks [18, 94, 96], and social and economic networks [42, 49, 59]. These examples can also be found in related thesis works including [12, 52, 120, 155] dealing with averaging dynamics and [47, 69, 73, 130, 144, 183] dealing with distributed optimization aspects.

In the sequel, we will often refer to networks as graphs, and we will use "agent" and "node" interchangeably. The rest of the chapter is organized as follows: in Sect. 1.2, we formally describe a multi-agent problem in a network and discuss some related aspects of the consensus protocol. Section 1.3 presents a distributed synchronous algorithm for solving the multi-agent problem in time-varying undirected graphs, while Sect. 1.4 deals with asynchronous implementations over a static undirected graph. Section 1.5 concludes this chapter by providing an overview of related literature including the most recent research directions.

1.2 Distributed Multi-Agent Problem

This section provides a formal multi-agent system problem description, introduces our basic notation and gives the underlying assumptions on the multi-agent problem. The agents are embedded in a communication graph which accommodates distributed computations through the use of consensus protocols. A basic consensus protocol for undirected time-varying graphs is presented, and its convergence result is provided for later use.

1.2.1 Problem and Assumptions

Throughout this chapter, we will be focused on solving distributed problems of the generic form

$$\min_{x \in X} f(x) \qquad \text{with} \quad f(x) = \sum_{i=1}^{m} f_i(x), \tag{1.1}$$

in a network of m agents, where each function f_i is known to agent i only while the constraint set $X \subseteq \mathbb{R}^n$ is known to all agents. We will assume that the problem (1.1) is convex.

Assumption 1 *The set $X \subseteq \mathbb{R}^n$ is closed and convex, and each function $f_i : \mathbb{R}^n \to \mathbb{R}$ is convex.*

We will explicitly state when we assume that problem (1.1) has a solution. In such cases, we will let f^* denote the optimal value of the problem and X^* denote the set of its solutions,

$$f^* = \min_{x \in X} f(x), \qquad X^* = \{x^* \in X \mid f(x^*) = f^*\}.$$

Throughout the chapter, we will work with the Euclidean norm, denoted by $\|\cdot\|$, unless otherwise explicitly stated. We use $\langle \cdot, \cdot \rangle$ to denote the inner product. We will view all vectors as column vectors, unless stated otherwise. We will use the prime to denote the transpose of a matrix and a vector.

We assume that the agents are embedded in a communication network, which allows the agents to exchange some limited information with their immediate (one-hop) neighbors. Multi-hop communications are not allowed in this setting. The agents' common goal is to solve the problem (1.1) collaboratively.

The communication network structure over time is captured with a sequence of time-varying undirected graphs. More specifically, we assume that the agents exchange their information (and perform some updates) at given discrete time instances, which are indexed by $k = 0, 1, 2, \ldots$. The communication network structure at time k is represented by an undirected graph $G_k = ([m], E_k)$, where $[m]$ is the agent (node) set, i.e., $[m] = \{1, \ldots, m\}$, while E_k is the set of edges. The edge $i \leftrightarrow j \in E_k$ indicates that agents i and j can communicate (send and receive messages) at time k.

Given a graph G_k at a time k, we let $N_i(k)$ denote the set of neighbors of agent i, at time k:

$$N_i(k) = \{j \in [m] \mid i \leftrightarrow j \in E_k\} \cup \{i\}.$$

Note that the neighbor set $N_i(k)$ includes agent i itself, which reflects the fact that agent i has access to some information from its one-hop neighbors and its own information.

The agents' desire to solve the problem (1.1) jointly through local communications translates to the following problem that agents are facing at time k:

$$\begin{aligned} \min \quad & \mathbf{f}(x_1,\ldots,x_m) \quad \text{with} \quad \mathbf{f}(x_1,\ldots,x_m) = \sum_{i=1}^{m} f_i(x_i) \\ \text{subject to} \quad & x_i = x_j \text{ for all } j \in N_i(k) \text{ and all } i \in [m], \\ & x_i \in X \text{ for all } i \in [m]. \end{aligned} \tag{1.2}$$

Thus, the agents are facing a sequence of optimization problems with time-varying constraints, which are capturing the time-varying structure of the underlying communication network. Since this is a nonstandard optimization problem, we need to specify what it means to solve the problem. To do so, we will impose some additional assumptions on the graphs G_k.

Throughout, we will assume that the graphs G_k are connected.

Assumption 2 *Each graph G_k is connected.*

This assumption can be relaxed to the requirement that the union of B consecutive graphs $G_k, \ldots, G_{k+B-1}$ is connected for all $k \geq 0$ and for some positive integer B. However, to keep the exposition simple, we will adopt Assumption 2.

Let C_k be the constraint set of problem (1.2) at time k, i.e.,

$$C_k = \{(x_1,\ldots,x_m) \in X^m \mid x_i = x_j \text{ for all } j \in N_i(k) \text{ and all } i \in [m]\}.$$

Under Assumption 2, the constraint sets C_k are all the same. Their description is given to the agents through a different set of equations at different time instances, as seen from the following lemma.

Lemma 1 *Let Assumption 2 hold. Then, for each k, we have*

$$C_k = \{(x_1,\ldots,x_m) \mid x_i = x \textit{ for some } x \in X \textit{ and all } i \in [m]\}.$$

The proof of Lemma 1 is straightforward and it is omitted. In fact, it can be seen that Lemma 1 also holds when the graphs G_k are directed and each of the graphs contains a directed rooted spanning tree,[1] where the neighbor set $N_i(k)$ is replaced with the in-neighbor set[2] $N_i^{\text{in}}(k)$ of agent i at time k.

[1] There exists a node i such that the graph contains a directed path from node i to any other node in the network.

[2] The set $N_i^{\text{in}}(k)$ of in-neighbors of agent i in a directed graph G_k is the set of all agents j such that the directed edge (j, i) exists in the graph.

In view of Lemma 1, it is now obvious that we can associate a limit problem with the sequence of problems (1.2), where the limit problem is given by:

$$\begin{array}{lll} \min & \mathbf{f}(x_1,\ldots,x_m) \quad \text{with} \quad \mathbf{f}(x_1,\ldots,x_m) = \sum_{i=1}^{m} f_i(x_i) \\ \text{subject to} & (x_1,\ldots,x_m) \in \cap_{k=1}^{\infty} C_k. \end{array} \tag{1.3}$$

As we noted, all sets C_k are the same under Assumption 2. However, we will keep the notation C_k to capture the fact that the agents have a different set of equations that describe the constraint set at different times. Furthermore, the preceding formulation of the limit problem is also suitable for the situations where the graphs G_k are not necessarily connected.

1.2.2 Consensus Problem and Algorithm

The consensus problem is a special case of the limit problem (1.3), where each $f_i \equiv 0$ and $X = \mathbb{R}^n$, i.e., the consensus problem is given by

$$\begin{array}{ll} \min & 0 \\ \text{subject to} & (x_1,\ldots,x_m) \in \cap_{k=1}^{\infty} C_k, \end{array} \tag{1.4}$$

with

$$C_k = \{(x_1,\ldots,x_m) \mid x_i = x \text{ for some } x \in \mathbb{R}^n \text{ and all } i \in [m]\} \qquad \text{for all } k \geq 1.$$

As one may observe, the consensus problem is a feasibility problem where the agents need to collectively determine an $\mathbf{x} = (x_1,\ldots,x_m)$ satisfying the constraint in (1.4), while obeying the communication structure imposed by graph G_k at each time k.

A possible way to solve the consensus problem is that each agent considers its own problem, at time k, of the following form:

$$\min_{x\in\mathbb{R}^n} \sum_{j\in N_i(k)} p_{ij}(k)\|x - x_j\|^2,$$

where $p_{ij}(k) > 0$ for all $j \in N_i(k)$ and for all $i \in [m]$. The values x_j are assumed to be communicated to agent i by its neighbors $j \in N_i(k)$. This problem can be viewed as a penalty problem associated with the constraints in the set C_k that involve agent i decision variable. The objective function is strongly convex and it has a unique solution, denoted by $\hat{x}_i$, i.e.,

$$\hat{x}_i(k) = \operatorname*{argmin}_{x\in\mathbb{R}^n} \sum_{j\in N_i(k)} p_{ij}(k)\|x - x_j\|^2.$$

In the following lemma, we provide the closed form of the solution $\hat{x}_i(k)$.

Lemma 2 *Let $X = \mathbb{R}^n$, and consider the feasible set*

$$C_{ik} = \{(x_j)_{j\in N_i(k)} \mid x_j = x \textit{ for all } j \in N_i(k) \textit{ and } x \in \mathbb{R}^n\} \tag{1.5}$$

corresponding to the constraints in C_k that involve agent i at time k. Then, the solution $\hat{x}_i(k)$ of the penalty problem $\min_{x\in\mathbb{R}^n} \sum_{j\in N_i(k)} p_{ij}(k)\|x - x_j\|^2$ *associated with the feasible set C_{ik} is given by*

$$\hat{x}_i(k) = \frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}.$$

Proof We note that

$$\begin{aligned}
&\sum_{j\in N_i(k)} p_{ij}(k)\|x - x_j\|^2 \\
&\quad= \sum_{j\in N_i(k)} p_{ij}(k)\|x\|^2 - 2\left\langle x, \sum_{j\in N_i(k)} p_{ij}(k)x_j \right\rangle + \sum_{j\in N_i(k)} p_{ij}(k)\|x_j\|^2 \\
&\quad= \sum_{j\in N_i(k)} p_{ij}(k)\left[\|x\|^2 - 2\left\langle x, \frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)} \right\rangle + \frac{\sum_{j\in N_i(k)} p_{ij}(k)\|x_j\|^2}{\sum_{j\in N_i(k)} p_{ij}(k)}\right] \\
&\quad= \sum_{j\in N_i(k)} p_{ij}(k)\left[\|x\|^2 - 2\left\langle x, \frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)} \right\rangle + \left\|\frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}\right\|^2 \right. \\
&\qquad\left. - \left\|\frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}\right\|^2 + \frac{\sum_{j\in N_i(k)} p_{ij}(k)\|x_j\|^2}{\sum_{j\in N_i(k)} p_{ij}(k)}\right].
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{j\in N_i(k)} p_{ij}(k)\|x - x_j\|^2 = \sum_{j\in N_i(k)} p_{ij}(k)&\left[\left\|x - \frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}\right\|^2 \right. \\
&\left. - \left\|\frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}\right\|^2 + \frac{\sum_{j\in N_i(k)} p_{ij}(k)\|x_j\|^2}{\sum_{j\in N_i(k)} p_{ij}(k)}\right].
\end{aligned}$$

Since the last two terms in the preceding sum do not depend on x, we see that

$$\begin{aligned}\hat{x}_i(k) &= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\left\{\sum_{j\in N_i(k)} p_{ij}(k)\|x-x_j\|^2\right\}\\ &= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\left\{\left(\sum_{j\in N_i(k)} p_{ij}(k)\right)\left\|x-\frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}\right\|^2\right\}.\end{aligned}$$

Furthermore, since $\sum_{j\in N_i(k)} p_{ij}(k) > 0$, we finally have

$$\hat{x}_i(k) = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\left\{\left\|x-\frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}\right\|^2\right\} = \frac{\sum_{j\in N_i(k)} p_{ij}(k)x_j}{\sum_{j\in N_i(k)} p_{ij}(k)}.$$

□

In view of Lemma 2, the penalty problem associated with agent i feasible set C_{ik} at time k can be equivalently be given by

$$\min_{x\in\mathbb{R}^n}\sum_{j\in N_i(k)} w_{ij}(k)\|x-x_j\|^2,$$

where the weights $w_{ij}(k)$, $j\in N_i(k)$, correspond to convex combinations, i.e.,

$$w_{ij}(k) > 0 \quad \text{for all } j\in N_i(k), \qquad \sum_{j\in N_i(k)} w_{ij}(k) = 1. \tag{1.6}$$

Obviously, for the equivalence of the two penalty problems, we need $w_{ij}(k) = p_{ij}(k)/\sum_{j\in N_i(k)} p_{ij}(k)$. In this case, the corresponding solution $\hat{x}_i(k)$ is given by

$$\hat{x}_i(k) = \sum_{j\in N_i(k)} w_{ij}(k)x_j.$$

The preceding discussion motivates the following algorithm, known as a *consensus algorithm* (with projections), for solving the constrained consensus problem (1.4): each agent has a variable $x_i(k)$ at time k. At time $k+1$, every agent i sends $x_i(k)$ to its neighboring agents $j\in N_i(k)$ and receives $x_j(k)$ from them. Then, every agent i updates its variable as follows:

$$x_i(k+1) = \sum_{j\in N_i(k)} w_{ij}(k)x_j(k),$$

where $w_{ij}(k) > 0$ for all $j \in N_i(k)$ and all $i \in [m]$, and $\sum_{j \in N_i(k)} w_{ij}(k) = 1$ for all $i \in [m]$. For a more compact representation, we define $w_{ij}(k) = 0$ for all $j \notin N_i(k)$ and all $i \in [m]$, so we have

$$x_i(k+1) = \sum_{j=1}^{m} w_{ij}(k) x_j(k) \qquad \text{for all } i \in [m] \text{ and all } k \geq 0. \tag{1.7}$$

The initial points $x_i(0) \in \mathbb{R}^n$, $i \in [m]$, are assumed to be arbitrary.

We note here that if a (convex) constraint set $X \subseteq \mathbb{R}^n$ is known to all agents, then the constrained consensus problem (1.4) can also be solved by the consensus algorithm in (1.7) with an adjustment of the initial selections $x_i(0)$ to satisfy $x_i(0) \in X$ for all i. This can be seen by noting that $x_i(k+1)$ is a convex combination of $x_j(k)$ for $j \in N_i(k)$ (see (1.6)), and it will lie in the set X as long as this set is convex and $x_j(k) \in X$ for all $j \in N_i(k)$.

The consensus algorithm in (1.7) has regained interest since the recent work [60], which attracted a significant attention to the consensus problem in various settings (for an overview of the consensus related literature see Sect. 1.5).

For the convergence of the consensus algorithm, some additional assumptions are typically needed for the weights $w_{ij}(k)$ aside from the "convex combination" requirement captured by relation (1.6). To state one such assumption, we will introduce some additional terminology and notation. We let $W(k)$ be the matrix with ijth entry equal to $w_{ij}(k)$. We will say that a matrix W is (row) *stochastic* if its entries are non-negative and the sum of its entries in each row is equal to 1. We will say that W is *doubly stochastic* if both W and its transpose W' are stochastic matrices.

Next, we state an assumption on the matrices $W(k)$ that we will use later on.

Assumption 3 *For every $k \geq 0$, the matrix $W(k)$ has the following properties:*

(a) $W(k)$ is doubly stochastic.

(b) $W(k)$ is compatible with the structure of the graph G_k, i.e.,

$$w_{ij}(k) = 0 \qquad iff \qquad i \leftrightarrow j \notin E_k.$$

(c) $W(k)$ has positive diagonal entries, i.e., $w_{ii}(k) > 0$ for all $i \in [m]$.

(d) There is an $\eta > 0$ such that

$$w_{ij}(k) \geq \eta \qquad iff \qquad i \leftrightarrow j \in E_k.$$

First, let us note that Assumption 3 is much stronger than what is typically assumed to guarantee the convergence of the consensus algorithm. In general, the graph G_k can be directed and the positive weights $w_{ij}(k)$ are assumed for the directed links $(j, i) \in E_k$, while the matrix $W(k)$ is assumed to be just (row) stochastic. We work with a stronger assumption since we want to address the optimization problem (1.3) which has a more general objective function than that of the consensus problem (1.4).

To provide insights into what motivates Assumption 3, consider the consensus algorithm in the case of an unconstrained scalar problem, i.e., $X = \mathbb{R}$. Then, for the consensus algorithm in (1.7), by stacking all the variables $x_i(k)$ into a single vector $x(k)$ at time k, we have

$$x(k+1) = W(k)x(k) = \cdots = W(k)W(k-1)\cdots W(0)x(0).$$

Furthermore, in the case of static graphs G_k, i.e., $G_k = G$ for some graph G, we can use $W(k) = W$ for all k, thus implying that

$$x(k) = W^k x(0).$$

When W is a stochastic matrix which is compatible with a connected graph G, then W is irreducible and, by Perron-Frobenius Theorem (see Theorem 4.2.1, page 101 of [46]), the spectral radius $\rho(W)$ (which is equal to 1 in this case) is a simple positive eigenvalue and the vector $\mathbf{1}$ with all entries equal to 1 is the unique right-eigenvector associated with eigenvalue 1, i.e.,

$$W\mathbf{1} = \mathbf{1}.$$

When, in addition, W has positive diagonal entries, then W is also primitive, so we have

$$\lim_{k\to\infty} W^k = \mathbf{1}v',$$

where v is the normalized (unique positive) left-eigenvector of W associated with eigenvalue 1, i.e., a unique vector satisfying

$$v'W = v' \qquad \text{where } v_i > 0 \text{ for all } i \quad \text{and} \quad \langle v, \mathbf{1}\rangle = 1$$

(see Theorem 4.3.1, page 106, and Theorem 4.4.4, page 119, both in [46]). Thus, when W is stochastic, compatible with a connected graph G, and has a positive diagonal, we obtain

$$\lim_{k\to\infty} x(k) = \left(\lim_{k\to\infty} W^k\right) x(0) = \mathbf{1}v'x(0) = \langle v, x(0)\rangle\,\mathbf{1}.$$

Hence, in this case, the consensus is reached, i.e., the iterates of the consensus algorithm converge to the value $\langle v, x(0)\rangle$, which is a convex combination of the initial agents' values $x_i(0)$. Observe that the behavior of the iterates in the limit, as k increases, is completely determined by the limit behavior of W^k as $k \to \infty$.

In the light of the preceding discussion, Assumption 3 guarantees that a similar behavior is exhibited in the case when the matrices are time-varying and the graphs G_k are connected. Specifically, in this case, we would like to have

$$\lim_{k\to\infty}[W(k)W(k-1)\cdots W(0)] = \mathbf{1}v'$$

for some vector v with all entries v_i positive and $\langle v, \mathbf{1}\rangle = 1$. This relation is guaranteed by Assumptions 2 and 3. In fact, under these assumptions we have a stronger result for the matrix sequence $\{W(k)\}$, as follows:

$$\lim_{k\to\infty}[W(k)W(k-1)\cdots W(s)] = \frac{1}{m}\mathbf{1}\mathbf{1}' \qquad \text{for all } s \geq 0.$$

This result is formalized in the following lemma, which also provides the rate of convergence for the matrix products $W(k)W(k-1)\cdots W(s)$ for all $k \geq s \geq 0$.

Lemma 3 (Lemma 5 in [110]) *Let the graph sequence $\{G_k\}$ satisfy Assumption 2, and let the matrix sequence $\{W(k)\}$ satisfy Assumption 3. Then, we have for all $s \geq 0$ and $k \geq s$,*

$$\sup_{x\in\mathbb{R}^n,\|x\|=1}\left\|\left(W(k)W(k-1)\cdots W(s+1)W(s) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)x\right\|^2 \leq \left(1-\frac{\eta}{2m^2}\right)^{k-s}.$$

In particular, for all $s \geq 0$ and $k \geq s$, and for all $i, j \in [m]$,

$$\left([W(k)W(k-1)\cdots W(s+1)W(s)]_{ij} - \frac{1}{m}\right)^2 \leq \left(1-\frac{\eta}{2m^2}\right)^{k-s}.$$

The first relation in Lemma 3 is a consequence of Lemma 5 in [110]. The second relation follows by letting x be any of the unit-vectors of the standard basis in $\mathbb{R}^n$.

Lemma 3 provides a key insight into the behavior of the products of the matrices $W(k)$, which implies that the consensus method in (1.7) converges geometrically to $\frac{1}{m}\sum_{i=1}^m x_i(0)$. We will use this lemma to show that consensus-based methods for solving a more general optimization problem (1.1) converge to a solution, as discussed in the next section.

1.3 Distributed Synchronous Algorithms for Time-Varying Undirected Graphs

We now consider a distributed algorithm for solving problem (1.3). We assume that the set X is closed and convex, and it has a simple structure so that the projection of a point on the set X is not computationally expensive. The idea is to construct an algorithm to be executed locally by each agent i that at every instant k involves two steps: one step aimed at satisfying agent i feasibility constraint C_{ik} in (1.5), and the other step aimed at minimizing its objective cost f_i over the set X. Thus, the first step is akin to consensus update in (1.7), while the second step is a simple projection-based (sub)gradient update using f_i.

To illustrate the idea, consider agent i and its surrogate objective function at time k:

$$F_{ik}(x) = f_i(x) + \delta_X(x) + \frac{1}{2}\sum_{j\in N_i(k)} w_{ij}(k)\|x - x_j\|^2,$$

where $\delta_X(x)$ is the indicator function of the set X, i.e.,

$$\delta_X(x) = \begin{cases} 0 & \text{if } x \in X, \\ +\infty & \text{otherwise.} \end{cases}$$

The weights $w_{ij}(k)$, $j \in N_i(k)$, are convex combinations (i.e., they are positive and they sum to 1; see (1.6)).

Having the vectors x_j, $j \in N_i(k)$, agent i may take the first step aimed at minimizing $\frac{1}{2}\sum_{j\in N_i(k)} w_{ij}(k)\|x - x_j\|^2$, which would result in setting

$$\hat{x}_i(k) = \sum_{j\in N_i(k)} w_{ij}(k)x_j.$$

In the second step, assuming for the moment that f_i is differentiable, agent i considers solving the problem

$$\min_{x\in\mathbb{R}^n}\left\{\langle\nabla f_i(\hat{x}_i(k)), x\rangle + \delta_X(x) + \frac{1}{2\alpha_k}\|x - \hat{x}_i(k)\|^2\right\},$$

which is equivalent to

$$\min_{x\in X}\left\{\langle\nabla f_i(\hat{x}_i(k)), x\rangle + \frac{1}{2\alpha_k}\|x - \hat{x}_i(k)\|^2\right\},$$

where $\alpha_k > 0$ is a stepsize. The preceding problem has a closed form solution given by

$$x_i^*(k) = \Pi_X[\hat{x}_i(k) - \alpha_k \nabla f(\hat{x}_i(k))],$$

where $\Pi_X[z]$ is the projection of a point z on the set X, i.e.,

$$\Pi_X[z] = \operatorname*{argmin}_{x\in X}\|x - z\|^2 \qquad \text{for all } z \in \mathbb{R}^n.$$

When the function f_i is not differentiable, we would replace the gradient $\nabla f(\hat{x}_i(k))$ with a subgradient $g_i(\hat{x}_i(k))$. Recall that a subgradient of a convex function $h : \mathbb{R}^n \to \mathbb{R}$ at a given point x is a vector $g(x) \in \mathbb{R}^n$ such that

$$h(x) + \langle g(x), y - x \rangle \leq h(y) \qquad \text{for all } y \in \mathbb{R}^n.$$

In what follows, we will use $g_i(k)$ to abbreviate the notation for a subgradient $g_i(\hat{x}_i(k))$ of the function $f_i(z)$ evaluated at $z = \hat{x}_i(k)$.

Now, based on the preceding discussion, we have the following algorithm: at every time k, each agent $i \in [m]$ maintains two vectors $y_i(k)$ and $x_i(k)$. The agent sends $x_i(k)$ to its neighbors $j \in N_i(k)$ and receives $x_j(k)$ from its neighbors $j \in N_i(k)$. Then, it updates as follows:

$$\begin{aligned} y_i(k+1) &= \sum_{j \in N_i(k)} w_{ij}(k) x_j(k), \\ x_i(k+1) &= \Pi_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)], \end{aligned} \tag{1.8}$$

where $\alpha_{k+1} > 0$ is a stepsize and $g_i(k+1)$ is a subgradient of $f_i(z)$ at point $z = y_i(k+1)$. The process is initialized with arbitrary points $x_i(0) \in X$ for all $i \in [m]$.

Note that the agents use the same stepsize value α_{k+1}. Note further that, due to the projection on the set X, we have $x_i(k) \in X$ for all i and k. Moreover, since $y_i(k+1)$ is a convex combination of points in X and since X is convex, we have $y_i(k+1) \in X$ for all i and k.

By introducing 0-weights for non-existing links in the graph G_k, i.e., by defining

$$w_{ij}(k) = 0 \qquad \text{when } j \notin N_i(k),$$

we can re-write (1.8) as follows: for all $k \geq 0$ and all $i \in [m]$,

$$\begin{aligned} y_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k) x_j(k), \\ x_i(k+1) &= \Pi_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)]. \end{aligned} \tag{1.9}$$

To illustrate the iterations of the algorithm in (1.9), consider a system of three agents in a connected graph, as illustrated in Fig. 1.4. Figure 1.4 shows a typical iteration of the algorithm. Since the graph is fully connected, all weights $w_{ij}(k)$ are positive, so the resulting points $y_i(k+1)$ lie inside the triangle formed by the points $x_i(k)$, $i = 1, 2, 3$. The new points $x_i(k+1)$, $i = 1, 2, 3$, obtained after the subgradient steps do not necessarily lie inside the triangle formed by the points $x_i(k)$, $i = 1, 2, 3$. Under some suitable assumptions on the stepsize and the subgradients, these triangles formed by $x_i(k)$, $i = 1, 2, 3$, as $k \to \infty$, can shrink into a single point, which is solution of the problem. Loosely speaking, while the consensus steps force the agents to agree on some point, the subgradient steps are

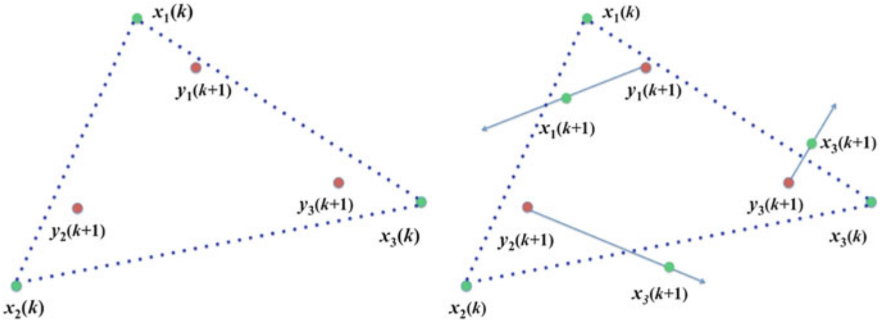


Fig. 1.4 At iteration k, agents hold values $x_i(k)$. The plot to the left illustrates the resulting points $y_i(k+1)$ of the iteration (1.9) which lie inside the triangle formed by the points $x_i(k)$, $i = 1, 2, 3$, (as all weights $w_{ij}(k)$ are positive in this case). The plot to the right depicts the iterates $x_i(k+1)$, $i = 1, 2, 3$, obtained through the subgradient steps of algorithm (1.9). These iterates do not necessarily lie inside the triangle formed by the prior iterates $x_i(k)$, $i = 1, 2, 3$

forcing the agreement point to be a solution of a given problem. Thus, one can think of the algorithm in (1.9) as a process that steers the consensus toward a particular region, in this case the region being the solution set of the agent optimization problem (1.3). To see this, note that from the definition of $x_i(k+1)$ we have

$$x_i(k+1) = y_i(k+1) - \alpha_{k+1} g_i(k+1) + \mathsf{e}_k,$$
$$\mathsf{e}_k = (\Pi_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)] - (y_i(k+1) - \alpha_{k+1} g_i(k+1)).$$

Assuming that the projection error e_k is small, and assuming that the functions are differentiable, we can approximate $x_i(k+1)$ as follows:

$$\begin{aligned} x_i(k+1) &\approx y_i(k+1) - \alpha_{k+1} \nabla f_i(y_i(k+1)) \\ &= \left(\sum_{j=1}^m w_{ij}(k) x_j(k) \right) - \alpha_{k+1} \nabla f_i \left(\sum_{j=1}^m w_{ij}(k) x_j(k) \right). \end{aligned} \tag{1.10}$$

Thus, the algorithm is similar to the consensus process $\sum_{j=1}^m w_{ij}(k)x_j(k)$ with an additional force coming from the gradient field, which steers the agreement point toward a solution of the problem $\min_{x \in X} \sum_{i=1}^m f_i(x)$. The preceding discussion sketches the approach that we will follow to establish the convergence properties of the method, which is the focus of the next section.

1.3.1 Convergence Analysis of Distributed Subgradient Method

In this section, we provide a main convergence result in Theorem 1 showing that the iterates $x_i(k)$, for all agents $i \in [m]$, converge to a solution of the problem (1.1), as $k \to \infty$. The proof of Theorem 1 relies on a basic relation satisfied by the algorithm in terms of all agents' iterates, as given in Proposition 1. The proof of this proposition is constructed through several auxiliary results that are provided in Lemmas 4–6. Specifically, Lemma 4 provides an elementary relation for the iterates $x_i(k)$ for a single agent, without the use of the network aspect. By viewing the algorithm (1.9) as a perturbation of the consensus algorithm, Lemma 5 establishes a relation for the distances between the iterates $x_i(k)$ and their averages taken across the agents (i.e., $\frac{1}{m}\sum_{j=1}^m x_j(k)$) in terms of the perturbation. The result of Lemma 5 is refined in Lemma 6 by taking into account that the perturbation to the consensus algorithm comes from a subgradient influence controlled by a stepsize choice.

Based on relation (1.10), we see that if $y_i(k+1)$ is close to the average of the points $x_j(k+1)$, $j \in [m]$, then for the iterate $x_i(k+1)$ we have

$$x_i(k+1) \approx x_{av}(k+1) - \alpha_{k+1}\nabla f_i(x_{av}(k+1)) + \epsilon_{k+1},$$

where $x_{av}(k+1) = \frac{1}{m}\sum_{j=1}^m x_j(k+1)$ and ϵ_{k+1} is an error due to using the gradient difference $\nabla f_i(x_{av}(k+1)) - \nabla f_i(y_i(k+1))$. When f_i is not differentiable, the iterates $x_i(k+1)$ would similarly correspond to an approximate subgradient update, where a subgradient $g_i(k+1)$ of $f_i(z)$ at $z = y_i(k+1)$ is used instead of a subgradient of $f_i(z)$ evaluated at $z = x_{av}(k+1)$ (which would have been used if the average $x_{av}(k+1)$ were available to all agents). Thus, the method (1.9) can be interpreted as an approximation of a centralized algorithm, where each agent would have access to the average vector $x_{av}(k+1)$ and could update by computing gradients of its own objective function f_i at the average $x_{av}(k+1)$.

1.3.1.1 Relation for a Single Agent Iterates

To start the analysis, for a single arbitrary agent, we will explore a basic relation for the distances between $x_i(k+1)$ and a point $x \in X$. In doing so, we will use the well-known property of the projection operator, namely

$$\|\Pi_X[z] - x\|^2 \le \|z - x\|^2 - \|\Pi_X[z] - z\|^2 \qquad \text{for all } x \in X \text{ and all } z \in \mathbb{R}^n. \tag{1.11}$$

The preceding projection relation follows from a more general relation which can be found in [44], in Volume II, 12.1.13 Lemma, on page 1120.

Lemma 4 *Let the problem be convex (Assumption 1 holds) and let* $\alpha_{k+1} > 0$*. Then, for the iterate* $x_i(k+1)$ *of the method* (1.9)*, we have for all* $x \in X$ *and all* $i \in [m]$*,*

$$\|x_i(k+1) - x\|^2 \le \|y_i(k+1) - x\|^2 - 2\alpha_{k+1}\left(f_i(y_i(k+1)) - f_i(x)\right) + \alpha_{k+1}^2\|g_i(k+1)\|^2.$$

Proof From the projection relation in (1.11) and the definition of $x_i(k+1)$ we obtain for any $x \in X$,

$$\begin{aligned}\|x_i(k+1) - x\|^2 \leq & \|y_i(k+1) - \alpha_{k+1} g_i(k+1) - x\|^2 \\ & - \|x_i(k+1) - y_i(k+1) + \alpha_{k+1} g_i(k+1)\|^2.\end{aligned}$$

By expanding the squared-norm terms, we further have

$$\begin{aligned}\|x_i(k+1) - x\|^2 \leq & \|y_i(k+1) - x\|^2 - 2\alpha_{k+1}\langle y_i(k+1) - x, g_i(k+1)\rangle \\ & + \alpha_{k+1}^2 \|g_i(k+1)\|^2 - \|x_i(k+1) - y_i(k+1)\|^2 \\ & - 2\alpha_{k+1}\langle x_i(k+1) - y_i(k+1), g_i(k+1)\rangle \\ & - \alpha_{k+1}^2 \|g_i(k+1)\|^2 \\ = & \|y_i(k+1) - x\|^2 - 2\alpha_{k+1}\langle y_i(k+1) - x, g_i(k+1)\rangle \\ & - \|x_i(k+1) - y_i(k+1)\|^2 \\ & - 2\alpha_{k+1}\langle x_i(k+1) - y_i(k+1), g_i(k+1)\rangle.\end{aligned}$$

Since $g_i(k+1)$ is a subgradient of f_i at $y_i(k+1)$, by convexity of f_i, we have

$$\langle y_i(k+1) - x, g_i(k+1)\rangle \geq f_i(y_i(k+1)) - f_i(x),$$

implying that

$$\begin{aligned}\|x_i(k+1) - x\|^2 \leq & \|y_i(k+1) - x\|^2 - 2\alpha_{k+1}\left(f_i(y_i(k+1)) - f_i(x)\right) \\ & - \|x_i(k+1) - y_i(k+1)\|^2 \\ & - 2\alpha_{k+1}\langle x_i(k+1) - y_i(k+1), g_i(k+1)\rangle.\end{aligned}$$

The last term in the preceding relation can be estimated by using Cauchy-Schwarz inequality, to obtain

$$\begin{aligned}& - 2\alpha_{k+1}\langle x_i(k+1) - y_i(k+1), g_i(k+1)\rangle \\ & \leq 2\|x_i(k+1) - y_i(k+1)\| \cdot \alpha_{k+1}\|g_i(k+1)\| \\ & \leq \|x_i(k+1) - y_i(k+1)\|^2 + \alpha_{k+1}^2\|g_i(k+1)\|^2.\end{aligned}$$

By combining the preceding two relations, we find that for any $x \in X$,

$$\begin{aligned}\|x_i(k+1) - x\|^2 \leq & \|y_i(k+1) - x\|^2 - 2\alpha_{k+1}\left(f_i(y_i(k+1)) - f_i(x)\right) \\ & + \alpha_{k+1}^2\|g_i(k+1)\|^2.\end{aligned}$$

□

1.3.1.2 Relation for Agents' Iterates and Their Averages Through Perturbed Consensus

We would like to estimate the difference between $x_i(k+1)$ and the average of these vectors, which can be then used in Lemma 4 to get some insights into the behavior of $\|x_i(k) - x^*\|$ for an optimal solution x^*. To do so, we will re-write the iterations of the method (1.9), as follows:

$$
\begin{aligned}
y_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k)x_j(k),\\
x_i(k+1) &= y_i(k+1) + \underbrace{(\Pi_X[y_i(k+1) - \alpha_{k+1}g_i(k+1)] - y_i(k+1))}_{\epsilon_i(k+1)}.
\end{aligned}
$$

Thus, we have for all i and $k \geq 0$,

$$
\begin{aligned}
x_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k)x_j(k) + \epsilon_i(k+1),\\
\epsilon_i(k+1) &= \Pi_X[y_i(k+1) - \alpha_{k+1}g_i(k+1)] - y_i(k+1),\\
y_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k)x_j(k).
\end{aligned}
\tag{1.12}
$$

In this representation, the iterates $x_i(k+1)$ can be viewed as obtained through a perturbed consensus algorithm, where $\epsilon_i(k+1)$ is a perturbation at agent i.

Under suitable conditions (cf. Assumption 3), by Lemma 3, we know that the matrix products $W(k)W(k-1)\cdots W(t)$ are converging as $k \to \infty$, for any t, to the matrix with all entries equal to $1/m$. We will use that result to establish a relation for the behavior of the iterates $x_i(k+1)$.

Lemma 5 *Let the graphs G_k satisfy Assumption 2 and the matrices $W(k)$ satisfy Assumption 3. Then, for the iterate process (1.12), we have for all $k \geq 0$,*

$$
\begin{aligned}
&\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2}\\
&\quad\leq mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m\left(\sum_{t=1}^{k} p^{k-t} \sqrt{\sum_{i=1}^{m} \|\epsilon_i(t)\|^2}\right)\\
&\qquad+\sqrt{m-1}\sqrt{\sum_{i=1}^{m} \|\epsilon_i(k+1)\|^2},
\end{aligned}
$$

where $x_{av}(k+1) = \frac{1}{m}\sum_{j=1}^{m} x_j(k+1)$, $p = 1 - \frac{\eta}{4m^2}$ *and* $\eta > 0$ *is a uniform lower bound on the entries of the matrices* $W(k)$ *(see Assumption 3(d)).*

Proof We write the evolution of the iterates $x_i(k+1)$ in (1.12) in a matrix representation. Letting $\ell \in [n]$ be any coordinate index, we can write for the ℓth coordinate (denoted by a superscript)

$$x_i^{\ell}(k+1) = \sum_{j=1}^{m} w_{ij}(k)x_j^{\ell}(k) + \epsilon_i^{\ell}(k+1) \qquad \text{for all } \ell \in [n].$$

Stacking all the ℓth coordinates in a column vector, denoted by $x^{\ell}(k+1)$, we have

$$x^{\ell}(k+1) = W(k)x_j^{\ell}(k) + \epsilon^{\ell}(k+1) \qquad \text{for all } \ell \in [n].$$

Next, we take the column vectors $x^{\ell}(k+1)$, $\ell \in [n]$, in a matrix $\mathbf{X}(k+1)$, for all k, and similarly, we construct the matrix $\mathbf{E}(k+1)$ from the perturbation vectors $\epsilon^{\ell}(k+1)$, $\ell \in [n]$. Thus, we have the following compact form representation for the evolution of the iterates $x_i(k+1)$:

$$\mathbf{X}(k+1) = W(k)\mathbf{X}(k) + \mathbf{E}(k+1) \qquad \text{for all } k \geq 0. \tag{1.13}$$

Using the recursion, from (1.13) we see that for all $k \geq 0$,

$$\begin{aligned}\mathbf{X}(k+1) &= W(k)\mathbf{X}(k) + \mathbf{E}(k+1)\\ &= W(k)W(k-1)\mathbf{X}(k-1) + W(k)\mathbf{E}(k) + \mathbf{E}(k+1)\\ &= \cdots\\ &= W(k:0)\mathbf{X}(0) + \left(\sum_{t=1}^{k} W(k:t)\mathbf{E}(t)\right) + \mathbf{E}(k+1),\end{aligned} \tag{1.14}$$

where

$$W(k:t) = W(k)W(k-1)\cdots W(t+1)W(t) \qquad \text{for all } k \geq t \geq 0.$$

By multiplying both sides of (1.14) with the matrix $\frac{1}{m}\mathbf{1}\mathbf{1}'$, we have

$$\begin{aligned}&\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\\ &\quad= \frac{1}{m}\mathbf{1}\mathbf{1}'W(k:0)\mathbf{X}(0) + \left(\sum_{t=1}^{k}\frac{1}{m}\mathbf{1}\mathbf{1}'W(k:t)\mathbf{E}(t)\right) + \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{E}(k+1)\\ &\quad= \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(0) + \left(\sum_{t=1}^{k}\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{E}(t)\right) + \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{E}(k+1),\end{aligned}$$

where the last equality follows from the fact that the matrices $W(k : t)$ are column-stochastic, as inherited from the matrices $W(k)$ being column-stochastic. By subtracting the preceding relation from (1.14), we obtain

$$\begin{aligned}
&\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1) \\
&\quad = \left(W(k:0) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{X}(0) + \sum_{t=1}^{k}\left(W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{E}(t) \\
&\qquad + \left(I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{E}(k+1), \qquad (1.15)
\end{aligned}$$

where I is the identity matrix. Let $\|A\|_F$ denote the Frobenius norm of an $m \times n$ matrix A, i.e.,

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}.$$

By taking the Frobenius norm of both sides in (1.15), we further obtain

$$\begin{aligned}
\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F &\le \left\|\left(W(k:0) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{X}(0)\right\|_F \\
+ \left(\sum_{t=1}^{k}\left\|\left(W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{E}(t)\right\|_F\right) &+ \left\|\left(I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{E}(k+1)\right\|_F.
\end{aligned}$$

Since the Frobenius norm is sub-multiplicative, i.e., $\|AB\|_F \le \|A\|_F \|B\|_F$, it follows that

$$\begin{aligned}
\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F &\le \left\|W(k:0) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \|\mathbf{X}(0)\|_F \\
+ \left(\sum_{t=1}^{k}\left\|W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \|\mathbf{E}(t)\|_F\right) &+ \left\|I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \|\mathbf{E}(k+1)\|_F. \quad (1.16)
\end{aligned}$$

By Lemma 3 we have

$$\left([W(k:t)]_{ij} - \frac{1}{m}\right)^2 \le q^{k-t} \qquad \text{for all } k \ge t \ge 0, \text{ with } q = 1 - \frac{\eta}{2m^2}.$$

Hence,

$$\left\|W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}\left([W(k:t)]_{ij} - \frac{1}{m}\right)^2} \le m\sqrt{q^{k-t}}.$$

Since $q = 1 - \frac{\eta}{2m^2}$, by using the fact $\sqrt{1-\mu} \le 1 - \mu/2$ for any $\mu \in (0, 1)$, we see that for all $k \ge t \ge 0$,

$$\left\| W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}' \right\|_F \le mp^{k-t} \qquad \text{with } \; p = 1 - \frac{\eta}{4m^2}. \tag{1.17}$$

For the norm $\left\| I - \frac{1}{m}\mathbf{1}\mathbf{1}' \right\|_F$ we have

$$\left\| I - \frac{1}{m}\mathbf{1}\mathbf{1}' \right\|_F = \sqrt{m\left(1 - \frac{1}{m}\right)^2 + (m-1)m\frac{1}{m^2}} = \sqrt{m-1}. \tag{1.18}$$

Using relations (1.17) and (1.18) in inequality (1.16), we obtain

$$\begin{aligned} \left\| \mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1) \right\|_F &\le mp^k \|\mathbf{X}(0)\|_F + m\left(\sum_{t=1}^{k} p^{k-t}\|\mathbf{E}(t)\|_F\right) \\ &\quad + \sqrt{m-1}\|\mathbf{E}(k+1)\|_F. \end{aligned} \tag{1.19}$$

We next interpret relation (1.19) in terms of the iterates $x_i(k+1)$ and the vectors $\epsilon_i(k+1)$, as given in (1.12). Recalling that the ℓth column of $\mathbf{X}(k)$ consists of the vector $x^\ell(k)$, with the entries $x_i^\ell(k)$, $i \in [m]$, for all $\ell \in [n]$, we can see that

$$\mathbf{1}'\mathbf{X}(k) = \left(\sum_{i=1}^{m} x_i^1(k), \ldots, \sum_{i=1}^{m} x_i^m(k)\right).$$

Thus,

$$\frac{1}{m}\mathbf{1}'\mathbf{X}(k) = x_{av}'(k) \qquad \text{where } \; x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k),$$

and

$$\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k) = \mathbf{1}x_{av}'(k) \qquad \text{for all } k.$$

Hence, $\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)$ is the matrix with all rows consisting of the vector $x_{av}'(k)$. Observing that the matrix $\mathbf{X}(k)$ has rows consisting of $x_1'(k), \ldots, x_m'(k)$, and using the definition of the Frobenius norm, we can see that

$$\left\| \mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k) \right\|_F = \sqrt{\sum_{i=1}^{m} \|x_i(k) - x_{av}(k)\|^2}.$$

Similarly, recalling that $\mathbf{E}(k)$ has rows consisting of $\epsilon_i'(k)$, $i \in [m]$, we also have

$$\|\mathbf{E}(k)\|_F = \sqrt{\sum_{i=1}^{m} \|\epsilon_i(k)\|^2}.$$

Therefore, relation (1.19) is equivalent to

$$\begin{aligned}
&\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} \\
&\quad \le mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m \left(\sum_{t=1}^{k} p^{k-t} \sqrt{\sum_{i=1}^{m} \|\epsilon_i(t)\|^2} \right) \\
&\quad\ + \sqrt{m-1} \sqrt{\sum_{i=1}^{m} \|\epsilon_i(k+1)\|^2}.
\end{aligned}$$

□

1.3.1.3 Basic Relation for Agents' Iterates

Recall that each $\epsilon_i(k+1)$ represents the difference between the projection point $\Pi_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)]$ and the point $y_i(k+1)$ (see (1.12)). Thus, there is a structure in $\epsilon_i(k+1)$ that can be further exploited. In particular, we can further refine the result of Lemma 5, under the assumption of bounded subgradients $g_i(k+1)$, as given in the following lemma.

Lemma 6 *Let the problem be convex (i.e., Assumption 1 holds). Also, assume that the subgradients of f_i are bounded over the set X for all i, i.e., there exists a constant C such that*

$$\|s\| \le C \qquad \text{for every subgradient } s \text{ of } f_i(z) \text{ at any } z \in X.$$

Furthermore, let Assumptions 2 and 3 hold for the graphs G_k and the matrices $W(k)$, respectively. Then, for the iterates $x_i(k)$ of the method (1.9) and their averages $x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k)$, we have for all $i \in [m]$ and $k \ge 0$,

$$\begin{aligned}
\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} &\le mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} \\
&\quad + m\sqrt{m}C \sum_{t=1}^{k} p^{k-t}\alpha_t + mC\alpha_{k+1},
\end{aligned}$$

where $p = 1 - \frac{\eta}{4m^2}$.

Proof By Lemma 5 we have for all $k \geq 0$,

$$\begin{aligned}
&\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} \\
&\quad \leq mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m \left(\sum_{t=1}^{k} p^{k-t} \sqrt{\sum_{i=1}^{m} \|\epsilon_i(t)\|^2} \right) \\
&\quad\quad + \sqrt{m-1} \sqrt{\sum_{i=1}^{m} \|\epsilon_i(k+1)\|^2}. \qquad (1.20)
\end{aligned}$$

Since $y_i(k+1)$ is a convex combination of points $x_j(k+1) \in X$, $j \in [m]$, by the convexity of the set X it follows that $y_i(k+1) \in X$ for all i, implying that for all $k \geq 0$,

$$\|\Pi_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)] - y_i(k+1)\| \leq \alpha_{k+1} \|g_i(k+1)\| \leq \alpha_{k+1} C.$$

Therefore, for all i and $k \geq 0$,

$$\|\epsilon_i(k+1)\|^2 \leq \alpha_{k+1}^2 C^2,$$

implying that

$$\sum_{i=1}^{m} \|\epsilon_i(k+1)\|^2 \leq m\alpha_{k+1}^2 C^2 \qquad \text{for all } k \geq 0.$$

By substituting the preceding estimate in (1.19), we obtain

$$\begin{aligned}
\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} &\leq mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m \left(\sum_{t=1}^{k} p^{k-t} \sqrt{m\alpha_t^2 C^2} \right) \\
&\quad + \sqrt{m-1} \sqrt{m\alpha_{k+1}^2 C^2} \\
&= mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m\sqrt{m} C \left(\sum_{t=1}^{k} p^{k-t} \alpha_t \right) \\
&\quad + \sqrt{m-1} \sqrt{m} C \alpha_{k+1}.
\end{aligned}$$

The desired relation follows by using $m - 1 \leq m$. □

We will now put together Lemmas 4 and 6 to provide a key result for establishing the convergence of the method. We assume some conditions on the stepsize, some of which are often used when analyzing the behavior of a subgradient algorithm.

Proposition 1 *Let Assumptions 1–3 hold. Assume that the subgradients of f_i are uniformly bounded over the set X for all i, i.e., there exists a constant C such that*

$$\|s\| \leq C \qquad \text{for every subgradient } s \text{ of } f_i(z) \text{ at any } z \in X.$$

Also, let the stepsize satisfy the following conditions

$$\alpha_{k+1} \leq \alpha_k \qquad \text{for all } k \geq 1, \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Then, for the iterates $x_i(k)$ of the method (1.9), we have for all $k \geq 0$ and all $x \in X$,

$$\sum_{i=1}^{m} \|x_i(k+1) - x\|^2 \leq \sum_{j=1}^{m} \|x_j(k) - x\|^2 - 2\alpha_{k+1}\left(f(x_{av}(k)) - f(x)\right) + s_k,$$

where $x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k)$, while s_k is given by

$$s_k = 2\alpha_{k+1} C\sqrt{m}\sqrt{\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2} + m\alpha_{k+1}^2 C^2,$$

and it satisfies

$$\sum_{k=0}^{\infty} s_k < \infty.$$

Proof By Lemma 4, we have for all i, all $k \geq 0$ and all $x \in X$,

$$\|x_i(k+1) - x\|^2 \leq \|y_i(k+1) - x\|^2 - 2\alpha_{k+1}\left(f_i(y_i(k+1)) - f_i(x)\right) + \alpha_{k+1}^2 \|g_i(k+1)\|^2.$$

Since $y_i(k+1)$ is a convex combination of the points $x_j(k)$, $j \in [m]$, by the convexity of the norm squared, it follows that

$$\|x_i(k+1) - x\|^2 \leq \sum_{j=1}^{m} w_{ij}(k)\|x_j(k) - x\|^2 - 2\alpha_{k+1}\left(f_i(y_i(k+1)) - f_i(x)\right) + \alpha_{k+1}^2 \|g_i(k+1)\|^2.$$

By summing these relations over i and by using the subgradient-boundedness property, we obtain

$$\begin{aligned}\sum_{i=1}^{m}\|x_i(k+1)-x\|^2 \le & \sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij}(k)\|x_j(k)-x\|^2\\ & -2\alpha_{k+1}\sum_{i=1}^{m}(f_i(y_i(k+1))-f_i(x))\\ & +m\alpha_{k+1}^2C^2.\end{aligned}$$

By exchanging the order of summation in the double-sum term, we see that

$$\sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij}(k)\|x_j(k)-x\|^2 = \sum_{j=1}^{m}\|x_j(k)-x\|^2\sum_{i=1}^{m} w_{ij}(k) = \sum_{j=1}^{m}\|x_j(k)-x\|^2,$$

where the last equality follows from $\mathbf{1}'W(k)=\mathbf{1}'$. Therefore,

$$\begin{aligned}\sum_{i=1}^{m}\|x_i(k+1)-x\|^2 \le & \sum_{j=1}^{m}\|x_j(k)-x\|^2 - 2\alpha_{k+1}\\ & \sum_{i=1}^{m}(f_i(y_i(k+1))-f_i(x)) + m\alpha_{k+1}^2C^2.\end{aligned}$$

We next estimate $f_i(y_i(k+1))-f_i(x)$ by using the average vector $x_{av}(k)$, as follows:

$$\begin{aligned}f_i(y_i(k+1))-f_i(x) &= f_i(y_i(k+1))-f_i(x_{av}(k))+f_i(x_{av}(k))-f_i(x)\\ &\ge -C\|y_i(k+1)-x_{av}(k)\|+f_i(x_{av}(k))-f_i(x),\end{aligned}$$

where the inequality follows by the Lipschitz continuity of f_i (due to the uniform subgradient-boundedness property on the set X and the fact that $y_i(k+1)\in X$ and $x_{av}(k)\in X$). By combining the preceding two relations and using $f=\sum_{i=1}^{m} f_i$, we have for all $k\ge 0$ and all $x\in X$,

$$\begin{aligned}\sum_{i=1}^{m}\|x_i(k+1)-x\|^2 \le & \sum_{j=1}^{m}\|x_j(k)-x\|^2 - 2\alpha_{k+1}\left(f(x_{av}(k))-f(x)\right)\\ & + 2\alpha_{k+1}C\sum_{i=1}^{m}\|y_i(k+1)-x_{av}(k)\| + m\alpha_{k+1}^2C^2. \quad (1.21)\end{aligned}$$

Consider now the vectors $y_i(k+1)$ and note that by the definition of $y_i(k+1)$, we have for any $y \in \mathbb{R}^n$,

$$\sum_{i=1}^{m} \|y_i(k+1) - y\| = \sum_{i=1}^{m} \left\| \sum_{j=1}^{m} w_{ij}(k)(x_j(k) - y) \right\|,$$

where we use $W(k)\mathbf{1} = \mathbf{1}$. By the convexity of the norm, it follows that

$$\sum_{i=1}^{m} \|y_i(k+1) - y\| \le \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij}(k) \|x_j(k) - y\| = \sum_{j=1}^{m} \|x_j(k) - y\|,$$

where the last equality is obtained by exchanging the order of summation and using $\mathbf{1}'W(k) = \mathbf{1}'$. Hence, for $y = x_{av}(k)$, we obtain

$$\sum_{i=1}^{m} \|y_i(k+1) - x_{av}(k)\| \le \sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\| \le \sqrt{m} \sqrt{\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2},$$

where the last inequality follows by Hölder's inequality. By substituting the preceding estimate in relation (1.21), we have for all $k \ge 0$ and all $x \in X$,

$$\begin{aligned} \sum_{i=1}^{m} \|x_i(k+1) - x\|^2 \le& \sum_{j=1}^{m} \|x_j(k) - x\|^2 - 2\alpha_{k+1} \left(f(x_{av}(k)) - f(x) \right) \\ &+ 2\alpha_{k+1} C \sqrt{m} \sqrt{\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2} + m\alpha_{k+1}^2 C^2. \end{aligned}$$

To simplify the notation, we let for all $k \ge 0$,

$$s_k = 2\alpha_{k+1} C \sqrt{m} \sqrt{\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2} + m\alpha_{k+1}^2 C^2, \tag{1.22}$$

so that we have for all $k \ge 0$ and all $x \in X$,

$$\sum_{i=1}^{m} \|x_i(k+1) - x\|^2 \le \sum_{j=1}^{m} \|x_j(k) - x\|^2 - 2\alpha_{k+1} \left(f(x_{av}(k)) - f(x) \right) + s_k.$$

We next show that the terms $\alpha_{k+1}\sqrt{\sum_{j=1}^m \|x_j(k) - x_{av}(k)\|^2}$ involved in the definition of s_k are summable over k. According to Lemma 6 we have for all $k \geq 1$,

$$\sqrt{\sum_{i=1}^m \|x_i(k) - x_{av}(k)\|^2} \leq mp^{k-1}\sqrt{\sum_{i=1}^m \|x_i(0)\|^2} + m\sqrt{m}C\sum_{t=1}^{k-1} p^{k-t-1}\alpha_t + mC\alpha_k.$$

Letting

$$r_k = \alpha_{k+1}\sqrt{\sum_{j=1}^m \|x_j(k) - x_{av}(k)\|^2}, \tag{1.23}$$

and using the assumption that the stepsize α_k is non-increasing, we see that

$$r_k \leq mp^{k-1}\alpha_1\sqrt{\sum_{i=1}^m \|x_i(0)\|^2} + m\sqrt{m}C\sum_{t=1}^{k-1} p^{k-t-1}\alpha_t^2 + mC\alpha_k^2.$$

By summing r_k over $k = 2, 3, \ldots, K$, for some $K \geq 2$, we have

$$\begin{aligned}\sum_{k=2}^K r_k &\leq m\left(\sum_{k=1}^K p^{k-1}\right)\alpha_1\sqrt{\sum_{i=1}^m \|x_i(0)\|^2} \\ &\quad + m\sqrt{m}C\sum_{k=2}^K\sum_{t=1}^{k-1} p^{k-t-1}\alpha_t^2 + mC\sum_{k=2}^K \alpha_k^2 \\ &< \frac{m}{1-p}\alpha_1\sqrt{\sum_{i=1}^m \|x_i(0)\|^2} + m\sqrt{m}C\sum_{s=1}^{K-1}\sum_{t=1}^{s} p^{s-t}\alpha_t^2 + mC\sum_{k=2}^K \alpha_k^2,\end{aligned}$$

where we use the fact that $p \in (0, 1)$ and we shift the indices in the double-sum term. Furthermore, by exchanging the order of summation, we see that

$$\sum_{s=1}^{K-1}\sum_{t=1}^{s} p^{s-t}\alpha_t^2 = \sum_{t=1}^{K-1}\alpha_t^2\sum_{s=t}^{K-1} p^{s-t} < \sum_{t=1}^{K-1}\alpha_t^2\frac{1}{1-p}.$$

Therefore,

$$\sum_{k=2}^{K} r_k < \frac{m}{1-p}\alpha_1 \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + \frac{m\sqrt{m}C}{1-p} \sum_{t=1}^{K-1} \alpha_t^2 + mC \sum_{k=2}^{K} \alpha_k^2.$$

In view of the assumption that $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, it follows that $\sum_{k=2}^{\infty} r_k < \infty$. Since

$$s_k = 2C\sqrt{m} r_k + m\alpha_{k+1}^2 C^2$$

(see (1.22) and (1.23)), it follows that

$$\sum_{k=0}^{\infty} s_k < \infty.$$

□

1.3.1.4 Convergence Result for Agents' Iterates

Using Proposition 1, we establish a convergence result for the iterates $x_i(k)$, as given in the following theorem.

Theorem 1 *Let Assumptions 1–3 hold. Assume that there is a constant C such that*

$$\|s\| \le C \qquad \textit{for every subgradient } s \textit{ of } f_i(z) \textit{ at any } z \in X.$$

Let the stepsize satisfy the following conditions

$$\alpha_{k+1} \le \alpha_k \qquad \textit{for all } k \ge 1, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty, \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty,$$

and assume that problem (1.1) has a solution. Then, the iterate sequences $\{x_i(k)\}$, $i \in [m]$, generated by the method (1.9), converge to an optimal solution of problem (1.1), i.e.,

$$\lim_{k\to\infty} \|x_i(k) - x^*\| = 0 \qquad \textit{for all } i \in [m] \textit{ and some } x^* \in X^*.$$

Proof By letting $x = x^*$ in Proposition 1, for an arbitrary $x^* \in X^*$, we obtain for all $i \in [m]$ and $k \ge 0$,

$$\sum_{i=1}^{m} \|x_i(k+1) - x^*\|^2 \le \sum_{j=1}^{m} \|x_j(k) - x^*\|^2 - 2\alpha_{k+1}\big(f(x_{av}(k)) - f(x^*)\big) + s_k,$$

with $s_k > 0$ satisfying $\sum_{k=0}^{\infty} s_k < \infty$. By summing these relations over $k = K, K + 1, \ldots, T$ for any $T \geq K \geq 0$, after re-arranging the terms, we further obtain for all $x^* \in X^*$ and all $T \geq K \geq 0$,

$$\sum_{i=1}^{m} \|x_i(T+1) - x^*\|^2 + 2\sum_{k=K}^{T} \alpha_{k+1} \left(f(x_{av}(k)) - f(x^*)\right)$$
$$\leq \sum_{j=1}^{m} \|x_j(K) - x^*\|^2 + \sum_{k=K}^{T} s_k. \tag{1.24}$$

Note that $f(x_{av}(k)) - f(x^*) > 0$ since $x_{av}(k) \in X$. Thus, the preceding relation implies that the sequences $\{x_i(k)\}$, $i \in [m]$, are bounded and, also, that

$$\sum_{k=0}^{\infty} \alpha_{k+1} \left(f(x_{av}(k)) - f^*\right) < \infty$$

since $\sum_{k=0}^{\infty} s_k < \infty$, where $f^* = f(x^*)$ for any $x^* \in X^*$. Thus, it follows that

$$\liminf_{k\to\infty} \left(f(x_{av}(k)) - f^*\right) = 0.$$

Let $\{k_\ell\}$ be a sequence of indices that attains the above limit inferior, i.e.,

$$\lim_{\ell\to\infty} f(x_{av}(k_\ell)) = f^*. \tag{1.25}$$

Since the sequences $\{x_i(k)\}$, $i \in [m]$, are bounded, so is the average sequence $\{x_{av}(k)\}$. Hence, $\{x_{av}(k_\ell)\}$ contains a converging subsequence. Without loss of generality, we may assume that $\{x_{av}(k_\ell)\}$ converges to some point $\hat{x}$, i.e.,

$$\lim_{\ell\to\infty} x_{av}(k_\ell) = \hat{x}.$$

Note that $\hat{x} \in X$ since $\{x_{av}(k)\} \subset X$ and the set X is assumed to be closed. Note further that f is continuous on $\mathbb{R}^n$ since it is convex on $\mathbb{R}^n$. Hence, we have

$$\lim_{\ell\to\infty} f(x_{av}(k_\ell)) = f(\hat{x}) \qquad \text{with} \qquad \hat{x} \in X,$$

which together with relation (1.25) yields $f(\hat{x}) = f^*$. Therefore, $\hat{x}$ is an optimal point.

Next, we show that $\{x_i(k_\ell)\}$ converges to $\hat{x}$ for all i. By Lemma 6 we have for all $k \geq 0$ and all $i \in [m]$,

$$\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} \leq mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m\sqrt{m}C \sum_{t=1}^{k} p^{k-t}\alpha_t + mC\alpha_{k+1}.$$

Letting $k = k_\ell - 1$ for any $k_\ell \geq 1$, we see that for all $i \in [m]$,

$$\sqrt{\sum_{i=1}^{m} \|x_i(k_\ell) - x_{av}(k_\ell)\|^2} \leq mp^{k_\ell - 1} \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m\sqrt{m}C \sum_{t=1}^{k_\ell - 1} p^{k_\ell - 1 - t}\alpha_t + mC\alpha_{k_\ell}.$$

Since $p \in (0, 1)$ and $\alpha_k \to 0$ (due to $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$), it follows that

$$\limsup_{\ell \to \infty} \sqrt{\sum_{i=1}^{m} \|x_i(k_\ell) - x_{av}(k_\ell)\|^2} \leq m\sqrt{m}C \limsup_{\ell \to \infty} \sum_{t=1}^{k_\ell - 1} p^{k_\ell - 1 - t}\alpha_t.$$

We note that

$$\begin{aligned}
\limsup_{\ell \to \infty} \sum_{t=1}^{k_\ell - 1} p^{k_\ell - 1 - t}\alpha_t &= \lim_{k \to \infty} \sum_{t=1}^{k-1} p^{k-1-t}\alpha_t \\
&= \lim_{k \to \infty} \left(\left(\sum_{\tau=1}^{k-1} p^{k-1-\tau} \right) \frac{1}{\sum_{\tau=1}^{k-1} p^{k-1-\tau}} \sum_{t=1}^{k-1} p^{k-1-t}\alpha_t \right) \\
&= \lim_{k \to \infty} \left(\sum_{\tau=1}^{k-1} p^{k-1-\tau} \right) \left(\lim_{k \to \infty} \frac{1}{\sum_{\tau=1}^{k-1} p^{k-1-\tau}} \sum_{t=1}^{k-1} p^{k-1-t}\alpha_t \right) \\
&= \frac{1}{1-p} \lim_{t \to \infty} \alpha_t, \qquad (1.26)
\end{aligned}$$

where in the last equality we use that fact that any convex combination of a convergent sequence $\{\alpha_k\}$ converges to the same limit as the sequence itself. Hence, we have

$$\limsup_{\ell \to \infty} \sum_{t=1}^{k_\ell - 1} p^{k_\ell - 1 - t}\alpha_t = 0,$$

implying that

$$\limsup_{\ell\to\infty} \sqrt{\sum_{i=1}^{m} \|x_i(k_\ell) - x_{av}(k_\ell)\|^2} = 0.$$

Therefore, since $\lim_{\ell\to\infty} x_{av}(k_\ell) = \hat{x}$, it follows that

$$\lim_{\ell\to\infty} x_i(k_\ell) = \hat{x} \quad \text{for all } i \in [m], \qquad \text{with} \quad \hat{x} \in X^*. \tag{1.27}$$

Now, since $\hat{x} \in X^*$, we let $x^* = \hat{x}$ in (1.24). Then, we let $K = k_\ell$ in (1.24) and by omitting the term involving the function values, from (1.24) we obtain for all $\ell \geq 1$,

$$\limsup_{T\to\infty} \sum_{i=1}^{m} \|x_i(T+1) - \hat{x}\|^2 \leq \sum_{j=1}^{m} \|x_j(k_\ell) - \hat{x}\|^2 + \sum_{k=k_\ell}^{\infty} s_k.$$

Letting $\ell \to \infty$, and using relation (1.27), we see that

$$\limsup_{T\to\infty} \sum_{i=1}^{m} \|x_i(T+1) - \hat{x}\|^2 \leq \lim_{\ell\to\infty} \sum_{k=k_\ell}^{\infty} s_k = 0,$$

where $\lim_{\ell\to\infty} \sum_{k=k_\ell}^{\infty} s_k = 0$ holds since $\sum_{k=0}^{\infty} s_k < \infty$. Thus, it follows that for $\hat{x} \in X^*$,

$$\lim_{k\to\infty} \|x_i(k) - \hat{x}\| = 0 \qquad \text{for all } i \in [m].$$

□

1.3.2 Numerical Examples

Here, we show some numerical results obtained for a variant of the algorithm in (1.9) as applied to a data classification problem from Example 3. We will consider an extension of that problem to the case when the data set is not perfectly separable. In this case, there is an additional slack variable u that enters the model, and the distributed version of the problem assumes the following form:

$$\min_{(x,u)\in\mathbb{R}^n\times\mathbb{R}} f(x,u) \qquad f(x,u) = \sum_{i=1}^{m} f_i(x,u),$$

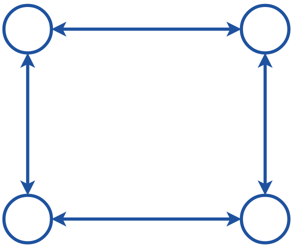


Fig. 1.5 An undirected network with four nodes

where each $f_i : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is given by:

$$f_i(x, u) = \frac{\rho}{2m}\|x\|^2 + \sum_{j\in D_i} \max\{0, 1 - y_j\langle x, z_j\rangle + u\},$$

where D_i is the collection of the data points at the center i (agent i).

Letting $\mathbf{x} = (x, u) \in \mathbb{R}^n \times \mathbb{R}$, we consider the following distributed algorithm[3] for this problem over a static graph $G = ([m], E)$:

$$\mathbf{y}_i(k+1) = \mathbf{x}_i(k) - \eta_{k+1}\sum_{j=1}^{m} r_{ij}\mathbf{x}_j(k) \qquad (r_{ij} = 0 \text{ when } i \leftrightarrow j \notin E),$$
$$\mathbf{x}_i(k+1) = \mathbf{y}_i(k+1) - \alpha_{k+1} g_i(k+1), \tag{1.28}$$

where $g_i(k+1)$ is a subgradient of f_i at $\mathbf{y}_i(k+1)$. We note that the weights used in the update of $\mathbf{y}_i(k+1)$ are different from the weights used in (1.9). The weights here are based on a Laplacian formulation of the consensus problem, which include another parameter $\eta_{k+1} > 0$. This parameter can be viewed as an additional stepsize that is associated with the feasibility step for the consensus constraints. Under some (boundedness) conditions on η_{k+1} and standard conditions on the stepsize (akin to those in Theorem 1), the method converges to a solution of the problem [144].

We illustrate the behavior of the method in (1.28) for the case of a network with four nodes organized in a ring, as depicted in Fig. 1.5. The simulations are generated for the regularization parameter $\rho = 6$. The stepsize values used in the experiment are: $\eta_k = 0.8$ and $\alpha_k = \frac{1}{k}$, for all $k \geq 1$. The behavior of the method is depicted in Fig. 1.6 where the resulting hyperplanes produced by agents are shown after 20 and after 500 iterations. The plots also show the true separating hyperplane that solves the centralized problem.

The algorithm (1.28) assumes perfect communication links, which is not typically the case in wireless networks. To capture the effect of communication noise,

[3] See [144, 146] for more details.

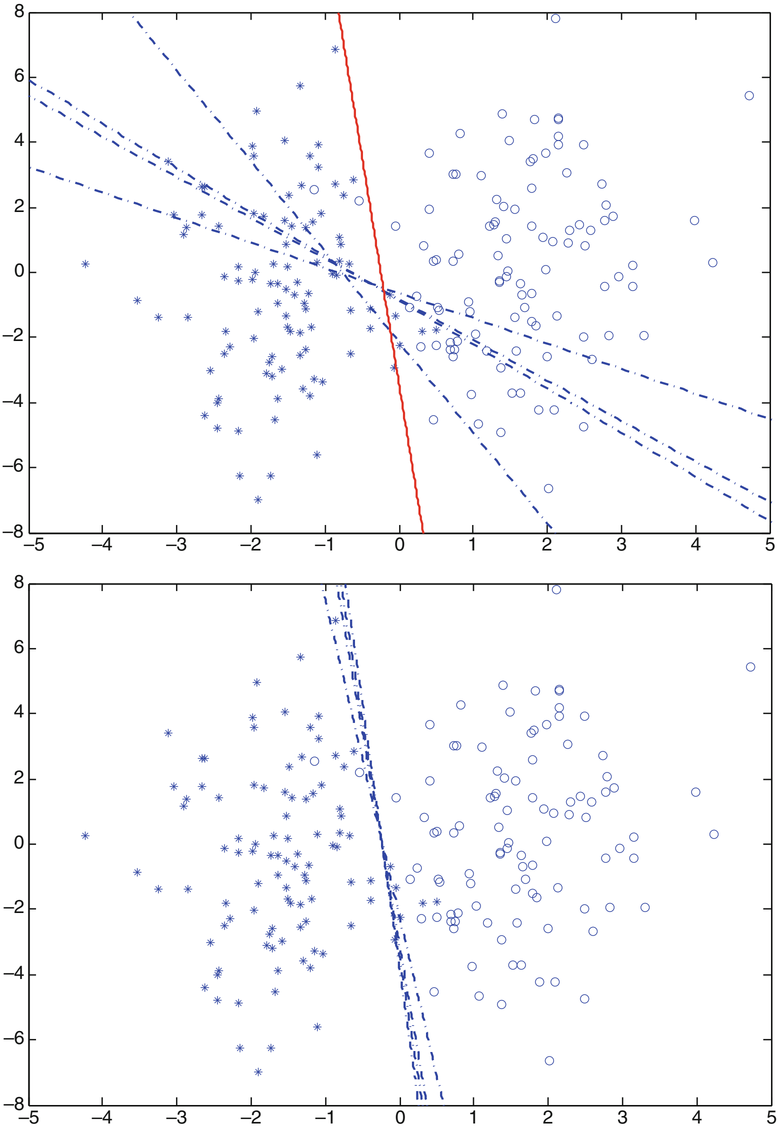


Fig. 1.6 The top plot shows the agents' iterates $\mathbf{x}_i(k)$ after 20 iterations and the true solution (the hyperplane in red color), while the bottom plot shows the agents' iterates after 500 iterations

consider the following variant of the method:

$$\mathbf{y}_i(k+1) = \mathbf{x}_i(k) - \eta_{k+1} \sum_{j=1}^{m} r_{ij}(\mathbf{x}_j(k) + \xi_{ij}(k)) \quad (r_{ij} = 0 \text{ when } i \leftrightarrow j \notin E),$$

$$\mathbf{x}_i(k+1) = \mathbf{y}_i(k+1) - \alpha_{k+1} g_i(k+1), \tag{1.29}$$

where $\xi_{ij}(k)$ is a link dependent noise associated with messages received by agent i from a neighbor j. The parameter $\eta_{k+1} > 0$ can be viewed as a noise-damping stepsize. In this case, for the method to converge to a solution of the problem, the noise-damping stepsize η_k has to be coordinated with the subgradient-related stepsize α_k. In particular, the following conditions are imposed:

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=1}^{\infty} \eta_k = \infty,$$

$$\sum_{k=1}^{\infty} \eta_k^2 < \infty, \quad \sum_{k=1}^{\infty} \alpha_k \eta_k < \infty, \quad \sum_{k=1}^{\infty} \frac{\alpha_k^2}{\eta_k} < \infty.$$

The simulations are performed for a different set of data and the ring graph shown in Fig. 1.5, where the links were assumed to be noisy. The noise is modeled by the i.i.d. zero mean Gaussian process with the variance equal to 1. The regularization parameter is $\rho = 6$, while the noise-damping parameter and the stepsize are $\eta_k = \frac{1}{k^{0.55}}$ and $\alpha_k = \frac{1}{k}$ for all $k \geq 1$. The results for one of the typical simulation run are shown in Fig. 1.7. These simulation results are taken from [144], where more simulation results can be found.

1.4 Distributed Asynchronous Algorithms for Static Undirected Graphs

There are several drawbacks of synchronous updates that limit their applications, including

- All agents have to update *at the same time*. Imagine that each agent has its own clock and it updates at each tick of its clock. Then, the requirement that the agents update synchronously, as in method (1.9), means that the agents must have their local clocks perfectly synchronized throughout the computation task. This is hard to ensure in practice for some networks, such as wireless networks where communication interference is an issue.
- The communication links in the connectivity graphs G_k have to be *perfectly activated* to transmit and receive information. Some communication protocols

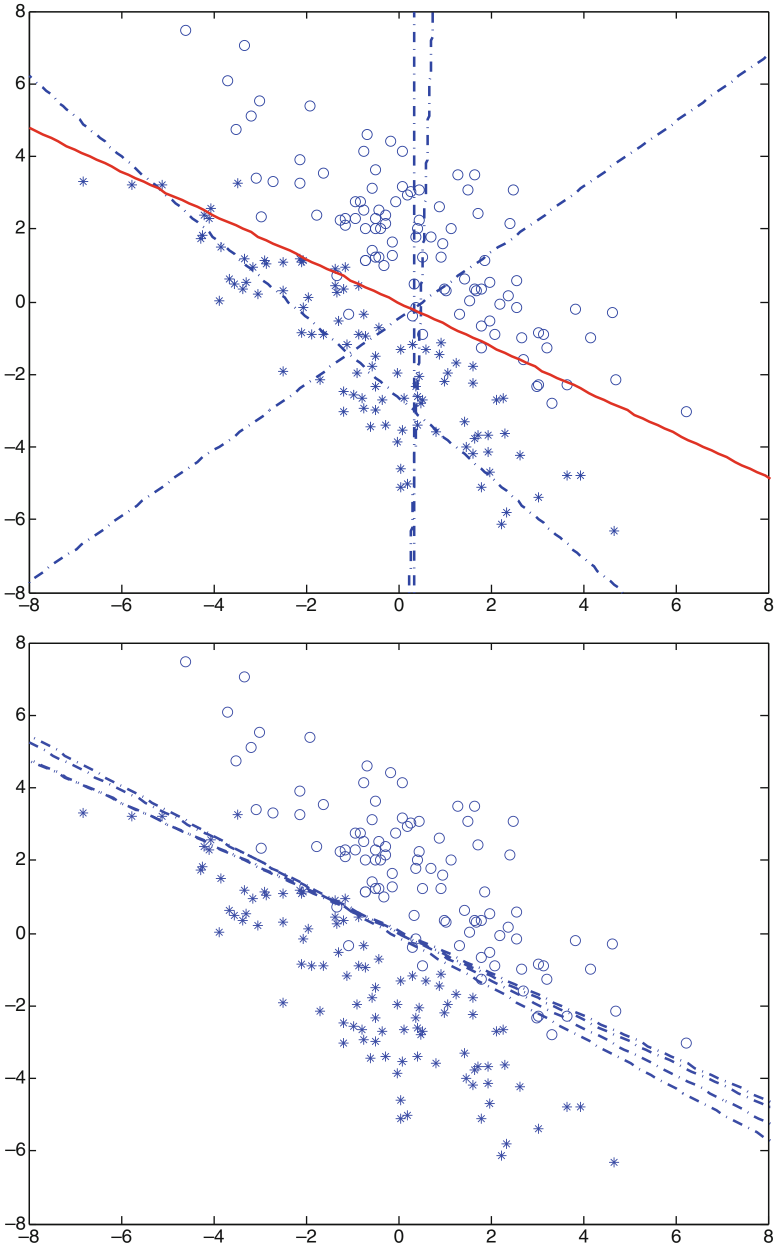


Fig. 1.7 The top plot shows the agents' iterates $\mathbf{x}_i(k)$ after the first iteration and the true solution (the hyperplane in red color), while the bottom plot shows the agents' iterates after 500 iterations

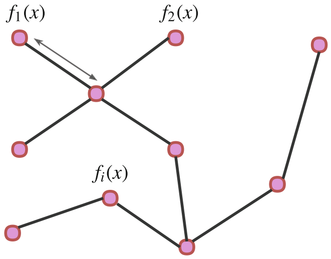


Fig. 1.8 The agent communication graph is static, undirected and connected. However, not all the links will necessarily be active at each time instance. A function f_i is a local objective of agent i

require receiving message acknowledgements (“acks"), which can lead to deadlocks when the links are not perfect.

- Communications can be *costly* (consume power) and it may not be efficient for agents to be activated too frequently to communicate, due to their limited power supply for example.

In order to alleviate these drawbacks of simultaneous updates, one possibility is to randomize the activation of communication links in the network or the activation of the agents. We will discuss two such random activations: gossip and broadcast protocols. Gossip could be viewed as a random link activation process, while broadcast is a random-agent activation process. We will here treat both as a random-agent activation, by assuming that the agents are equipped with local clocks that tick according to the same rate (the same inter-click time), but do not click synchronously.

Throughout this section, we assume that the underlying communication graph is static and undirected, denoted by $G = ([m], E)$, see Fig. 1.8 for an illustration of the graph. The randomization we will use to develop asynchronous algorithms will have some stepsizes that can be easily analyzed for a static graph. While their extensions to time-varying graphs may be possible, we will not consider them here.

To develop asynchronous algorithms for solving problem (1.1), we will use the asynchronous methods for consensus. Thus, we will at first discuss random asynchronous consensus methods in Sect. 1.4.1 and, then, we give the corresponding asynchronous optimization methods in Sect. 1.4.2.

1.4.1 Random Gossip and Random Broadcast for Consensus

Both random gossip and random broadcast algorithms can be used to achieve a consensus. These two approaches share in common the mechanism that triggers the update events, but they differ in the update rule specifications.

More concretely, these two approaches use the same random process to wake up an agent that will initiate a communication event that includes an iterate update. However, in the random gossip algorithm the agent that wakes up contacts one of its neighbors at random, thus, randomly activating a single undirected link for communication, and both the agent and the selected neighbor perform updates. Unlike this, in the random broadcast approach, the agent that wakes up broadcasts its information to all of its neighbors, thus resulting in using directed communication links (even-though the links are undirected). Moreover, upon the broadcast, the agent that triggered the communication event goes to an inactive mode and only its neighbors perform updates.

Let us now describe the random process that triggers the communication events for both gossip and broadcast models. Each agent has its own local Poisson clock that ticks with a rate equal to 1 (the rate can take any other positive value as long as all agents have the same rate). Each agent's clock ticks independently of the other agents' clocks. At a tick of its clock, an agent wakes up and initiates a communication event. The ticks of the local agents' clocks can be modeled as single virtual Poisson clock that ticks with a rate m. Letting $\{Z_k\}$ be the Poisson process of the virtual clock tick-times, we discretize the time according to $\{Z_k\}$ since these are the only times when a change will occur in some of the agent values $x_i(k)$. The inter-tick times $\{Z_{k+1} - Z_k\}$ are i.i.d. exponentially distributed with the rate m.

1.4.1.1 Gossip-Based Consensus Algorithm

In the gossip-model, the randomly activated agent wakes up and selects randomly one of its neighbors, as depicted in Fig. 1.9. The activated agent and its selected neighbor exchange their information and perform an iterate update. Upon the update, both agents go to sleep.

To formalize the process, we let I_k be the index of the agent that is activated at time Z_k, i.e., the agent whose clock ticks at time Z_k. The variables $\{I_k\}$ are i.i.d. with a uniform distribution over $\{1, \ldots, m\}$, i.e.,

$$\text{Prob}\{I_k = i\} = \frac{1}{m} \qquad \text{for all } i \in [m].$$

For agent i, let $p_{ij} > 0$ be the probability of contacting its neighbor $j \in N_i$, $j \neq i$. Let J_k be the index of a neighbor of agent I_k that is selected randomly for communication at time Z_k. Let $P = [p_{ij}]$ be the matrix of contact probabilities, where $p_{ij} = 0$ if $j \notin N_i$, and note that P is row-stochastic.

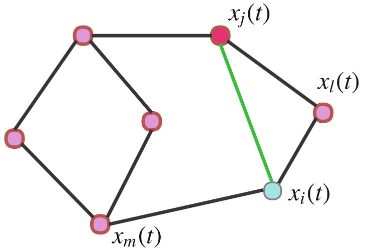


Fig. 1.9 Random gossip communication protocol: an agent that wakes up establishes a connection with a randomly selected neighbor. Thus, a random link is activated (shown in green color)

At time k, the active agents I_k and J_k exchange their current values $x_{I_k}(k)$ and $x_{J_k}(k)$, and then both update as follows:

$$x_{I_k}(k+1) = \frac{1}{2}\left(x_{I_k}(k) + x_{J_k}(k)\right), \qquad x_{J_k}(k+1) = \frac{1}{2}\left(x_{J_k}(k) + x_{I_k}(k)\right), \tag{1.30}$$

while the other agents do nothing (they sleep),

$$x_i(k+1) = x_i(k) \qquad \text{for all } i \notin \{I_k, J_k\}.$$

A value other than $1/2$ can be used in the updates in (1.30); however, we will work with $1/2$.

Assuming that the agent values $x_i(k)$ are scalars, the gossip iteration update can be compactly written, as follows:

$$x(k+1) = W_g(k)x(k) \qquad \text{for all } k \geq 0, \tag{1.31}$$

where $x(k)$ is a vector with components $x_i(k)$, $i \in [m]$, and the matrix $W_g(k)$ is symmetric with the entries given by

$$[W_g(k)]_{I_k,J_k} = [W_g(k)]_{J_k,I_k} = \frac{1}{2}, \qquad [W_g(k)]_{I_k,I_k} = \frac{1}{2}, \qquad [W_g(k)]_{J_k,J_k} = \frac{1}{2},$$

$$[W_g(k)]_{ii} = 1 \quad \text{for all } i \in [m] \setminus \{I_k, J_k\}, \qquad \text{and else} \quad [W_g(k)]_{ij} = 0.$$

Equivalently, the random matrix $W_g(k)$ is given by

$$W_g(k) = W^{(I_k J_k)}, \qquad \text{with} \quad W^{(ij)} = I - \frac{1}{2}(e_i - e_j)(e_i - e_j)' \quad \text{for all } i, j \in [m],$$

where e_i is the unit vector with its ith entry equal to 1 and the other entries equal to 0. Thus, the random matrix $W_g(k)$ takes values $W_g(k) = W^{(ij)}$ with the probability $(p_{ij} + p_{ji})/m$.

Every realization of $W_g(k)$ is a symmetric and stochastic matrix, hence, $W_g(k)$ is doubly stochastic. Furthermore, it can be seen that every realization $W^{(ij)}$ of $W_g(k)$ is a projection matrix[4] on the sub-space

$$S_{ij} = \{x \in \mathbb{R}^m \mid x_i = x_j\}.$$

Therefore, we have

$$W_g'(k)W_g(k) = W_g^2(k) = W_g(k) \qquad \text{for all } k \geq 0.$$

[4] A matrix A is a projection matrix if $A^2 = A$.

The convergence of the gossip algorithm has been shown in [16]. In the following theorem, we provide a statement based on the result in [16] that is relevant to our subsequent discussion on distributed asynchronous methods.

Theorem 2 *([16]) Assume that the graph G is connected. Then, the iterate sequences $\{x_i(k)\}$, $i \in [m]$, produced by the gossip algorithm (1.31) satisfy the following relation*

$$\mathbb{E}\left[\left\|x(k) - \frac{1}{m}\sum_{j=1}^m x_j(0)\,\mathbf{1}\right\|^2\right] \le \lambda_g^k \left\|x(0) - \frac{1}{m}\sum_{j=1}^m x_j(0)\,\mathbf{1}\right\|^2 \qquad \textit{for all } k \ge 0,$$

where $0 < \lambda_g < 1$ is the second largest eigenvalue of $\bar{W}_g = \mathbb{E}\left[W_g(k)\right]$.

Proof Defining

$$z(k) = x(k) - \frac{1}{m}\sum_{j=1}^m x_j(0)\,\mathbf{1},$$

and using the gossip updates in (1.31), the following relation has been shown in [16]:

$$\mathbb{E}\left[\|z(k+1)\|^2 \mid z(k)\right] = \langle z(k), \mathbb{E}\left[W_g'(k)W_g(k)\right]z(k)\rangle$$

(see equation (14) in [16]). Since $W_g'(k)W_g(k) = W_g(k)$ and since $\{W_g(k)\}$ is an i.i.d. matrix sequence, by letting $\mathbb{E}\left[W_g(k)\right] = \bar{W}_g$, we obtain

$$\mathbb{E}\left[\|z(k+1)\|^2 \mid z(k)\right] = \langle z(k), \bar{W}_g z(k)\rangle.$$

The matrix $\bar{W}_g$ is symmetric and doubly stochastic, since each realization $W^{(ij)}$ of any $W_g(k)$ is symmetric and doubly stochastic. Furthermore, since each realization $W^{(ij)}$ is a projection matrix, each $W^{(ij)}$ is positive semi-definite. Hence, $\bar{W}_g$ is also positive semi-definite and, consequently, all eigenvalues of $\bar{W}_g$ are non-negative. Moreover, we have

$$[\bar{W}_g]_{ii} > 0 \qquad \text{for all } i \in [m],$$

and for all $i \neq j$,

$$[\bar{W}_g]_{ij} > 0 \quad \Longleftrightarrow \quad i \leftrightarrow j \in E.$$

Since the graph G is connected, $\bar{W}_g$ is irreducible and by Theorem 4.3.1, page 106 in [46], the matrix $\bar{W}_g$ has 1 as the largest eigenvalue of multiplicity 1, with the associated eigenvector $\mathbf{1}$. Since $z(k) \perp \mathbf{1}$, it follows that

$$\langle z(k), \bar{W}_g z(k) \rangle \leq \lambda_g \|z(k)\|^2,$$

where $0 < \lambda_g < 1$ is the second largest eigenvalue of $\bar{W}_g$.

Therefore, we have

$$\mathbb{E}\left[\|z(k+1)\|^2 \mid z(k)\right] \leq \lambda_g \|z(k)\|^2 \qquad \text{for all } k \geq 0,$$

implying that

$$\mathbb{E}\left[\|z(k+1)\|^2\right] \leq \lambda_g \mathbb{E}\left[\|z(k)\|^2\right] \leq \cdots \leq \lambda_g^{k+1} \|z(0)\|^2 \qquad \text{for all } k \geq 0.$$

□

From Theorem 2 it follows that

$$\sum_{k=0}^{\infty} \mathbb{E}\left[\left\| x(k) - \frac{1}{m} \sum_{j=1}^{m} x_j(0)\, \mathbf{1} \right\|^2\right] < \infty,$$

which (by Fatou's Lemma) implies that with probability 1,

$$\lim_{k \to \infty} \left\| x(k) - \left(\frac{1}{m} \sum_{j=1}^{m} x_j(0) \right) \mathbf{1} \right\|^2 = 0,$$

showing that the iterates converge to the average of their initial values with probability 1. We note that the same result is true if the agents variables $x_j(0)$ were vectors due to the linearity of the update rule (1.31).

1.4.1.2 Broadcast-Based Consensus Algorithm

In the broadcast model, at time Z_k, the randomly activated agent I_k broadcasts its value $x_{I_k}(k)$ to all of its neighbors $j \in \breve{N}_{I_k}$ in the graph $G = ([m], E)$. Here, the neighbor set $\breve{N}_i$ of an agent i does not include the agent i itself,

$$\breve{N}_i = \{j \in [m] \mid i \leftrightarrow j \in E\}.$$

Thus, even though the graph G is undirected, the actual links that are used at any instance of time are virtually directed, as shown in Fig. 1.10.

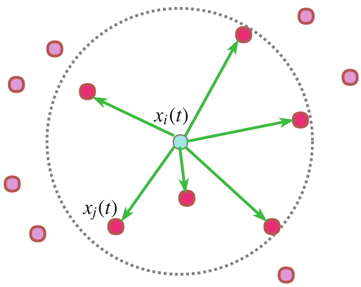


Fig. 1.10 Broadcast communication protocol: an agent that wakes up broadcasts its value to its neighbors, resulting in a random set of agents that are activated for performing an update. In a wireless network, the neighbors of an agent are typically defined as those agents that are within a certain radius of a given agent

Upon receiving the broadcasted value, the agents $j \in \breve{N}_{I_k}$ perform an update of their values, while the other agents do nothing (they sleep), including the agent I_k that broadcasted its information. Formally, the updates are given by

$$x_j(k+1) = (1-\beta)x_j(k) + \beta x_{I_k}(k) \qquad \text{for all } j \in \breve{N}_{I_k},$$
$$x_j(k+1) = x_j(k) \qquad \text{for all } j \notin \breve{N}_{I_k},$$

where $\beta \in (0,1)$.

We define the matrix $W_b(k)$, as follows:

$$[W_b(k)]_{ii} = 1-\beta \quad \text{for all } i \in \breve{N}_{I_k}, \qquad [W_b(k)]_{iI_k} = \beta \quad \text{for all } i \in \breve{N}_{I_k},$$
$$[W_b(k)]_{ii} = 1 \quad \text{for all } i \notin N_{I_k} \qquad \text{and else} \quad [W_b(k)]_{ij} = 0.$$

Using this matrix, the broadcast method can be written as:

$$x(k+1) = W_b(k)x(k) \qquad \text{for all } k \geq 0. \tag{1.32}$$

Note that the random matrix $W_b(k)$ is stochastic, but not necessarily doubly stochastic. Also, note that it is not symmetric. The expected matrix $\bar{W}_b = \mathbb{E}[W_b(k)]$ is in fact doubly stochastic. Specifically, as shown in [2, 3], $\bar{W}_b$ is given by

$$\bar{W}_b = I - \frac{\beta}{m} L_G,$$

where L_G is the Laplacian of the graph G, i.e., $L_G = D - A$ where A is the 0-1 adjacency matrix for the graph G and D is the diagonal matrix with entries $d_{ii} = |\breve{N}_i|$, $i \in [m]$. Since G is undirected, its Laplacian L_G is symmetric. Furthermore, since $L_G\mathbf{1} = 0$, it follows that $\bar{W}_b\mathbf{1} = \mathbf{1}$, which due to the symmetry of $\bar{W}_b$ also implies that $\mathbf{1}'\bar{W}_b = \mathbf{1}'$.

In addition, it has been shown in [2, 3] that, when the graph G is connected, the spectral norm of the matrix

$$\bar{W}_b - \frac{1}{m}\mathbf{1}\mathbf{1}'$$

is less than 1 (see Lemma 2 in [3]). This spectral property of the matrix $\bar{W}_b - \frac{1}{m}\mathbf{1}\mathbf{1}'$ is sufficient to guarantee the convergence of the random broadcast algorithm to a consensus in expectation only. Its convergence with probability 1 requires some additional analysis of the properties of the random matrices $W_b(k)$. In particular, the spectral norm of the matrix $\mathbb{E}\left[W_b'(k)(I - \frac{1}{m}\mathbf{1}\mathbf{1}')W_b(k)\right]$ plays a crucial role in establishing such a convergence result. Let Q denote this matrix, i.e.,

$$Q = \mathbb{E}\left[W_b'(k)\left(I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)W_b(k)\right], \tag{1.33}$$

where I denotes the identity matrix of the appropriate dimension.

It has been shown in Proposition 2 of [3] that, when the graph G is connected, then the matrix Q has a spectral radius less than 1 for any $\beta \in (0, 1)$ (see the role of β in the definition of matrix $W_b(k)$). This property is a key in proving the convergence of the method with probability 1, as given in Theorem 1 in [3]. In the next theorem, we summarize some key relations which have been established in [3].

Theorem 3 (Lemma 3 and Proposition 2 of [3]) *Assume that the graph G is connected. Then, for any $\beta \in (0, 1)$, we have*

(a) The spectral radius of the matrix Q in (1.33) is less than 1.

(b) The iterate sequences $\{x_i(k)\}$, $i \in [m]$, produced by the random broadcast algorithm (1.32) satisfy the following relation

$$\mathbb{E}\left[\left\|x(k) - \frac{1}{m}\sum_{j=1}^{m} x_j(k)\,\mathbf{1}\right\|^2\right] \le \lambda_b^k \left\|x(0) - \frac{1}{m}\sum_{j=1}^{m} x_j(0)\,\mathbf{1}\right\|^2 \quad \text{for all } k \ge 0,$$

where $0 < \lambda_b < 1$ is the spectral norm of the matrix Q given in (1.33).

An extension of Theorem 3 to the case when the links are unreliable can be found in [101]. We note that the random broadcast algorithm does not lead to the consensus on the average of the initial agents' values with probability 1. It guarantees, with probability 1, that the agents will reach a consensus on a random point whose expected value is the average of the initial agents' values. Concretely, as shown in Theorem 1 of [3], there holds

$$\mathrm{Prob}\left\{\lim_{k\to\infty} x(k) = c\mathbf{1}\right\} = 1,$$

where c is a random scalar satisfying

$$\mathbb{E}[c] = \frac{1}{m}\sum_{i=1}^{m} x_i(0).$$

1.4.2 Distributed Asynchronous Algorithm

In this section, we consider a general distributed asynchronous algorithm for optimization problem (1.1) based on random matrices. The random matrices are employed for the alignment of the agents iterates. As special cases of this algorithm, one can obtain the algorithms that use the random gossip and the random broadcast communications.

In particular, we will consider an algorithm with random asynchronous updates, as follows. We assume that there is some random i.i.d. process that triggers the update times Z_k (as for the cases of gossip and broadcast). Without going into details of such a process, we can simply keep a virtual index to count the update times (corresponding to the times when at least one agent is active). We also assume that the agents communicate over a network with connectivity structure captured by an undirected graph G.

At the time of the $k+1$st update, a random stochastic matrix $W(k)$ is available that captures the communication pattern among the agents, i.e., $w_{ij}(k) > 0$ if and only if agent i receives $x_j(k)$ from its neighbor $j \in N_i$. We let A_k be the set of agents that are active (perform an update) at time $k+1$. Then, the agents iterates at time $k+1$ are described through the following two steps:

$$\begin{aligned} v_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k) x_j(k), \\ x_i(k+1) &= \Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)]\chi_{\{i\in A_k\}} \\ &\quad + v_i(k+1)\chi_{\{i\notin A_k\}}, \end{aligned} \tag{1.34}$$

where $\alpha_{i,k+1} > 0$ is a stepsize of agent i, $g_i(k+1)$ is a subgradient of f_i at $v_i(k+1)$ and χ_{E} is the characteristic function of an event E. We will *assume that the initial points* $\{x_i(0), i \in [m]\} \subset X$ *are deterministic.*

Note that each agent uses its own stepsize $\alpha_{i,k+1}$. It is important to note that, since $W(k)$ is stochastic, the event $\{i \in A_k\}$ is equivalent to $\{w_{ii}(k) \neq 1\}$. Thus, when $i \notin A_k$, which is equivalent to $\{w_{ii}(k) = 1\}$, we have $v_i(k+1) = x_i(k)$

and $x_i(k+1) = x_i(k)$. Hence, the relation $i \notin A_k$ corresponds to agent i not updating at all, so the iterate updates in (1.34) are equivalent to the following update scheme:

$$v_i(k+1) = \sum_{j=1}^{m} w_{ij}(k) x_j(k),$$

$$x_i(k+1) = \Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] \qquad \text{for all } i \in A_k,$$

and otherwise

$$x_i(k+1) = x_i(k).$$

Moreover, when the matrices $W(k)$ are stochastic, we have

$$x_i(k+1) \in X \qquad \text{for all } k \geq 0 \text{ and all } i \in [m].$$

There is an alternative view for the updates of $x_i(k+1)$ in (1.34) that will be useful in our analysis. Specifically, noting that $\chi_{\{i \notin A_k\}} = 1 - \chi_{\{i \in A_k\}}$, from the definition of $x_i(k+1)$ in (1.34) it follows that

$$x_i(k+1) = \chi_{\{i \in A_k\}} \Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] + \left(1 - \chi_{\{i \in A_k\}}\right) v_i(k+1). \tag{1.35}$$

Thus, we can view $x_i(k+1)$ as a convex combination of two points, namely, a convex combination of $\Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] \in X$ and $v_i(k+1)$. When $W(k)$ is a stochastic matrix, the point $v_i(k+1)$ is in the set X.

If the random gossip protocol is used for communications, then $W(k) = W_g(k)$. Similarly, if the agents communicate using the random broadcast protocol, then $W(k) = W_b(k)$. Thus, the random gossip and random broadcast algorithms can be viewed as a special case of a more general random communication model, where the weight matrices $W(k)$ are random, drawn independently in time from the same distribution, and have the properties as specified in the following assumption.

Assumption 4 *Let $\{W(k)\}$ be a sequence of $m \times m$ random i.i.d. matrices such that the following conditions are satisfied:*

(a) Each realization of $W(k)$ is a stochastic matrix compatible with the graph $G = ([m], E)$, i.e., $w_{ij}(k) > 0$ only if $j \in N_i$.

(b) The spectral norm of the matrix $Q = \mathbb{E}\left[W'(k)\left(I - \frac{1}{m}\mathbf{11}'\right)W(k)\right]$ is less than 1.

(c) The expected matrix $\mathbb{E}[W(k)] = \bar{W}$ is doubly stochastic.

In view of Assumption 4(a), the entry $\bar{W}_{ij}$ of the expected matrix may be positive only if $j \in N_i$. We do not assume explicitly that the graph G is connected, however, this property of the graph is subsumed within Assumption 4(b).

Note that the random matrices corresponding to the gossip and the broadcast model satisfy Assumption 4 when the graph G is connected. In fact, the random matrices corresponding to the gossip model in (1.31) satisfy a stronger condition than that of Assumption 4(a), since each realization of $W_g(k)$ is a doubly stochastic matrix.

Under Assumption 4(a), the event $\{i \in A_k\}$ that agent i updates (is awaken) at time $k+1$ has a stationary probability, denoted by p_i, i.e.,

$$p_i = \text{Prob}\{i \in A_k\}.$$

We now specify the stepsize rule for the algorithm. We consider the case when every agent i choses its stepsize value $\alpha_{i,k+1}$ based on its own local count of the update times. Letting $\Gamma_i(k+1)$ be the number of times the agent was awaken up to (including) time k, i.e.,

$$\Gamma_i(k+1) = \sum_{t=0}^{k} \chi_{\{i \in A_t\}},$$

we define the stepsize $\alpha_{i,k+1}$, as follows

$$\alpha_{i,k+1} = \frac{1}{\Gamma_i(k+1)} \qquad \text{for all } i \in [m] \text{ and } k \geq 0. \tag{1.36}$$

We note that

$$\Gamma_i(k+1) \geq \Gamma_i(k) \qquad \text{for all } k \geq 0 \text{ and } i \in [m],$$

implying that

$$\alpha_{i,k+1} \leq \alpha_{i,k} \qquad \text{for all } k \geq 0 \text{ and } i \in [m]. \tag{1.37}$$

In what follows, we will work with the conditional expectations with respect to the past iterates of the algorithm. For this, we let F_k denote the history of the algorithm (1.34), i.e.,

$$\mathsf{F}_k = \{W(0), \ldots, W(k-1)\} \qquad \text{for all } k \geq 1,$$

and $\mathsf{F}_0 = \emptyset$.

1.4.3 Convergence Analysis of Asynchronous Algorithm

We investigate the convergence properties of the algorithm assuming that the stepsize $\alpha_{i,k+1}$ is selected by agent i based on its local information. Prior to specifying the stepsize, we provide a result that is valid for any stepsize choice. It is also valid for any matrix sequence $\{W(k)\}$.

Lemma 7 *Assume that the problem is convex (i.e., Assumption 1 holds). Then, for the iterates of the algorithm (1.34) with any stepsize $\alpha_{i,k+1} > 0$ we have for all $x \in X$, for all $i \in [m]$ and all $k \geq 0$,*

$$\sum_{i=1}^{m} \|x_i(k+1) - x\|^2 \leq \sum_{i=1}^{m} \|v_i(k+1) - x\|^2 \\ - 2\sum_{i=1}^{m} \alpha_{i,k+1} \chi_{\{i \in A_k\}} \left(f_i(v_i(k+1)) - f_i(x)\right) \\ + \sum_{i=1}^{m} \alpha_{i,k+1}^2 \chi_{\{i \in A_k\}} \|g_i(k+1)\|^2.$$

Proof From the relation in (1.35) by the convexity of the squared norm, it follows that for any $x \in X$, all $k \geq 0$ and all $i \in [m]$,

$$\|x_i(k+1) - x\|^2 \leq \chi_{\{i \in A_k\}} \|\Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] - x\|^2 \\ + \left(1 - \chi_{\{i \in A_k\}}\right) \|v_i(k+1) - x\|^2.$$

By Lemma 4, for the point $\Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)]$ and any $x \in X$, we have

$$\|\Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] - x\|^2 \leq \|v_i(k+1) - x\|^2 \\ - 2\alpha_{i,k+1} \left(f_i(v_i(k+1)) - f_i(x)\right) + \alpha_{i,k+1}^2 \|g_i(k+1)\|^2.$$

By combining the preceding two relations, we obtain

$$\|x_i(k+1) - x\|^2 \leq \|v_i(k+1) - x\|^2 - 2\alpha_{i,k+1} \chi_{\{i \in A_k\}} \left(f_i(v_i(k+1)) - f_i(x)\right) \\ + \alpha_{i,k+1}^2 \chi_{\{i \in A_k\}} \|g_i(k+1)\|^2.$$

The desired relation follows by summing the preceding inequalities over $i \in [m]$.

□

We have the following refinement of Lemma 7 for the random stepsizes $\alpha_{i,k+1}$ given by (1.36), which are measurable with respect to F_k for all $i \in [m]$. The result is developed under the assumption that the set X is compact, which is used to bound the error induced by the asynchronous updates and, in particular, the error due to a different frequency of agents' updates. The result assumes that the matrix sequence $\{W(k)\}$ is just an i.i.d. sequence.

Proposition 2 *Let the problem be convex (Assumption 1) and, also, assume that the set X is bounded. Let the random matrix sequence $\{W(k)\}$ be i.i.d. Consider the iterates produced by method (1.34) with the random stepsizes $\alpha_{i,k+1}$ as given in (1.36). Then, with probability 1, we have for all $k \geq 0$ and all $x \in X$,*

$$\begin{aligned}\sum_{i=1}^m \mathbb{E}\left[\|x_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \leq & \sum_{i=1}^m \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \\ & - \frac{2}{k+1}\left(f(x_{av}(k)) - f(x)\right) + r_k,\end{aligned}$$

where

$$\begin{aligned} r_k =& 2CD \sum_{i=1}^m \left| \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]\right| \\ & + \frac{2C\sqrt{\sum_{i=1}^m \frac{1}{p_i}}}{k+1}\sqrt{\sum_{i=1}^m \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]} \\ & + 2\sqrt{m}CD\sqrt{\sum_{i=1}^m \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]} + C^2\alpha_{i,k}^2,\end{aligned}$$

with p_i denoting the probability of the event $\chi_{\{i\in A_k\}}$, C being the uniform upper bound on the subgradient norms of f_i over the set X, and $D = \max_{x,y\in X}\|x-y\|$.

Proof In view of the compactness of the set X, it follows that the subgradients of f_i are uniformly bounded over the set X for all i, i.e., there exists a constant C such that

$$\|s\| \leq C \qquad \text{for every subgradient } s \text{ of } f_i(z) \text{ at any } z \in X. \tag{1.38}$$

Therefore, each function f_i is Lipschitz continuous on X, so that for all $x \in X$, all $k \geq 0$, and all $i \in [m]$,

$$\begin{aligned} f_i(v_i(k+1)) - f_i(x) &= f_i(v_i(k+1)) - f_i(x_{av}(k)) + f_i(x_{av}(k)) - f_i(x) \\ &\geq -C\|v_i(k+1) - x_{av}(k)\| + f_i(x_{av}(k)) - f_i(x), \end{aligned}$$

where $x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k)$.

By using the preceding estimate in Lemma 7 and the fact that the subgradients are bounded, we obtain

$$\begin{aligned} &\sum_{i=1}^{m} \|x_i(k+1) - x\|^2 \\ &\quad \leq \sum_{i=1}^{m} \|v_i(k+1) - x\|^2 - 2\sum_{i=1}^{m} \alpha_{i,k+1}\chi_{\{i \in A_k\}} \left(f_i(x_{av}(k)) - f_i(x)\right) \\ &\qquad 2C\sum_{i=1}^{m} \alpha_{i,k+1}\chi_{\{i \in A_k\}}\|v_i(k+1) - x_{av}(k)\| + C^2\sum_{i=1}^{m} \alpha_{i,k+1}^2\chi_{\{i \in A_k\}}. \end{aligned}$$

We take the conditional expectation with respect to F_k in both sides of the preceding relation and further obtain, with probability 1, for all $x \in X$ and all $k \geq 0$,

$$\begin{aligned} \sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] &\leq \sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \\ &- 2\sum_{i=1}^{m} \mathbb{E}\left[\alpha_{i,k+1}\chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]\left(f_i(x_{av}(k)) - f_i(x)\right) \\ &+ 2C\sum_{i=1}^{m} \mathbb{E}\left[\alpha_{i,k+1}\chi_{\{i \in A_k\}}\|v_i(k+1) - x_{av}(k)\| \mid \mathsf{F}_k\right] \\ &+ C^2\sum_{i=1}^{m} \mathbb{E}\left[\alpha_{i,k+1}^2\chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]. \end{aligned} \tag{1.39}$$

Since $\alpha_{i,k+1}\chi_{\{i \in A_k\}} \leq \alpha_{i,k+1}$ and the stepsize is non-increasing (see (1.37)), it follows that

$$\alpha_{i,k+1}\chi_{\{i \in A_k\}} \leq \alpha_{i,k} \qquad \text{for all } i \in [m] \text{ and all } k \geq 0.$$

Hence, with probability 1,

$$\mathbb{E}\left[\alpha_{i,k+1}^2\chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right] \leq \mathbb{E}\left[\alpha_{i,k}^2 \mid \mathsf{F}_k\right] = \alpha_{i,k}^2, \tag{1.40}$$

where in the last equality we use the fact that $\alpha_{i,k}$ is completely determined given the past F_k. By substituting relation (1.40) in inequality (1.39), we obtain

$$\begin{aligned}\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1)-x\|^2 \mid \mathsf{F}_k\right] &\leq \sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1)-x\|^2 \mid \mathsf{F}_k\right] \\ &- 2\sum_{i=1}^{m} \mathbb{E}\left[\alpha_{i,k+1}\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right](f_i(x_{av}(k)) - f_i(x)) \\ &+ 2C\sum_{i=1}^{m} \mathbb{E}\left[\alpha_{i,k+1}\chi_{\{i\in A_k\}}\|v_i(k+1)-x_{av}(k)\| \mid \mathsf{F}_k\right] + C^2\alpha_{i,k}^2. \qquad (1.41)\end{aligned}$$

By adding and subtracting $2\sum_{i=1}^{m} \frac{\mathbb{E}[\chi_{\{i\in A_k\}}|\mathsf{F}_k]}{(k+1)p_i}(f_i(x_{av}(k)) - f_i(x))$ to the second term on the right hand side of (1.41), and by doing similarly with a corresponding expression for the third term, we have

$$\begin{aligned}\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1)-x\|^2 \mid \mathsf{F}_k\right] &\leq \sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1)-x\|^2 \mid \mathsf{F}_k\right] \\ &- 2\sum_{i=1}^{m} \frac{\mathbb{E}\left[\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]}{(k+1)p_i}(f_i(x_{av}(k)) - f_i(x)) \\ &+ 2\sum_{i=1}^{m} \left|\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right](f_i(x_{av}(k)) - f_i(x))\right| \\ &+ \frac{2C}{k+1}\underbrace{\sum_{i=1}^{m}\frac{1}{p_i}\mathbb{E}\left[\chi_{\{i\in A_k\}}\|v_i(k+1)-x_{av}(k)\| \mid \mathsf{F}_k\right]}_{T_1} \\ &+ 2C\underbrace{\left|\sum_{i=1}^{m}\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)\chi_{\{i\in A_k\}}\|v_i(k+1)-x_{av}(k)\| \mid \mathsf{F}_k\right]\right|}_{T_2} \\ &+ C^2\alpha_{i,k}^2, \qquad (1.42)\end{aligned}$$

where p_i is the probability of the event that agent i is updating.

To estimate the term T_1, we use Hölder's inequality

$$\sum_{i=1}^{r}\mathbb{E}\left[|a_i b_i|\right] \leq \sqrt{\sum_{i=1}^{r}\mathbb{E}\left[a_i^2\right]}\sqrt{\sum_{i=1}^{r}\mathbb{E}\left[b_i^2\right]},$$

and obtain

$$\begin{aligned} T_1 &\le \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\frac{1}{p_i^2}\chi_{\{i\in A_k\}}\right]} \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]} \\ &\le \sqrt{\sum_{i=1}^{m} \frac{1}{p_i}} \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]}, \end{aligned} \tag{1.43}$$

where in the first inequality we also use the fact that the event $\{i \in A_k\}$ is independent from the past, while in the last inequality we use the fact that the probability that the event $\{i \in A_k\}$ occurs is p_i.

For the term T_2 in (1.42), by Hölder's inequality, we have

$$\begin{aligned} T_2 \le &\sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]} \\ &\times \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]}. \end{aligned} \tag{1.44}$$

We now substitute the estimates (1.43) and (1.44) in the inequality (1.42) and obtain that, with probability 1, there holds for all $x \in X$ and $k \ge 0$,

$$\begin{aligned} &\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \le \sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \\ &\quad - 2\sum_{i=1}^{m} \frac{\mathbb{E}\left[\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]}{(k+1)p_i} (f_i(x_{av}(k)) - f_i(x)) \\ &\quad + 2\sum_{i=1}^{m} \left|\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right) \chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right] (f_i(x_{av}(k)) - f_i(x))\right| \\ &\quad + \frac{2C\sqrt{\sum_{i=1}^{m} \frac{1}{p_i}}}{k+1} \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]} \\ &\quad + 2C \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]} \\ &\quad \times \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]} + C^2\, \alpha_{i,k}^2. \end{aligned}$$

In view of compactness of X, and $x_{av}(k) \in X$ and $v_i(k+1) \in X$ for all i and k, it follows that for all $x \in X$,

$$|f_i(x_{av}(k)) - f_i(x)| \leq C\|x_{av}(k) - x\| \leq CD, \qquad \|v_i(k+1) - x_{av}(k)\| \leq D,$$

where $D = \max_{y,z \in X} \|y - z\|$. We also note that

$$\frac{\mathbb{E}\left[\chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]}{(k+1)p_i} = \frac{1}{k+1}.$$

By using the preceding relations, we have that, with probability 1, there holds for all $x \in X$ and $k \geq 0$,

$$\begin{aligned}
&\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \\
&\quad \leq \sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] - \frac{2}{k+1}\left(f(x_{av}(k)) - f(x)\right) \\
&\qquad + 2CD \sum_{i=1}^{m} \left| \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right) \chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]\right| \\
&\qquad + \frac{2C\sqrt{\sum_{i=1}^{m} \frac{1}{p_i}}}{k+1} \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]} \\
&\qquad + 2\sqrt{m}CD \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]} + C^2 \alpha_{i,k}^2.
\end{aligned}$$

The desired relation follows by introducing the notation for the sum of the last four terms on the right hand side of the preceding relation. □

To establish the convergence of the method, one of the goals is to show that the error terms r_k in Proposition 2 are well behaved in the sense that $\sum_{k=0}^{\infty} r_k < \infty$ with probability 1. We note that the error r_k has two types of terms, one type related to the stepsize and the other related to the distances of iterates $v_i(k+1)$ and the average vector $x_{av}(k)$, which also involves the stepsize implicitly. So we start by investigating some properties of the stepsize.

1.4.3.1 Stepsize Analysis

We consider the random agent based stepsize defined in (1.36), which is the inverse of the number $\Gamma_i(k+1)$ of agent i updates from time $t = 0$ up to time $t = k$, inclusively. We establish some relations for the stepsize that involve expectations and a set of results for the stepsize sums.

We start with the relations involving the expectations of the stepsize in the term r_k of Proposition 2.

Lemma 8 *Let the matrix sequence* $\{W(k)\}$ *be an i.i.d. random sequence. Then, for the stepsize* $\alpha_{i,k}$ *in* (1.36)*, with probability 1, we have for all* $k \geq 0$ *and* $i \in [m]$*,*

$$\left|\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]\right| \leq p_i\left|\alpha_{i,k} - \frac{1}{kp_i}\right| + (1-p_i)\frac{\alpha_{i,k}}{k},$$

$$\sqrt{\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]} \leq \sqrt{2p_i}\left|\alpha_{i,k} - \frac{1}{kp_i}\right| + (1-p_i)\sqrt{\frac{2}{p_i}}\frac{\alpha_{i,k}}{k}.$$

Proof Recall that the event $\chi_{\{i\in A_k\}}$ that agent i updates has probability p_i. Thus, using the independence of the event $\chi_{\{i\in A_k\}}$ given the past F_k, we have with probability 1 for all $k \geq 0$ and $i \in [m]$,

$$\mathbb{E}\left[\alpha_{i,k+1}\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right] = \frac{p_i}{\Gamma_i(k)+1}.$$

Using the preceding relation and $\mathbb{E}\left[\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right] = p_i$, we obtain

$$\begin{aligned} M_1 &:= \left|\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)\chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right]\right| \\ &= \left|p_i\left(\frac{1}{\Gamma_i(k)+1} - \frac{1}{(k+1)p_i}\right)\right| \\ &= p_i\frac{|kp_i - \Gamma_i(k) + p - 1|}{(k+1)p_i(\Gamma_i(k)+1)}, \end{aligned}$$

where the last equality is obtained by re-grouping the terms in the numerator. Thus, it follows that

$$M_1 \leq p_i\frac{|kp_i - \Gamma_i(k)| + (1-p_i)}{(k+1)p_i(\Gamma_i(k)+1)} \leq p_i\frac{|kp_i - \Gamma_i(k)| + (1-p_i)}{kp_i\Gamma_i(k)}.$$

By separating the terms, we have

$$M_1 \leq p_i\frac{|kp_i - \Gamma_i(k)|}{kp_i\Gamma_i(k)} + \frac{(1-p_i)}{k\Gamma_i(k)} = p_i\left|\frac{1}{\Gamma_i(k)} - \frac{1}{kp_i}\right| + \frac{(1-p_i)}{k\Gamma_i(k)}.$$

Recognizing that $\alpha_{i,k} = \frac{1}{\Gamma_i(k)}$, we obtain

$$M_1 \leq p_i \left|\alpha_{i,k} - \frac{1}{kp_i}\right| + (1-p_i)\frac{\alpha_{i,k}}{k},$$

thus showing the first relation stated in the lemma.

For the second relation we have

$$\begin{aligned} M_2 :=& \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i\in A_k\}} \mid \mathsf{F}_k\right] = p_i\left(\frac{1}{\Gamma_i(k)+1} - \frac{1}{(k+1)p_i}\right)^2 \\ =& p_i\left(\frac{(k+1)p_i - \Gamma_i(k) - 1}{(k+1)p_i(\Gamma_i(k)+1)}\right)^2 = p_i\frac{(kp_i - \Gamma_i(k) + p - 1)^2}{(k+1)^2 p_i^2(\Gamma_i(k)+1)^2}. \end{aligned}$$

Now using the relation $(a+b)^2 \leq 2(a^2+b^2)$, which is valid for any scalars a and b, we obtain

$$M_2 \leq 2p_i\frac{(kp_i - \Gamma_i(k))^2 + (1-p_i)^2}{(k+1)^2 p_i^2(\Gamma_i(k)+1)^2} \leq 2p_i\frac{(kp_i - \Gamma_i(k))^2 + (1-p_i)^2}{k^2 p_i^2 \Gamma_i^2(k)}.$$

By separating the terms we further have

$$M_2 \leq 2p_i\frac{(kp_i - \Gamma_i(k))^2}{k^2 p_i^2 \Gamma_i^2(k)} + \frac{2(1-p_i)^2}{k^2 p_i \Gamma_i^2(k)} = 2p_i\left(\frac{1}{\Gamma_i(k)} - \frac{1}{kp_i}\right)^2 + \frac{2(1-p_i)^2}{k^2 p_i \Gamma_i^2(k)}.$$

By substituting $\alpha_{i,k} = \frac{1}{\Gamma_i(k)}$, it follows that

$$M_2 \leq 2p_i\left(\alpha_{i,k} - \frac{1}{kp_i}\right)^2 + \frac{2(1-p_i)^2\alpha_{i,k}^2}{k^2 p_i}.$$

Using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, which is valid for any $a, b \geq 0$, we have

$$\begin{aligned} \sqrt{M_2} &\leq \sqrt{2p_i\left(\alpha_{i,k} - \frac{1}{kp_i}\right)^2} + \sqrt{\frac{2(1-p_i)^2\alpha_{i,k}^2}{k^2 p_i}} \\ &= \sqrt{2p_i}\left|\alpha_{i,k} - \frac{1}{kp_i}\right| + (1-p_i)\sqrt{\frac{2}{p_i}}\frac{\alpha_{i,k}}{k}, \end{aligned}$$

which establishes the second relation of the lemma. □

We next investigate some properties of the stepsize sums under the assumption that the random matrix sequence $\{W(k)\}$ is i.i.d. In this case, for each $i \in [m]$, the events $\{i \in A_k\}$ are i.i.d., so that we have

$$\mathbb{E}[\Gamma_i(k)] = (k+1)p_i.$$

By the law of iterated logarithms [41] (pages 476–479), we have that for any $q > 0$,

$$\text{Prob}\left\{\lim_{k\to\infty} \frac{|\Gamma_i(k) - (k+1)p_i|}{(k+1)^{\frac{1}{2}+q}} = 0\right\} = 1 \qquad \text{for all } i \in [m]. \tag{1.45}$$

We use this relation to establish some results for the sums involving the stepsize, as given is the following lemma.

Lemma 9 *Let the random matrix sequence $\{W(k)\}$ be i.i.d., and consider the stepsize $\alpha_{i,k}$ as given in (1.36). Then, we have*

$$\text{Prob}\left\{\sum_{k=1}^{\infty} \alpha_{i,k}^2 < \infty, \ \textit{for all } i \in [m]\right\} = 1,$$

$$\text{Prob}\left\{\sum_{k=1}^{\infty} \left|\alpha_{i,k} - \frac{1}{kp_i}\right| < \infty, \ \textit{for all } i \in [m]\right\} = 1,$$

$$\text{Prob}\left\{\sum_{k=1}^{\infty} \frac{\alpha_{i,k}}{k} < \infty, \ \textit{for all } i \in [m]\right\} = 1.$$

Proof The proof is based on considering sample paths, where a sample path corresponds to a sequence of realizations of the matrices, which is denoted by ω. We fix a sample path ω for which the limit in (1.45) is zero. Then, using relation (1.45), we can show that for every $q \in (0, \frac{1}{2})$ there exists an index[5] $\tilde{k}(\omega)$ such that for all $k \geq \tilde{k}(\omega)$ and for all $i \in [m]$, we have[6]

$$\alpha_{i,k}(\omega) \leq \frac{2}{kp_i}, \qquad \left|\alpha_{i,k}(\omega) - \frac{1}{kp_i}\right| \leq \frac{1}{k^{\frac{3}{2}-q}p_i^2}. \tag{1.46}$$

Thus, there holds for all $i \in [m]$,

$$\sum_{k\geq\tilde{k}(\omega)} \alpha_{i,k}^2(\omega) < \infty, \qquad \sum_{k\geq\tilde{k}(\omega)} \left|\alpha_{i,k}(\omega) - \frac{1}{kp_i}\right| < \infty,$$

[5] The index $\tilde{k}(\omega)$ also depends on q, but this dependence is suppressed in the notation.

[6] The derivation of the relations in (1.46) can be found in the proof of Lemma 3 in [100], where the analysis is to be performed on a sample path.

where the last relation holds due to $q \in (0, \frac{1}{2})$. Furthermore, we have for all $k \geq \tilde{k}(\omega)$ and for all $i \in [m]$,

$$\frac{\alpha_{i,k}(\omega)}{k} \leq \frac{2}{k^2 p_i},$$

implying that

$$\sum_{k \geq \tilde{k}(\omega)} \frac{\alpha_{i,k}(\omega)}{k} < \infty \qquad \text{for all } i \in [m].$$

Since the preceding relations are true for almost all but zero measure sample paths ω that satisfy relation (1.45), the stated results follow. □

1.4.3.2 Relation for Agents' Iterates and Their Averages

We now turn our attention to the disagreements $\|x_i(k) - x_{av}(k)\|$. We establish a relation for these disagreements which will be combined with Proposition 2 to assert the convergence behavior of the distances $\|x_i(k) - x^*\|$ for an optimal solution x^*.

We start by re-writing the iterations of the method (1.34), as follows:

$$\begin{aligned} v_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k) x_j(k), \\ x_i(k+1) &= v_i(k+1) + \underbrace{\left(\Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] - v_i(k+1)\right) \chi\{i \in A_k\}}_{\phi_i(k+1)}. \end{aligned}$$

Hence, for all $i \in [m]$ and $k \geq 0$,

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k) x_j(k) + \phi_i(k+1), \\ \phi_i(k+1) &= \left(\Pi_X[v_i(k+1) - \alpha_{i,k+1} g_i(k+1)] - v_i(k+1)\right) \chi\{i \in A_k\}, \quad (1.47) \\ v_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k) x_j(k). \end{aligned}$$

Thus, we perceive the iterates $x_i(k+1)$ as obtained through a perturbed random consensus algorithm with random perturbations $\phi_i(k+1)$.

Under Assumption 4(a) and (c), in the following lemma, we establish a relation for the iterates $x_i(k+1)$.

Lemma 10 *Let the matrices $W(k)$ satisfy Assumptions 4(a) and (b). Then, for the iterate process in (1.47), we have for all $k \geq 0$ with probability 1,*

$$\sqrt{\mathbb{E}\left[\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2 \mid \mathsf{F}_k\right]} \leq \sqrt{\sum_{i=1}^{m} \rho \|x_i(k) - x_{av}(k)\|^2} + \sqrt{\sum_{i=1}^{m} \|\phi_i(k+1)\|^2},$$

where $x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k)$ and $\rho \in (0,1)$ is the spectral norm of the matrix $Q = \mathbb{E}\left[W'(k)(I - \frac{1}{m}\mathbf{1}\mathbf{1}')W(k)\right]$ (see Assumption 4(b)).

Proof We first write the iterates $x_i(k+1)$ in (1.47) in a matrix form. We construct a matrix $\mathbf{X}(k)$ by placing the vectors $x_i'(k)$ in its rows, and similarly, we construct the matrix $\Phi(k)$ by placing the vectors $\phi_i'(k)$ in its rows. By doing so, we have the following representation for the evolution of the iterates $x_i(k+1)$:

$$\mathbf{X}(k+1) = W(k)\mathbf{X}(k) + \Phi(k+1) \qquad \text{for all } k \geq 0. \tag{1.48}$$

By multiplying both sides of (1.48) with the matrix $\frac{1}{m}\mathbf{1}\mathbf{1}'$, we have

$$\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1) = \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\mathbf{X}(k) + \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1).$$

By subtracting the preceding relation from (1.48), we obtain for all $k \geq 0$,

$$\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1) = \left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\mathbf{X}(k) + \Phi(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1).$$

Since $W(k)$ is stochastic, we have $W(k)\mathbf{1} = \mathbf{1}$, implying that $\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)$ $\mathbf{1} = 0$, so that we have

$$\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\mathbf{X}(k) = \left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right).$$

Therefore, for all $k \geq 0$,

$$\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1) = \left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right) + \Phi(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1). \tag{1.49}$$

By taking the squared Frobenius norms of both sides in (1.49), and using the fact that $\|AB\|_F \le \|A\|_F\|B\|_F$ for any two (compatible) matrices, we obtain for all $k \ge 0$,

$$\begin{aligned}
&\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 \\
&\quad \le \left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \\
&\qquad + \left\|\Phi(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \\
&\qquad + 2\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F \\
&\qquad \times \left\|\Phi(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F.
\end{aligned}$$

Next, we take conditional expectation with respect to F_k and obtain for all $k \ge 0$ with probability 1,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\quad \le \mathbb{E}\left[\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\qquad + \mathbb{E}\left[\left\|\Phi(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\qquad + 2\mathbb{E}\left[\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F\right. \\
&\qquad \left.\times \left\|\Phi(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F \mid \mathsf{F}_k\right].
\end{aligned}$$

By using Hölder's inequality for expectations (see [8], page 242), we can see that

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\quad \le \mathbb{E}\left[\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right]
\end{aligned}$$

$$+\,\mathbb{E}\left[\left\|\Phi(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]$$
$$+\,2\sqrt{\mathbb{E}\left[\left\|\left(W(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right]}$$
$$\times\sqrt{\mathbb{E}\left[\left\|\Phi(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]}.$$

Hence, for all $k \geq 0$ with probability 1 we have

$$\mathbb{E}\left[\left\|\mathbf{X}(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]$$
$$\leq\left(\sqrt{\mathbb{E}\left[\left\|\left(W(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right]}\right.$$
$$\left.+\sqrt{\mathbb{E}\left[\left\|\Phi(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]}\right)^2. \tag{1.50}$$

Next, using the column vectors $\mathbf{x}^{\ell}(k)$ of the matrix $X(k)$ and the definition of the Frobenius norm, we can write

$$\left\|\left(W(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2$$
$$=\sum_{\ell=1}^{m}\left\|\left(W(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{x}^{\ell}(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^{\ell}(k)\right)\right\|^2$$
$$=\sum_{\ell=1}^{m}\left\langle\mathbf{x}^{\ell}(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^{\ell}(k),\,Q(k)\left(\mathbf{x}^{\ell}(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^{\ell}(k)\right)\right\rangle,$$

where

$$Q(k)=\left(W(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)'\left(W(k)-\frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right).$$

After some elementary algebra, it can be seen that for $Q(k)$ we have

$$Q(k) = W(k)'W(k) - \frac{1}{m}W'(k)\mathbf{1}\mathbf{1}'W(k) = W'(k)\left(I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)W(k).$$

Given the history F_k, the column vectors $\mathbf{x}^\ell(k)$ are deterministic, so that by using $Q = \mathbb{E}[Q(k)]$ (see Assumption 4(b)), we have with probability 1 for all $k \geq 0$,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\quad = \sum_{\ell=1}^m \left\langle \mathbf{x}^\ell(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^\ell(k), \mathbb{E}[Q(k)]\left(\mathbf{x}^\ell(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^\ell(k)\right)\right\rangle \\
&\quad \leq \sum_{\ell=1}^m \|Q\| \left\|\mathbf{x}^\ell(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^\ell(k)\right\|^2 \\
&\quad \leq \rho \sum_{\ell=1}^m \left\|\mathbf{x}^\ell(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{x}^\ell(k)\right\|^2 \\
&\quad = \rho \left\|\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right\|_F^2,
\end{aligned}$$

where $\rho \in (0, 1)$ is the spectral norm of the matrix Q (note that ρ is smaller than 1 by Assumption 4(b)). Hence, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\quad \leq \rho \left\|\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right\|_F^2.
\end{aligned}$$

Finally by noticing that the matrix $\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)$ has identical rows, where each its row is given by the vector $\frac{1}{m}\mathbf{1}'\mathbf{X}(k) = x_{av}'(k)$ with $x_{av}(k) = \frac{1}{m}\sum_{j=1}^m x_j(k)$, we see that

$$\begin{aligned}
&\mathbb{E}\left[\left\|\left(W(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'W(k)\right)\left(\mathbf{X}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k)\right)\right\|_F^2 \mid \mathsf{F}_k\right] \\
&\quad \leq \rho \left\|\mathbf{X}(k) - \mathbf{1}x_{av}'(k)\right\|_F^2 \\
&\quad = \rho \sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2. \qquad (1.51)
\end{aligned}$$

Similarly, using the definition of the Frobenius norm and the matrices $\Phi(k)$, we can see that

$$\mathbb{E}\left[\left\|\Phi(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]$$
$$=\mathbb{E}\left[\sum_{j=1}^{m}\left\|\phi_j(k+1)-\phi_{av}(k+1)\right\|^2 \mid \mathsf{F}_k\right],$$

where $\phi_{av}(k) = \sum_{\ell=1}^{m}\phi_\ell(k)$. Since the distance $\sum_{j=1}^{m}\|\phi_i(k+1)-y\|^2$ is minimized over all $y \in \mathbb{R}^n$ at $y^* = \phi_{av}(k+1)$, we obtain (using $y=0$)

$$\mathbb{E}\left[\left\|\Phi(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\Phi(k+1)\right\|_F^2 \mid \mathsf{F}_k\right] \leq \mathbb{E}\left[\sum_{j=1}^{m}\left\|\phi_j(k+1)\right\|^2 \mid \mathsf{F}_k\right]. \tag{1.52}$$

Using relations (1.51) and (1.52) in inequality (1.50), we have for all $k \geq 0$ with probability 1,

$$\mathbb{E}\left[\left\|\mathbf{X}(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]$$
$$\leq \left(\sqrt{\rho\sum_{j=1}^{m}\left\|x_j(k)-x_{av}(k)\right\|^2}+\sqrt{\mathbb{E}\left[\sum_{j=1}^{m}\left\|\phi_j(k+1)\right\|^2 \mid \mathsf{F}_k\right]}\right)^2.$$

Hence, by taking square roots on both sides, we see that

$$\sqrt{\mathbb{E}\left[\left\|\mathbf{X}(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 \mid \mathsf{F}_k\right]}$$
$$\leq \sqrt{\rho\sum_{j=1}^{m}\left\|x_j(k)-x_{av}(k)\right\|^2}+\sqrt{\mathbb{E}\left[\sum_{j=1}^{m}\left\|\phi_j(k+1)\right\|^2 \mid \mathsf{F}_k\right]}. \tag{1.53}$$

The desired relation follows from (1.53) by using the fact that

$$\left\|\mathbf{X}(k+1)-\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F^2 = \sum_{i=1}^{m}\|x_i(k+1)-x_{av}(k+1)\|^2.$$

□

Based on Lemma 10, we can prove that the agents' disagreements are well behaved. The proof makes use of an (almost) supermartingale convergence result. The result is due to Robbins and Siegmund [136], and it can also be found in [129] (Chapter 2.2, Lemma 11).

Lemma 11 ([136]) *Let V_k, u_k, β_k and γ_k be non-negative random variables adapted to some σ-algebra σ_k. If with probability 1 we have $\sum_{k=0}^{\infty} u_k < \infty$, $\sum_{k=0}^{\infty} \beta_k < \infty$, and*

$$\mathbb{E}\left[V_{k+1} \mid \sigma_k\right] \leq (1+u_k)V_k - \gamma_k + \beta_k \qquad \textit{for all } k \geq 0,$$

then V_k converges to some non-negative scalar and $\sum_{k=0}^{\infty} \gamma_k < \infty$ with probability 1.

We have the following result for the disagreement sum $\sum_{j=1}^{m} \left\|x_j(k) - x_{av}(k)\right\|^2$.

Lemma 12 *Let the problem be convex (Assumption 1 holds) and let the set X be bounded. Let the matrices $W(k)$ satisfy Assumption 4(a) and (b). Then, with probability 1,*

$$\sum_{k=0}^{\infty} \frac{1}{k+1} \sqrt{\sum_{i=1}^{m} \|x_i(k) - x_{av}(k)\|^2} < \infty,$$

$$\liminf_{k\to\infty} \sum_{j=1}^{m} \left\|x_j(k) - x_{av}(k)\right\|^2 = 0.$$

Proof By Lemma 10 and by using the relation $\mathbb{E}[a] \leq \sqrt{\mathbb{E}\left[a^2\right]}$, we have with probability 1 for all $k \geq 0$,

$$\mathbb{E}\left[\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} \mid \mathsf{F}_k\right] \leq \sqrt{\rho \sum_{j=1}^{m} \left\|x_j(k) - x_{av}(k)\right\|^2} + \sqrt{\mathbb{E}\left[\sum_{j=1}^{m} \left\|\phi_j(k+1)\right\|^2 \mid \mathsf{F}_k\right]}, \tag{1.54}$$

with $\rho \in (0, 1)$. From the definition of $\phi_j(k+1)$ in (1.47) and the non-expansiveness property of the projection operator, we can se that

$$\|\phi_i(k+1)\|^2 \le \alpha_{i,k+1}^2 \|g_i(k+1)\|^2 \chi\{i \in A_k\} \le \alpha_{i,k+1}^2 C^2 \chi\{i \in A_k\},$$

where the last inequality follows from the compactness of X, which implies that the subgradients of each f_i are uniformly bounded over the set X. Therefore, we have with probability 1,

$$\begin{aligned}\mathbb{E}\left[\|\phi_i(k+1)\|^2 \mid F_k\right] \le C^2 \mathbb{E}\left[\alpha_{i,k+1}^2 \chi\{i \in A_k\}\right] &= \frac{C^2 p_i}{(\Gamma_i(k)+1)^2} \le \frac{C^2 p_i}{\Gamma_i^2(k)} \\ &= C^2 p_i \alpha_{i,k}^2,\end{aligned}$$

implying that

$$\sqrt{\mathbb{E}\left[\sum_{j=1}^m \left\|\phi_j(k+1)\right\|^2 \mid \mathsf{F}_k\right]} \le C\sqrt{\sum_{i=1}^m p_i \alpha_{i,k}^2} \le C \sum_{i=1}^m \sqrt{p_i}\alpha_{i,k},$$

where the last inequality follows from relation $\sum_{i=1}^r a_i^2 \le (\sum_{i=1}^r a_i)^2$ which holds for any nonnegative scalars a_i. By using the preceding estimate in relation (1.54), we obtain with probability 1 for all $k \ge 0$,

$$\begin{aligned}&\mathbb{E}\left[\sqrt{\sum_{i=1}^m \|x_i(k+1) - x_{av}(k+1)\|^2 \mid \mathsf{F}_k}\right] \\ &\quad \le \sqrt{\rho \sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2} + C\sum_{i=1}^m \sqrt{p_i}\alpha_{i,k}.\end{aligned} \tag{1.55}$$

We divide both sides of (1.55) by $\frac{1}{k+1}$ and obtain with probability 1 for all $k \ge 0$,

$$\begin{aligned}&\frac{1}{k+1}\mathbb{E}\left[\sqrt{\sum_{i=1}^m \|x_i(k+1) - x_{av}(k+1)\|^2 \mid \mathsf{F}_k}\right] \\ &\quad \le \frac{\sqrt{\rho}}{k+1}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2} + C\sum_{i=1}^m \frac{\sqrt{p_i}\alpha_{i,k}}{k+1}.\end{aligned}$$

We add and subtract $\frac{1}{k}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2}$ to the right hand side of the preceding relation, so we can write

$$\frac{1}{k+1}\mathbb{E}\left[\sqrt{\sum_{i=1}^m \|x_i(k+1) - x_{av}(k+1)\|^2} \mid \mathsf{F}_k\right]$$
$$\leq \frac{1}{k}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2} - \left(\frac{1}{k} - \frac{\sqrt{\rho}}{k+1}\right)\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2}$$
$$+ C\sum_{i=1}^m \frac{\sqrt{p_i}\alpha_{i,k}}{k+1}.$$

We note that

$$\frac{1}{k} - \frac{\sqrt{\rho}}{k+1} = \frac{k+1-\sqrt{\rho}k}{k(k+1)} = \frac{k(1-\sqrt{\rho})+1}{k(k+1)} \geq \frac{(1-\sqrt{\rho})}{(k+1)},$$

where we use the fact that $\rho \in (0, 1)$. Therefore, we have with probability 1 for all $k \geq 0$,

$$\frac{1}{k+1}\mathbb{E}\left[\sqrt{\sum_{i=1}^m \|x_i(k+1) - x_{av}(k+1)\|^2} \mid \mathsf{F}_k\right] \leq \frac{1}{k}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2}$$
$$- \frac{1-\sqrt{\rho}}{k+1}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2} + C\sum_{i=1}^m \frac{\sqrt{p_i}\alpha_{i,k}}{k+1}. \tag{1.56}$$

We now apply Lemma 11 with the following identification

$$V_k = \frac{1}{k}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2}, \quad u_k = 0, \quad \beta_k = C\sum_{i=1}^m \frac{\sqrt{p_i}\alpha_{i,k}}{k+1},$$
$$\gamma_k = \frac{1-\sqrt{\rho}}{k+1}\sqrt{\sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2}.$$

Note that by Lemma 9 we have

$$\sum_{k=1}^\infty \sum_{i=1}^m \frac{\sqrt{p_i}\alpha_{i,k}}{k+1} < \infty \quad \text{with probability 1,}$$

so that the condition $\sum_{k=0}^{\infty}\beta_k < \infty$ with probability 1 is also satisfied. Thus, by the Robinson-Siegmund (almost) supermartingale convergence result in Lemma 11, from (1.56) we have that

$$\sum_{k=0}^{\infty}\frac{1-\sqrt{\rho}}{k+1}\sqrt{\sum_{j=1}^{m}\left\|x_j(k)-x_{av}(k)\right\|^2} < \infty \quad \text{with probability 1.}$$

Since $1-\sqrt{\rho} > 0$, the preceding relation implies that with probability 1,

$$\sum_{k=0}^{\infty}\frac{1}{k+1}\sqrt{\sum_{j=1}^{m}\left\|x_j(k)-x_{av}(k)\right\|^2} < \infty, \tag{1.57}$$

$$\liminf_{k\to\infty}\sum_{j=1}^{m}\left\|x_j(k)-x_{av}(k)\right\|^2 = 0,$$

where the last relation can be shown by arguing per sample path ω for which relation (1.57) is valid. □

1.4.3.3 Convergence Result for Asynchronous Algorithm

We are now in position to assert the convergence of the iterates of the algorithm (1.34) with probability 1, as stated in the following theorem. The proof is based on Proposition 2 and the almost supermartingale convergence result as given in Lemma 11. To verify that the conditions of Lemma 11 are satisfied, we rely on Lemmas 8, 9 and 12.

Theorem 4 *Let the problem be convex (Assumption 1) and, also, assume that the set X is bounded. Let the random matrix sequence $\{W(k)\}$ satisfy Assumption 4. Consider the iterates produced by method (1.34) with the random stepsizes $\alpha_{i,k+1}$ as given in (1.36). Then, with probability 1, the sequences $\{x_i(k)\}$, $i \in [m]$, converge to a common (random) optimal solution of the problem.*

Proof The proof has two main parts: (i) proving that $x_i(k) - x_{av}(k)$ converges to 0, as k tends to infinity, with probability 1 for all i, and (ii) proving that the iterate sequences $\{x_i(k)\}$, $i \in [m]$, converge to the same (random) optimal solution[7] with probability 1. We start by deriving a relation that will be used in both parts (i) and (ii) of the proof.

[7] Different sample paths may converge to different solutions.

By Proposition 2 we have with probability 1 for all $k \geq 0$ and all $x \in X$,

$$\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \leq \sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] - \frac{2}{k+1}\left(f(x_{av}(k)) - f(x)\right) + r_k, \tag{1.58}$$

with

$$\begin{aligned} r_k =& 2CD \sum_{i=1}^{m} \left| \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right) \chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]\right| \\ &+ \frac{2C\sqrt{\sum_{i=1}^{m} \frac{1}{p_i}}}{k+1} \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]} \\ &+ 2\sqrt{m}CD \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]} \\ &+ C^2 \alpha_{i,k}^2, \end{aligned} \tag{1.59}$$

where p_i is the probability of the event $\chi_{\{i \in A_k\}}$, C is a uniform upper bound on the subgradient norms of f_i over the set X, and $D = \max_{x,y \in X} \|x - y\|$. By the definition, $v_i(k+1)$ is a convex combination of $x_j(k)$, $j \in [m]$, so that by the convexity of the norm and the assumption that $W(k)$ is a stochastic matrix (see Assumption 4(a)), we have for any $x \in \mathbb{R}^n$,

$$\|v_i(k+1) - x\|^2 \leq \sum_{j=1}^{m} w_{ij}(k) \|x_j(k) - x\|^2,$$

implying that

$$\begin{aligned} \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] &\leq \sum_{j=1}^{m} \mathbb{E}\left[w_{ij}(k)\|x_j(k) - x\|^2 \mid \mathsf{F}_k\right] \\ &= \sum_{j=1}^{m} \mathbb{E}\left[w_{ij}(k)\right] \|x_j(k) - x\|^2 \\ &= \sum_{j=1}^{m} \bar{W}_{ij} \|x_j(k) - x\|^2. \end{aligned}$$

By summing the preceding inequalities over $i \in [m]$ and by using the assumption that $\bar{W}$ doubly stochastic (cf. Assumption 4(c)), we obtain for any $x \in \mathbb{R}^n$,

$$\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \leq \sum_{j=1}^{m} \|x_j(k) - x\|^2. \tag{1.60}$$

By substituting (1.60) in relation (1.58), we obtain with probability 1 for all $k \geq 0$ and all $x \in X$,

$$\begin{aligned}&\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x\|^2 \mid \mathsf{F}_k\right] \\ &\quad \leq \sum_{j=1}^{m} \|x_j(k) - x\|^2 - \frac{2}{k+1}\left(f(x_{av}(k)) - f(x)\right) + r_k. \end{aligned}\tag{1.61}$$

We will use relation (1.61) with $x = x_{av}(k)$ and $x = x^*$ for an arbitrary $x^* \in X^*$. In either case, we will use the almost supermartingale convergence result of Lemma 11. To use Lemma 11, we need to verify that $\sum_{k=0}^{\infty} r_k < \infty$ with probability 1. For this, we break down r_k in its terms, as follows:

$$\begin{aligned} r_k =& \underbrace{2CD \sum_{i=1}^{m} \left|\mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right) \chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]\right|}_{r_{1,k}} \\ &+ 2\sqrt{m}CD \underbrace{\sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\left(\alpha_{i,k+1} - \frac{1}{(k+1)p_i}\right)^2 \chi_{\{i \in A_k\}} \mid \mathsf{F}_k\right]}}_{r_{2,k}} \\ &+ \underbrace{\frac{2C\sqrt{\sum_{i=1}^{m} \frac{1}{p_i}}}{k+1} \sqrt{\sum_{i=1}^{m} \mathbb{E}\left[\|v_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right]}}_{r_{3,k}} + C^2 \alpha_{i,k}^2. \end{aligned}\tag{1.62}$$

By Lemmas 8 and 9 we can see that

$$\sum_{k=0}^{\infty} r_{1,k} < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} r_{2,k} < \infty \qquad \text{with probability 1.}$$

By Lemma 12 it follows that

$$\sum_{k=0}^{\infty} r_{3,k} < \infty \qquad \text{with probability 1,}$$

while by Lemma 9 we have

$$\sum_{k=0}^{\infty} \alpha_{i,k}^2 < \infty \qquad \text{with probability 1.}$$

Hence, from the preceding relations and relation (1.62) we have

$$\sum_{k=0}^{\infty} r_k < \infty \qquad \text{with probability 1.} \tag{1.63}$$

(i) We now use relation (1.61) with $x = x_{av}(k)$ and, thus, obtain with probability 1 for all $k \geq 0$,

$$\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x_{av}(k)\|^2 \mid \mathsf{F}_k\right] \leq \sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2 + r_k. \tag{1.64}$$

By noting that $\sum_{i=1}^m \|x_i(k+1) - x_{av}(k+1)\|^2 \leq \sum_{i=1}^m \|x_i(k+1) - y\|^2$ for any $y \in \mathbb{R}^n$, we have $\sum_{i=1}^m \|x_i(k+1) - x_{av}(k+1)\|^2 \leq \sum_{i=1}^m \|x_i(k+1) - x_{av}(k)\|^2$. Thus relation (1.64) implies that with probability 1 for all $k \geq 0$,

$$\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x_{av}(k+1)\|^2 \mid \mathsf{F}_k\right] \leq \sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2 + r_k. \tag{1.65}$$

By the almost supermartingale convergence of Lemma 11, we conclude that

$$\left\{\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2\right\} \quad \text{is convergent with probability 1.}$$

Since by Lemma 12 $\liminf_{k\to\infty} \sum_{j=1}^m \left\|x_j(k) - x_{av}(k)\right\|^2 = 0$, we must have that

$$\lim_{k\to\infty} \sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2 = 0 \;\; \text{with probability 1.} \tag{1.66}$$

(ii) Since the set X is compact, the optimal set X^* must be nonempty. Thus, by letting $x = x^*$ in relation (1.61) for any $x^* \in X^*$, we have with probability 1 for all $k \geq 0$ and all $x^* \in X^*$,

$$\sum_{i=1}^{m} \mathbb{E}\left[\|x_i(k+1) - x^*\|^2 \mid \mathsf{F}_k\right]$$
$$\leq \sum_{j=1}^{m} \|x_j(k) - x^*\|^2 - \frac{2}{k+1}\left(f(x_{av}(k)) - f^*\right) + r_k.$$

By the almost supermartingale convergence of Lemma 11, it follows that

$$\left\{\sum_{j=1}^{m} \|x_j(k) - x^*\|^2\right\} \text{ is convergent with probability 1 for all } x^* \in X^*, \tag{1.67}$$

$$\sum_{k=0}^{\infty} \frac{2}{k+1}\left(f(x_{av}(k)) - f^*\right) < \infty \quad \text{with probability 1.}$$

Thus, with probability 1 we must have

$$\liminf_{k\to\infty}\left(f(x_{av}(k)) - f^*\right) = 0.$$

Since X is bounded so is the sequence $\{x_{av}(k)\}$, and we can choose a subsequence $\{x_{av}(k_\ell)\}$ such that, with probability 1, $\lim_{\ell\to\infty} f(x_{av}(k_\ell)) = f^*$, $x_{av}(k_\ell) \to \tilde{x}$ where $\tilde{x} \in X$. By continuity of f, it follows that $f(\tilde{x}) = f^*$, i.e., $\tilde{x} \in X^*$ with probability 1.

Since we have shown that $\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2 \to 0$ with probability 1 (see relation (1.66)), it follows that

$$\lim_{\ell\to\infty} \sum_{j=1}^{m} \|x_j(k_\ell) - \tilde{x}\|^2 = 0 \quad \text{with probability 1.}$$

By using relation (1.67) with $x^* = \tilde{x}$, we find that $\lim_{k\to\infty} \sum_{j=1}^{m} \|x_j(k) - \tilde{x}\|^2 = 0$ with probability 1. □

Theorem 4 shows that the random updates driven by an i.i.d. matrix sequence $\{W(k)\}$, where agents are updating using their own stepsizes (based on their frequency of updates) leads to a convergence to some consensual (random) solution of the problem. Theorem 4 includes as special cases the random gossip and random broadcast methods, by setting $W(k) = W_g(k)$ and $W(k) = W_b(k)$. It can accommodate other asynchronous approaches whose corresponding matrices $W(k)$ satisfy Assumption 4.

Prior work on convex multi-agent problems that explored the use of random gossip method, with frequency-based stepsize, can be found in [131], while the random broadcast method with such a stepsize has been proposed and studied in [100]. Error rate results for these methods with a constant stepsize can be found in [132] (see also [130]). Another related work on distributed convex optimization that considers random matrices is reference [86].

1.5 Literature Overview and New Research Directions

1.5.1 Literature on Consensus Algorithms

The consensus algorithms have drawn a renewed interest in more recent years, which was inspired by the work [60]. There has been a large body of work on the stability property and the convergence behavior of consensus dynamic in continuous-time [98, 118, 119, 134, 135], including the variants with switching network topology and time delays. Also, there are many papers investigating convergence of consensus algorithm in discrete-time, such as [21–24, 97, 99, 135, 180–182], where different aspects are also considered including time-switching network topology and asynchrony. The convergence rate properties have been studied in [122, 124] (for a more detailed overview see [102]).

Consensus has been used in rendezvous problems [78, 79], flocking [11], opinion dynamics [90, 91], and parameter estimation [65]. There are several Ph.D. theses that provide a comprehensive study of the averaging dynamics and its properties for deterministic and random matrices, including [12, 52, 120, 144, 155, 156]. Moreover, various communication effects have been investigated such as the effects of the delay [9, 10, 107], quantization effects [20, 25, 26, 43, 66, 71, 110], and the effects of link failures and noise [56, 63, 64, 125, 127, 128, 158]. A question of (non)existence of quadratic Lyapunov function for consensus dynamic has been investigated in [123, 163], while in [53, 54, 166] some cut-related properties have been explored.

Consensus algorithms implemented in a network using a gossip-based or a broadcast-based communications have been studied in [2, 3, 16, 82], while a different consensus algorithm (the so called push-sum method) has been considered in [7, 67]. Paper [93] explores a connection between consensus problems and potential games. An averaging dynamic for opinion spread has been proposed in [51] (leading to a formation of agents' groups with each group reaching its own consensus value).

The robustness of consensus including robustness to adversarial agents, faulty nodes, resilience and privacy preserving are studied in [29, 72, 149–151, 185]. Fast convergence of consensus algorithms in networks with quantized communications has been recently investigated in [4, 30]. There are also recent works on complex consensus [84], consensus stability [83, 85], and the use of semi-norms [32, 81] to study the stability. The best known convergence rate (in terms of its dependence on the number of agents) has been accomplished in a recent paper [121], which provides a consensus algorithm inspired by the fast Nesterov method [117].

There is a stream of work focused on the consensus problem over random graphs and weighted (random) averaging dynamics [5, 13, 27, 45, 50, 55–58, 126, 152, 153, 156, 157, 159–162, 164–166]. One may refer to survey paper [38] for a detailed account of gossip algorithms and their applications to signal processing in sensor networks. An application of an asynchronous gossip algorithm to the problem of spectral ranking has been explored in [14], while a nonlinear gossip is investigated in [95]. A monograph [138] discusses applications of consensus-based approaches to parameter estimation, learning and adaptation in networks, while monograph [102] discusses weighted-averaging algorithms for solving constrained and unconstrained consensus problems over time-varying (deterministic) graphs.

1.5.2 Literature on Distributed Methods for Optimization in Networks

1.5.2.1 Weighted Averaging-Based Approaches

The approaches that use consensus models with stochastic matrices are often referred to as weighted-averaging methods. Thus, both algorithms (1.9) and (1.34) are based on weighted-averaging methods, where the former uses deterministic weights while the latter employs random weights.

There are various extensions and modifications to the algorithm (1.9) which were developed over the past few years. The early work on consensus-based optimization can be found in [171], where the agents share a common objective function. The first work on distributed optimization in a network with agent based local objective functions can be found in [88, 106, 108]. In [106, 108] a slightly different algorithm has been considered (with a fixed stepsize), namely

$$x_i(k+1) = \sum_{j=1}^{m} w_{ij}(k)x_j(k) - \alpha_k d_i(k), \tag{1.68}$$

where $d_i(k)$ is a subgradient of $f_i(x)$ at $x = x_i(k)$. The convergence rate of this algorithm has been investigated in [105]. An extension of this algorithm to the case of quantized messages has been investigated in [109], while its implementation over random networks has been studied in [86]. In [89, 172] both of these alternative approaches, namely, algorithm (1.9) and (1.68) have been studied for distributed estimation.

A variant of the distributed optimization problem, where agents want to solve the problem of minimizing $\sum_{i=1}^{m} f_i(x)$ over $X = \cap_{i=1}^{m} X_i$, with each agent i handling its own function f_i and its constraint set X_i, has been considered in [111]. To solve the problem, a modification of the algorithm (1.9) has been proposed, where each agent i updates using the projection on its constraint set X_i, instead of the common set X. Thus, agent i updates assume the following form:

$$x_i(k+1) = \Pi_{X_i}\left[\sum_{j=1}^{m} w_{ij}(k)x_j(k) - \alpha_k g_i(k)\right],$$

where $g_i(k)$ is a subgradient of $f_i(x)$ at $x = \sum_{j=1}^{m} w_{ij}(k)x_j(k)$. This algorithm has been studied in [73–75] for synchronous updates over time-varying graphs and for gossip-based asynchronous updates over a static graph. A different variant of this algorithm (using the Laplacian formulation of the consensus problem) for distributed optimization with distributed constraints in noisy networks has been studied in [144–146]. Distributed algorithms for special quadratic convex problems arising in parameter estimation in sensor networks have been developed and studied in [28, 33, 88, 89, 172, 173].

The consensus-based algorithms for other types of network objective functions have been considered in [133]. A Bregman-distance based distributed algorithm has been developed in [147], as well as consensus-based algorithms for solving certain min-max problems. Distributed algorithms, both synchronous and asynchronous, for solving special type of games (aggregative games) have been studied in [68].

A distributed dual Nesterov algorithm has been proposed in [40], while a distributed algorithm using the gradient differences has been recently proposed in [142]. A distributed algorithm that is based on the idea of preserving an optimality condition at every stage of the algorithm has been proposed in [92]. Distributed convex optimization algorithms for weight-balanced directed graphs have been investigated in continuous-time [48].

A different type of a distributed algorithm for convex optimization has been proposed in [77], where each agent keeps an estimate for all agents' decisions. This algorithm solves a problem where the agents have to minimize a global cost function $f(x_1, \ldots, x_m)$ while each agent i can control only its variable x_i. The algorithm of [77] has been recently extended to the online optimization setting in [76, 113].

Distributed algorithms using augmented Lagrangian approach with gossip-type communications have been studied in [61], while accelerated versions of distributed gradient methods have been proposed and studied in [62].

A consensus-based algorithm for solving problems with a separable constraint structure and the use of primal-dual distributed methods has been studied in [147, 189, 190], while a distributed primal-dual approach with perturbations has been explored in [31]. Work [176] provides algorithms for centralized and distributed convex optimization from control perspective, while [175] considers an event-triggered distributed optimization for sensor networks. In [19], a distributed simplex algorithm has been developed for linear programming problems, while a Newton-

Raphson consensus-based method has been proposed in [186] for distributed convex problems.

All of the prior work relies on the use of state-independent weights, i.e., the weights that do not depend on agents' iterates. A consensus-based algorithm employing state-dependent weights have been proposed and analyzed in [87].

The application of distributed methods to hypothesis testing problems in graphs has recently attracted attention, resulting in a stream of papers [70, 112, 139, 140]. While these works deal mainly with finitely many hypothesis, a recent paper [114] extends the framework to the case of infinitely many hypotheses. Paper [116] considers several algorithms that use different types of consensus models, namely, weighted-averaging as well as push-sum models, which are discussed in the next paragraph with some more details.

1.5.2.2 Push-Sum Based Approaches

Another class of distributed algorithms has recently been developed that employs a different type of consensus strategy known as push-sum model. It has also been referred to as doubly-linear iteration or ratio consensus algorithm, due to its form which involves ratio of two variables that evolve according to the same linear dynamics, but differ in the choice of the initial point. This algorithm was originally proposed in [67] for consensus problem over a static network (in a random gossip-based form), and has been recently investigated in [39] in a deterministic setting. It has been extended to time-varying networks in [7].

The first work that has employed the push-sum consensus model to develop distributed optimization methods is [169], which has been further investigated in [167, 168, 170]. This work has been focused on static graphs and it has been proposed as an alternative to the algorithm based on weighted averages in order to eliminate deadlocks and synchronization issues among others. The prior work also offers a push-sum algorithm that can deal with constraints by using Nesterov dual-averaging approach. Recently, this push-sum consensus-based algorithm has been extended to a *subgradient-push* algorithm in [103, 104] that can deal with convex optimization problems over time-varying directed graphs. More recently, the paper [148] has extended the push-sum algorithm to a larger class of distributed algorithms that are applicable to nonconvex objective functions, convex constraint sets, and time-varying graphs (see the subsequent paragraph on new directions for more details).

1.5.2.3 ADMM Based Approaches

Another approach for solving problem (1.1) in a distributed fashion over a static network can be constructed by using the Alternate Direction Multiplier Method (ADMM). This method is based on an equivalent formulation of consensus constraints. Unlike consensus-based (sub)-gradient method (1.9) which operates in the space of the primal-variables, the ADMM solves a corresponding Lagrangian dual problem (obtained by relaxing the equality constraints that are associated with consensus requirement). Just as any dual method, the ADMM is applicable to problems where the structure of the objective functions f_i is simple enough so that the ADMM updates can be executed efficiently. The algorithm has potential of solving the problem with geometric convergence rate, which requires global knowledge of some parameters including eigenvalues of a weight matrix associated with the graph. A recent survey on the ADMM and its various applications is given in [17].

The first works to address the development of distributed ADMM over a network are [80, 177, 178], while a linear convergence rate of the ADMM has been shown in [141]. In [1] the ADMM with linearization has been proposed for special composite optimization problems over graphs.

1.5.2.4 New Directions

Within the area of distributed (multi-agent) optimization over networks, loosely speaking, two main directions of research can be noted, namely toward efficiency improvements (to develop “fast” distributed algorithms whose performance can meet the best known performance guarantees in a centralized setting) and toward addressing non-convex problems over networks.

In the domain of efficiency improvements, there are new approaches that are rooted in the idea of distributing the optimality conditions for multi-agent problems and the approaches that investigate gradient-consensus models. The consensus-based primal algorithms with a constant step-size are not likely to reach geometric convergence rate even when the overall objective function is strongly convex. Papers [142, 143] develop the algorithm EXTRA and its proximal-gradient variant by employing a carefully selected gradient-difference scheme to cancel out the steady-state error that occurs in some distributed methods with a constant stepsize [106, 108]. The EXTRA algorithm converges at an $o(1/k)$ rate when the objective network function is convex, and it has a geometric rate when the objective function is strongly convex. These developments have considered a static and undirected graph.

References [179, 187] combine EXTRA with the push-sum protocol of [67] to produce DEXTRA (Directed Extra-Push) algorithm for optimization over a directed graph. It has been shown that DEXTRA converges at a geometric (R-linear) rate for a strongly convex objective function, but it requires a careful stepsize selection. It has been noted in [179] that the feasible region of stepsizes which guarantees this convergence rate can be empty in some cases.

The work [183, 184] utilizes an adapt-then-combine (ATC) strategy [137, 138] of dynamic weighted-average consensus approach [188] to develop a distributed algorithm, termed Aug-DGM algorithm. This algorithm can be used over static directed or undirected graphs (but requires doubly stochastic matrices). The most interesting aspect of the Aug-DGM algorithm is that it can produce convergent iterates even when different agents use different (constant) stepsizes.

Simultaneously and independently, the idea of tracking the gradient averages through the use of consensus has been proposed in [184] for convex unconstrained problems and in [35] for non-convex problems with convex constraints. The work in [35–37] develops a large class of distributed algorithms, referred to as NEXT, which utilizes various "function-surrogate modules" thus providing a great flexibility in its use and rendering a new class of algorithms that subsumes many of the existing distributed algorithms. The ideas in [36, 37] and in [183] have also been proposed independently, with the former preceding the latter. The algorithm framework of [35–37] is applicable to nonconvex problems with convex constraint sets over time-varying graphs, but requires the use of doubly stochastic matrices. This assumption was recently removed in [148] by using column-stochastic matrices, which are more general than the degree-based column-stochastic matrices of the push-sum method. Simultaneously and independently, the papers [37] and [154] have appeared to treat nonconvex problems over graphs. The work in [154] proposes and analyzes a distributed gradient method based on the push-sum consensus in deterministic and stochastic setting for unconstrained problems.

The idea of using a consensus process to track gradients has also been recently used in [115] to develop a distributed algorithm, referred to as DIGing, with a geometric convergence rate over time-varying graphs. This is the first paper to establish such a rate for a consensus-based algorithms for convex optimization over time-varying graphs. We note that the algorithm uses a fixed stepsize, and the rate result is applicable to the problems with a strongly convex smooth objective function.

Acknowledgment The work presented in this chapter has been partially supported by the NSF grant DMS-1312907.

References

1. N.S. Aybat, Z. Wang, T. Lin, S. Ma, Distributed linearized alternating direction method of multipliers for composite convex consensus optimization. IEEE Trans. Autom. Control **63**(1), 5–20 (2018)
2. T.C. Aysal, M.E. Yildiz, A.D. Sarwate, A. Scaglione, Broadcast gossip algorithms: design and analysis for consensus, in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 4843–4848 (2008)
3. T.C. Aysal, M.E. Yildriz, A.D. Sarwate, A. Scaglione, Broadcast gossip algorithms for consensus. IEEE Trans. Signal Process. **57**, 2748–2761 (2009)
4. T. Başar, S.R. Etesami, A. Olshevsky, Fast convergence of quantized consensus using Metropolis chains, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 1330–1334 (2014)
5. D. Bajović, J. Xavier, J.M.F. Moura, B. Sinopoli, Consensus and products of random stochastic matrices: exact rate for convergence in probability. IEEE Trans. Signal Process. **61**(10), 2557–2571 (2013)

6. B.A. Bash, P.J. Desnoyers, Exact distributed Voronoi cell computation in sensor networks, in *IPSN'07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks* (ACM Press, New York, 2007), pp. 236–243
7. F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, M. Vetterli, Weighted gossip: distributed averaging using non-doubly stochastic matrices, in *Proceedings of the 2010 IEEE International Symposium on Information Theory*, pp. 1753–1757 (2010)
8. P. Billingsley, *Probability and Measure*, 3rd edn. (Wiley, Hoboken, 1995)
9. P.A. Bliman, G. Ferrari-Trecate, Average consensus problems in networks of agents with delayed communications. Automatica **44**(8), 1985–1995 (2008)
10. P.A. Bliman, A. Nedich, A. Ozdaglar, Rate of convergence for consensus with delays, in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 4849–4854 (2008)
11. V.D. Blondel, J.M. Hendrickx, A. Olshevsky, J.N. Tsitsiklis, Convergence in multiagent coordination, consensus, and flocking, in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2996–3000 (2005)
12. S. Bolouki, Linear consensus algorithms: structural properties and connections with Markov chains, Ph.D. thesis, Department of Electrical Engineering, École Polytechnique de Montréal, 2014
13. S. Bolouki, R.P. Malhamé, Theorems about ergodicity and class-ergodicity of chains with applications in known consensus models, in *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1425–1431 (2012)
14. V.S. Borkar, R. Makhijani, R. Sundaresan, Asynchronous gossip for averaging and spectral ranking. IEEE J. Sel. Top. Sign. Proces. **8**(4), 703–716 (2014)
15. V. Borkar, P.P. Varaiya, Asymptotic agreement in distributed estimation. IEEE Trans. Autom. Control **27**(3), 650–655 (1982)
16. S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Gossip algorithms: design, analysis, and applications, in *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1653–1664 (2005)
17. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2010)
18. F. Bullo, J. Cortés, S. Martínez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms* (Princeton University Press, Princeton, 2009)
19. M. Bürger, G. Notarstefano, F. Bullo, F. Allgöwer, A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. Automatica **48**(9), 2298–2304 (2012)
20. K. Cai, H. Ishii, Quantized consensus and averaging on gossip digraphs. IEEE Trans. Autom. Control **56**(9), 2087–2100 (2011)
21. M. Cao, D.A. Spielman, A.S. Morse, A lower bound on convergence of a distributed network consensus algorithm, in *Proceedings of IEEE CDC*, pp. 2356–2361 (2005)
22. M. Cao, A.S. Morse, B.D.O. Anderson, Agreeing asynchronously. IEEE Trans. Autom. Control **53**(8), 1826–1838 (2008)
23. M. Cao, A.S. Morse, B.D.O. Anderson, Reaching a consensus in a dynamically changing environment: convergence rates, measurement delays, and asynchronous events. SIAM J. Control. Optim. 47(2), 601–623 (2008)
24. M. Cao, A.S. Morse, B.D.O. Anderson, Reaching a consensus in a dynamically changing environment: a graphical approach. SIAM J. Control. Optim. **47**(2), 575–600 (2008)
25. R. Carli, F. Fagnani, A. Speranzon, S. Zampieri, Communication constraints in coordinated consensus problem, in *Proceedings of IEEE American Control Conference*, pp. 4189–4194 (2006)
26. R. Carli, F. Fagnani, P. Frasca, T. Taylor, S. Zampieri, Average consensus on networks with transmission noise or quantization, in *Proceedings of European Control Conference*, pp. 1852–1857 (2007)
27. R. Carli, F. Fagnani, P. Frasca, S. Zampieri, Gossip consensus algorithms via quantized communication. Automatica **46**(1), 70–80 (2010)

28. F. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation. IEEE Trans. Signal Process. **58**(3), 1035–1048 (2010)
29. M. El Chamie, T. Başar, Optimal strategies for dynamic weight selection in consensus protocols in the presence of an adversary, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 735–740 (2014)
30. M. El Chamie, J. Liu, T. Başar, Design and analysis of distributed averaging with quantized communication, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 3860–3865 (2014)
31. T-H. Chang, A. Nedich, A. Scaglione, Distributed constrained optimization by consensus-based primal-dual perturbation method. IEEE Trans. Autom. Control **59**(6), 1524–1538 (2014)
32. B. Charron-Bost, Orientation and connectivity based criteria for asymptotic consensus (2013). Available at http://arxiv.org/abs/1303.2043
33. J. Chen, A.H. Sayed, Diffusion adaptation strategies for distributed optimization and learning over networks. IEEE Trans. Signal Process. **60**(8), 4289–4305 (2012)
34. M.H. DeGroot, Reaching a consensus. J. Am. Stat. Assoc. **69**(345), 118–121 (1974)
35. P. Di Lorenzo, G. Scutari, Distributed nonconvex optimization over networks, in *Proceedings of IEEE International Conference on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2015), Dec. 13–16, 2015, Cancun, Mexico* (2015)
36. P. Di Lorenzo, G. Scutari, Distributed nonconvex optimization over time-varying networks, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 16), March 20–25, 2016, Shanghai, China* (2016)
37. P. Di Lorenzo, G. Scutari, NEXT: in-network nonconvex optimization. IEEE Trans. Signal Inf. Process. Netw. **2**(2), 120–136 (2016)
38. A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, A. Scaglione, Gossip algorithms for distributed signal processing. Proc. IEEE **98**(11), 1847–1864 (2010)
39. A.D. Dominguez-Garcia, C. Hadjicostis, Distributed strategies for average consensus in directed graphs, in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2124–2129 (2011)
40. J.C. Duchi, A. Agarwal, M.J. Wainwright, Dual averaging for distributed optimization: convergence analysis and network scaling. IEEE Trans. Autom. Control **57**(3), 592 –606 (2012)
41. R. Dudley, *Real Analysis and Probability* (Cambridge University Press, Cambridge, 2002)
42. D. Easley, J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World* (Cambridge University Press, New York, 2010)
43. S.R. Etesami, T. Başar, Convergence time for unbiased quantized consensus, in *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 6190–6195 (2013)
44. F. Facchinei, J-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vols. I and II (Springer, New York, 2003)
45. F. Fagnani, S. Zampieri, Randomized consensus algorithms over large scale networks. IEEE J. Sel. Areas Commun. **26**(4), 634–649 (2008)
46. M. Fiedler, *Special Matrices and Their Applications in Numerical Mathematics*, 2nd edn. (Dover Publications, New York, 2008)
47. E. Ghadimi, Accelerating convergence of large-scale optimization algorithms, Ph.D. thesis, School of Electrical Engineering, Department of Automatic Control, KTH Royal Institute of Technology, 2015
48. B. Gharesifard, J. Cortés, Distributed continuous-time convex optimization on weight-balanced digraphs. Eur. J. Control. **18**(6), 539–557 (2012)
49. S. Goyal, *Connections: An Introduction to the Economics Networks* (Princeton University Press, Princeton, 2007)
50. Y. Hatano, A.K. Das, M. Mesbahi, Agreement in presence of noise: pseudogradients on random geometric networks, in *Proceedings of the 44th IEEE Conference on Decision and Control, and European Control Conference*, pp. 6382–6387 (2005)

51. R. Hegselmann, U. Krause, Opinion dynamics and bounded confidence models, analysis, and simulation. J. Artif. Soc. Soc. Simul. **5** (2002)
52. J.M. Hendrickx, Graphs and networks for the analysis of autonomous agent systems, Ph.D. thesis, Université Catholique de Louvain, 2008
53. J.M. Hendrickx, J.N. Tsitsiklis, A new condition for convergence in continuous-time consensus seeking systems, in *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 5070–5075 (2011)
54. J.M. Hendrickx, J.N. Tsitsiklis, Convergence of type-symmetric and cut-balanced consensus seeking systems. IEEE Trans. Autom. Control **58**(1), 214–218 (2013)
55. M. Huang, J.H. Manton, Stochastic approximation for consensus seeking: mean square and almost sure convergence, in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 306–311 (2007)
56. M. Huang, J.H. Manton, Stochastic lyapunov analysis for consensus algorithms with noisy measurements, in *Proceedings of the 2007 IEEE American Control Conference*, pp. 1419–1424 (2007)
57. M. Huang, J.H. Manton, Stochastic consensus seeking with measurement noise: convergence and asymptotic normality, in *Proceedings of the 2008 IEEE American Control Conference*, pp. 1337–1342 (2008)
58. M. Huang, J.H. Manton, Coordination and consensus of networked agents with noisy measurements: stochastic algorithms and asymptotic behavior. SIAM J. Control. Optim. **48**(1), 134–161 (2009)
59. M.O. Jackson, *Social and Economic Networks* (Princeton University Press, Princeton, 2008)
60. A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans. Autom. Control **48**(6), 988–1001 (2003)
61. D. Jakovetić, J. Xavier, J.M.F. Moura, Cooperative convex optimization in networked systems: augmented Lagrangian algorithms with directed gossip communication. IEEE Trans. Signal Process. **59**(8), 3889–3902 (2011)
62. D. Jakovetić, J. Xavier, J.M.F. Moura, Fast distributed gradient methods. IEEE Trans. Autom. Control **59**(5), 1131–1146 (2014)
63. S. Kar, J.M.F. Moura, Distributed consensus algorithms in sensor networks: link failures and channel noise. IEEE Trans. Signal Process. **57**(1), 355–369 (2009)
64. S. Kar, J.M.F. Moura, Distributed consensus algorithms in sensor networks: quantized data and random link failures. IEEE Trans. Signal Process. **58**(3), 1383 –1400 (2010)
65. S. Kar, J.M.F. Moura, Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental limits and tradeoffs. IEEE J. Sel. Top. Sign. Proces. **5**(4), 674–690 (2011)
66. A. Kashyap, T. Basar, R. Srikant, Quantized consensus. Automatica **43**(7), 1192–1203 (2007)
67. D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 482–491 (2003)
68. J. Koshl, A. Nedich, U.V. Shanbhag, Distributed algorithms for aggregative games on graphs. Oper. Res. **64**, 680–704 (2016)
69. K. Kvaternik, Decentralized coordination control for dynamic multiagent systems, Ph.D. thesis, Department of Electrical and Computer Engineering, University of Toronto, 2014
70. A. Lalitha, T. Javidi, A. Sarwate, Social learning and distributed hypothesis testing, in *Proceedings of 2014 IEEE International Symposium on Information Theory*, pp. 551–555 (2014)
71. J. Lavaei, R.M. Murray, Quantized consensus by means of gossip algorithm. IEEE Trans. Autom. Control **57**(1), 19–32 (2012)
72. H. LeBlanc, H. Zhang, X. Koutsoukos, S. Sundaram, Resilient asymptotic consensus in robust networks. IEEE J. Sel. Areas Commun.: Spec. Issue In-Netw. Comput. **31**(4), 766–781 (2013)
73. S. Lee, Optimization over networks: efficient algorithms and analysis, Ph.D. thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 2013

74. S. Lee, A. Nedich, Distributed random projection algorithm for convex optimization. IEEE J. Sel. Top. Sign. Proces. **48**(6), 988–1001 (2012)
75. S. Lee, A. Nedich, Asynchronous gossip-based random projection algorithms over networks. IEEE Trans. Autom. Control **61**(4), 953–968 (2016)
76. S. Lee, A. Nedich, M. Raginsky, Coordinate dual averaging for decentralized online optimization with non-separable global objectives. IEEE Trans. Control Netw. Syst. **5**(1), 34–44 (2018)
77. N. Li, J.R. Marden, Designing games for distributed optimization. IEEE J. Sel. Top. Sign. Proces. **7**(2), 230–242 (2013)
78. J. Lin, A.S. Morse, B.D.O. Anderson, The multi-agent rendezvous problem, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 1508–1513 (2003)
79. J. Lin, A.S. Morse, B.D.O. Anderson, The multi-agent rendezvous problem - the asynchronous case, in *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 1926–1931 (2004)
80. Q. Ling, A. Ribeiro, Decentralized dynamic optimization through the alternating direction method of multiplier. IEEE Trans. Signal Process. **62**(5), 1185–1197 (2014)
81. J. Liu, A.S. Morse, B.D.O. Anderson, C. Yu, Contractions for consensus processes, in *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 1974–1979 (2011)
82. J. Liu, S. Mou, A.S. Morse, B.D.O. Anderson, C. Yu, Deterministic gossiping. Proc. IEEE **99**(9), 1505–1524 (2011)
83. J. Liu, A.S. Morse, A. Nedich, T. Başar, Internal stability of linear consensus processes, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 922–927 (2014)
84. J. Liu, A. Nedich, T. Başar, Complex constrained consensus, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 1464–1469 (2014)
85. J. Liu, A.S. Morse, A. Nedich, T. Başar, Stability of a distributed algorithm for solving linear algebraic equations, in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 3707–3712 (2014)
86. I. Lobel, A. Ozdaglar, Distributed subgradient methods for convex optimization over random networks. IEEE Trans. Autom. Control **56**(6), 1291–1306 (2011)
87. I. Lobel, A. Ozdaglar, D. Feijer, Distributed multi-agent optimization with state-dependent communication. Math. Program. **129**(2), 255–284 (2011)
88. C. Lopes, A.H. Sayed, Distributed processing over adaptive networks, in *Adaptive Sensor Array Processing Workshop, MIT Lincoln Laboratory, MA*, pp. 1–5 (2006)
89. C.G. Lopes, A.H. Sayed, Diffusion least-mean squares over adaptive networks: formulation and performance analysis. IEEE Trans. Signal Process. **56**(7), 3122–3136 (2008)
90. J. Lorenz, A stabilization theorem for continuous opinion dynamics. Phys. A Stat. Mech. Appl. **355**, 217–223 (2005)
91. J. Lorenz, Repeated averaging and bounded confidence-modeling, analysis and simulation of continuous opinion dynamics, Ph.D. thesis, Universität Bremen, 2007
92. J. Lu, C.Y. Tang, Zero-gradient-sum algorithms for distributed convex optimization: the continuous-time case. IEEE Trans. Autom. Control **57**(9), 2348–2354 (2012)
93. J.R. Marden, G. Arslan, J.S. Shamma, Connections between cooperative control and potential games illustrated on the consensus problem, in *Proceedings of the European Control Conference*, pp. 4604–4611 (2007)
94. A. Martinoli, F. Mondada, G. Mermoud, N. Correll, M. Egerstedt, A. Hsieh, L. Parker, K. Stoy, *Distributed Autonomous Robotic Systems*. Springer Tracts in Advanced Robotics (Springer, Berlin, 2013)
95. A.S. Mathkar, V.S. Borkar, Nonlinear gossip (2014). Preprint
96. M. Mesbahi, M. Egerstedt, *Graph Theoretic Methods for Multiagent Networks* (Princeton University Press, Princeton, 2010)
97. C.C. Moallemi, B. Van Roy, Consensus propagation. IEEE Trans. Inf. Theory **52**(11), 4753–4766 (2006)
98. L. Moreau, Stability of continuous-time distributed consensus algorithms, in *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 3998–4003 (2004)

99. L. Moreau, Stability of multiagent systems with time-dependent communication links. IEEE Trans. Autom. Control **50**(2), 169–182 (2005)
100. A. Nedich, Asynchronous broadcast-based convex optimization over a network. IEEE Trans. Autom. Control **56**(6), 1337–1351 (2011)
101. A. Nedich, Random projection algorithms for convex minimization problems. Math. Program. Ser. B **129**, 225–253 (2011)
102. A. Nedich, *Convergence Rate of Distributed Averaging Dynamics and Optimization in Networks*. Found. Trends Syst. Control **2** (2015)
103. A. Nedich, A. Olshevsky, Distributed optimization over time-varying directed graphs. IEEE Trans. Autom. Control **60**(3), 601–615 (2015)
104. A. Nedich, A. Olshevsky, Stochastic gradient-push for strongly convex functions on time-varying directed graphs. IEEE Trans. Autom. Control **61**(12), 3936–3947 (2016)
105. A. Nedich, A. Ozdaglar, On the rate of convergence of distributed subgradient methods for multi-agent optimization, in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 4711–4716 (2007)
106. A. Nedich, A. Ozdaglar, Distributed subgradient methods for multi-agent optimization. IEEE Trans. Autom. Control **54**(1), 48–61 (2009)
107. A. Nedich, A. Ozdaglar, Convergence rate for consensus with delays. J. Glob. Optim. **47**(3), 437–456 (2010)
108. A. Nedich, A. Ozdaglar, Cooperative distributed multi-agent optimization, in *Convex Optimization in Signal Processing and Communications*, ed. by Y. Eldar, D. Palomar (Cambridge University Press, Cambridge, 2010), pp. 340–386
109. A. Nedich, A. Olshevsky, A. Ozdaglar, J.N. Tsitsiklis, Distributed subgradient methods and quantization effects, in *Proceedings of 47th IEEE Conference on Decision and Control*, pp. 4177–4184 (2008)
110. A. Nedich, A. Olshevsky, A. Ozdaglar, J.N. Tsitsiklis, On distributed averaging algorithms and quantization effects. IEEE Trans. Autom. Control **54**(11), 2506–2517 (2009)
111. A. Nedich, A. Ozdaglar, P.A. Parrilo, Constrained consensus and optimization in multi-agent networks. IEEE Trans. Autom. Control **55**(4), 922–938 (2010)
112. A. Nedich, A. Olshevsky, C.A. Uribe, Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs, in *Proceedings of American Control Conference 2015*, pp. 5884–5889 (2015)
113. A. Nedich, S. Lee, M. Raginsky, Decentralized online optimization with global objectives and local communication, in *Proceedings of the 2016 American Control Conference (ACC), Boston, MA, July 6–8, 2016*, pp. 4497–4503 (2016)
114. A. Nedich, A. Olshevsky, C.A. Uribe, Network independent rates in distributed learning, in *Proceedings of the 2016 American Control Conference (ACC), Boston, MA, July 6–8, 2016*, pp. 1072–1077 (2016)
115. A. Nedich, A. Olshevsky, W. Shi, Achieving geometric convergence for distributed optimization over time-varying graphs. SIAM J. Optim. **27**(4), 2597–2633 (2017)
116. A. Nedich, A. Olshevsky, C.A. Uribe, Fast convergence rates for distributed non-bayesian learning. IEEE Trans. Autom. Control **62**(11), 5538–5553 (2017)
117. Yu.E. Nesterov, A method of solving a convex programming problem with convergence rate of $o(1/k^2)$. Soviet Math. Dokl. **27**(2), 372–376 (1983)
118. R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. Autom. Control **49**(9), 1520–1533 (2004)
119. R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems. Proc. IEEE **95**(1), 215–233 (2007)
120. A. Olshevsky, Efficient information aggregation for distributed control and signal processing, Ph.D. thesis, MIT, 2010
121. A. Olshevsky, Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. SIAM J. Control. Optim. **55**(6), 3990–4014 (2017)
122. A. Olshevsky, J.N. Tsitsiklis, Convergence rates in distributed consensus averaging, in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3387–3392 (2006)

123. A. Olshevsky, J.N. Tsitsiklis, On the nonexistence of quadratic lyapunov functions for consensus algorithms. IEEE Trans. Autom. Control **53**(11), 2642–2645 (2008)
124. A. Olshevsky, J.N. Tsitsiklis, Convergence speed in distributed consensus and averaging. SIAM J. Control. Optim. **48**(1), 33–55 (2009)
125. S. Patterson, B. Bamieh, Distributed consensus with link failures as a structured stochastic uncertainty problem, in *Proceedings of the 46th Allerton Conference on Communication, Control, and Computing*, pp. 623–627 (2008)
126. S. Patterson, B. Bamieh, Convergence rates of consensus algorithms in stochastic networks, in *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 6608–6613 (2010)
127. S. Patterson, B. Bamieh, A. El Abbadi, Distributed average consensus with stochastic communication failures, in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 4215–4220 (2007)
128. S. Patterson, B. Bamieh, A. El Abbadi, Distributed average consensus with stochastic communication failures. IEEE Trans. Signal Process. **57**, 2748–2761 (2009)
129. B.T. Polyak, *Introduction to Optimisation* (Optimization Software, Inc., New York, 1987)
130. S.S. Ram, Distributed optimization in multi-agent systems: applications to distributed regression, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2009
131. S.S. Ram, A. Nedich, V.V. Veeravalli, Asynchronous gossip algorithms for stochastic optimization, in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC), Shanghai, China*, pp. 3581–3586 (2009)
132. S.S. Ram, A. Nedich, V.V. Veeravalli, Asynchronous gossip algorithms for stochastic optimization: constant stepsize analysis, in *Recent Advances in Optimization and its Applications in Engineering*. Volume of the 14th Belgian-French-German Conference on Optimization (BFG), pp. 51–60 (2010)
133. S.S. Ram, A. Nedich, V.V. Veeravalli, A new class of distributed optimization algorithms: application to regression of distributed data. Optim. Methods Softw. **27**(1), 71–88 (2012)
134. W. Ren, Consensus seeking in multi-vehicle systems with a time-varying reference state, in *Proceedings of IEEE American Control Conference*, pp. 717–722 (2007)
135. W. Ren, R. Beard, Consensus seeking in multi-agent systems under dynamically changing interaction topologies. IEEE Trans. Autom. Control **50**(5), 655–661 (2005)
136. H. Robbins, D. Siegmund, A convergence theorem for nonnegative almost supermartingales and some applications, in *Optimizing Methods in Statistics* (Academic, New York, 1971), pp. 233–257
137. A.H. Sayed, Diffusion adaptation over networks, in *Academic Press Library in Signal Processing*, vol. 3, pp. 323–454 (2013)
138. A.H. Sayed, Adaptation, learning, and optimization over networks. Found. Trends Mach. Learn. **7** (2014)
139. S. Shahrampour, A. Jadbabaie, Exponentially fast parameter estimation in networks using distributed dual averaging, in *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 6196–6201 (2013)
140. S. Shahrampour, A. Rakhlin, A. Jadbabaie, Distributed detection: finite-time analysis and impact of network topology. IEEE Trans. Autom. Control **61**(11), 3256–3268 (2016)
141. W. Shi, Q. Ling, K. Yuan, G. Wu, W. Yin, On the linear convergence of the admm in decentralized consensus optimization. IEEE Trans. Signal Process. **62**, 1750–1761 (2014)
142. W. Shi, Q. Ling, G. Wu, W. Yin, Extra: an exact first-order algorithm for decentralized consensus optimization. SIAM J. Optim. **25**, 944–966 (2015)
143. W. Shi, Q. Ling, G. Wu, W. Yin, A proximal gradient algorithm for decentralized composite optimization. IEEE Trans. Signal Process. **63**, 6013–6023 (2015)
144. K. Srivastava, Distributed optimization with applications to sensor networks and machine learning, Ph.D. thesis, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering, 2011
145. K. Srivastava, A. Nedich, Distributed asynchronous constrained stochastic optimization. IEEE J. Sel. Top. Sign. Proces. **5**(4), 772–790 (2011)

146. K. Srivastava, A. Nedich, D.M. Stipanović, Distributed constrained optimization over noisy networks, in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pp. 1945–1950 (2010)
147. K. Srivastava, A. Nedich, D. Stipanovic, Distributed Bregman-distance algorithms for min-max optimization, in *Agent-Based Optimization*. Springer Studies in Computational Intelligence (SCI) (Springer, Berlin, 2013), pp. 143–174
148. Y. Sun, G. Scutari, D. Palomar, Distributed nonconvex multiagent optimization over time-varying networks, in *Proceedings of the 50th Asilomar Conference on Signals, Systems, and Computers 2016* (2016). Available on arXiv at http://arxiv.org/abs/1607.00249
149. S. Sundaram, C.N. Hadjicostis, Distributed function calculation and consensus using linear iterative strategies. IEEE J. Sel. Areas Commun.: Issue Control Commun. **26**(4), 650–660 (2008)
150. S. Sundaram, C.N. Hadjicostis, Distributed function calculation via linear iterative strategies in the presence of malicious agents. IEEE Trans. Autom. Control **56**(7), 1495–1508 (2011)
151. S. Sundaram, S. Revzen, G.J. Pappas, A control-theoretic approach to disseminating values and overcoming malicious links in wireless networks. Automatica **48**(11), 2894–2901 (2012)
152. A. Tahbaz-Salehi, A. Jadbabaie, A necessary and sufficient condition for consensus over random networks. IEEE Trans. Autom. Control **53**(3), 791–795 (2008)
153. A. Tahbaz-Salehi, A. Jadbabaie, Consensus over ergodic stationary graph processes. IEEE Trans. Autom. Control **55**(1), 225–230 (2010)
154. T. Tatarenko, B. Touri, Non-convex distributed optimization. IEEE Trans. Autom. Control **62**(8), 3744–3757 (2017)
155. B. Touri, Product of random stochastic matrices and distributed averaging, Ph.D. thesis, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering, 2011
156. B. Touri, *Product of Random Stochastic Matrices and Distributed Averaging* (Springer, Berlin, 2012)
157. B. Touri, C. Langbort, On endogenous random consensus and averaging dynamics. IEEE Trans. Control Netw. Syst. **1**(3), 241–248 (2014)
158. B. Touri, A. Nedich, Distributed consensus over network with noisy links, in *Proceedings of the 12th International Conference on Information Fusion*, pp. 146–154 (2009)
159. B. Touri, A. Nedich, Approximation and limiting behavior of random models, in *Proceedings of 49th IEEE Conference on Decision and Control*, pp. 2656–2663 (2010)
160. B. Touri, A. Nedich, When infinite flow is sufficient for ergodicity, in *Proceedings of 49th IEEE Conference on Decision and Control*, pp. 7479–7486 (2010)
161. B. Touri, A. Nedich, Alternative characterization of ergodicity for doubly stochastic chains, in *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 5371–5376 (2011)
162. B. Touri, A. Nedich, On ergodicity, infinite flow and consensus in random models. IEEE Trans. Autom. Control **56**(7), 1593–1605 (2011)
163. B. Touri, A. Nedich, On existence of a quadratic comparison function for random weighted averaging dynamics and its implications, in *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 3806–3811 (2011)
164. B. Touri, A. Nedich, On approximations and ergodicity classes in random chains. IEEE Trans. Autom. Control **57**(11), 2718–2730 (2012)
165. B. Touri, A. Nedich, On backward product of stochastic matrices. Automatica **48**(8), 1477–1488 (2012)
166. B. Touri, A. Nedich, Product of random stochastic matrices and distributed averaging. IEEE Trans. Autom. Control **59**(2), 437–448 (2014)
167. K.I. Tsianos, The role of the network in distributed optimization algorithms: convergence rates, scalability, communication/computation tradeoffs and communication delays, Ph.D. thesis, McGill University, Dept. of Electrical and Computer Engineering, 2013
168. K.I. Tsianos, M.G. Rabbat, Distributed consensus and optimization under communication delays, in *Proceedings of the 49th IEEE Allerton Conference on Communication, Control, and Computing*, pp. 974–982 (2011)

169. K.I. Tsianos, S. Lawlor, M.G. Rabbat, Consensus-based distributed optimization: practical issues and applications in large-scale machine learning, in *Proceedings of the 50th Allerton Conference on Communication, Control, and Computing*, pp. 1543–1550 (2012)
170. K.I. Tsianos, S. Lawlor, M.G. Rabbat, Push-sum distributed dual averaging for convex optimization, in *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 5453–5458 (2012)
171. J.N. Tsitsiklis, Problems in decentralized decision making and computation, Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1984
172. S.-Y. Tu, A.H. Sayed, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. IEEE Trans. Signal Process. **60**(12), 6217–6234 (2012)
173. S.-Y. Tu, A.H. Sayed, On the influence of informed agents on learning and adaptation over networks. IEEE Trans. Signal Process. **61**(6), 1339–1356 (2012)
174. T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, O. Schochet, Novel type of phase transitions in a system of self-driven particles. Phys. Rev. Lett. **75**(6), 1226–1229 (1995)
175. P. Wan, M.D. Lemmon, Event-triggered distributed optimization in sensor networks, in *Symposium on Information Processing of Sensor Networks, (San Francisco, CA)*, pp. 49–60 (2009)
176. J. Wang, N. Elia, A control perspective for centralized and distributed convex optimization, in *Proceedings of the IEEE Conference on Decision and Control, (Florida, USA)*, pp. 3800–3805 (2011)
177. E. Wei, A. Ozdaglar, Distributed alternating direction method of multipliers, in *Proceedings of the 51st IEEE Conference on Decision and Control and European Control Conference*, pp. 5445–5450 (2012)
178. E. Wei, A. Ozdaglar, On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers, in *Proceedings of IEEE Global Conference on Signal and Information Processing*, pp. 551–554 (2013)
179. C. Xi, U. Khan, DEXTRA: a fast algorithm for optimization over directed graphs. IEEE Trans. Autom. Control **62**(10), 4980–4993 (2017)
180. L. Xiao, S. Boyd, Fast linear iterations for distributed averaging. Syst. Control Lett. **53**, 65–78 (2004)
181. L. Xiao, S. Boyd, S. Lall, A scheme for robust distributed sensor fusion based on average consensus, in *Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, pp. 63–70 (2005)
182. L. Xiao, S. Boyd, S.-J. Kim, Distributed average consensus with least mean square deviation. J. Parallel Distrib. Comput. **67**(1), 33–46 (2007)
183. J. Xu, Augmented distributed optimization for networked systems, Ph.D. Thesis, School of Electrical and Electronic Engineering, Nanyang Technological University, 2016
184. J. Xu, S. Zhu, Y. Soh, L. Xie, Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes, in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, pp. 2055–2060 (2015)
185. F. Yan, S. Sundaram, S.V.N. Vishwanathan, Y. Qi, Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties. IEEE Trans. Data Knowl. Eng. **25**(11), 2483–2493 (2013)
186. F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, L. Schenato, Newton-Raphson consensus for distributed convex optimization, in *Proceedings of the IEEE Conference on Decision and Control, (Florida, USA)*, pp. 5917–5922 (2011)
187. J. Zeng, W. Yin, Extrapush for convex smooth decentralized optimization over directed networks (2015). Available on arXiv at http://arxiv.org/abs/1511.02942
188. M. Zhu, S. Martínez, Discrete-time dynamic average consensus. Automatica **46**, 322–329 (2010)

189. M. Zhu, S. Martínez, On distributed convex optimization under inequality and equality constraints. IEEE Trans. Autom. Control **57**(1), 151–164 (2012)
190. M. Zhu, S. Martínez, An approximate dual subgradient algorithm for distributed non-convex constrained optimization. IEEE Trans. Autom. Control **58**(6), 1534–1539 (2013)

Chapter 2
Five Lectures on Differential Variational Inequalities

Jong-Shi Pang

Abstract Complex engineering and economic systems are often dynamic in nature and its time-varying variables are subject to algebraic restrictions that are cast in the form of equalities and inequalities; furthermore, these variables may be related to each other via some logical conditions. For such constrained dynamical systems, the classical approach based on ordinary differential equations (ODEs) alone is inadequate to fully capture the intricate details of the system evolution. Combining an ODE with a variational condition on an auxiliary algebraic variable that represents either a constrained optimization problem in finite dimensions or its generalization of a variational inequality, the mathematical paradigm of differential variational inequalities (DVIs) was born. On one hand, a DVI addresses the need to extend the century-old ODE paradigm to incorporate such elements as algebraic inequalities, mode changes, and logical relations, and on the other hand, introduces a temporal element into a static optimization or equilibrium problems. The resulting DVI paradigm transforms the basic science of dynamical systems by leveraging the modeling power and computational advances of constrained optimization and its extension of a variational inequality, bringing the former (dynamical systems) to new heights that necessitate renewed analysis and novel solution methods and endowing the latter (optimization) with a novel perspective that is much needed for its sustained development. Introducing the subject of the DVI and giving a survey of its recent developments, this report is an expanded version of five lectures that the author gave at the CIME Summer School on *Centralized and Distributed Multi-agent Optimization Models and Algorithms* held in Cetraro, Italy, June 23–27, 2014.

J.-S. Pang (✉)
The Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA, USA
e-mail: jongship@usc.edu

F. Facchinei, J.-S. Pang (eds.), *Multi-agent Optimization*, Lecture Notes
in Mathematics 2224, https://doi.org/10.1007/978-3-319-97142-1_2

2.1 Introduction

Complex engineering and economic systems are often dynamic in nature and its time-varying variables are subject to algebraic restrictions that are cast in the form of equalities and inequalities; furthermore, these variables may be related to each other via some logical conditions. For such constrained dynamical systems, the classical approach based on ordinary differential equations (ODEs) alone is inadequate to fully capture the intricate details of the system evolution. Combining an ODE with a variational condition on an auxiliary algebraic variable that represents either a constrained optimization problem in finite dimensions or its generalization of a variational inequality, the mathematical paradigm of differential variational inequalities (DVIs) was born. On one hand, a DVI addresses the need to extend the century-old ODE paradigm to incorporate such elements as algebraic inequalities, mode changes, and logical relations, and on the other hand, introduces a temporal element into a static optimization or equilibrium problems. The resulting DVI paradigm transforms the basic science of dynamical systems by leveraging the modeling power and computational advances of constrained optimization and its extension of a variational inequality, bringing the former (dynamical systems) to new heights that necessitate renewed analysis and novel solution methods and endowing the latter (optimization) with a novel perspective that is much needed for its sustained development. Thus, the DVI serves as a bridge between the worlds of optimization and dynamical systems.

Inspired by applications in stochastic queuing networks [60], the unpublished reference [91] is arguably the earliest publication where a dynamical system with explicit complementarity conditions, albeit in integral form, was formulated. In the optimization literature, a special instance of the DVI, called the *nonlinear differential complementarity problems*, was first studied in the Ph.D. thesis [68] as an extension of the (static) finite-dimensional nonlinear complementarity problem [45]. Independently of these two earlier works [68, 91], two doctoral theses [25, 62] from the Netherlands School have studied the linear complementarity systems (LCSs) and piecewise affine systems from the perspective of hybrid dynamical systems and control theory [108, 109]. With the goal of introducing a unified mathematical framework for these non-traditional dynamical systems, the author and his collaborator, David Stewart at the University of Iowa, formally defined the DVI in a 2008 paper [102]. From the beginning, it was recognized that the DVI, while providing a very broad setting for many applications, is a very challenging mathematical problem whose rigorous treatment requires the fusion of a variety of analytical tools and numerical techniques from both ODE and mathematical programming. Since the publication of these early works, there has been a steady growth in the study of the DVI and the closely related differential complementarity systems (DCSs); see the references [30–33, 55–57, 59, 65, 100, 114–117, 126, 127, 134, 135]. Increasingly, the importance of the DVIs and (DCSs) in applications is being recognized as documented in the surveys [21, 63, 111] and several recent publications [4, 85, 87, 88, 104, 127, 129], as well as the

Habilitation thesis [2] that contains many references discussing applications in mechanical, electrical, and biological systems [3, 5, 9, 13, 14, 23, 35, 39, 43, 47, 52–54, 64, 66, 67, 78, 80, 81, 93, 94, 124, 125, 128, 131, 138, 140]. In spite of these efforts to date, a comprehensive investigation of the DVI is very much in its infancy stage, with lots of open questions to be resolved, solution methods to be developed, and applications to be explored.

One of the most challenging aspects of the DVI and the complementarity system is the mode switches caused by the activation and inactivation of the constraints. The unknown timing of these switches and their frequencies are two major sources of difficulty for the long-time analysis (such as stability) and the development of provably fast convergent numerical methods. Such switches could give rise to the so-called Zeno phenomenon (i.e., infinite switches in finite time) whose existence would create all sorts of technical complications that need to be addressed. Besides the understanding of this phenomenon and the identification of *Zeno-free systems*, many system-theoretic properties remain to be investigated in full detail. The design of efficient provably convergent numerical methods for approximating a solution trajectory of these variational/complementarity constrained dynamical systems is another major topic that requires further investigation. To achieve practical efficiency of such methods it is important to leverage the advances of computational mathematical programming for solving the time-discretized subsystems. All in all, a systematic investigation of the DVI requires a long-term focused effort in order to fully develop this highly promising new discipline to fruition and for the subject to realize its impact on the applied domains. It is with these dual objectives that this chapter is written, hoping to introduce this class of novel dynamical systems to a broad audience spanning both the optimization and engineering communities.

This paper is organized according to the five lectures that the author delivered at the CIME Summer School on *Centralized and Distributed Multi-agent Optimization Models and Algorithms* held in Cetraro, Italy, June 23–27, 2014. These lectures are:

Lecture I: Introduction to differential variational systems
Lecture II: A study of non-Zenoness
Lecture III: Time-stepping methods and their convergence analysis
Lecture IV: Linear-quadratic optimal control with mixed state-control constraints
Lecture V: Open-loop differential Nash games with mixed state-control constraints.

These topics are drawn from the author's joint work with his collaborators as contained in the cited references. Consistent with the main theme of the Summer School, the lectures aim at presenting the DVI/DCS as a powerful framework for multi-agent optimization-based decision making in continuous time. A comprehensive survey of the topic of DVI addressed to the broad applied mathematics community is presently being prepared that contains many other topics including a host of engineering applications and details on system-theoretic issues.

2.2 Lecture I: Differential Variational Systems

On one hand, ODEs with smooth right-hand sides have a long history in mathematics:

$$\begin{array}{rll} \dot{x} \triangleq \dfrac{dx(t)}{dt} & = f(t; x(t)) & \text{modeling systems in evolution} \\ 0 & = \Psi(x(0); x(\mathscr{T})) & \text{prescribed initial and terminal conditions,} \end{array}$$

where $f : \mathbb{R}^{1+n} \to \mathbb{R}^n$ and $\Psi : \mathbb{R}^{2n} \to \mathbb{R}^n$ are differentiable vector functions and $\mathscr{T} > 0$ is the time horizon of the system. On the other hand, mathematical programming is a post-world war II subject, typically posed as a constrained optimization problem in finite dimensions:

$$\operatorname*{minimize}_{x \in X} \theta(x), \tag{2.1}$$

where X is a closed subset of $\mathbb{R}^n$, which in many applications is convex and $\theta : \Omega \to \mathbb{R}$ is a continuous scalar-valued function defined on an open superset of X. While there are several extensions of ODEs, most notably, differential algebraic equations [11, 77], equations with discontinuous right-hand sides [49, 121], multi-valued differential equations [42], differential inclusions [15, 118], studies in the differential world have been minimally connected to the contemporary subject of optimization.

Extending the first-order optimality condition of the optimization problem (2.1) when the objective function θ is differentiable and the feasible set X is closed and convex, the finite-dimensional variational inequality [45], denoted VI (X, Φ), where $\Phi : \Omega \to \mathbb{R}^n$ is a given continuous vector function, is to find a vector $x \in X$ such that

$$(x' - x)^T \Phi(x) \geq 0, \quad \forall x' \in X.$$

The solution set of this VI is denoted by SOL(X, Φ). The special case of X being a cone $\mathscr{C}$ leads to the complementarity problem that has the form:

$$\mathscr{C} \ni x \perp \Phi(x) \in \mathscr{C}^*,$$

where $\mathscr{C}^* \triangleq \{ v \in \mathbb{R}^n \mid v^T x \geq 0 \text{ for all } x \in C \}$ is the dual cone of C as in convex analysis. While there are also many extensions and special cases of the VI/CP, such as the quasi-VI (where the constant set X is replaced by the varying set $X(x)$), the generalized VI (where Φ is multi-valued), and the Karush-Kuhn-Tucker (KKT) conditions (when the set X is finitely representable satisfying certain constraint qualifications),the study of these variational problems remains in a static (at best discrete-time) setting where the temporal element is never a part of the problems. In spite of this deficiency, the strengths of the mathematical programming field are the fruitful treatment of algebraic inequalities and the emphasis on computational efficiency. As a consequence, piecewise properties, non-smoothness, multi-valuedness, disjunctions, and logical relations are nowadays all amenable to effective treatment in practical applications.

Formally, defined by the tuple $(K; F; H; \Psi; \mathscr{T})$, where K is a closed convex set in $\mathbb{R}^m$, F and H are continuous mappings from $\mathbb{R}^{1+n+m}$ into $\mathbb{R}^n$, and Ψ and T are the same as above, the differential variational inequality with initial and boundary conditions is to find continuous-time trajectories $x : [0, \mathscr{T}] \to \mathbb{R}^n$ and $u : [0, \mathscr{T}] \to \mathbb{R}^m$ such that

$$\begin{aligned} \dot{x}(t) &= F(t, x(t), u(t)) && \text{system dynamics} \\ u(t) &\in \text{SOL}\,(K, H(t, x(t), \bullet)) && \text{variational condition} \\ 0 &= \Psi(x(0), x(\mathscr{T})) && \text{two-point boundary condition.} \end{aligned} \tag{2.2}$$

When $\Psi(x, y) = x - b$ for some constant vector b, the system becomes one of the initial-value kind. A differential complementarity system is the special case where the set K is a cone. Among the special cases of the DCS is the time-invariant affine complementarity system (LCS) with affine two-point boundary conditions that is defined by the matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times m}$ and two matrices M and N that are both $n \times n$, as well as vectors $r \in \mathbb{R}^n$, $s \in \mathbb{R}^m$, and $b \in \mathbb{R}^n$:

$$\begin{aligned} \dot{x} \quad &= r + Ax + Bu \\ 0 \leq u &\perp s + Cx + Du \;\geq\; 0 \\ b \quad &= Mx + Nu. \end{aligned} \tag{2.3}$$

For ease of reference, we let SOL(q, R) denote the solution set of the linear complementarity problem, which we denote by LCP (q, R): $0 \leq u \perp q + Ru \geq 0$ for a given vector q and matrix R of the same order as the dimension of q. Exemplified by the LCS, DVIs form a non-traditional mathematical paradigm offering a broad, unifying framework for modeling many applied (dis)equilibrium problems containing:

- dynamics (pathway to equilibrium)
- inequalities (unilateral constraints), and
- disjunctions (paradigm switch).

As a consequence of these features, these dynamical systems exhibit the following characteristics:

- state-triggered mode transition that are distinct from arbitrary switchings
- non-smoothness of solution trajectories, extending smooth dynamical systems
- the presence of an endogenous auxiliary variable, satisfying a variational condition such as complementarity.

The result is a powerful framework that on one hand can exploit the advances in mathematical programming to benefit classical unconstrained dynamical systems, on the other hand extends many optimization and equilibrium problems to a (often more realistic) continuous-time setting.

An early source of applications of differential variational systems arises from three-dimensional frictional contact problems in engineering mechanics. Unlike the traditional setting of continuum mechanics that employs partial differential equations to describe the fine physics of body deformations under stresses and strains (see e.g. the work of Klarbring [74, 75] who advocated mathematical programming methods for solving these problems), discrete or discretized mechanical systems [76] are useful for the modeling of multi-rigid-body systems where body deformations are ignored. The complementarity approach for the treatment of such systems was initiated by the pioneering work of Lötstedt [82, 83] and has today developed into a very effective framework for the numerical simulation of rather complex mechanical systems under frictional contact. There are several monographs [20, 105, 127], reviews [22, 76, 123], and a vast literature including [1, 3, 6–8, 10, 36, 37, 51, 101, 122, 130, 136, 137] to name a sample of references. As an alternative to a rigid-body model, a local compliance model was introduced in the Ph.D. thesis [119] to circumvent many of the deficiencies of the former model; see also [97, 120]. In this resulting DVI, the algebraic variable $u \in \mathbb{R}^m$ is partitioned in 2 components: $y \in \mathbb{R}^{\ell_1}$ and $v \in \mathbb{R}^{\ell_2}$ with y being subject to a complementarity condition (modeling a non-penetration condition in the normal direction) and v satisfying a variational condition that is parameterized by (x, y) (modeling a friction principle in two tangential directions),

$$\begin{aligned} \dot{x} &= A(x, y) + B(x, y)v \\ 0 &\leq y \perp G(x, y) \geq 0 \\ v &\in \text{SOL}(K(x, y), H(x, y, \bullet)), \end{aligned} \tag{2.4}$$

where we have omitted the initial/boundary conditions. Incidentally, the particular form of the right-hand side in the ODE—linearity in the primary variable v of the VI—is essential to derive existence results and algorithmic convergence for the DVI; see the discussion following Proposition 1.

2.3 Lecture I: DVI in a Multi-Agent Paradigm

In order to discuss the role of the DVI in the context of optimization-based multi-agent decision making, it would be useful to first introduce the static non-cooperative game without the temporal aspect. In this multi-agent game, there are N selfish players each optimizing an objective function subject to constraints on his/her strategies while anticipating the rivals' strategies. For $i = 1, \cdots, N$, let $X^i \subseteq \mathbb{R}^{n_i}$ be a closed convex set of player i's admissible strategies and $\theta_i : \Omega \to \mathbb{R}$ be his/her objective function defined on the open set Ω containing $X \triangleq \prod_{i=1}^{N} X^i$. Anticipating $x^{-i} \triangleq \left(x^j\right)_{j \neq i}$, player i solves the following parametric optimization problem to determine a best response:

$$\underset{x^i \in X^i}{\text{minimize}}\, \theta(x^i, x^{-i}). \tag{2.5}$$

A Nash equilibrium (NE) is a tuple $x^* \triangleq \left(x^{*,j}\right)_{j=1}^{N}$ such that

$$x^{*,i} \in \underset{x^i \in X^i}{\text{minimize}}\, \theta_i(x^i, x^{*,-i}).$$

When each $\theta_i(\bullet, x^{-i})$ is convex and differentiable for every fixed $x^{-i} \in X^{-i} \triangleq \prod_{j \neq i} X^j$, it follows that x^* is a NE if and only if $x^* \in \text{SOL}(X, \Phi)$, where $\Phi(x) \triangleq \left(\nabla_{x^i} \theta_i(x)\right)_{i=1}^{N}$. For a recent summary of the Nash equilibrium problem, see [46] where the reader can find many references on distributed algorithms for solving this problem.

The continuous-time extension of the above static game replaces each player's finite-dimensional optimization problem (2.5) by an optimal control problem, which we formulate as follows [59]. Given a time horizon $\mathscr{T} > 0$ and an initial state $\xi \in \mathbb{R}^n$, find an *absolutely continuous* state trajectory $x : [0, \mathscr{T}] \to \mathbb{R}^n$ and an *integrable* control $u : [0, \mathscr{T}] \to \mathbb{R}^m$:

$$\begin{aligned}
&\underset{x,u}{\text{minimize}}\;\; V(x,u) \triangleq \varphi(x(\mathscr{T})) + \int_0^{\mathscr{T}} \phi(t, x(t), u(t))\, dt \\
&\text{subject to } x(0) = \xi \text{ and for almost all } t \in [\,0, \mathscr{T}\,]: \\
&\qquad \underbrace{\dot{x}(t) = r + Ax(t) + Bu(t)}_{\text{linear dynamics}} \quad \text{and} \quad \underbrace{f + Cx(t) + Du(t) \geq 0}_{\text{mixed state-control constraints}},
\end{aligned}$$

where (A, B, C, D) are constant matrices, (r, f) are constant vectors, and φ and ϕ are given functions. To simplify the discussion, we consider only linear dynamics but allow the algebraic constraints to contain both the state and control variables jointly. To formulate the optimality conditions of the above control problem as a DVI, let $\lambda(t)$ be the adjoint variable associated with the ODE. We then have the following system:

$$\begin{aligned}
\dot{\lambda}(t) &= -\nabla_x \phi(t, x(t), u(t)) - A^T \lambda(t), \quad \lambda(\mathscr{T}) = \nabla\varphi(x(\mathscr{T})) \\
\dot{x}(t) &= r + Ax(t) + Bu(t), \quad x(0) = \xi,
\end{aligned}$$

$$\text{and } u(t) \in \underset{u}{\operatorname{argmin}} \left[\underbrace{\phi(t, x(t), u) + \lambda(t)^T \, (\, r + Ax(t) + Bu\,)}_{\text{Hamiltonian}} \right]$$

$$\text{subject to } f + Cx(t) + Du \geq 0,$$

which has an initial condition $x(0) = \xi$ and a terminal condition $\lambda(\mathscr{T}) = \nabla\varphi(x(\mathscr{T}))$ on the (primal) state variable x and the (dual) adjoint variable λ, respectively. Extending the single optimal control problem to a multi-agent context yields the differential Nash game, which is formally defined as follows. While anticipating rivals' pairs $(x^{-i}, u^{-i}) \triangleq \left(x^j, u^j\right)_{j \neq i}$ of strategy trajectories, player i seeks a pair of state-control trajectories (x^i, u^i) to

$$\begin{aligned}
&\underset{x^i, u^i}{\text{minimize}} \;\; V(x^i, x^{-i}, u^i, u^{-i}) \\
&\text{subject to } x^i(0) = \xi^i \text{ and for almost all } t \in [\,0, \mathscr{T}\,]: \\
&\qquad\qquad \dot{x}^i(t) = r^i + A^i x^i(t) + B^i u^i(t) \\
&\text{and} \qquad f^i + C^i x^i(t) + D^i u^i(t) \geq 0.
\end{aligned}$$

A *differential Nash equilibrium* is a tuple $(x^*, u^*) \triangleq \left(x^{*,i}, u^{*,i}\right)_{i=1}^N$ of players' strategies such that for all $i = 1, \cdots, N$,

$$\begin{aligned}
(\, x^{*,i}, u^{*,i}\,) \in \underset{x^i, u^i}{\operatorname{argmin}} \quad & V(x^i, x^{*,-i}, u^i, u^{*,-i}) \\
& \text{subject to } (x^i, u^i) \text{ satisfies player } i\text{'s constraints.}
\end{aligned}$$

Concatenating the optimality conditions of the players' problems, we obtain an aggregated DVI whose solutions will be shown to be Nash equilibria. Unlike the static problem where the players' strategies are finite-dimensional vectors, the differential Nash problem considerably more complicated to analyze; for one thing, we have not even prescribed the regularity property of a solution trajectory to the differential variational problems introduced thus far. Unlike the treatment in the monograph [18] where computation is not emphasized, the optimization-based DVI framework allows us to develop effective algorithms for solving this continuous-time game as defined above. Such algorithms are in turn based on time discretizations of the interval $[0, \mathscr{T}]$ that result in finite-dimensional optimization subproblems which can be solved by well established algorithms.

2.4 Lecture I: Modeling Breadth

By way of a simple LCS, we illustrate the complexity of such a differential system coupled with complementarity constraints. It is well known that the (trivial) initial-value ODE: $\dot{x} = Ax$ with $x(0) = x^0$ has an explicit solution $x(t; x^0) = e^{At}x^0$ that is a linear function of x^0 for every fixed t; moreover, many other obvious properties can be said about this solution. For instance, $x(\bullet; x^0)$ is an analytic function for fixed x^0. Now, consider a simple variant with $A \neq B$

$$\dot{x} = \begin{cases} Ax & \text{if } c^T x < 0 \\ ? & \text{if } c^T x = 0 \\ Bx & \text{if } c^T x > 0, \end{cases} \tag{2.6}$$

which identifies one *mode* of the system on the "positive" side of the hyperplane $c^T x = 0$ and a possibly different mode on the other side of the same hyperplane; thus (2.6) is potentially a *bimodal system*. There are two cases depending on how the system evolves on the hyperplane. One case is that the trajectory crosses smoothly, which happens when $Ax = Bx$ on $c^T x = 0$ that corresponds to the situation where the right-hand side of the ODE is continuous; the other case is when the crossing is discontinuous, i.e., $Ax \neq Bx$ when $c^T x = 0$. In the former case, we must have $B = A + bc^T$ for some vector b. Hence, the ODE (2.6) becomes

$$\dot{x} = Ax + b \max(0, c^T x), \tag{2.7}$$

with the right-hand side being a simple piecewise linear function. Despite its simplicity, there is no explicit form for a solution to this piecewise linear ODE with initial condition x^0, although such a solution $x(t; x^0)$ still has the desirable property that $x(\bullet; x^0)$ is a C^1, albeit no longer analytic, function for fixed x^0. Moreover, to show that the continuous dependence of the solution on the initial condition x^0 is no longer as trivial as the previous case of a linear ODE. It turns out one can establish that $x(t; \bullet)$ is *semismooth* [100], a property that is fundamental in nonsmooth analysis; see [45, Definition 7.4.2].

The system (2.7) is a special piecewise ODE. Specifically, we recall [45, Definition 4.2.1] that a continuous function $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ is *piecewise affine (linear)* if there exists a *polyhedral (conic) subdivision* Ξ of $\mathbb{R}^n$ and a finite family of affine (linear) functions $\{G^i\}$ such that Φ coincides with one of the functions G^i on each polyhedron in Ξ. In turn, a polyhedral (conic) subdivision Ξ of $\mathbb{R}^n$ is a finite family of polyhedral sets (cones) that has the following three properties:

- the union of all polyhedra in the family is equal to $\mathbb{R}^n$;
- each polyhedron in the family is of dimension n, and
- the intersection of any two polyhedra in the family is either empty or a common proper face of both polyhedra.

A *conwise linear system* (CLS) is a piecewise system: $\dot{x} = \Phi(x)$ where Φ is a (continuous) piecewise linear function. Since the latter function must be (globally) Lipschitz continuous, the existence and uniqueness of a C^1 trajectory to a CLS for every initial condition is immediate.

Returning to the discontinuous case of (2.6), Filippov [49] proposed convexification of the right-hand side, expressing this bimodal system as a *differential inclusion*:

$$\dot{x} \in F(x) \triangleq \begin{cases} \{ Ax \} & \text{if } c^T x > 0 \\ \{ \lambda Ax + (1-\lambda)Bx \mid \lambda \in [0,1] \} & \text{if } c^T x = 0 \\ \{ Bx \} & \text{if } c^T x < 0. \end{cases} \tag{2.8}$$

In turn, one can convert the set-valued right-hand side into a single-valued using the complementarity condition. To do this, write $c^T x = \eta^+ - \eta^-$ as the difference of its nonnegative and nonpositive part, $\eta^\pm$, respectively so that the set-valued ODE (2.8) becomes a complementarity system with a bilinear ODE and a linear complementarity condition defined by a positive semidefinite, albeit asymmetric matrix:

$$\begin{aligned} &\dot{x} = \lambda Ax + (1-\lambda)Bx \\ &0 \leq \begin{pmatrix} \lambda \\ \eta^+ \end{pmatrix} \perp \begin{pmatrix} -c^T x \\ 1 \end{pmatrix} + \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\substack{\text{asymmetric positive} \\ \text{semidefinite}}} \begin{pmatrix} \lambda \\ \eta^+ \end{pmatrix} \geq 0. \end{aligned}$$

The multivalued signum function can be expressed by a complementarity condition as follows: for a scalar a, this function defined as $\operatorname{sgn}(a) \begin{cases} \triangleq 1 & \text{if } a > 0 \\ \in [-1,1] & \text{if } a = 0 \\ \triangleq -1 & \text{if } a < 0 \end{cases}$ is characterized as a scalar $\widehat{a}$ satisfying, for some λ, the two complementarity conditions: $0 \leq 1 + \widehat{a} \perp -a + \lambda \geq 0$ and $0 \leq \lambda \perp 1 - \widehat{a} \geq 0$. Scalar piecewise functions can also be expressed by the complementarity condition. For instance, consider the following univariate function f, which we assume, for notational convenience, is defined on the real line:

$$f(x) \triangleq \begin{cases} f_1(x) & \text{if } -\infty < x \leq a_1 \\ f_2(x) & \text{if } a_1 \leq x \leq a_2 \\ \vdots & \vdots \\ f_k(x) & \text{if } a_{k-1} \leq x \leq a_k \\ f_{k+1}(x) & \text{if } a_k \leq x < \infty, \end{cases}$$

with each f_i being a smooth function and the overall function f being continuous. It is not difficult to verify the following complementarity representation of this piecewise function:

$$f(x) \,=\, f_1(x_1) + \sum_{i=2}^{k+1} \left[\, f_i(a_{i-1} + x_i) - f_i(a_{i-1})\,\right] \quad \text{with} \quad x \,=\, \sum_{i=1}^{k+1} x_i,$$

where each x_i denotes the portion of x in the interval $[a_{i-1}, a_i]$, and satisfies

$$0 \,\leq\, x_2 \,\perp\, a_1 - x_1 \,\geq\, 0$$

and for $i = 3, \cdots, k+1$,

$$0 \,\leq\, x_i \,\perp\, (\,a_{i-1} - a_{i-2}\,) - x_{i-1} \,\geq\, 0.$$

As a result of the breadth of the variational and complementarity conditions in modeling mathematical properties and physical phenomena, the DVI and DCS provide a promising framework for the constructive treatment of nonsmooth dynamics, enabling the fruitful application of the advances in the fundamental theory and computational methods for solving the finite-dimensional VI/CP to the dynamic contexts.

We have so far described several contexts where the DVI and DCS have an important role to play; these include: optimal control problems with joint state and control constraints that are the basis for extension to differential non-cooperative multi-agent games, and the reformulation of ODEs with discontinuous and multi-valued right-hand sides. We have briefly mentioned the application to frictional contact problems; there are several other areas where the these non-traditional differential systems arise, in continuous-time dynamic user equilibrium in traffic planning [58, 85–89, 104], and in biological synthesis modeling [4, 41, 90].

2.5 Lecture I: Solution Concepts and Existence

The DVI (2.2) can be converted to an ODE with the same initial-boundary conditions under a strong monotonicity assumption of mapping $H(t, x, \bullet)$ in the variational condition. Specifically, if there exists a constant $\gamma > 0$ such that for all u and u' in K,

$$\left(u - u'\right)^T \left[\, H(t, x, u) - H(t, x, u')\,\right] \,\geq\, \gamma \,\|\, u - u'\|^2, \qquad \forall\,(\,t, x\,) \,\in\, [\,0, \mathscr{T}\,] \,\times\, \mathbb{R}^n,$$

then for every $(t, x) \in [0, \mathscr{T}] \times \mathbb{R}^n$, the VI $(K, H(t, x, \bullet))$ has a unique solution which we denote $u(t, x)$; moreover, this solution is Lipschitz continuous in (t, x) if $H(\bullet, \bullet, u)$ is Lipschitz continuous with a modulus that is independent of $u \in K$.

Thus under these assumptions, the DVI (2.2) become the (classic) ODE with a Lipschitz continuous right-hand side, provided that F is Lipschitz continuous in its arguments:

$$\dot{x}(t) = \mathbb{F}(t, x(t)) \triangleq F(t, x(t), u(t, x(t)))$$

with the same initial-boundary conditions. Using Robinson's theory of strong regularity [107] that since its initial publication has many extensions and refinements, one can obtain a similar conversion, albeit only locally near a tuple $(0, x^0, u^0)$, where u^0 is a *strongly regular* solution of the VI $(K, H(0, x^0, \bullet))$. We omit the details which can be found in [102]. For the LCS (2.3), a similar conversion holds provided that the matrix D is of the class P, i.e., all its principal minors are positive [38].

When the Lipschitz conversion fails, the next best thing would be to obtain a weak solution in the sense of Carathéodory; such is a pair of trajectory (x, u), with x being absolutely continuous and u being (Lebesque) integrable on $[0, \mathscr{T}]$ so that the ODE holds in the integral sense; i.e.,

$$x(t) = x(0) + \int_0^t F(s, x(s), u(s))\, ds, \qquad \forall\, t \in [\,0, \mathscr{T}\,],$$

and the membership $u(t) \in \text{SOL}(K, H(t, x(t), \bullet)$ holds for almost all $t \in [0, \mathscr{T}]$, or equivalently, $u(t) \in K$ for almost all $t \in [0, \mathscr{T}]$ and for all continuous function $v : [0, \mathscr{T}] \to K$,

$$\int_0^{\mathscr{T}} (\, u(s) - v(s)\,)^T H(s, x(s), u(s))\, ds \geq 0.$$

Conditions for the existence of a weak solution of the DVI can be found in [102]. In what follows, we present an existence result via the formulation of (2.2) as a differential inclusion (DI):

$$\dot{x}(t) \in \mathbb{F}(t, x(t)) \triangleq F(t, x(t), \text{SOL}(K, H(t, x(t), \bullet)), \qquad x(0) = x^0. \tag{2.9}$$

Specifically, the result [15, 118] provides two conditions on the abstract set-valued mapping $\mathbb{F}$ under which a weak solution in the sense of Carathéodory exists; such is an absolutely continuous (thus differentiable almost everywhere) function $x(t)$ satisfying the initial condition and the inclusion $\dot{x}(t) \in \mathbb{F}(t, x(t))$ for almost all $t \in [0, \mathscr{T}]$. One condition is the *upper semicontinuity* of $\mathbb{F}(t, x)$ on its domain. In general, a set-valued map $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ is upper semi-continuous at a point $\widehat{x} \in \text{dom}(\Phi)$ if for every open set $\mathscr{V}$ containing $\Phi(\widehat{x})$, an open neighborhood $\mathscr{N}$ of $\Phi(\widehat{x})$ exists such that, for each $x \in \mathscr{N}$, $\mathscr{V}$ contains $\Phi(x)$.

Proposition 1 *Suppose that* $\mathbb{F} : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ *is an upper semi-continuous set-valued map with nonempty closed convex values and satisfies*

- *the linear growth condition, i.e.,* $\exists\, \rho > 0$ *such that*

$$\sup\{\, \| y \| \mid y \in \mathbb{F}(t, x)\,\} \leq \rho\,(\,1 + \| x \|\,), \qquad \forall\,(t, x) \in [0, \mathscr{T}] \times \mathbb{R}^n.$$

Then for all x^0,

(a) the DI (2.9) has a weak solution in the sense of Carathéodory on $[0, \mathscr{T}]$*;*
(b) a measurable function $z : [0, \mathscr{T}] \to \mathbb{R}^m$ *exists such that for almost all* $t \in [0, T]$, $z(t) \in SOL(K, H(t, x(t), \bullet))$ *and* $\dot{x}(t) = F(t, x(t), z(t))$, *where* $x(t)$ *is the solution in (a).* □

Specialized to the VI $(K, H(t, x, \bullet))$, the two conditions (a) and (b) beg the following questions:

- When is the map: $\mathbb{F} : (t, x) \to F(t, x, \mathrm{SOL}(K, H(t, x, \bullet)))$ upper semi-continuous with nonempty closed convex values?
- When does the linear growth condition hold for the composite VI map?

In essence, the latter two questions can both be answered by invoking results form variational inequality theory [45]. Nevertheless, the convexity of the set $F(t, x, \mathrm{SOL}(K, H(t, x, \bullet)))$ restricts the function $F(t, x, \bullet)$ to be affine in the last argument so that $F(t, x, u) = A(t, x) + B(t, x)u$ for some vector-valued function $A : [0, \mathscr{T}] \times \mathbb{R}^n \to \mathbb{R}^n$ and matrix-valued function $B : [0, \mathscr{T}] \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$. Details can be found in [102] which also contains results for the two-point boundary-value problem. Uniqueness of solution for the (initial-value) DCS has been analyzed extensively in [126]. For the initial-value LCS (2.3) with $N = 0$ and $M = I$, a sufficient condition for the existence of a unique C^1 x-trajectory with no guarantee on uniqueness on the u-trajectory is when the set $B\mathrm{SOL}(Cx, D)$ is a singleton for all $x \in \mathbb{R}^n$; we call this a x-uniqueness property. An LCS with the latter singleton property is an instance of a conewise linear system. Parallel to the study of hybrid systems [29], the well-posedness (i.e., existence and uniqueness of solutions) of conewise linear systems have been studied in [27, 65, 134, 135]. It should be cautioned that all these well-posedness results are of the initial-value kind and are not directly applicable to either the mixed state-control constrained optimal control problem or its multi-agent generalization of a non-cooperative game, which, for one thing, are special problems with two-point boundary conditions. Details of the latter two problems will be presented subsequently. Finally, we refer to [100] where results on the dependence on initial conditions of the solution trajectory can be found, under a uniqueness hypothesis of the solution.

2.6 Lecture I: Summary

In this lecture, we have

- motivated and formally defined several differential variational/complementarity systems;
- presented several applications of such systems, including the differential Nash game involving multiple competitive, optimizing decision makers in a continuous-time setting;
- briefly described a number of technical issues, discussed the Lipschitz case, introduced the concept of a weak solution and provided a general existence result based on the formulation as a differential inclusion.

2.7 Lecture II: The Zeno Phenomenon

There are many paradoxes due to the Greek philosopher Zeno of Elea (ca. 490–430 BC); a famous one is the time-motion paradox having to do with a race between Tortoise and Archilles. The paradox is as follows.

> The Tortoise challenged Achilles to a race, claiming that he would win as long as Achilles gave him a small head start. Achilles laughed at this, for of course he was a mighty warrior and swift of foot, whereas the Tortoise was heavy and slow. "How big a head start do you need?" he asked the Tortoise with a smile. "Ten meters," the latter replied. Achilles laughed louder than ever. "You will surely lose, my friend, in that case," he told the Tortoise, "but let us race, if you wish it." "On the contrary," said the Tortoise, "I will win, and I can prove it to you by a simple argument." "Go on then,"; Achilles replied, with less confidence than he felt before. He knew he was the superior athlete, but he also knew the Tortoise had the sharper wits, and he had lost many a bewildering argument with him before this.
> [http://platonicrealms.com/encyclopedia/Zenos-Paradox-of-the-Tortoise-and-Achilles]

Mathematically, the Zeno phenomenon is probably the most fundamental property of a dynamical system subject to mode changes. The phenomenon refers to the possibility that there exist infinitely many such changes in a finite time interval. If a particular state of a solution trajectory is of the Zeno type, i.e., if this phenomenon occurs in a time interval surrounding this state, it will lead to great difficulty in faithfully analyzing and simulating such a trajectory in practice; the reason is simple: it is not possible to capture and predict the infinitely many mode transitions. The Zenoness of a state is a local property that arises at a finite time instance; there is also the asymptotic Zenoness that one needs to be concerned with if one is interested in investigating the long-time behavior (such as stability) of a solution trajectory; for such a solution trajectory, there is not a single mode in which the trajectory will remain in no matter how long time passes. For both theoretical and practical considerations, it is important to gain a clear understanding of the Zeno property, both long and short-time, of a constrained dynamical systems, particularly to identify systems where Zenoness is absent in their solutions.

In the case of the DCS where algebraic inequalities and logical disjunctions are present, mode switches are the result of activation and de-activation of these inequalities and the realization of the disjunctions. Historically, mode changes in (smooth) ODEs with piecewise analytic right-hand sides have been studied in [24, 133]. There are subsequent studies of "one-side non-Zenoness" for certain complementarity systems [26] and hybrid systems in control theory [71, 141]. A systematic study of (two-sided) non-Zenoness for complementarity systems was initiated in [114] that has led to several extensions [30, 55, 99, 113, 115–117]. We summarize the results in these references in the next several sections.

Before doing so, we return to the Zeno's paradox of the Tortoise and Archilles. What's the Tortoise's argument? Who wins the race? Define an event as a moment when Achilles catches up to the Tortoise's previous position. How many events are there during the race? Can all such events be tracked? How is this paradox related to the DVI? How do we formalize events mathematically and analyze the Zeno phenomenon for the DVI? Let's translate these questions to the bimodal ODE given by

$$\dot{x} \,=\, Ax + b\,\max(\,0, c^T x\,), \qquad x(0) \,=\, x^0,$$

whose unique solution we denote $x(t; x^0)$. The following are the above questions rephrased for this trajectory. How often does the trajectory $x(t; x^0)$ switch between the two halfspaces? In finite time? In infinite time? What does "switch" mean formally? Is touching the hyperplane considered a switch? Does the system exhibit the Zeno behavior, i.e., are there infinitely many mode switches in finite time? Are there bimodal systems with (in)finite many switches in long time? Can we characterize bimodal systems with finite numbers of switches, including zero, in infinite time? These are questions that the study of (non)-Zenoness aims to address.

2.8 Lecture II: Non-Zenoness of Complementarity Systems

Consider the time-invariant nonlinear complementarity system (NCS):

$$\begin{aligned} \dot{x} &= F(x, y) \\ 0 \le u \perp G(x, u) &\ge 0. \end{aligned} \tag{2.10}$$

Let $(x(t), u(t))$ denote a given solution trajectory. Associated with this solution, define three fundamental index sets at time t:

$$\begin{aligned} \alpha(t) &\triangleq \{\, i \mid u_i(t) > 0 = G_i(x(t), u(t))\,\}, \text{ inactive } u\text{-indices} \\ \beta(t) &\triangleq \{\, i \mid u_i(t) = 0 = G_i(x(t), u(t))\,\}, \text{ degenerate } u\text{-indices} \\ \gamma(t) &\triangleq \{\, i \mid u_i(t) = 0 < G_i(x(t), u(t))\,\}, \text{ strongly active } u\text{-indices.} \end{aligned}$$

The *switchings* of the index sets amount to the transitions among the differential algebraic equations (DAEs):

$$\begin{aligned} \dot{x} &= F(x,u) \\ 0 &= u_{\mathscr{I}} \\ 0 &= G_{\mathscr{J}}(x,u), \end{aligned}$$

each called a *mode* of the NCS, for various pairs of index sets $\mathscr{I}$ and $\mathscr{J}$ that partition $\{1, \cdots, m\}$. There are 2^m such pairs of index sets. Phrasing the discussion in the last section in this specific context, we re-iterate that mode switchings are a major concern in

- the design and analysis of the convergence rates of numerical schemes, particularly for methods with high-order methods (for $T < \infty$), and
- establishing any kind of asymptotic system-theoretic properties, such as Lyapunov stability, observability, reachability (for $T = \infty$).

Practically, it is impossible to simulate all mode switchings near a Zeno state. Analytically, all classical results in systems theory are invalidated due to the mode switches. Throughout the absence of Zeno states is key. The two main challenges in an analysis of Zenoness are:

- nonsmoothness: solution trajectory is at best once continuously differentiable,
- unknown switchings: dependent on initial conditions, implicit and at unknown times.

2.8.1 Solution Expansion

In what follows, we summarize the steps in the analysis that lead to the demonstration of non-Zenoness of the LCS that has $F(x,u) = Ax + Bu$ and $G(x,u) = Cx + Du$; cf. (2.3). First is the assumption that D is a P-matrix [38], which yields the existence and uniqueness of a solution to the LCP (q, D) for all vectors $q \in \mathbb{R}^m$. This then implies the existence and uniqueness of a solution pair $(x(t), u(t))$ of the initial-value LCS:

$$\begin{aligned} \dot{x} &= Ax + Bu, \qquad x(0) \;=\; x^0 \\ 0 &\le\; u \perp Cx + Du \;\ge\; 0. \end{aligned} \tag{2.11}$$

The P-property of D further implies for that every x, the LCP (Cx, D) has a unique solution which we denote $u(x)$; moreover this solution function is piecewise linear. Thus it is directionally differentiable everywhere with directional derivative, denoted $u'(x; d)$ at x at a direction $d \in \mathbb{R}^n$ being the unique vector v satisfying the mixed linear complementarity conditions:

$$\begin{aligned} v_\alpha &= 0 \leq (\,Cx + Du\,)_\alpha \\ 0 &\leq v_\beta \perp (\,Cx + Du\,)_\beta \leq 0 \\ v_\gamma &\geq 0 = (\,Cx + Du\,)_\alpha, \end{aligned}$$

Fix a time $t_* > 0$ (the following analysis applies also to the initial time $t = 0$ except that there is no backward time to speak about in this case). Suppose that the state $x^* \triangleq x(t_*)$ is *unobservable* for the linear system; that is, suppose that $CA^j x^* = 0$ for all $j = 0, 1, \cdots$. In this case the trivial solution $x(t) = e^{A(t-t_*)}x^*$ derived from $u(t) = 0$ is the unique solution trajectory; thus the three index sets $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ remain constant throughout the time duration.

Theorem 1 *Let D be a P-matrix and $(x(t), u(t))$ be the unique pair of solutions to the initial-value LCS (2.11). For every $t_* > 0$, there exist $\varepsilon > 0$ and index sets $(\alpha^\pm_{t_*}, \beta^\pm_{t_*}, \gamma^\pm_{t_*})$ so that*

$$(\,\alpha(t), \beta(t), \gamma(t)\,) = \left(\alpha^-_{t_*}, \beta^-_{t_*}, \gamma^-_{t_*}\,\right),\ \forall\, t\, \in\, [\,t_* - \varepsilon, t_*\,)$$

$$(\,\alpha(t), \beta(t), \gamma(t)\,) = \left(\alpha^+_{t_*}, \beta^+_{t_*}, \gamma^+_{t_*}\,\right),\ \forall\, t\, \in\, (\,t_*, t_* + \varepsilon\,].$$

Hence for every $\mathscr{T} > 0$, both $x(t)$ and $u(t)$ are continuous, piecewise analytic in $[0, \mathscr{T}]$; more precisely, there exists a finite partition of this time interval:

$$0 = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N \triangleq \mathscr{T} \tag{2.12}$$

such that both $x(t)$ and $u(t)$ are analytic functions in each open subinterval (t_{i-1}, t_i) for $i = 1, \cdots, N$. □

The proof of the above result is based on an expansion of the solution trajectory $x(t)$ near time t_*: Let $x^* = x(t_*)$ be the state of the solution trajectory $(x(t), u(t))$ at this time. Without loss of generality, we may assume that a nonnegative integer k exists such that $CA^j x^* = 0$ for all $j = 0, \cdots k - 1$. For all $t > t_*$,

$$\begin{aligned} x(t) = &\sum_{j=0}^{k+2} \frac{(t-t_*)^j}{j!}\, A^j x^* + \frac{(t-t_*)^{k+1}}{(k+1)!}\, Bu(CA^k x^*) \\ &+ \frac{(t-t_*)^{k+2}}{(k+2)!}\, Bu'(CA^k x^*; CA^{(k+1)}x^* + CBu(CA^k x^*)) + o((t-t_*)^{k+2}) \end{aligned}$$

$$\begin{aligned} u(t) = &\underbrace{\frac{(t-t_*)^k}{k!}}_{\text{dominant term}}\ u(CA^k x^*) \\ &+ \frac{(t-t_*)^{k+1}}{(k+1)!}\, u'(CA^k x^*; CA^{(k+1)}x^* + CBu(CA^k x^*)) + o((t-t_*)^{k+1}). \end{aligned}$$

Here $u(CA^k x^*)$ is the unique solution of the LCP $(CA^k x^*, D)$. The latter expansion establishes that locally near t_*, the sign of $u_i(t)$ is dictated by the sign of $u_i(CA^k x^*)$, where k is the first nonnegative index for which $CA^k x^* \neq 0$; similarly for $w(t) \triangleq Cx(t) + Du(t)$. The proof is completed by an inductive argument via a dimension reduction of the LCS. A similar expansion and argument can be derived for $t < t_*$. The existence of the partition (2.12) and the analyticity of $(x(t), u(t))$ on each of the (open) subintervals follow from the piecewise constancy of the triple of index sets $(\alpha(t), \beta(t), \gamma(t))$ which implies that each of the subintervals, the pair of trajectories coincide with the solution of the linear DAE:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ 0 &= C_{\alpha\bullet} x + D_{\alpha\alpha} u_\alpha \\ 0 &= u_\beta, \quad \text{and} \quad u_\gamma = 0 \end{aligned}$$

with the principal submatrix $D_{\alpha\alpha}$ being nonsingular.

Calling a state $x^* = x(t_*)$ for which the triple of index set $(\alpha(t), \beta(t), \gamma(t))$ remains constant for all times t sufficiently close to the nominal time t_* in the sense of Theorem 1 *non-Zeno*, we deduce from this theorem that the unique solution trajectory of an LCS defined by the tuple (A, B, C, D) with D being a P-matrix has no Zeno states. A similar result holds for the NCS (2.10) for analytic functions F and G under the assumption that the state u^* is a *strongly regular* solution of the NCP

$$0 \leq u \perp G(x^*, u) \geq 0. \tag{2.13}$$

Strong regularity can be characterized in a number of ways. In particular, a matrix-theoretic characterization of the condition is as follows. Writing $(\alpha_*, \beta_*, \gamma_*) \triangleq (\alpha(t_*), \beta(t_*), \gamma(t_*))$, we may partition the Jacobian matrix $J_u G(x^*, u^*)$ as

$$\begin{bmatrix} J_{\alpha_*} G_{\alpha_*}(x^*, u^*) & J_{\beta_*} G_{\alpha_*}(x^*, u^*) & J_{\gamma_*} G_{\alpha_*}(x^*, u^*) \\ J_{\alpha_*} G_{\beta_*}(x^*, u^*) & J_{\beta_*} G_{\beta_*}(x^*, u^*) & J_{\gamma_*} G_{\beta_*}(x^*, u^*) \\ J_{\alpha_*} G_{\gamma_*}(x^*, u^*) & J_{\beta_*} G_{\gamma_*}(x^*, u^*) & J_{\gamma_*} G_{\gamma_*}(x^*, u^*) \end{bmatrix}.$$

It is known [45, Corollary 5.3.20] that u^* is a strongly regular solution of the NCP (2.13) if and only if: a) the principal submatrix $J_{\alpha_*} G_{\alpha_*}(x^*, u^*)$ is nonsingular, and b) the Schur complement

$$J_{\beta_*} G_{\beta_*}(x^*, u^*) - J_{\alpha_*} G_{\beta_*}(x^*, u^*) \left[J_{\alpha_*} G_{\alpha_*}(x^*, u^*) \right]^{-1} J_{\beta_*} G_{\alpha_*}(x^*, u^*)$$

is a P-matrix. Under this assumption, it follows that there exist open neighborhoods $\mathscr{V}$ and $\mathscr{U}$ of x^* and u^*, respectively, and a Lipschitz continuous function $u : \mathscr{V} \to \mathscr{U}$ such that for every $x \in \mathscr{V}$, $u(x)$ is the unique vector u in $\mathscr{U}$ that satisfies the

NCP: $0 \leq u \perp G(x, u) \geq 0$. Employing this NCP solution function, it follows that near the time t_*, there is a unique solution trajectory $(x(t), u(t))$ of the NCS (2.10) passing through the pair $(x(t_*), u(t_*))$ at time t_* and staying near this base pair. Moreover, we can derive a similar expansion of this solution trajectory similar to that of the LCS. To describe this expansion, let $L_f^k C(x)$ denote the Lie derivative [73, 95] of a smooth vector-valued function $C(x)$ with respect to the vector field $f(x)$, that is, $L_f^0 C(x) \triangleq C(x)$, and inductively,

$$L_f^k C(x) = \left[J L_f^{k-1} C(x) \right] f(x), \qquad \text{for } k \geq 1,$$

where $J L_f^{k-1} C(x)$ denote the Jacobian matrix of the vector function $L_f^{k-1} C(x)$. See the cited references for the fundamental role the Lie derivatives play in nonlinear systems theory. In deriving the solution expansion of the solution pair $(x(t), u(t))$ near time t_*, we take $u^* = G(x^*, u^*) = 0$ without loss of generality because otherwise, i.e., if either $u^* \neq 0$ or $G(x^*, u^*) \neq 0$, we may then reduce the dimension of the algebraic variable u by at least one and obtain a system locally equivalent to the original NCS for which the induction hypothesis is applicable to establish the constancy of the index sets $(\alpha(t), \beta(t), \gamma(t))$ near t_*. With $u^* = G(x^*, u^*) = 0$, let $f(x) \triangleq F(x, 0)$ and $C(x) \triangleq G(x, 0)$. The strong regularity of u^* implies that the Jacobian matrix $D_* \triangleq JG(x^*, 0)$ is a P-matrix. Suppose that $L_f^i C(x^*) = 0$ for all $i = 1, \cdots, k-1$, then for all $t > t_*$ sufficiently near t_*,

$$\begin{aligned} x^*(t) &= x^* + \sum_{j=0}^{k} \frac{(t-t_*)^{(j+1)}}{(j+1)!} L_f^j f(x^*) \\ &\quad + \frac{(t-t_*)^{(k+1)}}{(k+1)!} J_u F(x^*, u^*) v^{k*} + o((t-t_*)^{k+1}) \\ u^*(t) &= \frac{(t-t_*)^k}{k!} v^{k*}, \end{aligned}$$

where v^{k*} is the unique solution to the LCP $(L_f^k C(x^*), D)$. Based on this assumption and dividing the argument into two cases: $L_f^k C(x^*) = 0$ for all nonnegative integer k or otherwise, we can complete the proof of the desired invariance of the index triple $(\alpha(t), \beta(t), \gamma(t))$ near t_*.

The above non-Zeno results have been extended to the Karush-Kuhn-Tucker (KKT) system derived from the DVI (2.4) by assuming that $K(x, y)$ is a polyhedron given by $\{v \mid D(x, y) + Ev \geq 0\}$ and $H(x, y, v) \triangleq C(x, y) + N(x)v$:

$$\begin{aligned} \dot{x} &= A(x, y) + B(x, y)v, \\ 0 &\leq y \perp G(x, y) \geq 0, \\ 0 &= C(x, y) + N(x)v - E^T \lambda, \\ 0 &\leq \lambda \perp D(x, y) + Ev \geq 0, \end{aligned} \tag{2.14}$$

where E is a constant matrix, and λ represents the Lagrange multiplier of the constraint $D(x, y) + Ev \geq 0$, which is non-unique in general. The condition $0 = C(x, y) + N(x)v - E^T\lambda$ implies that the vector $C(x, y) + N(x)v$ belongs to the conical hull of the columns of E^T, which we denote $\text{pos}(E^T)$. In addition to the fundamental index sets $\alpha(t)$, $\beta(t)$, and $\gamma(t)$ associated with the complementarity condition: $0 \leq y \perp G(x, y) \geq 0$, we also have the following index sets for the second complementarity condition:

$$\mathscr{I}(t) \triangleq \{ i \mid [D(x(t), y(t)) + Ev(t)]_i = 0 \},$$
$$\mathscr{J}(t) \triangleq \{ i \mid [D(x(t), y(t)) + Ev(t)]_i > 0 \}.$$

Furthermore, for a given state (x^*, y^*, v^*) of a solution trajectory $(x(t), y(t), v(t))$ at time t_*, we assume that (i) the functions A, B, G, C, D, and N are analytic in a neighborhood of (x^*, y^*); (ii) v^* is a strongly regular solution of the NCP: $0 \leq y \perp G(x^*, y) \geq 0$; (iii) the matrix $N(x^*)$ is positive definite; and (iv) $K(x, y) \neq \emptyset$ for any (x, y) with $y \geq 0$ in a neighborhood of (x^*, y^*). Under these assumptions, the following two properties can be proved:

- For all (x, y) near (x^*, y^*) with $y \geq 0$, $\text{SOL}(K(x, y); H(x, y, \bullet))$ is a singleton whose unique element we denote $v(x, y)$;
- The solution function $v(x, y)$ for $y \geq 0$ is Lipschitz continuous and piecewise analytic near (x^*, y^*).

In terms of the above implicit functions $y(x)$ and $v(x, y)$, the DVI is equivalent to $\dot{x} = \Upsilon(x) \triangleq A(x, y(x)) + B(x, y(x))v(x, y(x))$, with Υ being Lipschitz near x^*.

Like the LCS, we first treat the case where the unique solution, denoted $\widehat{x}^f$, to the ODE: $\dot{x} = f(x) \triangleq A(x, 0)$ with $x(0) = x^*$ derived from $u = 0$ and $y = 0$ is the (unique) solution of (2.14). Let

$$g(x) \triangleq G(x, 0); \;\; h(x) \triangleq D(x, 0), \;\; \text{and}\, c(x) \triangleq C(x; 0).$$

If (a) $L_f^j g(x^*) = 0$ and $L_f^j h(x^*) = 0$ for all j, and (b) $c(\widehat{x}^f(t)) \in \text{pos}(E^T)$ for all $t \geq 0$, then $(\widehat{x}^f, 0, 0)$ is the unique solution to (2.14). The remaining analysis treats the case where the conditions (a) and (b) do not hold; a solution expansion is derived that enables an inductive argument to complete the proof of the following theorem. For details, see [55].

Theorem 2 *Under the above assumptions (i)–(iv), there exist a scalar $\varepsilon_* > 0$ and two tuples of index sets $(\alpha_\pm, \beta_\pm, \gamma_\pm, \mathscr{I}_\pm, \mathscr{J}_\pm)$ such that*

$$\big(\alpha(t), \beta(t), \gamma(t), \mathscr{I}(t), \mathscr{J}(t)\big) = \big(\alpha_+, \beta_+, \gamma_+, \mathscr{I}_+, \mathscr{J}_+\big), \; \forall\, t \in (t_*, t_* + \varepsilon_*],$$
$$\big(\alpha(t), \beta(t), \gamma(t), \mathscr{I}(t), \mathscr{J}(t)\big) = \big(\alpha_-, \beta_-, \gamma_-, \mathscr{I}_-, \mathscr{J}_-\big), \; \forall\, t \in [t_* - \varepsilon_*, t_*).$$

In the application of the system (2.4) to contact problems with Coulomb friction [120], the set $K(x, y)$ is the Lorentz cone. Presently, the extension of Theorem 2 to this non-polyhedral case remains unsolved.

2.8.2 Non-Zenoness of CLSs

Consider the CLS defined by:

$$\dot{x} = A^i x, \quad \text{if } x \in \mathscr{C}^i \tag{2.15}$$

where the family of polyhedral cones $\{\mathscr{C}^i\}_{i=1}^I$ is a polyhedral conic subdivision of $\mathbb{R}^n$. This system is said to satisfy the (forward and backward) non-Zeno property if for any initial condition $x^0 \in \mathbb{R}^n$ and any $t_* \geq 0$, there exist a scalar $\varepsilon_+ > 0$ and indices $i_\pm \in \{1, \cdots, I\}$ such that $x(t; x^0) \in \mathscr{C}^{i_+}$ for all $t \in [t_*, t_* + \varepsilon_*]$, and for any $t_* > 0$, $x(t; x^0) \in \mathscr{C}^{i_-}$ for all $t \in [t_* - \varepsilon_*, t_*]$ (backward-time non-Zeno). A time $t_* \in [0, \mathscr{T}]$ is *non-switching in the weak sense* if there exist $\varepsilon_* > 0$ and an index $i_* \in \{1, \cdots, I\}$ such that $x(t) \in \mathscr{C}^{i_*}$ for all $t \in [t_* - \varepsilon_*, \ t_* + \varepsilon_*]$.

Proposition 2 *The CLS (2.15) has the non-Zeno property. Moreover, every solution trajectory of the CLS has at most a finite number of switching times in* $[0, \mathscr{T}]$. □

Not surprisingly, we can also establish the constancy of index sets for the CLS similar to that for the P-matrix case of the LCS. We refer the readers to [30] where a proof to Proposition 2 and many other related results for the CLS can be found.

2.9 Lecture II: Summary

In this lecture, relying heavily on the theories of the LCP and strong regularity, we have

- explained the Zeno phenomenon and present its formal definition;
- sketched how the non-Zenoness of certain LCS/DVI can be analyzed;
- presented a solution expansion of the trajectory near a nominal state; and
- briefly touched on the property of switching of cone-wise linear complementarity systems.

2.10 Lecture III: Numerical Time-Stepping

We next discuss numerical methods for approximating a solution trajectory to the initial-value, time-invariant DVI:

$$\begin{aligned} \dot{x} &= F(x, y, v), \quad x(0) = x^0 \\ 0 &\leq y \perp G(x, y) \geq 0 \\ v &\in \text{SOL}(K(x, y), H(x, y, \bullet)). \end{aligned} \tag{2.16}$$

Let $h > 0$ be a time step so that $N_h \triangleq \mathscr{T}/h$ is an integer. Let $t_{h,0} \triangleq 0$ and inductively $t_{h,i+1} \triangleq t_{h,i} + h$, for $i = 0, 1, \cdots, N_h - 1$. We approximate the time derivative by the forward divided difference:

$$\dot{x}(t) \approx \frac{x(t+h) - x(t)}{h}$$

and let $x^{h,i} \approx x(t_{h,i})$, $y^{h,i} \approx x(t_{h,i})$, and $v^{h,i} \approx v(t_{h,i})$ be discrete-time iterates approximating the continuous-time solution trajectory at the discrete time sequence:

$$0 = t_{h,0} < t_{h,1} < \cdots < t_{h,N_h-1} < t_{h,N_h} = \mathscr{T}.$$

For a given step size $h > 0$, we generate the discrete-time iterates

$$\{ x^{h,i};\, y^{h,i};\, v^{h,i} \}_{i=0}^{N_h} \tag{2.17}$$

by solving a finite-dimensional subproblem. Using the above iterates, we then construct discrete-time trajectories by interpolation as follows:

- The state trajectory $x^h(t)$ is obtained by piecewise linear interpolation joining consecutive iterates; specifically, for $i = 0, 1, \cdots, N_h - 1$,

$$x^h(t) \triangleq x^{h,i} + \frac{x^{h,i+1} - x^{h,i}}{h}, \quad \text{for } t \in [\, t_{h,i}, t_{h,i+1}\,].$$

- The algebraic trajectories $y^h(t)$ and $v^h(t)$ are obtained as piecewise constant functions; specifically, for $i = 0, 1, \cdots, N_h - 1$,

$$\left.\begin{aligned} y^h(t) &\triangleq y^{h,i+1} \\ v^h(t) &\triangleq v^{h,i+1} \end{aligned}\right\} \quad \text{for } t \in [\, t_{h,i}, t_{h,i+1}\,].$$

It is desirable that these numerical trajectories converge in a sense to be specified, at least subsequentially, to some limiting trajectories that constitute a weak solution of the DVI.

To facilitate the implementation and analysis of the iterations, the discrete-time subproblems need to be defined carefully. For this purpose, we focus on the case where the function $F(x, y, v)$ is given by $F(x, y, v) = A(x) + B(x)y + C(x)v$; cf. (2.4) for some vector-valued function A and matrix-valued functions B and C. Note the linearity in the pair $u \triangleq (y, v)$ for fixed x. Let $x^{h,0} = x^0$. At time step $i \geq 0$, we generate the next iterate $(x^{h,i+1}, y^{h,i+1}, v^{h,i+1})$ by a *backward Euler semi-implicit scheme*:

$$
\begin{aligned}
x^{h,i+1} &= x^{h,i} + h \left[\underbrace{A(x^{h,i+1}) + B(x^{h,i})y^{h,i+1} + C(x^{h,i})v^{h,i+1}}_{\text{note the presence of the unknown iterates}} \right] \\
0 &\leq y^{h,i+1} \perp G(x^{h,i+1}, y^{h,i+1}) \geq 0 \\
v^{h,i+1} &\in \text{SOL}(K(x^{h,i+1}, y^{h,i+1}), H(x^{h,i+1}, y^{h,i+1}, \bullet)).
\end{aligned}
$$

From the first equation, we may solve for $x^{h,i+1}$ in terms of $(x^{h,i}, y^{h,i+1}, u^{h,i+1})$ for all $h > 0$ sufficiently small, provided that $\|A(x)\|$ is bounded uniformly for all x. Specifically, we have

$$
x^{h,i+1} = [\mathbf{I} - h\, A(\bullet)\,]^{-1} \left\{ x^{h,i} + h \left(B(x^{h,i})y^{h,i+1} + C(x^{h,i})v^{h,i+1} \right) \right\}. \tag{2.18}
$$

This solution function may then be substituted into the complementarity and variational conditions, yielding: a *quasi-variational inequality* (QVI):

$$
u^{h,i+1} \triangleq \begin{pmatrix} y^{h,i+1} \\ v^{h,i+1} \end{pmatrix} \in \text{SOL}\left(\widehat{L}^{h,i} \left(\underbrace{u^{h,i+1}}_{\text{quasi nature}} \right), \widehat{\Phi}^{h,i} \right),
$$

where $\widehat{L}^{h,i}(u^{h,i+1}) \triangleq \mathbb{R}^{\ell_1} \times K(x^{h,i+1}, y^{h,i+1})$ and

$$
\widehat{\Phi}^{h,i}(u^{h,i+1}) \triangleq \begin{pmatrix} \widehat{G}^{h,i}(u^{h,i+1}) \\ \widehat{H}^{h,i}(u^{h,i+1}) \end{pmatrix} \triangleq \begin{pmatrix} G(x^{h,i+1}, y^{h,i+1}) \\ H(x^{h,i+1}, y^{h,i+1}, v^{h,i+1}) \end{pmatrix}
$$

with $x^{h,i+1}$ substituted by the right-hand side in (2.18). Needless to say, the solvability of the latter QVI is key to the well-definedness of the iterations; more importantly, we need to demonstrate that the above QVI has a solution for all $h > 0$ sufficiently small. Such a demonstration is based on [45, Corollary 2.8.4] specialized to the QVI on hand; namely, with $m = \ell_1 + \ell_2$ where ℓ_1 and ℓ_2 are the dimensions of the vectors y and v, respectively, and with "cl" and "bd" denoting, respectively, the closure and boundary of a set, there exists $\bar{h} > 0$ such that for all $h \in (0, \bar{h}]$,

- $\widehat{L}^{h,i} : \mathbb{R}^m \to \mathbb{R}^m$ is closed-valued and convex-valued;
- there exist a bounded open set $\Omega \subset \mathbb{R}^m$ and a vector $u^{h,i;\text{ref}} \in \Omega$ such that

(a) for every $\bar{u} \in \text{cl}\, \Omega$, $\widehat{L}^{h,i}(\bar{u}) \neq \emptyset$ and the set limit holds: $\lim\limits_{u \to \bar{u}} \widehat{L}^{h,i}(u) = \widehat{L}^{h,i}(\bar{u})$;
(b) $z^{h;\text{ref}} \in \widehat{L}^{h,i}(u)$ for every $u \in \text{cl}\, \Omega$; and
(c) the set $\left\{ u \in \widehat{L}^{h,i}(u) \mid (u - u^{h,i;\text{ref}})^T \widehat{\Psi}^{h,i}(u) < 0 \right\} \cap \text{bd}\, \Omega = \emptyset$.

The above conditions can be shown to hold under the following postulates on the DVI (2.4):

- $A : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and $\sup\limits_{x \in \mathbb{R}^n} \| A(x) \| < \infty$;
- $K : \mathbb{R}^n \times \mathbb{R}^{\ell_1}_+ \to \mathbb{R}^{\ell_2}$ is a nonempty-valued, closed-valued, convex-valued, and continuous set-valued map;
- there exists $v^{\text{ref}} \in K(x, y)$ for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^{\ell_1}_+$;
- the map $(y, v) \mapsto \begin{pmatrix} G(x, y) \\ H(x, y, v) \end{pmatrix}$ is strongly monotone on $\mathbb{R}^{\ell_1}_+ \times \mathbb{R}^{\ell_2}$ with a modulus that is independent of $x \in \mathbb{R}^n$.

These postulates are very loose and can be sharpened; they are meant to illustrate the solvability of the subproblems under a set of reasonable assumptions. We will return to discuss more about this issue for the LCS subsequently.

Having constructed the numerical trajectories $\left\{ (x^h(t), y^h(t), v^h(t)) \mid h > 0 \right\}$, by piecewise interpolation, we next address the convergence of these trajectories. The result below summarizes the criteria for convergence; for a proof, see [102].

Proposition 3 *Suppose that there exist positive constants $\bar{h}$, c_x, and c_u such that for all $h \in (0, \bar{h}]$, and all integers $i = 0, 1, \cdots, N_h - 1$,*

$$\| u^{h,i+1} \| \leq c_u \left(1 + \| x^{h,i} \| \right) \quad \textit{and} \quad \| x^{h,i+1} - x^{h,i} \| \leq h\, c_x \left(1 + \| x^{h,i} \| \right).$$

The following two statements hold:

- *boundedness: there exist positive constants c_{0x}, c_{1x}, c_{0u}, and c_{1u} such that for all $h \in (0, \bar{h}]$,*

$$\max_{0 \leq i \leq N_h} \| x^{h,i} \| \leq c_{0x} + c_{1x} \| x^{h,0} \| \quad \textit{and} \quad \max_{1 \leq i \leq N_h} \| x^{h,i} \| \leq c_{0u} + c_{1u} \| x^{h,0} \|;$$

- *convergence: there exists a subsequence $\{h_\nu\} \downarrow 0$ such that the following two limits exists: $x^{h_\nu} \to \widehat{x}^\infty$ uniformly in $[0, \mathscr{T}]$ and $u^{h_\nu} \to \widehat{u}^\infty = (\widehat{y}^\infty, \widehat{v}^\infty)$ weakly in $L^2(0, \mathscr{T})$ as $\nu \to \infty$.* □

The steps in the proof of the above theorem are as follows. Apply the Arzelá-Ascoli Theorem [79, page 57–59] to deduce the uniform convergence of a subsequence $\{\widehat{x}^{h_\nu}\}_{\nu \in \kappa}$ to a limit $\widehat{x}^\infty$ in the supremum, i.e., L$^\infty$-norm:

$$\lim_{\nu (\in \kappa) \to \infty} \sup_{t \in [0, \mathscr{T}]} \| \widehat{x}^{h_\nu}(t) - \widehat{x}^\infty(t) \| = 0.$$

Next, apply Alaoglu's Theorem [79, page 71–72] to deduce the weak convergence of a further subsequence $\{\widehat{u}^{h_\nu}\}_{\nu\in\kappa'}$, where $\kappa' \subseteq \kappa$ to a limit $\widehat{u}^\infty$, which implies: for any function $\varphi \in \mathrm{L}^2(0, \mathscr{T})$

$$\lim_{\nu(\in\kappa')\to\infty} \int_0^{\mathscr{T}} \varphi(s)^T \widehat{u}^{h_{\nu'}}(s)\, ds \;=\; \int_0^{\mathscr{T}} \varphi(s)^T \widehat{u}^{\infty}(s)\, ds.$$

By Mazur's Theorem [79, page 88], we deduce the strong convergence of a sequence of convex combinations of $\{\widehat{u}^{h_\nu}\}_{\nu\in\kappa'}$. A subsequence of such convex combinations converges pointwise to $\widehat{u}^\infty$ for almost all times in $[0, \mathscr{T}]$; hence by convexity of the graph $\mathrm{Gr}(K)$ of the set-valued map K, it follows that $(\widehat{x}^\infty(t), \widehat{u}^\infty(t)) \in \mathrm{Gr}(K)$ for almost all $t \in [0, \mathscr{T}]$.

Ideally, one would want to establish that every limit tuple $(\widehat{x}^\infty, \widehat{y}^\infty, \widehat{v}^\infty)$ is a weak solution of the DVI (2.4). Nevertheless, the generality of the setting makes this not easy; the case where $G(x, y)$ and $H(x, y, v)$ are separable in their arguments and the set-valued map K is a constant has been analyzed in detail in [102]; the paper [103] analyzed the convergence of a time-stepping method for solving the DVI arising from a frictional contact problem with local compliance. In the next section, we present detailed results for the initial-value LCS (2.3).

2.11 Lecture III: The LCS

Consider the time-invariant, initial-value LCS (2.3):

$$\begin{aligned} \dot{x} &= Ax + Bu, \qquad x(0) \;=\; x^0 \\ 0 \le{}& u \perp Cx + Du \;\ge\; 0. \end{aligned} \tag{2.19}$$

For this system, the semi-implicit scheme becomes a (fully) implicit scheme:

$$\begin{aligned} x^{h,i+1} &= x^{h,i} + h\left(Ax^{h,i+1} + Bu^{h,i+1}\right), \qquad i \;=\; 0, 1, \cdots, N_h - 1 \\ 0 \qquad &\le u^{h,i+1} \perp Cx^{h,i+1} + Du^{h,i+1} \;\ge\; 0 \end{aligned}$$

Solving for $x^{h,i+1}$ in the first equation and substituting into the complementarity condition yields the LCP:

$$0 \le u^{h,i+1} \;\perp\; q^{h,i+1} + D^h u^{h,i+1} \;\ge\; 0, \tag{2.20}$$

where properties of the matrix $D^h \triangleq D - C\,[\,\mathbf{I} - hA\,]^{-1}\,B$ are central to the well-definedness and convergence of the scheme. The first thing to point out regarding the convergence of the numerical trajectories $(\widehat{x}^h, \widehat{u}^h)$ is that the matrix D is required to be positive semidefinite, albeit not necessarily definite.

We make several remarks regarding the above iterative algorithm. In general, the iteration matrix D^h is not necessarily positive semidefinite in spite of the same property of D. If the LCP (2.20) has multiple solutions, to ensure the boundedness of at least one such solution, use the least-norm solution obtained from:

$$\underset{u}{\text{minimize}} \parallel u \parallel^2 \quad \text{subject to (2.20);}$$

boundedness means $\|u^{h,i}\| \leq \rho(1 + \|x^{h,i}\|)$ for some constant $\rho > 0$. In general, the above LCP-constrained least-norm problem is a *quadratic program with linear complementarity constraints* [16, 17]. When D^h is positive semidefinite as in the case of a "passive" LCS (see definition below), such a least-norm solution can be obtained by an iterative procedure that solves a sequence of linear complementarity subproblems. The same matrix D^h appears in each time step $i = 0, 1, \cdots, N_h - 1$; thus some kind of warm-start algorithm can be exploited in practical implementation, if possible, for an initial-value problem. Nevertheless, as with all time-stepping algorithms, the stepwise procedure is not applicable for two-point boundary problems such as (instead of $x(0) = x^0$) $Mx(0) + Nx(\mathscr{T}) = b$, which couples all the iterates $\left\{x(0) \triangleq x^{h,0}, x^{h,1}, \cdots, x^{h,N_h} \triangleq x(\mathscr{T})\right\}$.

In order to present a general convergence result, we need to summarize some key LCP concepts that are drawn from [38]. Given a real square matrix M, the *LCP-Range* of M, denoted LCP-Range(M), is the set of all vectors q for which the LCP(q, M) is solvable, i.e., SOL(q, M) $\neq \emptyset$; the *LCP-Kernel of M*, which we denote LCP-Kernel(M), is the solution set SOL($0, M$) of the homogeneous LCP: $0 \leq v \perp Mv \geq 0$. An R$_0$-matrix is a matrix M for which LCP-Kernel(M) = $\{0\}$. For a given pair of matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times n}$, let $\overline{\mathscr{O}}(C, A)$ denote the unobservability space of the pair of matrices (C, A); i.e., $v \in \overline{\mathscr{O}}(C, A)$ if and only if $CA^i v = 0$ for all $i = 0, 1, \cdots, n - 1$. The result below employs these concepts; a proof can be found in [57].

Theorem 3 *Suppose the following assumptions hold:*

(A) D is positive semidefinite,
(B) Range(C) $\subseteq$ LCP-Range(D^h) for all $h > 0$ sufficiently small, and
(C) the implication below holds:

$$\left.\begin{array}{r} \textit{LCP-Kernel}(D) \ni u^\infty \perp s^\infty + CBu^\infty \in [\,\textit{LCP-Kernel}(D)]^* \\ \textit{for some } s^\infty \in \textit{LCP-Range}(D) \end{array}\right\} \tag{2.21}$$
$$\Rightarrow Bu^\infty \in \overline{\mathscr{O}}(C_{\beta\bullet}, A), \quad \textit{where } \beta \equiv \{i : (Du^\infty)_i = 0\}.$$

Then there exist an $\bar{h} > 0$ such that, for every x^0 satisfying $Cx^0 \in$ LCP-Range(D), the two trajectories $\widehat{x}^h(t)$ and $\widehat{u}^h(t)$ generated by the least-norm time-stepping scheme are well defined for all $h \in (0, \bar{h}]$ and there is a sequence $\{h_\nu\} \downarrow 0$ such that the following two limits exist: $\widehat{x}^{h_\nu}(\cdot) \to \widehat{x}(\cdot)$ uniformly on $[0, \mathscr{T}]$ and $\widehat{u}^{h_\nu}(\cdot) \to \widehat{u}(\cdot)$

weakly in $L^2(0,T)$. *Moreover, all such limits* $(\widehat{x}(\cdot), \widehat{u}(\cdot))$ *are weak solutions of the initial-value LCS (2.19).* □

A large class of LCSs that satisfy the assumptions of the above theorem is of *passive type* [28, 34]. A linear system $\Sigma(A, B, C, D)$ given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bz(t) \\ w(t) &= Cx(t) + Dz(t) \end{aligned} \tag{2.22}$$

is *passive* if there exits a nonnegative-valued function $V : \mathbb{R}^n \to \mathbb{R}_+$ such that for all $t_0 \leq t_1$ and all trajectories (z, x, w) satisfying the system (2.22), the following inequality holds:

$$V(x(t_0)) + \int_{t_0}^{t_1} z^T(t)w(t)dt \geq V(x(t_1)).$$

Passivity is a fundamental property in linear systems theory; see the two cited references. Moreover, it is well known that the system $\Sigma(A, B, C, D)$ is passive if and only if there exists a symmetric positive semidefinite matrix K such that the following symmetric matrix:

$$\begin{bmatrix} A^T K + KA & KB - C^T \\ B^T K - C & -(D + D^T) \end{bmatrix} \tag{2.23}$$

is negative semidefinite. Checking passivity can be accomplished by solving a linear matrix inequality using methods of semidefinite programming [19].

Corollary 1 *If* $\Sigma(A, B, C, D)$ *is passive, then the conclusion of Theorem 3 holds.* □

2.12 Lecture III: Boundary-Value Problems

A classic family of methods for solving boundary-value ODEs [11, 12, 72] (see also [132, Section 7.3]) is that of shooting methods. The basic idea behind these methods is to cast the boundary-value problem as a system of algebraic equations whose unknown is the initial condition that defines an initial-value ODE. An iterative method, such as a bisection or Newton method, for solving the algebraic equations is then applied; each evaluation of the function defining the algebraic equations requires solving an initial-value ODE. Multiple-shooting methods refine this basic idea by first partitioning the time interval of the problem into smaller sub-intervals on each of which a boundary-value problem is solved sequentially. Convergence of the overall method requires differentiability of the algebraic equations if a Newton-type method is employed to facilitate speed-up. In what follows, we apply the basic idea of a shooting method to the time-invariant, boundary-value DVI:

$$\begin{aligned} \dot{x} &= F(x,u) \\ u &\in \text{SOL}\,(K, H(x,\bullet)) \\ 0 &= \Psi(x(0), x(\mathscr{T})). \end{aligned} \tag{2.24}$$

For simplicity, suppose that the mapping $H(x,\bullet)$ is strongly monotone with a modulus independent of x. This implies for every x^0, the ODE $\dot{x} = F(x, u(x))$ with $x(0) = x^0$, where $u(x)$ is the unique solution of the VI $(K, H(x,\bullet))$, has a unique solution which we denote $x(t; x^0)$. This solution determines the terminal condition $x(\mathscr{T})$ uniquely as $x(\mathscr{T}; x^0)$. Thus the two-point boundary-value problem becomes the algebraic equation:

$$\Phi(x^0) \triangleq \Psi(x^0, x(\mathscr{T}; x^0)) = 0. \tag{2.25}$$

As a function of its argument, Φ is typically not differentiable; thus a fast method such as the classical Newton method is not applicable. Nevertheless, a "semismooth Newton method" [45, Chapter 7] can be applied provided that Φ is a semismooth function. In turn, this will be so if for instance the boundary function Ψ is differentiable and the solution $x(\mathscr{T};\bullet)$ depends semismoothly on the initial condition x^0. In what follows, we present the semismoothness concept [106] and the resulting Newton method for finding a root of Φ, which constitutes the basic shooting method for solving the boundary-value DVI (2.24). We warn the reader that this approach has not been tested in practice and refinements are surely needed for the method to be effective.

There are several equivalent definitions of semismoothness. The following one based on the directional derivative is probably the most elementary without requiring advanced concepts in nonsmooth analysis.

Definition 1 Let $G : \Omega \subseteq \Re^n \to \Re^m$, with Ω open, be a locally Lipschitz continuous function on Ω. We say that G is *semismooth* at a point $\bar{x} \in \Omega$ if G is directionally differentiable (thus B(ouligand)-differentiable) near $\bar{x}$ and such that

$$\lim_{\bar{x} \neq x \to \bar{x}} \frac{\| G'(x; x - \bar{x}) - G'(\bar{x}; x - \bar{x}) \|}{\| x - \bar{x} \|} = 0. \tag{2.26}$$

If the above requirement is strengthened to

$$\limsup_{\bar{x} \neq x \to \bar{x}} \frac{\| G'(x; x - \bar{x}) - G'(\bar{x}; x - \bar{x}) \|}{\| x - \bar{x} \|^2} < \infty, \tag{2.27}$$

we say that G is *strongly semismooth* at $\bar{x}$. If G is (strongly) semismooth at each point of Ω, then we say that G is (strongly) semismooth on Ω. □

Let $\Xi_{\mathscr{T}} \triangleq \{x(t,\xi^0) \mid t \in [0,\mathscr{T}]\}$, where $x(t,\xi)$ is a solution of the ODE with initial condition: $\dot{x} = f(x)$ and $x(0) = \xi^0$. Suppose that f is Lipschitz continuous and directionally differentiable, thus B(ouligand)-differentiable, in an open neighborhood $\mathscr{N}_T$ containing $\Xi_{\mathscr{T}}$. The following two statements hold:

- $x(t,\bullet)$ is B-differentiable at ξ^0 for all $t \in [0,\mathscr{T}]$ with the directional derivative $x(t,\bullet)'(\xi^0;\eta)$ being the unique solution $y(t)$ of the *directional ODE*:

$$\dot{y}(t) = f'(x(t,\xi^0);y); \qquad y(0) = \eta; \tag{2.28}$$

- if f is semismooth at all points in $\mathscr{N}_{\mathscr{T}}$, then $x(t,\bullet)$ is semismooth at ξ^0 for all $t \in [0,\mathscr{T}]$.

Before introducing the semismooth-based shooting method, we need to address the question of when the solution $u(x)$ of the VI $(K, H(x,\bullet))$ is a semismooth function of x. A broad class of parametric VIs that possess this property is when K is a polyhedron, or more generally, a finitely representable convex set satisfying the constant rank constraint qualification (CRCQ) [45, Chapter 4]. For simplicity, we present the results for the case where K is a polyhedron. Suppose that u^* is a strongly regular solution of the VI $(K, H(x^*,\bullet))$. Similar to the NCP, there exist open neighborhoods $\mathscr{V}$ and $\mathscr{U}$ of x^* and u^*, respectively, and a B-differentiable function $u : \mathscr{V} \to \mathscr{U}$ such that for every $x \in \mathscr{V}$, $u(x)$ is the unique solution in $\mathscr{U}$ that is a solution of the VI $(K, H(x,\bullet))$; moreover, the directional derivative $u'(x^*;dx)$ at x^* along a direction dx is given as the unique solution du of the CP:

$$\begin{aligned} du &\in \underbrace{\mathscr{T}(K;x^*) \cap H(x^*,u^*)^{\perp}}_{\text{critical cone } \mathscr{C}(x^*)} \\ J_xH(x^*,u^*)dx + J_uH(x^*,u^*)du &\in \left[\, \mathscr{T}(K;x^*) \cap H(x^*,u^*)^{\perp} \,\right]^* \\ du &\perp J_xH(x^*,u^*)dx + J_uH(x^*,u^*)du, \end{aligned} \tag{2.29}$$

where $\mathscr{T}(K;x^*)$ is the tangent cone of K at x^* and $H(x^*,u^*)^{\perp}$ denotes the orthogonal complement of the vector $H(x^*,u^*)$.

Returning to the DVI (2.24), we may deduce that, assuming the differentiability of F and H, the strong regularity of the solution $u(x(t,\xi^0))$ of the VI $(K, H(x(t,\xi^0),\bullet))$, and by combining (2.28) with (2.29), $x(t,\bullet)'(\xi^0;\eta)$ is the unique solution $y(t)$ of:

$$\begin{aligned} &\dot{y}(t) = J_xF(x(t,\xi^0),u(x(t,\xi^0)))y + J_uF(x(t,\xi^0),u(x(t,\xi^0)))v, \\ &y(0) = \eta v \in \mathscr{C}(x(t,\xi^0)) \\ &J_xH(x(t,\xi^0),u(x(t,\xi^0)))y + J_uH(x(t,\xi^0),u(x(t,\xi^0)))v \in \left[\mathscr{C}(x(t,\xi^0))\right]^* \\ &v \perp J_xH(x(t,\xi^0),u(x(t,\xi^0)))y + J_uH(x(t,\xi^0),u(x(t,\xi^0)))v, \end{aligned}$$

or equivalently,

$$\begin{aligned}
\dot{x} &= F(x,u), \quad x(0) = \xi^0 \\
\dot{y} &= J_x F(x,u) y + J_u F(x,u) v, \quad y(0) = \eta \\
u &\in \text{SOL}(K, H(x,\bullet)) \\
\mathscr{C}(x) \ni v &\perp J_x H(x,u) y + J_u H(x,u) v \in [\mathscr{C}(x(t))]^* .
\end{aligned}$$

Let

$$z \triangleq \begin{pmatrix} x \\ y \end{pmatrix}, \; w \triangleq \begin{pmatrix} u \\ v \end{pmatrix}, \; z^0 \triangleq \begin{pmatrix} \xi^0 \\ \eta \end{pmatrix}, \; \widehat{F}(z,w) \triangleq \begin{pmatrix} F(x,u) \\ J_x F(x,u) y + J_u F(x,u) v \end{pmatrix}$$

$$\widehat{K}(z) \triangleq K \times \mathscr{C}(x), \quad \text{and} \quad \widehat{H}(z,w) \triangleq \begin{pmatrix} \Psi(x,u) \\ J_x H(x,u) y + J_u H(x,u) v \end{pmatrix};$$

we deduce that the triple $(x(t,\xi^0), u(x(t,\xi^0)), x(t,\bullet)'(\xi^0;\eta))$ is the unique triplet (x,u,y), which together with an auxiliary variable z, satisfies the DVI:

$$\begin{aligned}
\dot{z} &= \widehat{F}(z,w); \quad z(0) = (\xi^0, \eta) \\
w &\in \text{SOL}(\widehat{K}(z), \widehat{H}).
\end{aligned}$$

This is a further instance of a DVI where the defining set of the variational condition varies with the state variable; cf. (2.4).

Returning to the algebraic equation reformulation (2.25) of the boundary-value DVI (2.4), we sketch the semismooth Newton method for finding a zero x^0 of the composite function $\Phi(x^0) = \Psi(x^0, x(\mathscr{T}, x^0))$. This is an iterative method wherein at each iteration ν, given a candidate zero ξ^ν, we compute a *generalized Jacobian matrix* $A(\xi^\nu)$ of Φ at ξ^ν and then let the next iterate $x^{\nu+1}$ be the (unique) solution of the (algebraic) linear equation: $\Phi(\xi^\nu) + A(\xi^\nu)(\xi - \xi^0) = 0$. This is the version of the method where a constant unit step size is employed. Under a suitable nonsingularity assumption at an isolated zero of Φ whose semismoothness is assumed, local superlinear convergence of the generated sequence of (vector) iterates can be proved; see [45, Chapter 7]. To complete the description of the method, the matrix $A(\xi^\nu)$ needs to be specified. As a composite function of Ψ and the solution function $x(\mathscr{T},\bullet)$, $A(\xi^\nu)$ can be obtained by the chain rule provided that a generalized Jacobian matrix of the latter function at the current iterate ξ^ν is available. Details of this can be found in [100] which also contains a statement of convergence of the overall method.

2.13 Lecture III: Summary

In this lecture, we have

- introduced a basic numerical time-stepping method for solving the DVI (2.4),
- provided sufficient conditions for the weak, subsequential convergence of the method,
- specialized the method to the LCS and presented sharpened convergence results, including the case of a passive system, and
- outlined a semismooth Newton method for solving an algebraic equation reformulation of a boundary-value problem, whose practical implementation requires further study.

2.14 Lecture IV: Linear-Quadratic Optimal Control Problems

An optimal control problem is the building block of a multi-agent optimization problem in continuous time. In this lecture, we discuss the linear-quadratic (LQ) case of the (single-agent) optimal control problem in preparation for the exposition of the multi-agent problem in the next lecture. The LQ optimal control problem with mixed state-control constraints is to find continuous-time trajectories $(x, u) : [0, \mathscr{T}] \to \mathbb{R}^{n+m}$ to

$$\begin{aligned}
&\underset{x,u}{\text{minimize}}\ V(x,u) \triangleq c^T x(\mathscr{T}) + \tfrac{1}{2}\, x(\mathscr{T})^T S x(\mathscr{T}) \\
&\quad + \int_0^{\mathscr{T}} \left[\begin{pmatrix} x(t) \\ u(t) \end{pmatrix}^T \begin{pmatrix} p(t) \\ q(t) \end{pmatrix} + \tfrac{1}{2} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}^T \begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \right] dt \\
&\text{subject to } x(0) = \xi, \text{ and for almost all } t \in [\,0, \mathscr{T}\,] \\
&\quad \frac{dx(t)}{dt} \triangleq \dot{x}(t) = Ax(t) + Bu(t) + r(t) \\
&\text{and} \quad \underbrace{Cx(t) + Du(t) + f \geq 0}_{\text{mixed state-control constraints}},
\end{aligned} \tag{2.30}$$

where the matrices $\Xi \triangleq \begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix}$ and S are symmetric positive semidefinite. Unlike these time-invariant matrices and the constant vector f, $(p; q; r)$ is a triple of properly dimensioned Lipschitz continuous vector functions. The *semi-coercivity* assumption on the objective function, as opposed to coercivity, is a major departure of this setting from the voluminous literature on this problem. For one thing, the existence of an optimal solution is no longer a trivial matter. Ideally, we should aim at deriving an analog of Proposition 4 below for a convex quadratic program in finite dimensions, which we denote QP$(Z(b), e, M)$:

$$\underset{z \in Z(b)}{\text{minimize}}\ e^T z + \tfrac{1}{2} z^T M z,$$

where M is an $m \times m$ symmetric positive semidefinite matrix, $Z(b) \triangleq \{z \in \mathbb{R}^m \mid Ez + b \geq 0\}$, where E is a given matrix and e and b are given vectors, all of appropriate orders. It is well known that the polyhedron $Z(b)$ has the same recession cone $Z_\infty \triangleq \{v \in \mathbb{R}^m \mid Ev \geq 0\}$ for all b for which $Z(b) \neq \emptyset$. Let SOL$(Z(b), e, M)$ denote the optimal solution set of the above QP.

Proposition 4 *Let M be symmetric positive semidefinite and let E be given. The following three statements hold.*

(a) For any vector b for which $Z(b) \neq \emptyset$, a necessary and sufficient condition for the QP$(Z(b), e, M)$ to have an optimal solution is that $e^T d \geq 0$ for all d in $Z_\infty \cap ker(M)$.
(b) If $Z_\infty \cap ker(M) = \{0\}$, then SOL$(Z(b), e, M) \neq \emptyset$ for all (e, b) for which $Z(b) \neq \emptyset$.
(c) If SOL$(Z(b), e, M) \neq \emptyset$, then SOL$(Z(b), e, M) = \{z \in Z(b) \mid Mz = M\widehat{z},\ e^T\widehat{z} = e^T\widehat{z}\}$ for any optimal solution $\widehat{z}$; thus MSOL$(Z(b), e, M)$ is a singleton whenever it is nonempty.

Extending the KKT conditions for the QP$(Z(b), e, M)$:

$$\begin{aligned} 0 &= e + Mz - E^T \mu \\ 0 &\leq \mu \perp Ez + b \geq 0, \end{aligned}$$

we can derive a 2-point BV-LCS formulation of (2.30) as follows. We start by defining the Hamiltonian function:

$$H(x, u, \lambda) \triangleq x^T p + u^T q + \tfrac{1}{2} x^T P x + x^T Q u + \tfrac{1}{2} u^T R u + \lambda^T\, (\, Ax + Bu + r\,),$$

where λ is the costate (also called adjoint) variable of the ODE $\dot{x}(t) = Ax(t) + Bu(t) + r$, and the Lagrangian function:

$$L(x, u, \lambda, \mu) \triangleq H(x, u, \lambda) - \mu^T\, (\, Cx + Du + f\,),$$

where μ is the Lagrange multiplier of the algebraic constraint: $Cx + Du + f \geq 0$. Inspired by the Pontryagin Principle [139, Section 6.2] and [61, 112], we introduce the following DAVI:

$$\begin{aligned}
&\begin{pmatrix} \dot{\lambda}(t) \\ \dot{x}(t) \end{pmatrix} = \begin{pmatrix} -p(t) \\ r(t) \end{pmatrix} + \begin{bmatrix} -A^T & -P \\ 0 & A \end{bmatrix} \begin{pmatrix} \lambda(t) \\ x(t) \end{pmatrix} + \begin{bmatrix} -Q \\ B \end{bmatrix} u(t) + \begin{bmatrix} C^T \\ 0 \end{bmatrix} \mu(t) \\
&\qquad \left.\begin{aligned} 0 &= q(t) + Q^T x(t) + Ru(t) + B^T \lambda(t) - D^T \mu(t) \\ 0 &\le \mu(t) \perp Cx(t) + Du(t) + f \ge 0 \end{aligned}\right\} \\
&\qquad \left[\Rightarrow u(t) \in \operatorname*{argmin}_{u \in U(x(t))} H(x(t), u, \lambda(t)) \right] \qquad (2.31) \\
&\qquad x(0) = \xi \quad \text{and} \quad \lambda(T) = c + Sx(T),
\end{aligned}$$

where $U(x) \triangleq \{u \in \mathbb{R}^m \mid Cx + Du + f \ge 0\}$. Note that the above is a DAVI with a boundary condition: $\lambda(T) = c + Sx(T)$; this is one challenge of this system. Another challenge is due to the mixed state-control constraint: $Cx(t) + Du(t) + f \ge 0$. While the membership $u(t) \in \operatorname*{argmin}_{u \in U(x(t))} H(x(t), u, \lambda(t))$ implies the existence of a multiplier $\widehat{\mu}(t)$ such that

$$\begin{aligned}
0 &= q(t) + Q^T x(t) + Ru(t) + B^T \lambda(t) - D^T \widehat{\mu}(t) \\
0 &\le \widehat{\mu}(t) \perp Cx(t) + Du(t) + f \ge 0,
\end{aligned}$$

we seek in (2.31) a particular multiplier $\mu(t)$ that also satisfies the ODE. So far, we have only formally written down the formulation (2.31) without connecting it to the optimal control problem (2.30). As a DAVI with (x, λ) as the pair of differential variables and (u, μ) as the pair of algebraic variables, the tuple (x, u, λ, μ) is a weak solution of (2.31) if (i) (x, λ) is absolutely continuous and (u, μ) is integrable on $[0, \mathscr{T}]$, (ii) the differential equation and the two algebraic conditions hold for almost all $t \in (0, \mathscr{T})$, and (iii) the initial and boundary conditions are satisfied.
In addition to the positive semidefiniteness assumption of the matrices Ξ and S, we need three more blanket conditions assumed to hold throughout the following discussion:

- (a feasibility assumption) a continuously differentiable function $\widehat{x}_{\text{fs}}$ with $\widehat{x}_{\text{fs}}(0) = \xi$ and a continuous function $\widehat{u}_{\text{fs}}$ exist such that for all $t \in [0, T]$: $d\widehat{x}_{\text{fs}}(t)/dt = A\widehat{x}_{\text{fs}}(t) + B\widehat{u}_{\text{fs}}(t) + r(t)$ and $\widehat{u}_{\text{fs}}(t) \in U(\widehat{x}_{\text{fs}}(t))$;
- (a primal condition) $[Ru = 0,\ Du \ge 0]$ implies $u = 0$;
- (a dual condition) $[D^T \mu = 0,\ \mu \ge 0]$ implies $(CA^i B)^T \mu = 0$ for all integers $i = 0, \cdots, n-1$, or equivalently, for all nonnegative integers i.

It is easy to see that the primal Slater condition: $[\exists\, \widehat{u}$ such that $D\widehat{u} > 0]$ implies the dual condition, but not conversely. For more discussion of the above conditions, particularly the last one, see [59].

Here is a roadmap of the main Theorem 4 below. It starts with the above postulates, under which part (I) asserts the existence of a weak solution of the DAVI (2.31) in the sense of Carathéodory. The proof of part (I) is based on a constructive numerical method. Part (II) of Theorem 4 asserts that any weak solution of the DAVI yields an optimal solution of (2.30); this establishes the sufficiency of the Pontryagin optimality principle. The proof of this part is based on a direct verification of the assertion, from which we can immediately obtain several properties characterizing an optimal solution of (2.30). These properties are summarized in part (III) of the theorem. From these properties, part (IV) shows that any optimal solution of (2.30) must be a weak solution of the DAVI (2.31), thereby establishing the necessity of the Pontryagin optimality principle. Finally, part (V) asserts the uniqueness of the solution obtained from part (I) under the positive definiteness of the matrix R.

Theorem 4 *Under the above setting and assumptions, the following five statements (I–V) hold.*
(I: Solvability of the DAVI) The DAVI (2.31) has a weak solution $(x^*, \lambda^*, u^*, \mu^*)$.
(II: Sufficiency of Pontryagin) If $(x^*, \lambda^*, u^*, \mu^*)$ *is any weak solution of (2.31), then the pair* (x^*, u^*) *is an optimal solution of the problem (2.30).*
(III: Gradient characterization of optimal solutions) If $(\widehat{x}, \widehat{u})$ *and* $(\widetilde{x}, \widetilde{u})$ *are any two optimal solutions of (2.30), then the following three properties hold:*

(a) for almost all $t \in [0, T]$,

$$\begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix} \begin{pmatrix} \widehat{x}(t) - \widetilde{x}(t) \\ \widehat{u}(t) - \widetilde{u}(t) \end{pmatrix} = 0,$$

(b) $S\widehat{x}(T) = S\widetilde{x}(T)$, *and*

(c) $c^T(\widehat{x}(T) - \widetilde{x}(T)) + \displaystyle\int_0^T \begin{pmatrix} p(t) \\ q(t) \end{pmatrix}^T \begin{pmatrix} \widehat{x}(t) - \widetilde{x}(t) \\ \widehat{u}(t) - \widetilde{u}(t) \end{pmatrix} dt = 0.$

Thus given any optimal solution $(\widehat{x}, \widehat{u})$ *of (2.30), a feasible tuple* $(\widetilde{x}, \widetilde{u})$ *of (2.30) is optimal if and only if conditions (a), (b), and (c) hold.*
(IV: Necessity of Pontryagin) Let $(x^*, \lambda^*, u^*, \mu^*)$ *be the tuple obtained from part (I). A feasible tuple* $(\widetilde{x}, \widetilde{u})$ *of (2.30) is optimal if and only if* $(\widetilde{x}, \lambda^*, \widetilde{u}, \mu^*)$ *is a weak solution of (2.31).*
(V: Uniqueness under positive definiteness) If R *is positive definite, then for any two optimal solutions* $(\widehat{x}, \widehat{u})$ *and* $(\widetilde{x}, \widetilde{u})$ *of (2.30),* $\widehat{x} = \widetilde{x}$ *everywhere on* $[0, T]$ *and* $\widehat{u} = \widetilde{u}$ *almost everywhere on* $[0, T]$. *In this case (2.30) has a unique optimal solution*

$(\widehat{x}, \widehat{u})$ *such that* $\widehat{x}$ *is continuously differentiable and* $\widehat{u}$ *is continuous on* $[0, T]$, *and for any optimal* $\widehat{\lambda}$, $\widehat{u}(t) \in \underset{u \in U(\widehat{x}(t))}{\operatorname{argmin}} \; H(\widehat{x}(t), u, \widehat{\lambda}(t))$ *for all* $t \in [0, T]$. □

It should be noted that in part (V) of the above, the uniqueness requires the positive definiteness of the principal block R of Ξ, and not of this entire matrix Ξ, which nonetheless is assumed to be positive semidefinite. The reason is twofold: one: via the ODE, the state variable x can be solved uniquely in terms of the control variable u; two: part (I) implies that the difference $\widehat{u} - \widetilde{u}$ is equal to $R^{-1} Q\widehat{x} - \widetilde{x}$ for any two solution pairs $(\widetilde{x}, \widetilde{u})$ and $(\widetilde{x}, \widetilde{u})$. Combining these two consequences yields the uniqueness in part (V) under the positive definiteness assumption of R. This is the common case treated in the optimal control literature.

2.14.1 The Time-Discretized Subproblems

A general time-stepping method for solving the LQ problem (2.30) proceeds similarly to (2.16), albeith with suitable modifications as described below. Let $h > 0$ be an arbitrary step size such that $N_h \triangleq \dfrac{T}{h}$ is a positive integer. We partition the interval $[0, \mathscr{T}]$ into N_h subintervals each of equal length h:

$$0 \triangleq t_{h,0} < t_{h,1} < t_{h,2} < \cdots < t_{h,N_h - 1} < t_{h,N_h} \triangleq \mathscr{T}.$$

Thus $t_{h,i+1} = t_{h,i} + h$ for all $i = 0, 1, \cdots, N_h - 1$. We step forward in time and calculate the state iterates $\mathbf{x}^h \triangleq \left\{ x^{h,i} \right\}_{i=0}^{N_h}$ and control iterates $\mathbf{u}^h \triangleq \left\{ u^{h,i} \right\}_{i=1}^{N_h}$ by solving N_h finite-dimensional convex quadratic subprograms, provided that the latter are feasible. From these discrete-time iterates, continuous-time numerical trajectories are constructed by piecewise linear and piecewise constant interpolation, respectively. Specifically, define the functions $\widehat{x}^h$ and $\widehat{u}^h$ on the interval $[0, T]$: for all $i = 0, \cdots, N_h - 1$:

$$\begin{aligned} \widehat{x}^h(t) &\triangleq x^{h,i} + \frac{t - t_{h,i}}{h} \left(x^{h,i+1} - x^{h,i} \right), \; \forall\, t \in [\, t_{h,i}, t_{h,i+1} \,] \\ \widehat{u}^h(t) &\triangleq u^{h,i+1}, \qquad\qquad\qquad\qquad\quad \forall\, t \in (\, t_{h,i}, t_{h,i+1} \,]. \end{aligned} \tag{2.32}$$

The convergence of these trajectories as the step size $h \downarrow 0$ to an optimal solution of the LQ control problem (2.30) is a main concern in the subsequent analysis. Neverthless, since the DAVI (2.31) is essentially a boundary-value problem, care is needed to define the discretized subproblems so that the iterates $(\mathbf{x}^h, \mathbf{u}^h)$ are well defined. Furthermore, since the original problem (2.30) is an infinite-dimensional quadratic program, one would like the subproblems to be finite-dimensional QPs. The multipliers of the constraints in the latter QPs will then define the adjoint trajectory $\widehat{\lambda}^h(t)$ and the algebraic trajectory $\widehat{\mu}^h(t)$; see below.

In an attempt to provide a common framework for the analysis of the backward Euler discretization and the model predictive control scheme [48, 50, 92] that has a long tradition in optimal control problems, a unified discretization method was proposed in [59] that employed several families of matrices $\{B(h)\}$, $\{A(h)\}$, $\{E(h)\}$ and $\{\widehat{E}(h)\}$ parametrized by the step $h > 0$; these matrices satisfy the following limit properties:

$$\lim_{h\downarrow 0} \frac{A(h) - I}{h} = A; \quad \lim_{h\downarrow 0} \frac{B(h)}{h} = B; \quad \text{and} \quad \lim_{h\downarrow 0} \frac{E(h)}{h} = \lim_{h\downarrow 0} \frac{\widehat{E}(h)}{h} = I.$$

Furthermore, in order to ensure that each discretized subproblem is solvable, a two-step procedure was introduced: the first step computes the least residual of feasibility of such a subproblem by solving the linear program:

$$\begin{array}{lll} \rho_h(\xi) \triangleq & \underset{\rho;\ \{x^{h,i},u^{h,i}\}_{i=1}^{N_h}}{\text{minimum}} & \rho \\ & \text{subject to} & x^{h,0} = \xi, \quad \rho \geq 0 \\ \text{and} & \text{for } i = 0, 1, \cdots, N_h : & \end{array} \tag{2.33}$$

$$\left\{ \begin{array}{l} x^{h,i+1} = \left[\theta\, E(h)\, r^{h,i} + (1-\theta)\, \widehat{E}(h)\, r^{h,i+1} \right] + A(h)x^{h,i} + B(h)u^{h,i+1} \\ Cx^{h,i+1} + Du^{h,i+1} + f + \rho\,\mathbf{1} \geq 0 \end{array} \right\},$$

where $r^{h,i} \triangleq r(t_{h,i})$ for all $i = 0, 1, \cdots, N_h$, and $\mathbf{1}$ is the vector of all ones. The above linear program must have a finite optimal solution and the optimal objective value $\rho_h(\xi)$ satisfies $\lim\limits_{h\downarrow 0} \rho_h(\xi) = 0$; this limit ensures that the pair of limit trajectories $\widehat{x}^h(t)$ and $\widehat{u}^h(t)$ constructed from the discrete-time iterates will be feasible to (2.30) for almost all times $t \in [0, \mathscr{T}]$. The scalar $\theta \in [0, 1]$ adds flexibility to the above formulation and leads to different specific schemes with proper choices. For instance, when $\theta = 0$, by letting $\widehat{E}(h) \triangleq hA(h)$, $A(h) \triangleq (I - hA)^{-1}$, and $B(h) \triangleq hA(h)B$, we obtain a standard backward Euler discretization of the ODE constraint in (2.30). When $\theta = 1$, by letting $E(h) \triangleq \int_0^h e^{As} ds$, $A(h) \triangleq e^{Ah}$, and $B(h) \triangleq \int_0^h e^{As} ds B$, we obtain the MPC approximation of this ODE.

Employing the minimum residual $\rho_h(\xi)$, the relaxed, unified time-stepping method solves the following (feasible) convex quadratic program at time $t_{h,i+1}$:

$$(\widehat{\mathrm{QP}}^h): \underset{\{x^{h,i},u^{h,i}\}_{i=1}^{N_h+1}}{\text{minimize}} \quad V_h(\mathbf{x}^h,\mathbf{u}^h) \triangleq (x^{h,N_h+1})^T\left(c+\frac{1}{2}Sx^{h,N_h+1}\right)$$
$$+\frac{h}{2}\sum_{i=0}^{N_h}\left\{2\begin{pmatrix}\theta x^{h,i}+(1-\theta)x^{h,i+1}\\ u^{h,i+1}\end{pmatrix}^T\begin{pmatrix}p^{h,i+1}\\ q^{h,i+1}\end{pmatrix}\right.$$
$$\left.+\begin{pmatrix}\theta x^{h,i}+(1-\theta)x^{h,i+1}\\ u^{h,i+1}\end{pmatrix}^T\begin{bmatrix}P & Q\\ Q^T & R\end{bmatrix}\begin{pmatrix}\theta x^{h,i}+(1-\theta)x^{h,i+1}\\ u^{h,i+1}\end{pmatrix}\right\}$$

subject to $x^{h,0}=\xi$, and for $i=0,1,\cdots,N_h$:

$$\left\{\begin{array}{l} x^{h,i+1} = \left[\theta\, E(h)\, r^{h,i}+(1-\theta)\,\widehat{E}(h)\, r^{h,i+1}\right]+A(h)x^{h,i}+B(h)u^{h,i+1}\\ f+Cx^{h,i+1}+Du^{h,i+1}+\underbrace{\rho_h(\xi)\,\mathbf{1}}_{\text{relaxed feasibility}}\ \geq 0 \end{array}\right\}.$$

The primal condition on the pair (R,D) and the use of the least residual $\rho_h(\xi)$ are sufficient to ensure that an optimal solution to the above QP exists. Notice that unlike the case in Lecture III of an initial-value DVI (2.16) or the LCS (2.19), the $\widehat{\mathrm{QP}}^h$ is coupled in the discrete-time iterates and does not decompose into individual subproblems according to the time steps. Thus this quadratic subprogram is potentially of very large size and its practical solution requires care.

Letting $\{\lambda^{h,i}\}_{i=0}^{N_h}$ be the multipliers of the discrete-time state constraints and $\{\mu^{h,i+1}\}_{i=0}^{N_h}$ be the multipliers of the algebraic state-control inequalities, we define the λ-trajectory similarly to the x-trajectory; namely, for $i=0,\cdots,N_h$,

$$\widehat{\lambda}^h(t) \triangleq \lambda^{h,i}+\frac{t-t_{h,i}}{h}(\lambda^{h,i+1}-\lambda^{h,i}), \quad \forall t\in[t_{h,i},t_{h,i+1}],$$

with $\lambda^{h,N_h+1}\triangleq c+Sx^{h,N_h+1}$, and the μ-trajectory similarly to the u-trajectory; namely, for $i=0,\cdots,N_h$,

$$\widehat{\mu}^h(t) \triangleq \frac{\mu^{h,i+1}}{h}, \quad \forall t\in(t_{h,i},t_{h,i+1}].$$

The convergence of the numerical trajectories is formally stated in the theorem below.

Theorem 5 Assume the setting state above. Let $\widehat{x}^h(t)$ and $\widehat{u}^h(t)$ be as defined by (2.32) and $\widehat{\lambda}^h(t)$ and $\widehat{\mu}^h(t)$ as above. The following four statements hold.

(a) There exists a sequence of step sizes $\{h_\nu\} \downarrow 0$ such that the two limits exist: $\left(\widehat{x}^{h_\nu}, \widehat{\lambda}^{h_\nu}\right) \to \left(\widehat{x}, \widehat{\lambda}\right)$ uniformly on $[0, T]$ and $\left(\widehat{u}^{h_\nu}, \widehat{\mu}^{h_\nu}\right) \to (\widehat{u}, \widehat{\mu})$ weakly in $L^2(0, T)$; moreover, $\widehat{x}$ and $\widehat{\lambda}$ are Lipschitz continuous.
(b) The sequences $\left\{\begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix} \begin{pmatrix} \widehat{x}^h \\ \widehat{u}^h \end{pmatrix}\right\}$ and $\left\{D^T \widehat{\mu}^h\right\}$ converge to $\begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix} \begin{pmatrix} \widehat{x} \\ \widehat{u} \end{pmatrix}$ and $D^T \widehat{\mu}$ uniformly on $[0, T]$, respectively.
(c) Any limit tuple $(\widehat{x}, \widehat{u}, \widehat{\lambda}, \widehat{\mu})$ from (a) is a weak solution of (2.31); thus $(\widehat{x}, \widehat{u})$ is an optimal solution of (2.30).
(d) Part (I) of Theorem 4 holds. □

The proof of the above theorem hinges on establishing the bounds in Proposition 3 for the differential iterates $(x^{h,i}, \lambda^{h,i})$ and the algebraic iterates $(u^{h,i}, \mu^{h,i})$. This is highly technical, partly due the relaxed assumptions we have made—e.g., the positive semidefiniteness, instead of positive definiteness of the matrix $\Xi \triangleq \begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix}$—and partly due to the boundary-value nature of the DAVI. Details can be found in [59].

When R is positive definite, we can establish the uniform convergence of the u-variable by redefining the discrete-time trajectory $\widehat{u}^h$ using piecewise linear interpolation instead of the piecewise constant interpolation in the semidefinite case. First notice that $u^{h,0}$ is not included in the $(\widehat{\text{QP}}^h)$. By letting $u^{h,0}$ be the unique solution of the QP at the initial time $t = 0$,

$$\underset{u \in U(\xi)}{\text{minimize}} \left[q^{h,0} + h^{-1} B(h)^T \lambda^{h,0} + Q^T \xi \right]^T u + \tfrac{1}{2} u^T R u,$$

we redefine

$$\widehat{u}^h(t) \triangleq u^{h,i} + \frac{t - t_{h,i}}{h} \left(u^{h,i+1} - u^{h,i} \right) \quad \forall\, t \in [\, t_{h,i}, t_{h,i+1} \,]. \tag{2.34}$$

It can be shown that the sequences of state and control trajectories $\{\widehat{x}^h\}$ and $\{\widehat{u}^h\}$ converge, respectively, to the unique optimal solution $(\widehat{x}, \widehat{u})$ of the problem (2.30) with $\widehat{x}$ being continuously differentiable and $\widehat{u}$ Lipschitz continuous on $[0, T]$. We omit the details.

2.15 Lecture IV: Summary

In this lecture, we have

- introduced the linear-quadratic optimal control problem with mixed state-control constraints
- described a time-stepping method for solving the problem that unifies time stepping and model predictive control, and
- presented a convergence result under a set of assumptions.

This development is the basis for extension to a multi-agent non-cooperative game where each player solves such an LQ optimal control problem parameterized by the rivals' time-dependent strategies.

2.16 Lecture V: Open-Loop Differential Nash Games

Non-cooperative game theory provides a mathematical framework for conflict resolution among multiple selfish players each seeking to optimize his/her individual objective function that is impacted by the rivals' decisions and subject to constraints that may be private, coupled, or shared. A solution concept of such a game was introduced by John Nash and has been known as a *Nash equilibrium*. There have been several recent surveys on the static Nash equilibrium problem defined in terms of multiple finite-dimensional optimization problems, one for each player; see e.g. [44, 46] where the reader can also find an extensive literature including historical account, theory, algorithms, and selected applications. This lecture pertains to the Nash equilibrium problem (NEP) defined in continuous time wherein each player's optimization problem is the single-agent LQ optimal control problem discussed in the last lecture and extended herein to the case of multiple agents. The discussion below is drawn from the recently published paper [110].

Specifically, we consider a linear-quadratic (LQ) N-player noncooperative game on the time interval $[0, \mathscr{T}]$, where $\mathscr{T} < \infty$ is the finite horizon and N is the number of players of the game. Each player $i \in \{1, \cdots, N\}$ chooses an absolutely continuous state function $x_i : [0, \mathscr{T}] \to \mathbb{R}^{n_i}$ and a bounded measurable (thus integrable) control function $u_i : [0, \mathscr{T}] \to \mathbb{R}^{m_i}$ for some positive integers n_i and m_i via the solution of a LQ optimal control problem. These state and control variables are constrained by a player-specific ODE and a linear inequality system.

Notation $x_{-i} \triangleq \left(x_j\right)_{i \neq j=1}^N$; $u_{-i} \triangleq \left(u_j\right)_{i \neq j=1}^N$ denote the rivals' pairs of state and control variables, respectively.

Anticipating the pair (x_{-i}, u_{-i}) of rivals' trajectory and treating only private constraints, player i solves

$$\underset{x_i, u_i}{\text{minimize}}\ \theta_i(x_i, x_{-i}, u_i, u_{-i}) \triangleq x_i(\mathscr{T})^T \left(c_i + \sum_{i'=1}^N W_{ii'}\, x_{i'}(\mathscr{T}) \right) +$$

$$\int_0^{\mathscr{T}} \begin{pmatrix} x_i(t) \\ u_i(t) \end{pmatrix}^T \left(\begin{pmatrix} p_i(t) \\ q_i(t) \end{pmatrix} + \sum_{i'=1}^N \begin{bmatrix} P_{ii'} & Q_{ii'} \\ R_{ii'} & S_{ii'} \end{bmatrix} \begin{pmatrix} x_{i'}(t) \\ u_{i'}(t) \end{pmatrix} \right) dt \tag{2.35}$$

subject to $x_i(0) = \xi_i$ and for almost all $t \in [\,0, \mathscr{T}\,]$:

$$\dot{x}_i(t) = r_i(t) + A_i x_i(t) + B_i u_i(t)$$

and $\quad f_i + C_i x_i(t) + D_i u_i(t) \geq 0,$

where each W_{ii} and $\begin{bmatrix} P_{ii} & Q_{ii} \\ R_{ii} & S_{ii} \end{bmatrix}$ are symmetric positive semidefinite, resulting in (2.35) being a convex LQ optimal control problem in player i's variables. The other notations are extensions of (2.30) that is the case of a single player. Moreover, the assumptions previously introduced for (2.30) extend to each player's problem in the game.

An aggregated pair of trajectories $(\mathbf{x}^*, \mathbf{u}^*)$, where $\mathbf{x}^* \triangleq \left(x_i^*\right)_{i=1}^N$ and $\mathbf{u}^* \triangleq \left(u_i^*\right)_{i=1}^N$, is a *Nash equilibrium* (NE) of the above game if for each $i = 1, \cdots, N$,

$$\begin{array}{rl} (x_i^*, u_i^*) \in \underset{(x_i,u_i)}{\operatorname{argmin}} & \theta_i(x_i, x_{-i}^*, u_i, u_{-i}^*) \\ & \text{subject to } (x_i, u_i) \text{ feasible to (2.35).} \end{array}$$

Toward the analysis of this game, we distinguish two cases:

- $\Xi_{ii'} \triangleq \begin{bmatrix} P_{ii'} & Q_{ii'} \\ R_{ii'} & S_{ii'} \end{bmatrix} = \begin{bmatrix} P_{i'i} & Q_{i'i} \\ R_{i'i} & S_{i'i} \end{bmatrix}^T \triangleq \Xi_{i'i}^T$ and $W_{ii'} = W_{i'i}^T$ for all $i \neq i'$, reflecting the *symmetric* impact of the strategy of player i' on player i's objective function and vice versa.
- $(W_{ii'}, \Xi_{ii'}) \neq \left(W_{i'i}^T, \Xi_{i'i}^T\right)$ for some $i \neq i'$, reflecting the *asymmetric* impact of the strategy of player i' on player i's objective function and vice versa.

The treatment of these two cases is different. In the symmetric case, we show that a NE of the game can be obtained as a stationary solution of a single optimal control problem with an objective function that is the sum of the players' objectives and whose constraints are the Cartesian product of the individual players' constraints. In the asymmetric case, we provide a best-response algorithm to constructively establish the existence of a NE to the game; such an algorithm iteratively solves single-player LQ optimal control problems by fixing the rivals' variables at their current iterates.

2.16.1 The Symmetric Case

Writing the symmetric assumption more succinctly, we assume that the matrices $\mathbf{W}$ and Ξ are symmetric positive semidefinite, where

$$\mathbf{W} \triangleq \left[[W_{ii'}]_{i,i'=1}^N + \operatorname{diag}(W_{ii})_{i=1}^N \right]$$

$$\Xi \triangleq \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix} \triangleq \begin{bmatrix} [P_{ii'}]_{i,i'=1}^N + \operatorname{diag}(P_{ii})_{i=1}^N & [Q_{ii'}]_{i,i'=1}^N + \operatorname{diag}(Q_{ii})_{i=1}^N \\ [R_{ii'}]_{i,i'=1}^N + \operatorname{diag}(R_{ii})_{i=1}^N & [S_{ii'}]_{i,i'=1}^N + \operatorname{diag}(S_{ii})_{i=1}^N \end{bmatrix}$$

The aggregated LQ optimal control problem in the variables $(\mathbf{x}, \mathbf{u})$ is then

$$\begin{aligned} &\underset{\mathbf{x},\mathbf{u}}{\text{minimize}}\ \mathbf{x}(\mathscr{T})^T \left[\mathbf{c} + \tfrac{1}{2}\, \mathbf{W}\mathbf{x}(\mathscr{T}) \right] + \\ &\int_0^{\mathscr{T}} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}^T \left(\begin{pmatrix} \mathbf{p}(t) \\ \mathbf{q}(t) \end{pmatrix} + \tfrac{1}{2} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \right) dt \end{aligned} \tag{2.36}$$

$$\begin{aligned} &\text{subject to } x_i(0) = \xi_i \text{ for all } i \in \{1, \cdots, N\} \text{ and almost all } t \in [0, \mathscr{T}] : \\ &\qquad\qquad \dot{x}_i(t) \;=\; r_i(t) + A_i x_i(t) + B_i u_i(t), \\ &\text{and} \qquad f_i + C_i x_i(t) + D_i u_i(t) \;\geq\; 0. \end{aligned}$$

Theorem 6 Under the above symmetry assumption and the conditions on each of the players' problems set forth in Lecture IV, the following statements hold between the N-person differential Nash game and the aggregated optimal control problem (2.36):

- Equivalence: A pair $(\mathbf{x}^*, \mathbf{u}^*)$ is a NE if and only if $(\mathbf{x}^*, \mathbf{u}^*)$ is an optimal solution of the aggregate optimal control problem.
- Existence: A NE exists such that $\mathbf{x}^*$ is absolutely continuous and $\mathbf{u}^*$ is square-integrable on $[\,0, \mathscr{T}\,]$.
- Uniqueness: If in addition $\mathbf{S}$ is positive definite, then $(\mathbf{x}^*, \mathbf{u}^*)$ is the unique NE such that $\mathbf{x}^*$ is continuously differentiable and $\mathbf{u}^*$ is Lipschitz continuous on $[\,0, \mathscr{T}\,]$.
- Computation: A NE can be obtained as the limit of a sequence of numerical trajectories obtained by discretizing the optimal control problem (2.36) as described in Lecture IV. □

It should be noted that while the symmetry assumption of the matrices $\mathbf{W}$ and $\boldsymbol{\Xi}$ are essential for the equivalence between the game and the single optimal control formulation, the positive semidefiniteness of these matrices makes the problem (2.36) a convex problem, albeit in continuous time, to which the time-discretization method is applicable for its numerical solution. Without the positive semidefiniteness condition, we should settle for a solution to the DAVI formulation of (2.36) that is only a stationary solution but not necessarily a globally optimal solution. In this case, the solution method of the last Lecture needs to be examined for applicability and its convergence requires an extended proof.

2.16.2 The Asymmetric Case

In addition to the assumptions for the individual players' problems, the asymmetric case requires a few more conditions that are motivated by the convergence analysis of the best-response scheme for the static NEP. These additional conditions are stated below:

$(\widehat{\mathbf{A}})$ For all $i = 1, \cdots, N$, the matrices Ξ_{ii} are positive definite with minimum eigenvalues $\sigma_i^{\Xi} > 0$; the matrices W_{ii} remain (symmetric) positive semidefinite.

(W) For all $i = 1, \cdots, N$, the matrices $W_{ii'} = 0$ for all $i' \neq i$; (to somewhat simplify the notation). [Otherwise, the matrix $\boldsymbol{\Gamma}$ needs to be modified.]

$(\widehat{\mathbf{D}})$ For all $i = 1, \cdots, N$, the following implication holds: $D_i u_i \geq 0 \Rightarrow u_i = 0$, implying the boundedness of the feasible controls

$$u_i \in U_i(x_i) \triangleq \left\{ u \in \mathbb{R}^{m_i} \mid f_i + C_i x_i + D_i u_i \geq 0 \right\}, \quad \text{for all } x.$$

Define the matrix $\boldsymbol{\Gamma} \triangleq [\, \Gamma_{ii'} \,]_{i,i'=1}^{N}$, where

$$\Gamma_{ii'} \triangleq \begin{cases} 0 & \text{if } i = i' \\ \dfrac{1}{\sqrt{\sigma_i^{\Xi}\,\sigma_{i'}^{\Xi}}} \, \| \, \Xi_{ii'} \, \| & \text{if } i \neq i', \end{cases}$$

A Key Postulate The spectral radius $\rho(\boldsymbol{\Gamma}) < 1$.

Dating back to the convergence analysis of fixed-point iterations for solving systems of linear equations [96] and extending to splitting methods for the linear complementarity problems [38], the spectral radius condition generalizes the well-known property of (strictly) diagonally dominance and has been key to the convergence of best-response algorithms for static games; see e.g. [84, 98]. The interesting fact is that this spectral radius condition remains key in the continuous-time game.

The following is a Jacobi-type iterative algorithm for solving the continuous-time non-cooperative game. A particular feature of the algorithm is that it is of the distributed type, meaning that each player can update his/her response simultaneously and independently of the other players; after such an update a synchronization occurs leading to a new iteration. A sequential Gauss-Seidel type algorithm can be stated; we omit the details.

A Continuous-Time Best-Response Algorithm Given a pair of state-control trajectories $(\mathbf{x}^{\nu}, \mathbf{u}^{\nu})$ at the beginning of iteration $\nu + 1$, where $\mathbf{x}^{\nu}$ is continuously differentiable and $\mathbf{u}^{\nu}$ is Lipschitz continuous, we compute the next pair of such trajectory $(\mathbf{x}^{\nu+1}, \mathbf{u}^{\nu+1})$ by solving N LQ optimal control problems (2.35), where for $i = 1, \cdots, N$, the i-th such LQ problem solves for the pair $(x_i^{\nu+1}, u_i^{\nu+1})$ from (2.35) by fixing (x_j, u_j) at (x_j^{ν}, u_j^{ν}) for all $j \neq i$. □

The above is a continuous-time distributed algorithm that requires solving LQ subproblems in parallel; in turn each such subproblem is solved by time discretization that leads to the solution of finite-dimensional quadratic programs. This is in contrast to first discretization that results in solving finite-dimensional subgames, which can in turn be solved by a distributed algorithm (in discrete time). The relative efficiency of these two approaches: first best-response (in continuous time) followed by discretization versus first discretization followed by best-response (in discrete tim) on applied problem has yet to be understood.

The convergence of the above algorithm is summarized in the following result whose proof can be found in [110].

Theorem 7 *In the above setting, the following two statements hold for the sequence* $\{(x_i^\nu, u_i^\nu)\}$ *generated by the Jacobi iterative algorithm.*

- *(Well-definedness) The sequence is well-defined with* x_i^ν *being continuously differentiable and* u_i^ν *Lipschitz continuous on* $[\,0,\,\mathscr{T}\,]$ *for all* ν.
- *(Contraction and strong convergence) The sequence contracts and converges strongly to a square-integrable, thus integrable, limit* $(x_i^\infty(t), u_i^\infty(t))_{i=1}^N$ *in the space* $L^2[\,0,\,\mathscr{T}\,]$ *that is the unique NE of the differential LQ game. Indeed, it holds that*

$$\mathbf{e}^\nu \,\leq\, \boldsymbol{\Gamma}\,\mathbf{e}^{\nu-1}, \qquad \forall\,\nu \,=\, 1, 2, \cdots,$$

where $\mathbf{e}^\nu \,\triangleq\, \left(e_i^\nu\right)_{i=1}^N$, *with* $e_i^\nu \,\triangleq\, \sqrt{\sigma_i^\Xi}\,\sqrt{\int_0^{\mathscr{T}} \left\| \begin{pmatrix} x_i^\nu(t) - x_i^{\nu+1}(t) \\ u_i^\nu(t) - u_i^{\nu+1}(t) \end{pmatrix} \right\|^2 dt;}$

moreover, strong convergence means that

$$\lim_{\nu\to\infty} \int_0^{\mathscr{T}} \left\| \begin{pmatrix} x_i^\nu(t) - x_i^\infty(t) \\ u_i^\nu(t) - u_i^\infty(t) \end{pmatrix} \right\| dt \,=\, 0.$$

2.16.3 Two Illustrative Examples

Illustrating the abstract framing of the symmetric and asymmetric problems in the previous two sections, we present two concrete examples of how such problems may arise in applied game theory. The first example model is an adaptation of the well-known Nash-Cournot equilibrium problem while the second is a conjectured supply function equilibrium problem. Although these types of problems are typically studied in a static setting, the differential formulations presented herein represent natural problem extensions for which solution existence can be established from the previous results. In the Nash-Cournot version of this problem, each player believes that their output affects the commodity price which is represented as a function of total output. For a two-player, two-node problem with a linear pricing function and quadratic cost functions, let player 1's optimal control problem be

$$\underset{\mathbf{g}_1,\,\mathbf{s}_1,\,\mathbf{r}_1}{\text{minimize}}$$

$$\int_0^{\mathscr{T}} \begin{pmatrix} g_{11}(t) \\ g_{12}(t) \\ s_{11}(t) \\ s_{12}(t) \\ r_{11}(t) \\ r_{12}(t) \end{pmatrix}^T \left(\begin{pmatrix} a_{11} - w(t) \\ a_{12} \\ -P_1^0 + w(t) \\ -P_2^0 \\ 0 \\ 0 \end{pmatrix} + \begin{bmatrix} b_{11} & & & & & \\ & b_{12} & & & & \\ & & \dfrac{P_1^0}{Q_1^0} & & & \\ & & & \dfrac{P_2^0}{Q_2^0} & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} \begin{pmatrix} g_{11}(t) \\ g_{12}(t) \\ s_{11}(t) \\ s_{12}(t) \\ r_{11}(t) \\ r_{12}(t) \end{pmatrix} \right.$$

$$\left. + \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \dfrac{P_1^0}{Q_1^0} & & & \\ & & & \dfrac{P_2^0}{Q_2^0} & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} \begin{pmatrix} g_{21}(t) \\ g_{22}(t) \\ s_{21}(t) \\ s_{22}(t) \\ r_{21}(t) \\ r_{22}(t) \end{pmatrix} \right) dt$$

$$= \int_0^{\mathscr{T}} \left\{ \sum_{j=1}^{2} \left(a_{1j} g_{1j}(t) + b_{1j} g_{1j}(t)^2 - \left[P_j^0 - \frac{P_j^0}{Q_j^0}(s_{1j}(t) + s_{2j}(t)) \right] s_{1j}(t) \right) \right.$$

$$\left. + (s_{11}(t) - g_{11}(t)) w(t) \right\} dt$$

subject to $g_{11}(0) = g_{11}^0$, $g_{12}(0) = g_{12}^0$, and for almost all $t \in [0, \mathscr{T}]$:

$$\left. \begin{array}{ll} \dot{g}_{11}(t) = r_{11}(t) & \\ \dot{g}_{12}(t) = r_{12}(t) & \\ -\underline{r}_{1j} + r_{1j}(t) \geq 0 & \text{for } j = 1,2 \\ \overline{r}_{1j} - r_{1j}(t) \geq 0 & \text{for } j = 1,2 \end{array} \right\} \Rightarrow \begin{array}{l} \underline{r}_{11} \leq \dot{g}_{11}(t) \leq \overline{r}_{11} \\ \underline{r}_{12} \leq \dot{g}_{12}(t) \leq \overline{r}_{12} \end{array}$$

and $$- g_{11}(t) - g_{12}(t) + s_{11}(t) + s_{12}(t) \geq 0,$$

where the state variables are $\{g_{ij}(t)\}_{i,j=1}^{2}$, representing player i's production at node j at time t, and the control variables are $\{s_{ij}(t), r_{ij}(t)\}_{i,j=1}^{2}$, representing player i's sales and ramp rate (instantaneous change in production) at node j at time t, respectively. The term $a_{ij}g_{ij}(t) + b_{ij}g_{ij}(t)^2$ is the quadratic production cost function, $P_j^0 - \dfrac{P_j^0}{Q_j^0}\left(\displaystyle\sum_{i=1}^{2} s_{ij}(t)\right)$ is the linear nodal pricing equation at time t with intercept P_j^0 and slope $\dfrac{P_j^0}{Q_j^0}$ where P_j^0 and Q_j^0 are positive, and $(s_{ij}(t) - g_{ij}(t))w(t)$ is the transportation cost with $w(t)$ being the marginal directional shipment cost at time t. The first group of constraints describes generation ramp rates, namely that the rate of generation change for player i at node j is bounded by $\underline{r}_{ij}$ and $\overline{r}_{ij}$. The last two constraints equate total generation with total sales.

Player 2's objective function is easily shown to be identical to that given above except with 1 and 2 interchanged in player index i. Therefore, it is apparent that $\boldsymbol{\Xi} \triangleq \begin{bmatrix} \Xi_{11} & \Xi_{12} \\ \Xi_{21} & \Xi_{22} \end{bmatrix}$ is the symmetric matrix

$$\begin{bmatrix}
2b_{11} &&&&&&&&&&& \\
& 2b_{12} &&&&&&&&&& \\
&& 2\dfrac{P_1^0}{Q_1^0} &&&&&& \dfrac{P_1^0}{Q_1^0} &&& \\
&&& 2\dfrac{P_2^0}{Q_2^0} &&&&&& \dfrac{P_2^0}{Q_2^0} && \\
&&&& 0 &&&&&&& \\
&&&&& 0 &&&&&& \\
&&&&&& 2b_{21} &&&&& \\
&&&&&&& 2b_{22} &&&& \\
&& \dfrac{P_1^0}{Q_1^0} &&&&&& 2\dfrac{P_1^0}{Q_1^0} &&& \\
&&& \dfrac{P_2^0}{Q_2^0} &&&&&& 2\dfrac{P_2^0}{Q_2^0} && \\
&&&&&&&&&& 0 & \\
&&&&&&&&&&& 0
\end{bmatrix}.$$

It is not difficult to verify that the matrix $\boldsymbol{\Xi}$ is positive semidefinite.

We next turn our attention to a conjectured supply function (CSF) problem [40, 69, 70] to demonstrate the existence of games that take an asymmetric form. In the Nash-Cournot problem, symmetry arises from the assumptions that each player uses the same commodity pricing function and that no player anticipates competitor production/sales changes with respect to price. In a conjectured supply function equilibrium problem, players instead use a function to predict how total competitor production will change based on price. For this example, we will simplify the model to include only one node so that generation and sales quantities are equivalent and transmission is not needed.

For player i, let the function $\sigma_i(G_{-i}(t), p_i(t), t)$ represent the relationship between price and total competitor production in time t. For our linear-quadratic problem, we will define

$$\sigma_i(G_{-i}(t), p_i(t), t) \triangleq G_{-i}(t) + \beta_i(G_{-i}(t), p_i^*(t), t)(p_i(t) - p_i^*(t)),$$

where $G_{-i}(t)$ is the total amount of competitor generation expected at the specified equilibrium price $p_i^*(t)$ at time t. Notice that players may expect different equilibrium price trajectories here; this setting generalizes the case in which players use the same equilibrium price trajectory where $p_i^*(t) = p^*(t)$ for $i = 1, 2$. It follows that, depending on the specification of $\beta_i(G_{-i}(t), p_i^*(t), t)$, the conjectured total production from other players will rise or fall if the realized price $p_i(t)$ does not equal the equilibrium price $p_i^*(t)$. Upon substitution into the production-pricing relationship

$$g_i(t) + \sigma_i(G_{-i}(t), p_i(t), t) = Q^0 - \frac{Q^0}{P^0} p_i(t),$$

invertibility of $\dfrac{Q^0}{P^0} + \beta_i(G_{-i}(t), p_i^*(t), t)$ provides an explicit equation for player i's conjectured price $p_i(t)$. This invertibility will hold in realistic market settings since $\beta_i(G_{-i}(t), p_i^*(t), t)$ should be nonnegative so total competitor production levels are believed to change in the same direction as price differences (i.e., higher prices than expected at equilibrium should not decrease conjectured production). In the special case assumed here where $\beta_i(G_{-i}(t), p_i^*(t), t) \triangleq B_{-i}$ for some positive constant B_{-i}, we obtain

$$p_i(t) = \frac{Q^0 - G_i(t) + B_{-i} p_i^*(t)}{\dfrac{Q^0}{P^0} + B_{-i}}.$$

Using this conjectured price, we can formulate player 1's optimal control problem as a cost minimization problem in which the conjectured supply function price is used for determining revenue and costs include a quadratic production cost and a quadratic ramp rate cost:

$$\begin{aligned}
&\operatorname*{minimize}_{g_1,\, r_1} \\
&\int_0^{\mathscr{T}} \begin{pmatrix} g_1(t) \\ r_1(t) \end{pmatrix}^T \left(\begin{pmatrix} a_{11} - \dfrac{Q^0 + B_{-1} p_1^*(t)}{\dfrac{Q^0}{P^0} + B_{-1}} \\ 0 \end{pmatrix} \right. \\
&\qquad \left. + \begin{bmatrix} b_{11} + \dfrac{1}{\dfrac{Q^0}{P^0} + B_{-1}} & 0 \\ 0 & a_{12} \end{bmatrix} \begin{pmatrix} g_1(t) \\ r_1(t) \end{pmatrix} + \begin{bmatrix} \dfrac{1}{\dfrac{Q^0}{P^0} + B_{-1}} & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} g_2(t) \\ r_2(t) \end{pmatrix} \right) dt
\end{aligned}$$

subject to $g_1(0) = g_1^0$ and for almost all $t \in [\,0, \mathscr{T}\,]$:

$$\left. \begin{aligned} \dot{g}_1(t) &= r_1(t) \\ -\underline{r}_1 + r_1(t) &\geq 0 \\ \overline{r}_1 - r_1(t) &\geq 0 \end{aligned} \right\} \Rightarrow \underline{r}_1 \leq \dot{g}_1(t) \leq \overline{r}_1$$

Similarly, player 2's optimal control problem just interchanges 1 and 2 for the player index. If the player supply conjectures are not identical (i.e., $B_{-1} \neq B_{-2}$),

$$\varXi_{12} = \begin{bmatrix} \dfrac{1}{\dfrac{Q^0}{P^0} + B_{-1}} & 0 \\ 0 & 0 \end{bmatrix} \neq \begin{bmatrix} \dfrac{1}{\dfrac{Q^0}{P^0} + B_{-2}} & 0 \\ 0 & 0 \end{bmatrix} = \varXi_{21}^T.$$

It follows that a conjectured supply function game in which players have different conjectures is not a symmetric game.

To prove $\rho(\boldsymbol{\Gamma}) < 1$, we can use the fact that $\rho(\boldsymbol{\Gamma}) \leq \|\boldsymbol{\Gamma}^k\|^{\frac{1}{k}}$ for all natural numbers k. With $k = 1$ and employing the Euclidean norm, $\|\boldsymbol{\Gamma}\|$ is the largest eigenvalue of $(\boldsymbol{\Gamma}^T \boldsymbol{\Gamma})^{\frac{1}{2}}$, which is equal to

$$\left(\begin{bmatrix} 0 & \dfrac{1}{\sqrt{\sigma_2^{\varXi} \sigma_1^{\varXi}}} \|\varXi_{21}\| \\ \dfrac{1}{\sqrt{\sigma_1^{\varXi} \sigma_2^{\varXi}}} \|\varXi_{12}\| & 0 \end{bmatrix} \begin{bmatrix} 0 & \dfrac{1}{\sqrt{\sigma_1^{\varXi} \sigma_2^{\varXi}}} \|\varXi_{12}\| \\ \dfrac{1}{\sqrt{\sigma_2^{\varXi} \sigma_1^{\varXi}}} \|\varXi_{21}\| & 0 \end{bmatrix} \right)^{\frac{1}{2}}$$

$$= \frac{1}{\sqrt{\sigma_1^{\varXi} \sigma_2^{\varXi}}} \begin{bmatrix} \|\varXi_{12}\| & 0 \\ 0 & \|\varXi_{21}\| \end{bmatrix}$$

where σ_i^{Ξ} is the minimum eigenvalue of Ξ_{ii}. For this problem,

$$\sigma_1^{\Xi} \triangleq \min\left(b_{11} + \frac{1}{\dfrac{Q_0}{P_0} + B_{-1}}, \, a_{12} \right) \quad \text{and} \quad \sigma_2^{\Xi} \triangleq \min\left(b_{21} + \frac{1}{\dfrac{Q_0}{P_0} + B_{-2}}, \, a_{22} \right).$$

Hence, if

$$\begin{array}{ccc}
\| \, \Xi_{12} \, \| & & \\
\| & & \\
\dfrac{1}{\dfrac{Q^0}{P^0} + B_{-1}} & < & \sqrt{\min\left(b_{11} + \dfrac{1}{\dfrac{Q_0}{P_0} + B_{-1}}, \, a_{12} \right) \min\left(b_{21} + \dfrac{1}{\dfrac{Q_0}{P_0} + B_{-2}}, \, a_{22} \right)} \\
 & & \| \\
 & & \sqrt{\sigma_1^{\Xi} \sigma_2^{\Xi}} \\
 & & \| \\
\dfrac{1}{\dfrac{Q^0}{P^0} + B_{-2}} & < & \sqrt{\min\left(b_{11} + \dfrac{1}{\dfrac{Q_0}{P_0} + B_{-1}}, \, a_{12} \right) \min\left(b_{21} + \dfrac{1}{\dfrac{Q_0}{P_0} + B_{-2}}, \, a_{22} \right)} \\
\| & & \\
\| \, \Xi_{21} \, \| , & &
\end{array}$$

then $\rho(\boldsymbol{\Gamma}) < 1$. The above condition can clearly be satisfied for a wide variety of parameter values. We have thus proven that Theorem 7 holds for the above CSF problem specification and the presented Jacobi iterative algorithm will converge to the unique differential Nash equilibrium.

2.17 Lecture V: Summary

In this lecture, we have

- presented an open-loop differential LQ Nash game,
- shown the equivalence in the symmetric case of the game with a single concatenated linear-quadratic optimal control problem,

- discussed, in the asymmetric case, a Jacobi-type iterative solution scheme and presented a converge result under certain conditions to a unique differential Nash equilibrium, and
- illustrated the results using two simple instances of a Nash production game.

2.18 Closing Remarks

We close these lectures with the following remarks:

- Based on a differential variational framework, these lectures lay down the foundation for distributed, competitive multi-agent optimal decision making problems in continuous time.
- The first four lectures prepare the background for the fifth lecture which extends the other four lectures of this summer school to a continuous-time setting.
- In general, many real-life systems are dynamic in nature and subject to unilateral constraints and variational principles.
- The dynamics has to be recognized in the modeling and solution of the systems.
- The DVI provides a very powerful framework for this purpose, in particular, for the study of non-cooperative games in continuous times.
- Some extensive results are available, but there remain many questions and issues of the DVI to be studied.

Acknowledgements These lectures are based on joint work with many of the author's collaborators, starting from the basic paper [100] joint with David Stewart, extending to collaborative work with Kanat Camlibel, Jinglai Shen, Lanshan Han, and Dan Schiro. We thank all these individuals for their contributions to this area of research. The author is indebted to Professor Fabio Schoen for having orchestrated this Summer School and for his leadership within the Fondazione Centro Internazionale Matematico Estivo that has made the School a success. The Staff of CIME has made the daily operations of the School very smooth and the stay of the School participants very enjoyable. Last but not least, the author is most grateful to the Co-Director of the School, Francisco Facchinei, for his long-lasting collaboration, valuable friendship, and enthusiastic support of the School.

This work is based on research partially supported by the National Science Foundation under grant CMMI-0969600.

References

1. V. Acary, Higher order event capturing time-stepping schemes for nonsmooth multibody systems with unilateral constraints and impacts. Appl. Numer. Math. **62**, 1259–1275 (2012)
2. V. Acary, Analysis, simulation and control of nonsmooth dynamical systems, Habilitation thesis, L'INRIA Grenoble Rhâne-Alpes et Laboratoire Jean Kuntzmann, University of Grenoble, July 2015
3. V. Acary, O. Bonnefon, B. Brogliato, *Nonsmooth Modeling and Simulation for Switched Circuits*. Lecture Notes in Electrical Engineering, vol. 69 (Springer, Berlin, 2011)

4. V. Acary, H. de Jong, B. Brogliato, Numerical simulation of piecewise-linear models of gene regulatory networks using complementarity systems. Physica D **269**, 103–119 (2014)
5. F. Alvarez, On the minimizing property of a second order dissipative system in Hilbert spaces. SIAM J. Control Optim. **38**, 1102–1119 (2000)
6. M. Anitescu, G.D. Hart, Solving nonconvex problems of multibody dynamics with contact and small friction by sequential convex relaxation. Mech. Based Des. Mach. Struct. **31**, 335–356 (2003)
7. M. Anitescu, G.D. Hart, A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. Int. J. Numer. Methods Eng. **60**, 2335–2371 (2004)
8. M. Anitescu, G.D. Hart, A fixed-point iteration approach for multibody dynamics with contact and small friction. Math. Program. **101**, 3–32 (2004)
9. M. Anitescu, A. Tasora, An iterative approach for cone complementarity problems for nonsmooth dynamics. Comput. Optim. Appl. **47**, 207–235 (2010)
10. M. Anitescu, F.A. Potra, D.E. Stewart, Time-stepping for three-dimensional rigid body dynamics. Comput. Methods Appl. Mech. Eng. **177**, 183–197 (1999)
11. U.M. Archer, L.R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations* (SIAM Publications, Philadelphia, 1998)
12. U.M. Ascher, R.M.M. Mattheij, R.D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM Classics in Applied Mathematics, vol. 13 (Society for Industrial and Applied Mathematics, Philadelphia, 1995)
13. H. Attouch, R. Cominetti, A dynamical approach to convex minimization coupling approximation with the steepest descent method. J. Differ. Equ. **128**, 519–540 (1996)
14. H. Attouch, X. Goudou, P. Redont, The heavy ball with friction method, I. The continuous dynamical systems: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. Commun. Contemp. Math. **2**, 1–34 (2000)
15. J.P. Aubin, A. Cellina, *Differential Inclusions: Set-Valued Maps And Viability Theory* (Springer, New York, 1984)
16. L. Bai, J.E. Mitchell, J.S. Pang, Using quadratic convex reformulation to tighten the convex relaxation of a quadratic program with complementarity constraints. Optim. Lett. **8**, 811–822 (2014)
17. L. Bai, J.E. Mitchell, J.S. Pang, On convex quadratic programs with linear complementarity constraints. Comput. Optim. Appl. **54**, 517–544 (2013)
18. T. Basar, G.J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd edn. Classics in Applied Mathematics, vol. 23 (SIAM Publications, Philadelphia, 1998)
19. S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Studies in Applied Mathematics, vol. 15 (Society for Industrial and Applied Mathematics, Philadelphia, 1994)
20. B. Brogliato, *Nonsmooth Mechanics*, 2nd edn. (Springer, London, 1999)
21. B. Brogliato, Some perspectives on the analysis and control of complementarity systems. IEEE Trans. Autom. Control **48**, 918–935 (2003)
22. B. Brogliato, A.A. ten Dam, L. Paoli, F. Génot, M. Abadie, Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. ASME Appl. Mech. Rev. **55**, 107–150 (2002)
23. B. Brogliato, A. Daniilidis, C. Lemaréchal, V. Acary, On the equivalence between complementarity systems, projected systems and unilateral differential inclusions. Syst. Control Lett. **55**, 45–51 (2006)
24. P. Brunovsky, Regular synthesis for the linear-quadratic optimal control problem with linear control constraints. J. Differ. Equ. **38**, 344–360 (1980)
25. M.K. Camlibel, Complementarity methods in the analysis of piecewise linear dynamical systems, Ph.D. thesis, Center for Economic Research, Tilburg University, The Netherlands, May 2001

26. M.K. Camlibel, J.M. Schumacher, On the Zeno behavior of linear complementarity systems, in *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, vol. 1, pp. 346–351 (2001). https://doi.org/10.1109/.2001.9801242001
27. M.K. Camlibel, J.M. Schumacher, Existence and uniqueness of solutions for a class of piecewise linear dynamical systems. Linear Algebra Appl. **351**, 147–184 (2002)
28. M.K. Camlibel, J.M. Schumacher, Linear passive systems and maximal monotone mappings. Math. Program. Ser. B **157**, 397–420 (2016)
29. M.K. Camlibel, W.P.M.H. Heemels, A.J. van der Schaft, J.M. Schumacher, Well-posedness of hybrid systems, in *Theme 6.43 Control Systems, Robotics and Automation – UNESCO Encyclopedia of Life Support Systems (EOLSS)*, ed. by R. Unbehauen, E6-43-28-02 (2004)
30. M.K. Camlibel, J.S. Pang, J.L. Shen, Conewise linear systems, non-Zenoness and observability. SIAM J. Control Optim. **45**, 1769–1800 (2006)
31. M.K. Camlibel, J.S. Pang, J.L. Shen, Lyapunov stability of complementarity and extended systems. SIAM J. Optim. **17**, 1056–1101 (2006)
32. M.K. Camlibel, W.P.M.H. Heemels, J.M. Schumacher, Algebraic necessary and sufficient conditions for the controllability of conewise linear systems. IEEE Trans. Autom. Control **53**, 762–774 (2008)
33. M.K. Camlibel, W.P.M.H. Heemels, J.M. Schumacher, A full characteriztion of stabilizability of bimodal piecewise linear systems with scalar inputs. Automatica **44**, 1261–1267 (2008)
34. M.K. Camlibel, L. Iannelli, F. Vasca, Passivity and complementarity. Math. Program. **145**, 531–563 (2014)
35. X.D. Chen, D.F. Sun, J. Sun, Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems. Comput. Optim. Appl. **25**, 39–56 (2003)
36. P.W. Christensen, J.S. Pang, Frictional contact algorithms based on semismooth Newton methods, In *Reformulation, Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, ed. by M. Fukushima, L. Qi, pp. 81–116 (Kluwer Academic Publishers, Boston, 1999)
37. P.W. Christensen, A. Klarbring, J.S. Pang, N. Strömberg, Formulation and comparison of algorithms for frictional contact problems. Int. J. Numer. Methods Eng. **42**, 145–173 (1998)
38. R.W. Cottle, J.S. Pang, R.E. Stone, *The Linear Complementarity Problem*. SIAM Classics in Applied Mathematics, vol. 60 (Society for Industrial and Applied Mathematics, Philadelphia, 2009). [Originally published by Academic Press, Boston (1992)]
39. J. Dai, J.M. Harrison, Reflecting Brownian motion in three dimensions: A new proof of sufficient conditions for positive recurrence. Math. Methods Oper. Res. **75**, 135–147 (2012)
40. C.J. Day, B.F. Hobbs, J.S. Pang, Oligopolistic competition in power networks: a conjectured supply function approach. IEEE Trans. Power Syst. **17**, 597–607 (2002)
41. H. De Jong, J.L. Gouzé, C. Hermandez, M. Page, T. Sari, J. Geiselmann, Qualitative simulation of genetic regulatory networks using piecewise-linear models. Bull. Math. Biol. **66**, 301–340 (2004)
42. K. Deimling, *Multivalued Differential Equations* (Walter de Gruyter, Berlin, 1992)
43. P. Dupuis, A. Nagurney, Dynamical systems and variational inequalities. Ann. Oper. Res. **44**, 7–42 (1993)
44. F. Facchinei, C. Kanzow, Generalized Nash equilibrium problems. 4OR **5**, 173–210 (2007)
45. F. Facchinei, J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vols. I and II (Springer, New York, 2003)
46. F. Facchinei, J.S. Pang, Nash equilibria: the variational approach. In *Convex Optimization in Signal Processing and Communications*, ed. by Y. Eldar and D. Palomar. Cambridge University Press (Cambridge, England, 2009), pp. 443–493
47. F. Facchinei, J.S. Pang, G. Scutari, L. Lampariello, VI-constrained hemivariational inequalities: distributed algorithms and power control in ad-hoc networks. Math. Program. Ser. A **145**, 59–96 (2014)
48. R. Findeisen, L. Imsland, F. Allgower, B.A. Foss, State and output feedback nonlinear model predictive control: an overview. Eur. J. Control **9**, 190–206 (2003)

49. A.F. Filippov, *Differential Equations with Discontinuous Right-Hand Sides* (Kluwer Academic Publishers, The Netherlands, 1988)
50. C.E. Garcia, D.M. Prett, M. Morari, Model predictive control: theory and practice–a survey. Automatica **25**, 335–348 (1989)
51. C. Glocker, C. Studer, Formulation and preparation for numerical evaluation of linear complementarity systems in dynamics. Multibody Syst. Dyn. **13**, 447–463 (2005)
52. R. Goebel, R.G. Sanfelice, A. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness* (Princeton University Press, New Jersey, 2012)
53. X. Goudou, J. Munier, The gradient and heavy ball with friction dynamical systems: the quasiconvex case. Math. Program. Ser. B **116**, 173–191 (2009)
54. L. Han, Topics in differential variational systems, Ph.D. thesis, Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy 2007
55. L. Han, J.S. Pang, Non-Zenoness of a class of differential quasi-variational inequalities. Math. Program. Ser. A **121**, 171–199 (2010)
56. L. Han, J.S. Pang, Time-stepping methods for linear complementarity systems, in *Proceedings of the International Conference of Chinese Mathematicians, Part 2*, vol. 51, ed. by L. Ji, Y.S. Poon, L. Yang, S.T. Yau (American Mathematical Society, Providence, RI, 2012), pp. 731–746
57. L. Han, A. Tiwari, K. Camlibel, J.S. Pang, Convergence of time-stepping schemes for passive and extended linear complementarity systems. SIAM J. Numer. Anal. **47**, 1974–1985 (2009)
58. L. Han, S.V. Ukkusuri, K. Doan, Complementarity formulations for the cell transmission model based dynamic user equilibrium with departure time choice, elastic demand and user heterogeneity. Transp. Res. Part B: Methodol. **45**, 1749–1767 (2011)
59. L. Han, M.K. Camlibel, W.P.M.H. Heemels, J.S. Pang, A unified numerical scheme for linear-quadratic optimal control problems with joint control and state constraints. Optim. Methods Softw. **27**, 761–799 (2012)
60. J.M. Harrison, M.I. Reiman, Reflected Brownian motion on an orthant. Ann. Probab. **9**, 302–308 (1981)
61. R.F. Hartl, R. Vickson, S. Sethi, A survey of the maximum principles for optimal control problems with state constraints. SIAM Rev. **37**, 181–218 (1995)
62. W.P.H. Heemels, Linear complementarity systems: a study in hybrid dynamics, Ph.D. thesis, Department of Electrical Engineering, Eindhoven University of Technology, Nov 1999
63. W.P.M.H. Heemels, B. Brogliato, The complementarity class of dynamical systems. Eur. J. Control **9**, 322–360 (2003)
64. W.P.M.H. Heemels, J.M. Schumacher, S. Weiland, The rational complementarity problem. Linear Algebra Appl. **294**, 93–135 (1999)
65. W.P.M.H. Heemels, J.M. Schumacher, S. Weiland, Linear complementarity systems. SIAM J. Appl. Math. **60**, 1234–1269 (2000)
66. W.P.M.H. Heemels, J.M. Schumacher, S. Weiland, Projected dynamical systems in a complementarity formalism. Oper. Res. Lett. **27**, 83–91 (2000)
67. W.P.M.H. Heemels, B. Schutter, A. Bemporad, Equivalence of hybrid dynamical models. Automatica **37**, 1085–1091 (2001)
68. D. Hipfel, The nonlinear differential complementarity problem, Ph.D. thesis, Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 1993
69. B.F. Hobbs, J.S. Pang, Spatial oligopolistic equilibria with arbitrage, shared resources, and price function conjectures. Math. Program. Ser. B **101**, 57–94 (2004)
70. B.F. Hobbs, F.A.M. Rijkers, Strategic generation with conjectured transmission price responses in a mixed transmission pricing system–Part I: Formulation. IEEE Trans. Power Syst. **19**, 707–717 (2004)
71. K.H. Johansson, M. Egersted, J. Lygeros, S.S. Sastry, On the regularization of Zeno hybrid automata. Syst. Control Lett. **38**, 141–150 (1999)
72. H.B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems* (Dover, New York 1992)
73. H. Khalil, *Nonlinear Systems*, 2nd edn. Upper Saddle River (Prentice-Hall, New Jersey, 1996)

74. A. Klarbring, Contact problems with friction – using a finite dimensional description and the theory of linear complementarity. Linköping Studies in Science and Technology, Thesis No. 20, LIU-TEK-LIC-1984:3 (1984)
75. A. Klarbring, A mathematical programming approach to three dimensional contact problems with friction. Comput. Methods Appl. Mech. Eng. **58**, 175–200 (1986)
76. A. Klarbring, Contact, friction, discrete mechanical structures and mathematical programming, in *New Developments in Contact Problems*, ed. by P. Wriggers, P. Panagiotopoulos (Springer, Vienna, 1999), pp. 55–100
77. P. Kunkel, V.L. Mehrmann, *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society Textbooks in Mathematics (European Mathematical Society, Zürich, 2006)
78. M. Kunze, M.D.P. Monteiro Marques, An introduction to Moreau's sweeping process, in *Impact in Mechanical systems: Analysis and Modelling*, ed. by B. Brogliato. Lecture Notes in Physics, vol. 551 (Springer, Berlin, 2000), pp. 1–60
79. S. Lang, *Real and Functional Analysis*, 3rd edn. (Springer, Berlin 1993)
80. A. Lin, A high-order path-following method for locating the least 2-norm solution of monotone LCPs. SIAM J. Optim. **18**, 1414–1435 (2008)
81. Y.J. Lootsma, A.J. van der Schaft, M.K. Camlibel, Uniqueness of solutions of linear relay systems. Automatica **35**, 467–478 (1999)
82. P. Lötstedt, Coulomb friction in two-dimensional rigid body systems. Zeitschrift Angewandte Mathematik und Mechanik **61**, 605–615 (1981)
83. P. Lötstedt, Mechanical systems of rigid bodies subject to unilateral constraints. SIAM J. Appl. Math. **42**, 281–296 (1982)
84. Z.Q. Luo, J.S. Pang, Analysis of iterative waterfilling algorithm for multiuser power control in digital subscriber lines. EURASIP J. Appl. Signal Process. Article ID 24012, 10 pp. (2006)
85. R. Ma, X.J. Ban, J.S. Pang, Continuous-time dynamic system optimum for single-destination traffic networks with queue spillbacks. Transp. Res. Part B **68**, 98–122 (2014)
86. R. Ma, X.J. Ban, J.S. Pang, A link-based dynamic complementarity system formulation for continuous-time dynamic user equilibria with queue spillbacks. Transp. Sci. https://doi.org/10.1287/trsc.2017.0752
87. R. Ma, X.J. Ban, J.S. Pang, H.X. Liu, Continuous-time point-queue models in dynamic network loading. Transp. Res. Part B: Methodol. **46**, 360–380 (2012)
88. R. Ma, X.J. Ban, J.S. Pang, H.X. Liu, Modeling and solving continuous-time instantaneous dynamic user equilibria: a differential complementarity systems approach. Transp. Res. Part B: Methodol. 46, 389–408 (2012)
89. R. Ma, X.J. Ban, J.S. Pang, H.X. Liu, Approximating time delays in solving continuous-time dynamic user equilibria. Netw. Spat. Econ. **15**, 443–463 (2015)
90. A. Machina, A. Ponosov, Filippov solutions in the analysis of piecewise linear models describing gene regulatory networks. Nonlinear Anal. Theory Methods Appl. **74**, 882–900 (2011)
91. A. Mandelbaum, The dynamic complementarity problem. Accepted by *Mathematics of Operations Research*, but never resubmitted
92. M. Morari, J. Lee, Model predictive control: past, present and future. Comput. Chem. Eng. **23**, 667–682 (1999)
93. J.J. Moreau, Evolution problem associated with a moving convex set in a Hilbert space. J. Differ. Equ. **26**, 347–374 (1977)
94. X. Nie, H.M. Zhang, A comparative study of some macroscopic link models used in dynamic traffic assignment. Netw. Spat. Econ. **5**, 89-115 (2005)
95. H. Nijmeijer, A.J. van der Schaft, *Nonlinear Dynamical Control Systems* (Springer, New York, 1990)
96. J.M. Ortega, *Numerical Analysis: A Second Course*. SIAM Classics in Applied Mathematics, vol. 3 (Society for Industrial and Applied Mathematics, Philadelphia, 1990). [Originally published by Academic Press, New York (1970)]

97. J.S. Pang, Frictional contact models with local compliance: semismooth formulation. Zeitschrift für Angewandte Mathematik und Mechanik **88**, 454–471 (2008)
98. J.S. Pang, M. Razaviyayn, A unified distributed algorithm for non-cooperative games with non-convex and non-differentiable objectives, in *Big Data over Networks*, ed. by S. Cui, A. Hero, Z.Q. Luo, J.M.F. Moura. Cambridge University Press (Cambridge, England, 2016), pp. 101–134
99. J.S. Pang, J. Shen, Strongly regular differential variational systems. IEEE Trans. Autom. Control **52**, 242–255 (2007)
100. J.S. Pang, D.E. Stewart, Solution dependence on initial conditions in differential variational inequalities. Math. Program. Ser. B **116**, 429–460 (2009)
101. J.S. Pang, D.E. Stewart, A unified approach to frictional contact problems. Int. J. Eng. Sci. **37**, 1747–1768 (1999)
102. J.S. Pang, D.E. Stewart, Differential variational inequalities. Math. Program. Ser. A **113**, 345–424 (2008)
103. J.S. Pang, V. Kumar, P. Song, Convergence of time-stepping methods for initial and boundary value frictional compliant contact problems. SIAM J. Numer. Anal. **43**, 2200–2226 (2005)
104. J.S. Pang, L. Han, G. Ramadurai, S. Ukkusuri, A continuous-time dynamic equilibrium model for multi-user class single bottleneck traffic flows. Math. Program. Ser. A **133**, 437–460 (2012)
105. F. Pfeiffer, C. Glocker, *Multibody Dynamics with Unilateral Contacts*. Wiley Series in Nonlinear Science (E-book, 2008)
106. L. Qi, J. Sun, A nonsmooth version of Newton's method. Math. Program. **58**, 353–368 (1993)
107. S.M. Robinson, Generalized equations and their solutions. I. Basic theory. Math. Program. Study **10**, 128–141 (1979)
108. A.J. van der Schaft, J.M. Schumacher, Complementarity modeling of hybrid systems. IEEE Trans. Autom. Control **43**, 483–490 (1998)
109. A.J. van der Schaft, J.M. Schumacher, A.J. van der Schaft, *An Introduction to Hybrid Dynamical Systems* (Springer, London, 2000)
110. D. Schiro, J.S. Pang, On differential linear-quadratic Nash games with mixed state-control constraints, in *Proceedings of the IMU-AMS Special Session on Nonlinear Analysis and Optimization, June 2014*, ed. by B. Mordukhovich, S. Reich, and A.J. Zaslavski. Contemporay Mathematics (American Mathematical Society, Providence, RI, 2016), pp. 221–242
111. J.M. Schumacher, Complementarity systems in optimization. Math. Program. Ser. B **101**, 263–295 (2004)
112. S.P. Sethi, G.L. Thompson, *Optimal Control Theory: Applications to Management Science and Economics*, 2nd edn. (Kluwer Academic Publishers, Boston, 2000)
113. J. Shen, Robust non-Zenoness of piecewise analytic systems with applications to complementarity systems, in *Proceedings of the 2010 American Control Conference*, Baltimore, pp. 148–153 (2010)
114. J. Shen, J.S. Pang, Linear Complementarity systems: Zeno states. SIAM J. Control Optim. **44**, 1040–1066 (2005)
115. J. Shen, J.S. Pang, Semicopositive linear complementarity systems. Int. J. Robust Nonlinear Control **17**, 1367–1386 (2007)
116. J. Shen, J.S. Pang, Linear complementarity systems with singleton properties: Non-Zenoness, in *Proceedings of the 2007 American Control Conference*, New York, pp. 2769-2774 (2007)
117. J. Shen, L. Han, J.S. Pang, Switching and stability properties of conewise linear systems. ESAIM: Control Optim. Calc. Var. **16**, 764–793 (2009)
118. G.V. Smirnov, *Introduction to the Theory of Differential Inclusions* (American Mathematical Society, Providence, RI, 2002)
119. P. Song, Modeling, analysis and simulation of multibody systems with contact and friction, Ph.D. thesis, Department of Mechanical Engineering, University of Pennsylvania, 2002
120. P. Song, J.S. Pang, V. Kumar, A semi-implicit time-stepping model for frictional compliant contact problems. Int. J. Numer. Methods Eng. **60**, 2231–2261 (2004)

121. D.E. Stewart, A high accuracy method for solving ODEs with discontinuous right-hand side. Numer. Math. **58**, 299–328 (1990)
122. D.E. Stewart, Convergence of a time-stepping scheme for rigid body dynamics and resolution of Painlevé's problems. Arch. Ration. Mech. Anal. **145**, 215–260 (1998)
123. D.E. Stewart, Rigid-body dynamics with friction and impact. SIAM Rev. **42**, 3–39 (2000)
124. D.E. Stewart, Convolution complementarity problems with application to impact problems. IMA J. Appl. Math. **71**, 92–119 (2006)
125. D.E. Stewart, Differentiating complementarity problems and fractional index convolution complementarity problems. Houst. J. Math. **33**, 301–322 (2006)
126. D.E. Stewart, Uniqueness for solutions of differential complementarity problems. Math. Program. Ser. A **118**, 327–345 (2009)
127. D.E. Stewart, *Dynamics with Inequalities: Impacts and Hard Constraints* (SIAM Publishers, Philadelphia, 2011)
128. D.E. Stewart, Runge-Kutta methods for differential variational inequalities, in *Presentation given at the SIOPT Darmstadt*, Germany (May 2011)
129. D.E. Stewart, Differential variational inequalities and mechanical contact problems. Tutorial presented at the BIRS Meeting on Computational Contact Mechanics: Advances and Frontiers in Modeling Contact 14w5147, Banff, Canada (2014). https://www.birs.ca/workshops/2014/14w5147/files/Stewart-tutorial.pdf
130. D.E. Stewart, J.C. Trinkle, An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. Int. J. Numer. Methods Eng. **39**, 2673–2691 (1996)
131. D.E. Stewart, T.J. Wendt, Fractional index convolution complementarity problems. Nonlinear Anal. Hybrid Syst. **1**, 124–134 (2007)
132. J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis* (Springer, New York, 1980). (See Section 7.3.)
133. H.J. Sussmann, Bounds on the number of switchings for trajectories of piecewise analytic vector fields. J. Differ. Equ. **43**, 399–418 (1982)
134. L.Q. Thuan, Piecewise affine dynamical systems: well-posedness, controllability, and stabilizability, Ph.D. thesis, Department of Mathematics, University of Groningen, 2013
135. L.Q. Thuan, M.K. Camlibel, On the existence, uniqueness and nature of Caratheodory and Filippov solutions for bimodal piecewise affine dynamical systems. Syst. Control Lett. **68**, 76–85 (2014)
136. J.C. Trinkle, J.S. Pang, S. Sudarsky, G. Lo, On dynamic multi-rigid-body contact problems with Coulomb friction. Zeitschrift für Angewandte Mathematik und Mechanik **77**, 267–279 (1997)
137. J. Tzitzouris, J.C. Trinkle, J.S. Pang, Multi-rigid-systems with concurrent distributed contacts. Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. **359**, 2575–2593 (2001)
138. W.S. Vickrey, Congestion theory and transport investment. Am. Econ. Rev. **59**, 251–261 (1969)
139. R. Vinter, *Optimal Control* (Birkhäuser, Boston, 2000)
140. A.A. Vladimirov, V.S. Kozyakin, N.A. Kuznetsov, A. Mandelbaum, An investigation of the dynamic complementarity problem by methods of the theory of desynchronized systems. Russ. Acad. Sci. Dokl. Math. **47**, 169–173 (1993)
141. J. Zhang, K.H. Johansson, J. Lygeros, S.S. Sastry, Zeno hybrid systems. Int. J. Robust Nonlinear Control **11**, 435–451 (2001)

Chapter 3
Parallel and Distributed Successive Convex Approximation Methods for Big-Data Optimization

Gesualdo Scutari and Ying Sun

Abstract Recent years have witnessed a surge of interest in parallel and distributed optimization methods for large-scale systems. In particular, nonconvex large-scale optimization problems have found a wide range of applications in several engineering fields. The design and the analysis of such complex, large-scale, systems pose several challenges and call for the development of new optimization models and algorithms. First, many of the aforementioned applications lead to huge-scale optimization problems. These problems are often referred to as big-data. This calls for the development of solution methods that operate in parallel, exploiting hierarchical computational architectures. Second, many networked systems are spatially (or virtually) distributed. Due to the size of such networks and often to the proprietary regulations, these systems do not possess a single central coordinator or access point that is able to solve alone the entire optimization problem. In this setting, the goal is to develop distributed solution methods that operate seamless in-network. Third, many formulations of interest are nonconvex, with nonconvex objective functions and/or constraints. Except for very special cases, computing the global optimal solution of nonconvex problems might be computationally prohibitive in several practical applications. The desiderata is designing (parallel/distributed) solution methods that are easy to implement (in the sense that the computations performed by the workers are not expensive), with provable convergence to stationary solutions (e.g., local optima) of the nonconvex problem under consideration. To this regard, a powerful and general tool is offered by the so-called Successive Convex Approximation (SCA) techniques: as proxy of the nonconvex problem, a sequence of "more tractable" (possibly convex) subproblems is solved, wherein the original nonconvex functions are replaced by properly chosen "simpler" surrogates.

In this contribution, we put forth a general, unified, algorithmic framework, based on Successive Convex Approximation techniques, for the parallel and distributed solution of a general class of non-convex constrained (non-separable, networked)

G. Scutari (✉) · Y. Sun
Purdue University, West Lafayette, IN, USA
e-mail: gscutari@purdue.edu; sun578@purdue.edu

F. Facchinei, J.-S. Pang (eds.), *Multi-agent Optimization*, Lecture Notes in Mathematics 2224, https://doi.org/10.1007/978-3-319-97142-1_3

problems. The presented framework unifies and generalizes several existing SCA methods, making them appealing for a parallel/distributed implementation while offering a flexible selection of function approximants, step size schedules, and control of the computation/communication efficiency. This contribution is organized according to the lectures that one of the authors delivered at the CIME Summer School on *Centralized and Distributed Multi-agent Optimization Models and Algorithms* held in Cetraro, Italy, June 23–27, 2014. These lectures are: I) Successive Convex Approximation Methods: Basics; II) Parallel Successive Convex Approximation Methods; and III) Distributed Successive Convex Approximation Methods.

3.1 Introduction

Recent years have witnessed a surge of interest in parallel and distributed optimization methods for large-scale systems. In particular, *nonconvex large-scale* optimization problems have found a wide range of applications in several engineering fields as diverse as (networked) information processing (e.g., parameter estimation, detection, localization, graph signal processing), communication networks (e.g., resource allocation in peer-to-peer/multi-cellular systems), sensor networks, data-based networks (including Facebook, Google, Twitter, and YouTube), swarm robotic, and machine learning (e.g., nonlinear least squares, dictionary learning, matrix completion, tensor factorization), just to name a few—see Fig. 3.1.

The design and the analysis of such complex, large-scale, systems pose several challenges and call for the development of new optimization models and algorithms.

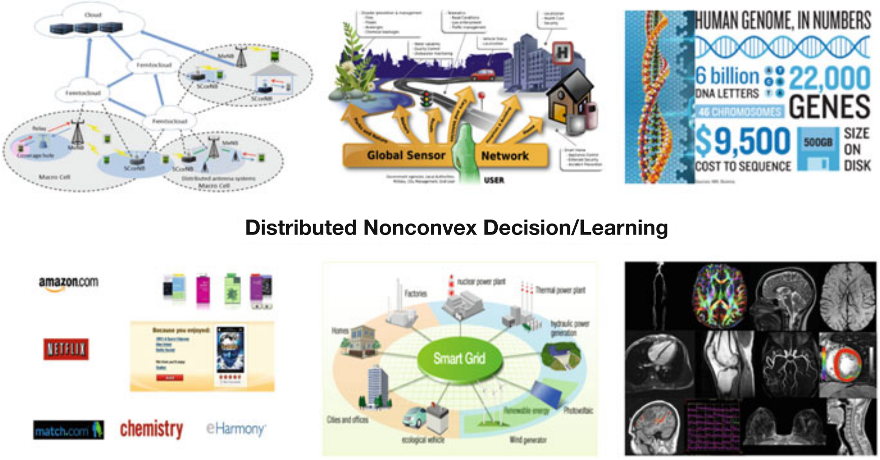


Fig. 3.1 A bird's-eye view of some relevant applications generating nonconvex large-scale (networked) optimization problems

- **Big-Data:** Many of the aforementioned applications lead to *huge-scale* optimization problems (i.e., problems with a very large number of variables). These problems are often referred to as *big-data*. This calls for the development of solution methods that operate *in parallel*, exploiting hierarchical computational architectures (e.g., multicore systems, cluster computers, cloud-based networks), if available, to cope with the curse of dimensionality and accommodate the need of fast (real-time) processing and optimization. The challenge is that such optimization problems are in general not separable in the optimization variables, which makes the design of parallel schemes not a trivial task.
- **In-network optimization:** The networked systems under consideration are typically spatially distributed over a large area (or virtually distributed). Due to the size of such networks (hundreds to millions of agents), and often to the proprietary regulations, these systems do not possess a single central coordinator or access point with the complete system information, which is thus able to solve alone the entire optimization problem. Network/data information is instead distributed among the entities comprising the network. Furthermore, there are some networks such as surveillance networks or some cyber-physical systems, where a centralized architecture is not desirable, as it makes the system prone to central entity fails and external attacks. Additional challenges are encountered from the network topology and connectivity that can be time-varying, due, e.g., to link failures, power outage, and agents' mobility. In this setting, the goal is to develop *distributed* solution methods that operate seamless *in-network*, by leveraging the network connectivity and local information (e.g., neighbor information) to cope with the lack of global knowledge on the optimization problem and offer robustness to possible failures/attacks of central units and/or to time-varying connectivity.
- **Nonconvexity:** Many formulations of interest are *nonconvex*, with nonconvex objective functions and/or constraints. Except for very special classes of nonconvex problems, whose solution can be obtained in closed form, computing the global optimal solution might be computationally prohibitive in several practical applications. This is the case, for instance, of distributed systems composed of workers with limited computational capabilities and power (e.g., motes or smart dust sensors). The desiderata is designing (parallel/distributed) solution methods that are easy to implement (in the sense that the computations performed by the workers are not expensive), with provable convergence to stationary solutions of the nonconvex problem under consideration (e.g., local optimal solutions). To this regard, a powerful and general tool is offered by the so-called *Successive Convex Approximation* (SCA) techniques: as proxy of the nonconvex problem, a sequence of "more tractable" (possibly convex) subproblems is solved, wherein the original nonconvex functions are replaced by properly chosen "simpler" surrogates. By tailoring the choice of the surrogate functions to the specific structure of the optimization problem under consideration, SCA techniques offer a lot of freedom and flexibility in the algorithmic design.

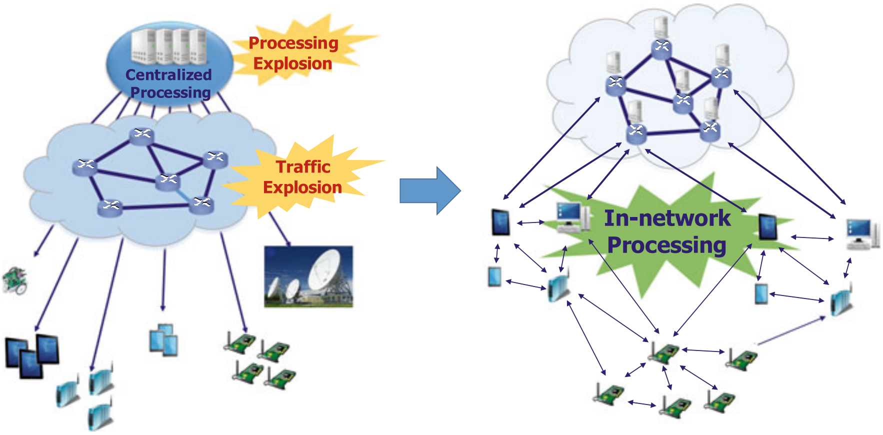


Fig. 3.2 In-network big-data analytics: Traditional centralized processing and optimization are often infeasible or inefficient when dealing with large volumes of data distributed over large-scale networks. There is a necessity to develop fully decentralized algorithms that operate seamless in-network

As a concrete example, consider the emerging field of *in-network big-data analytics*: the goal is to preform some, generally nonconvex, analytic tasks from a sheer volume of data, distributed over a network—see Fig. 3.2—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.

The above challenges make the traditional (centralized) optimization and control techniques inapplicable, thus calling for the development of new computational models and algorithms that support efficient, *parallel* and *distributed nonconvex* optimization over networks. The major contribution of this paper is to put forth a general, unified, algorithmic framework, based on *SCA techniques*, for the parallel and distributed solution of a general class of non-convex constrained (non-separable) problems. The presented framework unifies and generalizes several existing SCA methods, making them appealing for a parallel/distributed implementation while offering a flexible selection of function approximants, step size schedules, and control of the computation/communication efficiency.

This chapter is organized according to the lectures that one of the authors delivered at the CIME Summer School on *Centralized and Distributed Multi-agent*

Optimization Models and Algorithms held in Cetraro, Italy, June 23–27, 2014. These lectures are:

Lecture I—Successive Convex Approximation Methods: Basics (Sect. 3.2)
Lecture II—Parallel Successive Convex Approximation Methods (Sect. 3.3)
Lecture III—Distributed Successive Convex Approximation Methods (Sect. 3.4)

Omissions Consistent with the main theme of the Summer School, the lectures aim at presenting SCA-based algorithms as a powerful framework for parallel and distributed, nonconvex multi-agent optimization. Of course, other algorithms have been proposed in the literature for parallel and distributed optimization. This paper does not cover schemes that are not directly related to SCA-methods or provably applicable to nonconvex problems. Examples of omissions are: primal-dual methods; augmented Lagrangian methods, including the alternating direction methods of multipliers (ADMM); and Newton methods and their inexact versions. When relevant, we provide citations of the omitted algorithms at the end of each lecture, in the section of "Source and Notes".

3.2 Successive Convex Approximation Methods: Basics

This lecture overviews the majorization-minimization (MM) algorithmic framework, a particular instance of Successive Convex Approximation (SCA) Methods. The MM basic principle is introduced along with its convergence properties, which will set the ground for the design and analysis of SCA-based algorithms in the subsequent lectures. Several examples and applications are also discussed.

Consider the following general class of nonconvex optimization problems

$$\underset{\mathbf{x}\in X}{\text{minimize}}\ \ V(\mathbf{x}), \tag{3.1}$$

where $X \subseteq \mathbb{R}^m$ is a nonempty closed convex set and $V : O \to \mathbb{R}$ is continuous (possibly nonconvex and nonsmooth) on O, an open set containing X. Further assumptions on V are introduced as needed.

The MM method applied to Problem (3.1) is based on the solution of a sequence of "more tractable" subproblems whereby the objective function V is replaced by a "simpler" suitably chosen surrogate function. At each iteration k, a subproblem is solved of the type

$$\mathbf{x}^{k+1} \in \underset{\mathbf{x}\in X}{\operatorname{argmin}}\, \widetilde{V}(\mathbf{x} \mid \mathbf{x}^k), \tag{3.2}$$

where $\widetilde{V}(\bullet \mid \mathbf{x}^k)$ is a surrogate function (generally dependent on the current iterate $\mathbf{x}^k$) that upperbounds V globally (further assumptions on $\widetilde{V}$ are introduced as needed). The sequence of majorization-minimization steps are pictorially shown in Fig. 3.3. The underlying idea of the approach is that the surrogate function $\widetilde{V}$

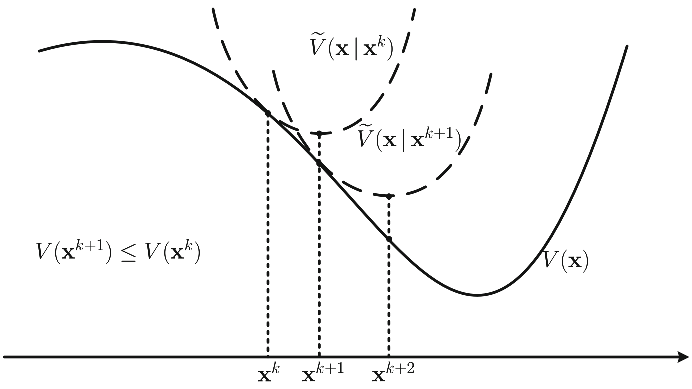

$\widetilde{V}(\mathbf{x}\,|\,\mathbf{x}^k)$
$\widetilde{V}(\mathbf{x}\,|\,\mathbf{x}^{k+1})$
$V(\mathbf{x}^{k+1}) \leq V(\mathbf{x}^k)$
$V(\mathbf{x})$
$\mathbf{x}^k$
$\mathbf{x}^{k+1}$
$\mathbf{x}^{k+2}$

Fig. 3.3 Pictorial description of the MM procedure [230]

is chosen so that the resulting subproblem (3.2) can be efficiently solved. Roughly speaking, surrogate functions enjoying the following features are desirable:

- (Strongly) Convexity: this would lead to (strongly) convex subproblems (3.2);
- (Additively) Block-separability in the optimization variables: this is a key enabler for parallel/distributed solution methods, which are desirable to solve large-scale problems;
- Minimizer over X in closed-form: this reduces the cost per iteration of the MM algorithm.

Finding the "right" surrogate function for the problem under consideration (possibly enjoying the properties above) might not be an easy task. A major goal of this section is to put forth general construction techniques for $\widetilde{V}$ and show their application to some representative problems in signal processing, data analysis, and communications. Some instances of $\widetilde{V}$ are drawn from the literature, e.g., [230], while some others are new and introduced for the first time in this chapter. The rest of this lecture is organized as follows. After introducing in Sect. 3.2.1 some basic results which will lay the foundations for the analysis of SCA methods in the subsequent sections, in Sect. 3.2.2 we describe in details the MM framework along with its convergence properties; several examples of valid surrogate functions are also discussed (cf. Sect. 3.2.2.1). When the surrogate function $\widetilde{V}$ is block separable and so are the constraints in (3.2), subproblems (3.2) can be solved leveraging parallel algorithms. For unstructured functions V, in general separable surrogates are difficult to be found. When dealing with large scale optimization problems, solving (3.2) with respect to *all* variables might not be efficient or even possible; in all these cases, parallel block schemes are mandatory. This motivates the study of so-called "block MM" algorithms—only some blocks of the variables are selected and optimized at a time. Section 3.2.3 is devoted to the study of such algorithms. In Sect. 3.2.4 we will present several applications of MM methods to problems in signal processing, machine learning, and communications. Finally, in Sect. 3.2.5 we overview the main literature and highlight some extensions and generalizations of the methods described in this lecture.

3.2.1 Preliminaries

We introduce here some preliminary basic results which will be extensively used through the whole paper.

We begin with the definition of directional derivative of a function and some basic properties of directional derivatives.

Definition 2.1 (Directional Derivative) A function $f : \mathbb{R}^m \to (-\infty, \infty]$ is directionally differentiable at $\mathbf{x} \in \mathrm{dom} f \triangleq \{\mathbf{x} \in \mathbb{R}^m : f(\mathbf{x}) < \infty\}$ along a direction $\mathbf{d} \in \mathbb{R}^m$ if the following limit

$$f'(\mathbf{x}; \mathbf{d}) \triangleq \lim_{\lambda \downarrow 0} \frac{f(\mathbf{x} + \lambda \mathbf{d}) - f(\mathbf{x})}{\lambda} \tag{3.3}$$

exists; this limit $f'(\mathbf{x}; \mathbf{d})$ is called the directional derivative of f at $\mathbf{x}$ along $\mathbf{d}$. If f is directionally differentiable at $\mathbf{x}$ along all directions, then we say that f is directionally differentiable at $\mathbf{x}$. □

If f is differentiable at $\mathbf{x}$, then $f'(\mathbf{x}; \mathbf{d})$ reads: $f'(\mathbf{x}; \mathbf{d}) = \nabla f(\mathbf{x})^T \mathbf{d}$, where $\nabla f(\mathbf{x})$ is the gradient of f at $\mathbf{x}$. Some examples of directional derivatives of some structured functions (including convex functions) are discussed next.

• **Case Study 1: Convex Functions** Throughout this example, we assume that $f : \mathbb{R}^m \to (-\infty, \infty]$ is a convex, closed, proper function; and $\mathrm{int}(\mathrm{dom} f) \neq \emptyset$ (otherwise, one can work with the relative interior of $\mathrm{dom} f$), with $\mathrm{int}(\mathrm{dom} f)$ denoting the interior of $\mathrm{dom} f$.

We show next that if $\mathbf{x} \in \mathrm{dom} f$, $f'(\mathbf{x}; \mathbf{d})$ is well defined, taking values in $[-\infty, +\infty]$. In particular, if $\mathbf{x} \in \mathrm{dom} f$ can be approached by the direction $\mathbf{d} \in \mathbb{R}^m$, then $f'(\mathbf{x}; \mathbf{d})$ is finite. For $\mathbf{x} \in \mathrm{dom} f$, $\mathbf{d} \in \mathbb{R}^m$ and nonzero $\lambda \in \mathbb{R}$, define

$$\lambda \mapsto g_\lambda(\mathbf{x}; \mathbf{d}) \triangleq \frac{f(\mathbf{x} + \lambda \mathbf{d}) - f(\mathbf{x})}{\lambda}.$$

A simple argument by convexity (increasing slopes) shows that $g(\mathbf{d}; \lambda)$ is increasing in λ. Therefore, the limit in (3.3) exists in $[-\infty, \infty]$ and can be replaced by

$$f'(\mathbf{x}; \mathbf{d}) = \inf_{\lambda > 0} \frac{1}{\lambda} \left[f(\mathbf{x} + \lambda \mathbf{d}) - f(\mathbf{x}) \right].$$

Moreover, for $0 < \lambda \leq \beta \in \mathbb{R}$, it holds

$$g_{-\beta}(\mathbf{x}; \mathbf{d}) \leq g_{-\lambda}(\mathbf{x}; \mathbf{d}) \leq g_\lambda(\mathbf{x}; \mathbf{d}) \leq g_\beta(\mathbf{x}; \mathbf{d}).$$

If $\mathbf{x} \in \text{int}(\text{dom} f)$, both $g_{-\beta}(\mathbf{x}; \mathbf{d})$ and $g_{\beta}(\mathbf{x}; \mathbf{d})$ are finite, for sufficiently small $\beta > 0$; therefore, we have

$$-\infty < g_{-\beta}(\mathbf{x}; \mathbf{d}) \leq f'(\mathbf{x}; \mathbf{d}) = \inf_{\lambda > 0} g_{\lambda}(\mathbf{d}; \mathbf{x}) \leq g_{\beta}(\mathbf{x}; \mathbf{d}) < +\infty.$$

Finally, since f is convex, it is locally Lipschitz continuous: for sufficiently small $\beta > 0$, there exists some finite $L > 0$ such that $g_{\beta}(\mathbf{x}; \mathbf{d}) \leq L\|\mathbf{d}\|$ and $g_{-\beta}(\mathbf{x}; \mathbf{d}) \geq -L\|\mathbf{d}\|$. We have proved the following result.

Proposition 2.2 *For convex functions $f : \mathbb{R}^m \to (-\infty, \infty]$, at any $\mathbf{x} \in \text{dom} f$ and for any $\mathbf{d} \in \mathbb{R}^m$, the directional derivative $f'(\mathbf{x}; \mathbf{d})$ exists in $[-\infty, +\infty]$ and it is given by*

$$f'(\mathbf{x}; \mathbf{d}) = \inf_{\lambda > 0} \frac{1}{\lambda} \left[f(\mathbf{x} + \lambda \mathbf{d}) - f(\mathbf{x}) \right].$$

If $\mathbf{x} \in$ int(dom$f$), there exists a finite constant $L > 0$ such that $|f'(\mathbf{x}; \mathbf{d})| \leq L\|\mathbf{d}\|$, for all $\mathbf{d} \in \mathbb{R}^m$. □

Directional Derivative and Subgradients The directional derivative of a convex function can be also written in terms of its subgradients, as outlined next. We first introduce the definition of subgradient along with some of its properties.

Definition 2.3 (Subgradient) A vector $\boldsymbol{\xi} \in \mathbb{R}^m$ is a subgradient of f at a point $\mathbf{x} \in \text{dom} f$ if

$$f(\mathbf{x} + \mathbf{d}) \geq f(\mathbf{x}) + \boldsymbol{\xi}^T \mathbf{d}, \ \forall \mathbf{d} \in \mathbb{R}^m. \tag{3.4}$$

The *subgradient set* (a.k.a. *subdifferential*) of f at $\mathbf{x} \in \text{dom} f$ is defined as

$$\partial f(\mathbf{x}) \triangleq \left\{ \boldsymbol{\xi} \in \mathbb{R}^m \ : \ f(\mathbf{x} + \mathbf{d}) \geq f(\mathbf{x}) + \boldsymbol{\xi}^T \mathbf{d}, \ \forall \mathbf{d} \in \mathbb{R}^m \right\}. \tag{3.5}$$

Partitioning $\mathbf{x}$ in blocks, $\mathbf{x} = (\mathbf{x}_i)_{i=1}^n$, with $\mathbf{x}_i \in \mathbb{R}^{m_i}$ and $\sum_{i=1}^n m_i = m$, similarly to (3.5), we can define the *block-subdifferential* with respect to each $\mathbf{x}_i$, as given below, where $(\mathbf{x})_i \triangleq (\mathbf{0}^T, \ldots, \mathbf{x}_i^T, \ldots, \mathbf{0}^T)^T \in \mathbb{R}^m$.

Definition 2.4 (Block-Subgradient) The subgradient set $\partial_i f(\mathbf{x})$ of f at $\mathbf{x} = (\mathbf{x}_i)_{i=1}^n \in \text{dom} f$ with respect to $\mathbf{x}_i$ is defined as

$$\partial_i f(\mathbf{x}) \triangleq \left\{ \boldsymbol{\xi}_i \in \mathbb{R}^{m_i} \ : \ f(\mathbf{x} + (\mathbf{d})_i) \geq f(\mathbf{x}) + \boldsymbol{\xi}_i^T \mathbf{d}_i, \ \forall \mathbf{d}_i \in \mathbb{R}^{m_i} \right\}. \tag{3.6}$$

Intuitively, when a function f is convex, the subgradient generalizes the derivative of f. Since a convex function has global linear underestimators of itself, the subgradient set $\partial f(\mathbf{x})$ should be non-empty and consist of supporting hyperplanes to the epigraph of f. This is formally stated in the next result (see, e.g., [24, 104] for the proof).

Theorem 2.5 *Let* $\mathbf{x} \in int(dom f)$*. Then,* $\partial f(\mathbf{x})$ *is nonempty, compact, and convex.*

Note that, in the above theorem, we cannot relax the assumption $\mathbf{x} \in \text{int}(\text{dom} f)$ with $\mathbf{x} \in \text{dom} f$. For instance, consider the function $f(x) = -\sqrt{x}$, with $\text{dom} f = [0, \infty)$. We have $\partial f(0) = \emptyset$.

The subgradient definition describes a global properties of the function whereas the (directional) derivative is a local property. The connection between a directional derivative and the subdifferential of a convex function is contained in the next two results, whose proof can be found in [24, Ch.3].

Lemma 2.6 *The subgradient set (3.5) at* $\mathbf{x} \in dom f$ *can be equivalently written as*

$$\partial f(\mathbf{x}) \triangleq \left\{ \boldsymbol{\xi} \in \mathbb{R}^m \; : \; f'(\mathbf{x}; \mathbf{d}) \geq \boldsymbol{\xi}^T \mathbf{d}, \quad \forall \mathbf{d} \in \mathbb{R}^m \right\}. \tag{3.7}$$

Note that, since $f'(\mathbf{x}; \mathbf{d})$ is finite for all $\mathbf{d} \in \mathbb{R}^m$ (cf. Proposition 2.2), the above representation readily shows that $\partial f(\mathbf{x})$, $\mathbf{x} \in \text{int}(\text{dom} f)$, is a compact set (as proved already in Theorem 2.5). Furthermore, $\boldsymbol{\xi} \in \partial f(\mathbf{x})$ satisfies

$$\|\boldsymbol{\xi}\|_2 = \sup_{\mathbf{d} : \|\mathbf{d}\|_2 \leq 1} \boldsymbol{\xi}^T \mathbf{d} \leq \sup_{\mathbf{d} : \|\mathbf{d}\|_2 \leq 1} f'(\mathbf{x}; \mathbf{d}) < \infty.$$

Lemma 2.6 above showed how to identify subgradients from directional derivative. Lemma 2.7 below shows how to move in the reverse direction.

Lemma 2.7 (Max Formula) *At any* $\mathbf{x} \in int(dom f)$ *and all* $\mathbf{d} \in \mathbb{R}^m$*, it holds*

$$f'(\mathbf{x}; \mathbf{d}) = \sup_{\boldsymbol{\xi} \in \partial f(\mathbf{x})} \boldsymbol{\xi}^T \mathbf{d}. \tag{3.8}$$

Lastly, we recall a straightforward result, stating that the subgradient is simply the gradient of differentiable convex functions. This is a direct consequence of Lemma 2.6. Indeed, if f is differentiable at $\mathbf{x}$, we can write [cf. (3.7)]

$$\boldsymbol{\xi}^T \mathbf{d} \leq f'(\mathbf{x}; \mathbf{d}) = \nabla f(\mathbf{x})^T \mathbf{d}, \qquad \forall \boldsymbol{\xi} \in \partial f(\mathbf{x}).$$

Since the above inequality holds for all $\mathbf{d} \in \mathbb{R}^m$, we also have $\boldsymbol{\xi}^T(-\mathbf{d}) \leq f'(\mathbf{x}; -\mathbf{d}) = \nabla f(\mathbf{x})^T(-\mathbf{d})$, and thus $\boldsymbol{\xi}^T \mathbf{d} = \nabla f(\mathbf{x})^T \mathbf{d}$, for all $\mathbf{d} \in \mathbb{R}^m$. This proves $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$.

The subgradient is also intimately related to optimality conditions for convex minimization. We discuss this relationship in the next subsection. We conclude this brief review with some basic examples of calculus of subgradient.

Examples of Subgradients As the first example, consider

$$f(x) = |x|.$$

It is not difficult to check that

$$\partial |x| = \begin{cases} \text{sign}(x), & \text{if } x \neq 0; \\ [-1,\ 1] & \text{if } x = 0; \end{cases} \tag{3.9}$$

where $\text{sign}(x) = 1$, if $x > 0$; $\text{sign}(x) = 0$, if $x = 0$; and $\text{sign}(x) = -1$, if $x < 0$.

Similarly, consider the ℓ_1 norm function, $f(\mathbf{x}) = \|\mathbf{x}\|_1$. We have

$$\begin{aligned} \partial \|\mathbf{x}\|_1 &= \sum_{i=1}^{m} \partial |x_i| = \sum_{i=1}^{m} \begin{cases} \mathbf{e}_i \cdot \text{sign}(x_i), & \text{if } x_i \neq 0; \\ \mathbf{e}_i \cdot [-1,\ 1], & \text{if } x_i = 0; \end{cases} \\ &= \sum_{x_i > 0} \mathbf{e}_i - \sum_{x_i < 0} \mathbf{e}_i + \sum_{x_i = 0} [-\mathbf{e}_i, \mathbf{e}_i], \end{aligned} \tag{3.10}$$

where $\mathbf{e}_i$ denotes the i-th standard basis vector of $\mathbb{R}^m$; and the sum for $x_i = 0$ is the Minkowski sum. Therefore,

$$\partial \|\mathbf{0}\|_1 = \sum_{i=1}^{m} [-\mathbf{e}_i, \mathbf{e}_i] = \left\{ \mathbf{x} \in \mathbb{R}^m \ : \ \|\mathbf{x}\|_\infty \leq 1 \right\}.$$

A more complex example is given by considering any norm function $\| \bullet \|$. Introducing the dual norm

$$\|\mathbf{x}\|_* \triangleq \sup_{\mathbf{y}\,:\,\|\mathbf{y}\| \leq 1} \mathbf{x}^T \mathbf{y},$$

one can show that

$$\partial \|\mathbf{x}\| = \left\{ \boldsymbol{\xi} \in \mathbb{R}^m \ : \ \|\boldsymbol{\xi}\|_* \leq 1,\ \boldsymbol{\xi}^T \mathbf{x} = \|\mathbf{x}\| \right\}. \tag{3.11}$$

As a concrete example, consider the ℓ_2 norm, $f(\mathbf{x}) = \|\mathbf{x}\|_2$. Observing that

$$\|\mathbf{x}\|_2 = \sup_{\|\mathbf{y}\|_2 \leq 1} \mathbf{x}^T \mathbf{y},$$

a direct application of (3.11) yields

$$\begin{aligned} \partial \|\mathbf{x}\|_2 &= \left\{ \boldsymbol{\xi} \in \mathbb{R}^m \ : \ \|\boldsymbol{\xi}\|_2 \leq 1,\ \boldsymbol{\xi}^T \mathbf{x} = \|\mathbf{x}\|_2 \right\} \\ &= \begin{cases} \dfrac{\mathbf{x}}{\|\mathbf{x}\|_2}, & \text{if } \mathbf{x} \neq \mathbf{0}; \\ \{\boldsymbol{\xi} \in \mathbb{R}^m \ : \ \|\boldsymbol{\xi}\|_2 \leq 1\}, & \text{if } \mathbf{x} = \mathbf{0}. \end{cases} \end{aligned}$$

• **Case Study 2: Pointwise Max of Functions** Consider the pointwise maximum of (possibly) nonconvex functions

$$g(\mathbf{x}) \triangleq \max_{i=1,\ldots,I} f_i(\mathbf{x}), \tag{3.12}$$

where each $f_i : \mathbb{R}^m \to (-\infty, +\infty]$ is assumed to be directionally differentiable at a given $\bar{\mathbf{x}}$ along the direction $\mathbf{d}$ (with finite directional derivative). For notational simplicity, we assume that all f_i have the same effective domain. The following lemma shows that $g(\mathbf{x})$ is directional differentiable at $\bar{\mathbf{x}}$ along $\mathbf{d}$ and provides an explicit expression for $g'(\bar{\mathbf{x}}; \mathbf{d})$.

Lemma 2.8 *In the above setting, the function g defined in (3.12) is directionally differentiable at $\bar{\mathbf{x}}$ along $\mathbf{d}$, with*

$$g'(\bar{\mathbf{x}}; \mathbf{d}) = \max_{i \in A(\bar{\mathbf{x}})} f_i'(\bar{\mathbf{x}}; \mathbf{d}), \tag{3.13}$$

where $A(\mathbf{x}) \triangleq \{i = 1, \ldots, I \ : \ f_i(\mathbf{x}) = g(\mathbf{x})\}$.

Proof The proof follows similar steps of that of Danskin's theorem [57]. Let $\{t^k\}$ be a sequence of positive numbers t^k such that $t^k \to 0$ as $k \to \infty$. Define $\mathbf{x}^k = \bar{\mathbf{x}} + t^k\, \mathbf{d}$; and let i^k and $\bar{i}$ be two indices in $A(\mathbf{x}^k)$ and $A(\bar{\mathbf{x}})$, respectively. We prove (3.13) by showing that

$$\limsup_{k\to\infty} \frac{g(\mathbf{x}^k) - g(\bar{\mathbf{x}})}{t^k} \le \max_{i \in A(\bar{\mathbf{x}})} f_i'(\bar{\mathbf{x}}; \mathbf{d}) \le \liminf_{k\to\infty} \frac{g(\mathbf{x}^k) - g(\bar{\mathbf{x}})}{t^k}. \tag{3.14}$$

We prove the right inequality first. We have

$$\begin{aligned} \liminf_{k\to\infty} \frac{g(\mathbf{x}^k) - g(\bar{\mathbf{x}})}{t^k} &= \liminf_{k\to\infty} \frac{f_{i^k}(\mathbf{x}^k) - f_{\bar{i}}(\bar{\mathbf{x}})}{t^k} \\ &= \liminf_{k\to\infty} \frac{f_{i^k}(\mathbf{x}^k) - f_{\bar{i}}(\mathbf{x}^k) + f_{\bar{i}}(\mathbf{x}^k) - f_{\bar{i}}(\bar{\mathbf{x}})}{t^k} \\ &\overset{(a)}{\ge} \liminf_{k\to\infty} \frac{f_{\bar{i}}(\mathbf{x}^k) - f_{\bar{i}}(\bar{\mathbf{x}})}{t^k} \overset{(b)}{=} f_{\bar{i}}'(\bar{\mathbf{x}}; \mathbf{d}), \end{aligned} \tag{3.15}$$

where (a) follows from $f_{i^k}(\mathbf{x}^k) - f_{\bar{i}}(\mathbf{x}^k) \ge 0$; and in (b) we used the fact that each f_i is directionally differentiable at $\bar{x}$ along $\mathbf{d}$. Since $\bar{i}$ is any arbitrary index in $A(\bar{\mathbf{x}})$, we have

$$\liminf_{k\to\infty} \frac{g(\mathbf{x}^k) - g(\bar{\mathbf{x}})}{t^k} \ge \max_{i \in A(\bar{\mathbf{x}})} f_i'(\bar{\mathbf{x}}; \mathbf{d}). \tag{3.16}$$

We prove next the inequality on the left of (3.14). Following similar steps, we have

$$\begin{aligned}\limsup_{k\to\infty}\frac{g(\mathbf{x}^k)-g(\bar{\mathbf{x}})}{t^k}&=\limsup_{k\to\infty}\frac{f_{i^k}(\mathbf{x}^k)-f_{\bar{i}}(\bar{\mathbf{x}})}{t^k}\\&=\limsup_{k\to\infty}\frac{f_{i^k}(\mathbf{x}^k)-f_{i^k}(\bar{\mathbf{x}})+f_{i^k}(\bar{\mathbf{x}})-f_{\bar{i}}(\bar{\mathbf{x}})}{t^k}\\&\overset{(a)}{\le}\limsup_{k\to\infty}\frac{f_{i^k}(\mathbf{x}^k)-f_{i^k}(\bar{\mathbf{x}})}{t^k}\\&\overset{(b)}{\le}\max_{i\in A(\bar{\mathbf{x}})} f_i'(\bar{\mathbf{x}};\mathbf{d}),\end{aligned}\tag{3.17}$$

where (a) comes from $f_{i^k}(\bar{\mathbf{x}})-f_{\bar{i}}(\bar{\mathbf{x}})\le 0$; and (b) is due to $\lim_{k\to\infty}\min_{j\in A(\bar{\mathbf{x}})}|i^k-j|=0$, which is a consequence of $\lim_{k\to\infty}\|\mathbf{x}^k-\bar{\mathbf{x}}\|=0$, as showed next.

Suppose that the above statement is not true. Then, there exits a subsequence of i^k—say i^{k_ℓ}, with $i^{k_\ell}\in A(\mathbf{x}^{k_\ell})$—such that $\lim_{\ell\to\infty} i^{k_\ell}=i^\infty\notin A(\bar{\mathbf{x}})$ (note that $A(\bar{\mathbf{x}})$ has a finite cardinality). Therefore, for sufficiently large ℓ, it holds $g(\mathbf{x}^{k_\ell})=f_{i^{k_\ell}}(\mathbf{x}^{k_\ell})=f_{i^\infty}(\mathbf{x}^{k_\ell})$. Letting $\ell\to+\infty$ and invoking continuity of g (it is the point-wise maximum of finitely many continuous functions), we get $g(\bar{\mathbf{x}})=f_{i^\infty}(\bar{\mathbf{x}})$, which is in contradiction with $i^\infty\notin A(\bar{\mathbf{x}})$. □

Optimality Conditions As a non-convex optimization problem, globally optimal solutions of Problem (3.1) are in general not possible to be computed. Thus, one has to settle for computing a "stationary" solution in practice. Even in this case, there are many kinds of stationary solutions for Problem (3.1). Ideally, one would like to identify a stationary solution of the sharpest kind. Arguably, for the convex constrained nonconvex program (3.1), a d(irectional)-stationary solution defined in terms of the directional derivatives of the objective function would qualify for this purpose. For the sake of simplicity, we will make the following blanket assumptions on Problem (3.1): (1) V is directionally differentiable on X; and (2) X is closed and convex. We introduce next two concepts of stationarity, namely: d-stationarity and *coordinate-wise* d-stationarity.

Definition 2.9 (d-Stationarity) Given Problem (3.1) in the above setting, $\mathbf{x}^*\in X$ is a d-stationary solution of (3.1) if

$$V'\left(\mathbf{x}^*;\mathbf{y}-\mathbf{x}^*\right)\ge 0,\quad \forall \mathbf{y}\in X.\tag{3.18}$$

Two remarks are in order. When V is convex, it follows from Lemma 2.7 that $\mathbf{x}^*$ is a d-stationary (and thus a global optimal) solution of Problem (3.1) if there exists a $\boldsymbol{\xi}\in\partial f\,(\mathbf{x}^*)$ such that $\boldsymbol{\xi}^T\,(\mathbf{y}-\mathbf{x}^*)\ge 0$, $\forall\mathbf{y}\in X$. Furthermore, if V is differentiable, since $V'(\mathbf{x};\mathbf{d})=\nabla V(\mathbf{x})^T\mathbf{d}$, (3.18) reads $(\mathbf{y}-\mathbf{x}^*)^T\nabla V(\mathbf{x}^*)\ge 0$, $\forall\mathbf{y}\in X$.

Definition 2.10 (Coordinate-Wise d-Stationary) Given Problem (3.1), with $X = X_1 \times \cdots \times X_n$, $X_i \subseteq \mathbb{R}^{m_i}$, and $\sum_{i=1}^n m_i = m$, $\mathbf{x}^*$ is a coordinate-wise d-stationary solution of Problem (3.1) if $V'\left(\mathbf{x}^*; (\mathbf{y}-\mathbf{x}^*)_i\right) \geq 0,\ \forall \mathbf{y} \in X$ and all $i = 1, \ldots, n$.

In words, a coordinate-wise stationary solution is a point for which $\mathbf{x}^*$ is stationary w.r.t. every block of variables. Coordinate-wise stationarity is a weaker form of stationarity. It is the standard property of a limit point of a convergent coordinate-wise scheme (see, e.g., [250]). It is clear that a stationary point is always a coordinate-wise stationary point; the converse however is not always true, unless extra conditions on V are satisfied.

Definition 2.11 (Regularity) Problem (3.1) is regular at a coordinate-wise d-stationary solution $\mathbf{x}^* \in X = X_1 \times \cdots \times X_n$, if $\mathbf{x}^*$ is also a d-stationary point of the problem.

The regularity condition is readily satisfied in the following two simple cases:

(a) V is additively separable (possibly nonsmooth), i.e., $V(\mathbf{x}) = \sum_{i=1}^n V_i(\mathbf{x}_i)$;
(b) V is differentiable around $\mathbf{x}^*$.

Note that (a) is due to the separability of the directional derivative, that is, $V'(\mathbf{x};\mathbf{d}) = \sum_{i=1}^n V_i'(\mathbf{x}_i;\mathbf{d}_i)$; and so does (b).

Of course the two cases above are not at all inclusive of situations for which regularity holds. As an example of a nonseparable function for which regularity holds at a point at which is not continuously differentiable, consider the function arising in logistic regression problems $V(\mathbf{x}) = \sum_{i=1}^I \log(1 + \mathrm{e}^{-a_i \mathbf{y}_i^T \mathbf{x}}) + c \cdot \|\mathbf{x}\|_2$, where $X = \mathbb{R}^m$, and $\mathbf{y}_i \in \mathbb{R}^m$ and $a_i \in \{-1,\ 1\}$ are given constants. Such a function V is continuously differentiable, and thus regular, at any stationary point but $\mathbf{x}^* \neq \mathbf{0}$. It is easy to check that V is regular also at $\mathbf{x}^* = \mathbf{0}$, if $c < \log 2$.

Finally, an example of a nonsmooth, nonseparable function that is not regular is $V(\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|_1$, with $\mathbf{A} = [3\ 4; 2\ 1]$ and $X = \mathbb{R}^2$. Point $\mathbf{x}^* = [-4\ 3]^T$ is a coordinate-wise d-stationary point, but not d-stationary [cf. Fig. 3.4].

3.2.2 The Majorization-Minimization (MM) Algorithm

We study Problem (3.1) under the following blanket assumptions.

Assumption 2.12 *Given Problem* (3.1)*, we assume that:*

1. *$X \neq \emptyset$ is a closed and convex set in $\mathbb{R}^m$;*
2. *$V : O \to \mathbb{R}$ is continuous on the open set $O \supseteq X$;*
3. *$V'(\mathbf{x};\mathbf{d})$ exists at any $\mathbf{x} \in X$ and for all feasible directions $\mathbf{d} \in \mathbb{R}^m$ at $\mathbf{x}$;*
4. *V is bounded from below.*

Note that the above assumptions are quite standard and are satisfied by most of the problems of practical interest; see Sect. 3.2.4 for some illustrative examples.

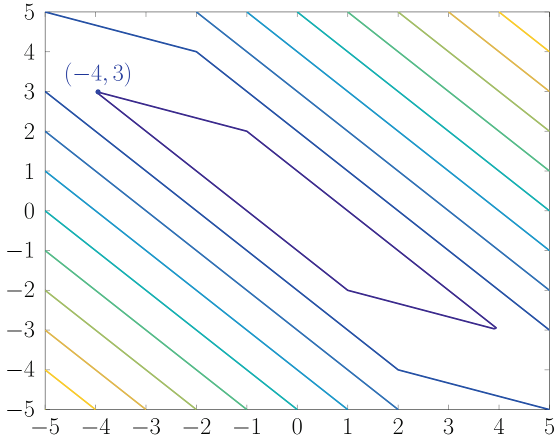


Fig. 3.4 Contour of $V(\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|_1$, with $\mathbf{A} = [3\ 4;\ 2\ 1]$. The function is not regular at $\mathbf{x}^* = [-4, 3]^T$ [194]

As already anticipated at the beginning of this section, the idea of the MM procedure is to approximate, at each iteration, the objective function V in (3.1) by a "simpler" properly chosen surrogate function $\widetilde{V}(\bullet \,|\, \mathbf{x}^k)$ and solve the resulting optimization problem (3.2). Convergence of this iterative method is guaranteed if the following conditions are satisfied in the choice of $\widetilde{V}$. In what follows, we denote by $\widetilde{V}'(\mathbf{y}; \mathbf{d} \,|\, \mathbf{x})$ the directional derivative of $\widetilde{V}(\bullet \,|\, \mathbf{x})$ at $\mathbf{y}$ along the direction $\mathbf{d}$.

Assumption 2.13 *The surrogate function $\widetilde{V} : O \times O \to \mathbb{R}$ satisfies the following conditions:*

1. $\widetilde{V}(\bullet \,|\, \bullet)$ *is continuous on $X \times X$;*
2. $\mathbf{y} \in \operatorname{argmin}_{\mathbf{x} \in X} \widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) - V(\mathbf{x})$*;*
3. *The directional derivative of $\widetilde{V}$ satisfies $\widetilde{V}'(\mathbf{x}; \mathbf{d} \,|\, \mathbf{x}) = V'(\mathbf{x}; \mathbf{d})$, for all $\mathbf{x} \in X$ and feasible directions $\mathbf{d} \in \mathbb{R}^m$ at $\mathbf{x}$.*

Assumption 2.13.2 states that, at any feasible $\mathbf{y}$, $\widetilde{V}(\bullet \,|\, \mathbf{y})$ upperbounds $V(\bullet)$ on X, in the following sense:

$$\widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) \geq V(\mathbf{x}) + c_y, \quad \forall \mathbf{x} \in X, \tag{3.19}$$

with $c_y \triangleq \widetilde{V}(\mathbf{y} \,|\, \mathbf{y}) - V(\mathbf{y})$, where the equality is achieved when $\mathbf{x} = \mathbf{y}$. Assumption 2.13.3 is a derivative consistency condition: roughly speaking, it ensures that $\widetilde{V}(\bullet \,|\, \mathbf{x})$ has the same first order properties of $V(\bullet)$ at $\mathbf{x} \in X$.

The MM algorithm is summarized in Algorithm 1 and its convergence is stated in Theorem 2.14.

Theorem 2.14 *Let $\{\mathbf{x}^k\}_{k \in \mathbb{N}_+}$ be the sequence generated by Algorithm 1 under Assumptions 2.12 and 2.13. Then, every limit point of $\{\mathbf{x}^k\}_{k \in \mathbb{N}_+}$ (if exists) is a d-stationary solution of Problem (3.1).*

Algorithm 1 The majorization-minimization (MM) algorithm

```
Data : $\mathbf{x}^0 \in X$. Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Update $\mathbf{x}$ as
        $\mathbf{x}^{k+1} \in \operatorname*{argmin}_{\mathbf{x} \in X} \widetilde{V}(\mathbf{x} \,|\, \mathbf{x}^k);$   (3.20)
(S.3) : $k \leftarrow k + 1$, and go to (S.1).
```

Proof The main properties of Algorithm 1 is to generate a nonincreasing sequence $\{V(\mathbf{x}^k)\}$. Indeed, we have

$$V(\mathbf{x}^{k+1}) \overset{(3.19)}{\leq} \widetilde{V}(\mathbf{x}^{k+1} \,|\, \mathbf{x}^k) - c^k \overset{(3.20)}{\leq} \widetilde{V}(\mathbf{x}^k \,|\, \mathbf{x}^k) - c^k \overset{(a)}{=} V(\mathbf{x}^k), \tag{3.21}$$

where $c^k \triangleq \widetilde{V}(\mathbf{x}^k \,|\, \mathbf{x}^k) - V(\mathbf{x}^k)$, and (a) follows from the fact that the inequality (3.19) is achieved with equality at $\mathbf{x} = \mathbf{x}^k$.

Let $\mathbf{x}^*$ be a limit point of $\{\mathbf{x}^k\}_{k \in \mathbb{N}_+}$, that is, $\lim_{t \to \infty} \mathbf{x}^{k_t} = \mathbf{x}^* \in X$. We have

$$\begin{aligned} \widetilde{V}(\mathbf{x}^{k_{t+1}} \,|\, \mathbf{x}^{k_{t+1}}) - c^{k_{t+1}} &= V(\mathbf{x}^{k_{t+1}}) \overset{(3.21)}{\leq} V(\mathbf{x}^{k_t+1}) \\ &\leq \widetilde{V}(\mathbf{x}^{k_t+1} \,|\, \mathbf{x}^{k_t}) - c^{k_t} \leq \widetilde{V}(\mathbf{x} \,|\, \mathbf{x}^{k_t}) - c^{k_t}, \end{aligned}$$

for all $\mathbf{x} \in X$. Let $t \to +\infty$; invoking the continuity of $\widetilde{V}(\bullet \,|\, \bullet)$ and $V(\bullet)$, we have that the sequence $\{c^{k_t}\}_{t \in \mathbb{N}_+}$ converges (to a finite value). Therefore,

$$\widetilde{V}\left(\mathbf{x}^* \,|\, \mathbf{x}^*\right) \leq \widetilde{V}\left(\mathbf{x} \,|\, \mathbf{x}^*\right), \quad \forall \mathbf{x} \in X, \tag{3.22}$$

which implies

$$0 \leq \widetilde{V}'\left(\mathbf{x}^*; \mathbf{d} \,|\, \mathbf{x}^*\right) \overset{(a)}{=} V'\left(\mathbf{x}^*; \mathbf{d}\right), \quad \forall\, \mathbf{d} \in \mathbb{R}^m \quad \text{such that} \quad \mathbf{x}^* + \mathbf{d} \in X, \tag{3.23}$$

where (a) follows from Assumption 2.13.3. This shows that $\mathbf{x}^*$ is a d-stationary solution of Problem (3.1). □

Note that, since the sequence $\{V(\mathbf{x}^k)\}$ is nonincreasing, a sufficient condition for $\{\mathbf{x}^k\}$ to admit a limit point is that the set $\{\mathbf{x} \in X \;:\; V(\mathbf{x}) \leq V(\mathbf{x}^0)\}$ is compact. A sufficient condition for that is the coercivity of V on X.

On the Termination Criterion We briefly discuss how to choose the termination criterion in Step 2 of Algorithm 1; we refer the interested reader to [241] for more details. Let $M : X \to X$ be a map such that $M(\mathbf{x}^k) \in \operatorname{argmin}_{\mathbf{x} \in X} \widetilde{V}(\mathbf{x} \,|\, \mathbf{x}^k)$ and $\mathbf{x}^{k+1} = M(\mathbf{x}^k)$ [cf. (3.2)]. In words, among all the global minimizers of $\widetilde{V}(\bullet \,|\, \mathbf{x}^k)$ on X, M uniquely selects the one, $\mathbf{x}^{k+1}$, used in Step 2 of the MM algorithm. We show next that $V(M(\mathbf{x})) = V(\mathbf{x})$ is a sufficient condition of $\mathbf{x}$ being a d-stationary solution of Problem (3.1). It follows from (3.21) that $V(M(\mathbf{x})) = V(\mathbf{x})$ forces $\widetilde{V}(M(\mathbf{x}) \,|\, \mathbf{x}) = \widetilde{V}(\mathbf{x} \,|\, \mathbf{x})$, which implies that $\mathbf{x}$ is a minimizer of $\widetilde{V}(\bullet \,|\, \mathbf{x})$ on X, and thus (by Assumption 2.13.3) a d-stationary point of (3.1). Based on the above observation and assuming that $M(\bullet)$ is *continuous*, the following is a valid merit function to measure distance from stationarity of the iterate $\mathbf{x}^k$:

$$J^{k+1} \triangleq \frac{V(\mathbf{x}^k) - V(\mathbf{x}^{k+1})}{\max(1, |V(\mathbf{x}^k)|)}. \tag{3.24}$$

The continuity assumption of M maybe hard to check directly. A stronger condition implying continuity of M is that the minimizer of $\widetilde{V}(\bullet \,|\, \mathbf{x}^k)$ over X is unique, for all $\mathbf{x}^k \in X$ [241, Lemma 1]. Note that, in such a case, it is not difficult to check that, if $\mathbf{x}$ is a fixed-point of M, then it must be a d-stationary point of Problem (3.1). Therefore, in the aforementioned setting, an alternative merit function is

$$J^{k+1} \triangleq \|\mathbf{x}^{k+1} - \mathbf{x}^k\|. \tag{3.25}$$

Other termination criteria are discussed in Lecture II (cf. Sect. 3.3) for SCA-based algorithms.

3.2.2.1 Discussion on Algorithm 1

The following comments on Algorithm 1 are in order.

On the Choice of Surrogate Function

The successful application of the MM algorithms relies on the possibility of finding a valid surrogate function $\widetilde{V}$. The critical assumption to be satisfied is undoubtedly the upperbound condition, as stated in Assumption 2.13.2. We provide next some systematic rules which help to build a surrogate function that meets this condition (and the other required assumptions); several illustrating examples are also discussed. Although for specific (structured) problems it is possible to find a nonconvex surrogate function whose minimizer can be computed efficiently, a convex surrogate is in general preferred, since it leads to a convex subproblem (3.2). Therefore, next we mainly focus on convex surrogates.

1) First Order Taylor Expansion Suppose V is a differentiable *concave* function on X. A natural choice for $\widetilde{V}$ satisfying Assumption 2.13 is then: given $\mathbf{y} \in X$,

$$\widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) = V(\mathbf{y}) + \nabla V(\mathbf{y})^T\,(\mathbf{x} - \mathbf{y}). \tag{3.26}$$

More generally, $\widetilde{V}$ can be chosen as any convex differentiable function on X, say $\widetilde{V}_{\mathrm{CVX}}$, satisfying the gradient consistency condition $\nabla \widetilde{V}_{\mathrm{CVX}}(\mathbf{x} \,|\, \mathbf{x}) = \nabla V(\mathbf{x})$, for all $\mathbf{x} \in X$; this is enough for Assumption 2.13.2 (and thus Assumption 2.13) to be satisfied, as shown by the following chain of inequalities

$$V(\mathbf{x}) - V(\mathbf{y}) \overset{(a)}{\leq} \nabla V(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \overset{(b)}{=} \nabla \widetilde{V}_{\text{cvx}}(\mathbf{y}\,|\,\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \overset{(c)}{\leq} \widetilde{V}_{\text{cvx}}(\mathbf{x}\,|\,\mathbf{y}) - \widetilde{V}_{\text{cvx}}(\mathbf{y}\,|\,\mathbf{y}), \tag{3.27}$$

where (a) follows from the concavity of V; in (b) we used $\nabla \widetilde{V}_{\text{cvx}}(\mathbf{y}\,|\,\mathbf{y}) = \nabla V(\mathbf{y})$; and (c) follows from the convexity of $\widetilde{V}_{\text{cvx}}$. Equation (3.27) shows that (3.19) (thus Assumption 2.13.2) is satisfied by such a $\widetilde{V}_{\text{cvx}}$.

Example 1 $V(x) = \log(x)$ is concave on $(0, +\infty]$. Hence, it can be majorized by $\widetilde{V}(x\,|\,y) = x/y$, for any given $y > 0$.

Example 2 $V(x) = |x|^p$, with $p \in (0, 1)$, is concave on $(-\infty, 0)$ and $(0, +\infty)$. It thus can be majorized by the quadratic function $\widetilde{V}(x\,|\,y) = \frac{p}{2}\,|y|^{p-2}\,x^2$, for any given $y \neq 0$.

2) Second Order Taylor Expansion Suppose V is C^1, with L-Lipschitz gradient on X. Then V can be majorized by the surrogate function: given $\mathbf{y} \in X$,

$$\widetilde{V}(\mathbf{x}\,|\,\mathbf{y}) = V(\mathbf{y}) + \nabla V(\mathbf{y})^T\,(\mathbf{x}-\mathbf{y}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2. \tag{3.28}$$

Moreover, if V is twice differentiable and there exists a matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ such that $\mathbf{M} - \nabla^2 V(\mathbf{x}) \succeq \mathbf{0}$, for all $\mathbf{x} \in X$, then V can be majorized by the following valid surrogate function

$$\widetilde{V}(\mathbf{x}\,|\,\mathbf{y}) = V(\mathbf{y}) + \nabla V(\mathbf{y})^T\,(\mathbf{x}-\mathbf{y}) + \frac{1}{2}(\mathbf{x}-\mathbf{y})^T\mathbf{M}\,(\mathbf{x}-\mathbf{y}). \tag{3.29}$$

Example 3 (The Proximal Gradient Algorithm) Suppose that V admits the structure $V = F + G$, where $F : \mathbb{R}^m \to \mathbb{R}$ is C^1, with L-Lipschitz gradient on X, and $G : \mathbb{R}^m \to \mathbb{R}$ is convex (possibly nonsmooth) on X. Using (3.28) to majorize F, a valid surrogate for V is: given $\mathbf{y} \in X$,

$$\widetilde{V}(\mathbf{x}\,|\,\mathbf{y}) = F(\mathbf{x}^k) + \nabla F(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2 + G(\mathbf{x}). \tag{3.30}$$

Quite interestingly, the above choice leads to a strongly convex subproblem (3.2), whose minimizer has the following closed form:

$$\mathbf{x}^{k+1} = \text{prox}_{1/L,\,G}\left(\mathbf{x}^k - \frac{1}{L}\nabla F(\mathbf{x}^k)\right), \tag{3.31}$$

where $\text{prox}_{\gamma,G}(\bullet)$ is the proximal response, defined as

$$\text{prox}_{\gamma,\,G}(\mathbf{x}) \triangleq \underset{\mathbf{z}}{\text{argmin}}\left\{G(\mathbf{z}) + \frac{1}{2\,\gamma}\;\|\mathbf{z}-\mathbf{x}\|^2\right\}.$$

The resulting MM algorithm (Algorithm 1) turns out to be the renowned *proximal gradient algorithm*, with step-size $\gamma = 1/L$.

3) Pointwise Maximum Suppose $V : \mathbb{R}^m \to \mathbb{R}$ can be written as the pointwise maximum of functions $\{f_i\}_{i=1}^I$, i.e.,

$$V(\mathbf{x}) \triangleq \max_{i=1,\ldots,I} f_i(\mathbf{x}),$$

where each $f_i : \mathbb{R}^m \to \mathbb{R}$ satisfies Assumption 2.12.2 and 2.12.3. Then V can be majorized by

$$\widetilde{V}(\mathbf{x}\,|\,\mathbf{y}) = \max_{i=1,\ldots,I} \widetilde{f}_i(\mathbf{x}\,|\,\mathbf{y}), \tag{3.32}$$

for any given $\mathbf{y} \in X$, where $\widetilde{f}_i : X \times X \to \mathbb{R}$ is a surrogate function of f_i satisfying Assumption 2.13 and $\widetilde{f}_i(\mathbf{y}\,|\,\mathbf{y}) = f_i(\mathbf{y})$. It is not difficult to verify that $\widetilde{V}$ above satisfies Assumption 2.13. Indeed, the continuity of $\widetilde{V}$ follows from that of $\widetilde{f}_i$. Moreover, we have $\widetilde{V}(\mathbf{y}\,|\,\mathbf{y}) = V(\mathbf{y})$. Finally, condition 2.13.3 is a direct consequence of Lemma 2.8:

$$V'(\mathbf{x};\mathbf{d}) \overset{(3.13)}{=} \max_{i\in A(\mathbf{x})} f_i'(\mathbf{x};\mathbf{d}) = \max_{i\in A(\mathbf{x})} \widetilde{f}_i'(\mathbf{x};\mathbf{d}\,|\,\mathbf{x}) = \widetilde{V}'(\mathbf{x};\mathbf{d}\,|\,\mathbf{x}), \tag{3.33}$$

where $A(\mathbf{x}) = \{i : f_i(\mathbf{x}) = V(\mathbf{x})\} = \{i : \widetilde{f}_i(\mathbf{x}\,|\,\mathbf{x}) = \widetilde{V}(\mathbf{x})\}$.

4) Composition by a Convex Function Suppose $V : \mathbb{R}^m \to \mathbb{R}$ can be expressed as $V(\mathbf{x}) \triangleq f\left(\sum_{i=1}^n \mathbf{A}_i\mathbf{x}\right)$, where $f : \mathbb{R}^m \to \mathbb{R}$ is a convex function and $\mathbf{A}_i \in \mathbb{R}^{m\times m}$ are given matrices. Then, one can construct a surrogate function of V leveraging the following inequality due to convexity of f:

$$f\left(\sum_{i=1}^n w_i\mathbf{x}_i\right) \le \sum_{i=1}^n w_i f(\mathbf{x}_i) \tag{3.34}$$

for all $\sum_{i=1}^n w_i = 1$ and each $w_i > 0$. Specifically, rewrite first V as: given $\mathbf{y} \in \mathbb{R}^m$,

$$V(\mathbf{x}) = f\left(\sum_{i=1}^n \mathbf{A}_i\mathbf{x}\right) = f\left(\sum_{i=1}^n w_i\left(\frac{\mathbf{A}_i(\mathbf{x}-\mathbf{y})}{w_i} + \sum_{i=1}^n \mathbf{A}_i\mathbf{y}\right)\right).$$

Then, using (3.34) we can upperbound V as

$$V(\mathbf{x}) \le \widetilde{V}(\mathbf{x}|\mathbf{y}) = \sum_{i=1}^n w_i\, f\left(\frac{\mathbf{A}_i(\mathbf{x}-\mathbf{y})}{w_i} + \sum_{i=1}^n \mathbf{A}_i\,\mathbf{y}\right). \tag{3.35}$$

It is not difficult to check that $\widetilde{V}$ satisfies Assumption 2.13. Equation (3.35) is particularly useful to construct surrogate functions that are additively separable in the (block) variables, which opens the way to parallel solution methods wherein (blocks of) variables are updated in parallel. Two examples are discussed next.

Example 4 Let $V : \mathbb{R}^m \to \mathbb{R}$ be convex and let $\mathbf{x} \in \mathbb{R}^m$ be partitioned as $\mathbf{x} = (\mathbf{x}_i)_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^{m_i}$ and $\sum_{i=1}^n m_i = m$. Let us rewrite $\mathbf{x}$ in terms of its block as $\mathbf{x} = \sum_{i=1}^n \mathbf{A}_i \mathbf{x}$, where $\mathbf{A}_i \in \mathbb{R}^{m \times m}$ is the block diagonal matrix such that $\mathbf{A}_i \mathbf{x} = (\mathbf{x})_i$, with $(\mathbf{x})_i \triangleq [\mathbf{0}^T, \ldots, \mathbf{0}^T, \mathbf{x}_i^T, \mathbf{0}^T, \ldots, \mathbf{0}^T]^T$ denoting the operator nulling all the blocks of $\mathbf{x}$ except the i-th one. Then using (3.35) one can choose the following surrogate function: given $\mathbf{y} = (\mathbf{y}_i)_{i=1}^n$, with each $\mathbf{y}_i \in \mathbb{R}^{m_i}$,

$$V(\mathbf{x}) = V\left(\sum_{i=1}^n \mathbf{A}_i \mathbf{x}\right) \leq \widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) \triangleq \sum_{i=1}^n w_i \, V\left(\frac{1}{w_i}\left((\mathbf{x})_i - (\mathbf{y})_i\right) + \mathbf{y}\right).$$

It is easy to check that such a $\widetilde{V}$ is separable in the blocks $\mathbf{x}_i$'s.

Example 5 Let $V : \mathbb{R} \to \mathbb{R}$ be convex and let vectors $\mathbf{x}$, $\mathbf{a} \in \mathbb{R}^m$ be partitioned as $\mathbf{x} = (\mathbf{x}_i)_{i=1}^n$ and $\mathbf{a} = (\mathbf{a}_i)_{i=1}^n$, respectively, with $\mathbf{x}_i$ and $\mathbf{a}_i$ having the same size. Then, invoking (3.35), a valid surrogate function of the composite function $V(\mathbf{a}^T\mathbf{x})$ is: given $\mathbf{y} = (\mathbf{y}_i)_{i=1}^n$, partitioned according to $\mathbf{x}$,

$$\begin{aligned} V(\mathbf{a}^T\mathbf{x}) = V\left(\sum_{i=1}^n (\mathbf{a})_i^T \mathbf{x}\right) \leq \widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) &\triangleq \sum_{i=1}^n w_i \, V\left(\frac{(\mathbf{a})_i^T(\mathbf{x} - \mathbf{y})}{w_i} + \sum_{i=1}^n (\mathbf{a})_i^T \mathbf{y}\right) \\ &= \sum_{i=1}^n w_i \, V\left(\frac{\mathbf{a}_i^T(\mathbf{x}_i - \mathbf{y}_i)}{w_i} + \mathbf{a}^T\mathbf{y}\right). \end{aligned} \tag{3.36}$$

This is another example of additively (block) separable surrogate function.

5) Surrogates Based on Special Inequalities Other techniques often used to construct valid surrogate functions leverage specific inequalities, such as the Jensen's inequality, the arithmetic-geometric-mean inequality, and the Cauchy-Schwartz inequality. The way of using these inequalities, however, depends on the specific expression of the objective function V under consideration; generalizing these approaches to arbitrary V's seems not possible. We provide next two illustrative (nontrivial) case studies based on the Jensen's inequality and the arithmetic-geometric-mean inequality while we refer the interested reader to [230] for more examples building on this approach.

Example 6 (The Expectation-Maximization Algorithm) Given a pair of random (vector) variables $(\mathbf{s}, \mathbf{z})$ whose joint probability distribution $p(\mathbf{s}, \mathbf{z}|\mathbf{x})$ is parametrized by $\mathbf{x}$, we consider the maximum likelihood estimation problem of estimating $\mathbf{x}$ *only* from $\mathbf{s}$ while the random variable $\mathbf{z}$ is unobserved/hidden. The problem is formulated as

$$\hat{\mathbf{x}}_{\text{ML}} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \left\{ V(\mathbf{x}) \triangleq -\log p(\mathbf{s}|\mathbf{x}) \right\}, \tag{3.37}$$

where $p(\mathbf{s}|\mathbf{x})$ is the (conditional) marginal distribution of $\mathbf{s}$.

In general, the expression of $p\,(\mathbf{s}|\mathbf{x})$ is not available in closed form; moreover numerical evaluations of the integration of $p(\mathbf{s}, \mathbf{z}|\mathbf{x})$ with respect to $\mathbf{z}$ can be computationally too costly, especially if the dimension of $\mathbf{z}$ is large. In the following we show how to attach Problem (3.37) using the MM framework. We build next a valid surrogate function for V leading to a simpler optimization problem to solve.

Specifically, we can rewrite V as

$$\begin{aligned}
V(\mathbf{x}) &= -\log p\,(\mathbf{s}|\mathbf{x}) \\
&= -\log \int p(\mathbf{s}|\mathbf{z}, \mathbf{x})\, p(\mathbf{z}|\mathbf{x}) \mathrm{d}\mathbf{z} \\
&= -\log \int \left(\frac{p(\mathbf{s}|\mathbf{z}, \mathbf{x})\, p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k)}{p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k)} \right) p(\mathbf{z}|\mathbf{x}) \mathrm{d}\mathbf{z} \\
&= -\log \int \left(\frac{p(\mathbf{s}|\mathbf{z}, \mathbf{x})\, p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k)} \right) p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k) \mathrm{d}\mathbf{z} \\
&\overset{(a)}{\leq} -\int \log \left(\frac{p(\mathbf{s}|\mathbf{z}, \mathbf{x})\, p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k)} \right) p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k) \mathrm{d}\mathbf{z} \\
&= -\int \log\left(p(\mathbf{s}, \mathbf{z}|\mathbf{x})\right)\, p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k) \mathrm{d}\mathbf{z} + \underbrace{\int \log\left(p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k)\right) p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k) \mathrm{d}\mathbf{z}}_{\text{constant}},
\end{aligned}$$

where (a) follows from the Jensen's inequality. This naturally suggests the following surrogate function of V: given $\mathbf{y}$,

$$\widetilde{V}(\mathbf{x} \mid \mathbf{y}) = -\int \log\left(p(\mathbf{s}, \mathbf{z}|\mathbf{x})\right)\, p(\mathbf{z}|\mathbf{s}, \mathbf{y}) \mathrm{d}\mathbf{z}. \tag{3.38}$$

In fact, it is not difficult to check that such a $\widetilde{V}$ satisfies Assumption 2.13.

The update of $\mathbf{x}$ resulting from the MM algorithm then reads

$$\mathbf{x}^{k+1} \in \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ -\int \log\left(p(\mathbf{s}, \mathbf{z}|\mathbf{x})\right)\, p(\mathbf{z}|\mathbf{s}, \mathbf{x}^k) \mathrm{d}\mathbf{z} \right\}. \tag{3.39}$$

Problem (3.39) can be efficiently solved for specific probabilistic models, including those belonging to the exponential family, the Gaussian/multinomial mixture model, and the linear Gaussian latent model.

Quite interestingly, the resulting MM algorithm [Algorithm 1 based on the update (3.39)] turns out to be the renowned *Expectation-Maximization* (EM) algorithm

[66]. The EM algorithm starts from an initial estimate $\mathbf{x}^0$ and generates a sequence $\mathbf{x}^k$ by repeating the following two steps:

- *E-step*: Calculate $\widetilde{V}(\mathbf{x} \,|\, \mathbf{x}^k)$ as in (3.38);
- *M-step*: Update $\mathbf{x}^{k+1}$ as $\mathbf{x}^{k+1} \in \operatorname{argmax}_{\mathbf{x}} -\widetilde{V}(\mathbf{x} \,|\, \mathbf{x}^k)$.

Note that these two steps correspond exactly to the majorization and minimization steps (3.38) and (3.39), respectively, showing that the EM algorithm belongs to the family of MM schemes, based on the surrogate function (3.38).

Example 7 (Geometric Programming) Consider the problem of minimizing a signomial

$$V(\mathbf{x}) = \sum_{j=1}^{J} c_j \prod_{i=1}^{n} x_i^{\alpha_{ij}}$$

on the nonnegative orthant $\mathbb{R}^n_+$, with $c_j, \alpha_{ij} \in \mathbb{R}$. In the following, we assume that V is coercive on $\mathbb{R}^n_+$. A sufficient condition for this to hold is that for all $i = 1, \ldots, n$ there exists at least one j such that $c_j > 0$ and $\alpha_{ij} > 0$, and at least one j such that $c_j > 0$ and $\alpha_{ij} < 0$ [136].

We construct a separable surrogate function of V at given $\mathbf{y} \in \mathbb{R}^n_{++}$. We first derive an upperbound for the summand in V with $c_j > 0$ and a lowerbound for those with $c_j < 0$, using the arithmetic-geometric mean inequality and the concavity of the log function (cf. Example 1), respectively.

Let z_i and α_i be nonnegative scalars, the arithmetic-geometric mean inequality reads

$$\prod_{i=1}^{n} z_i^{\alpha_i} \leq \sum_{i=1}^{n} \frac{\alpha_i}{\|\boldsymbol{\alpha}\|_1} z_i^{\|\boldsymbol{\alpha}\|_1}. \tag{3.40}$$

Since $y_i > 0$ for all $i = 1, \ldots, n$, let $z_i = x_i/y_i$ for $\alpha_i > 0$ and $z_i = y_i/x_i$ for $\alpha_i < 0$. Then (3.40) implies that the monomial $\prod_{i=1}^n x_i^{\alpha_i}$ can be upperbounded on $\mathbb{R}^n_{++}$ as

$$\prod_{i=1}^{n} x_i^{\alpha_i} \leq \left(\prod_{i=1}^{n} (y_i)^{\alpha_i}\right) \sum_{i=1}^{n} \frac{|\alpha_i|}{\|\boldsymbol{\alpha}\|_1} \left(\frac{x_i}{y_i}\right)^{\|\boldsymbol{\alpha}\|_1 \operatorname{sign}(\alpha_i)}. \tag{3.41}$$

To upperbound the terms in V with negative c_j on $\mathbb{R}^n_{++}$, which is equivalent to find a lowerbound of $\prod_{i=1}^n x_i^{\alpha_i}$, we use the bound introduced in Example 1, with $x = \prod_{i=1}^n x_i^{\alpha_i}$, which yields

$$\log\left(\prod_{i=1}^{n} x_i^{\alpha_i}\right) \leq \log\left(\prod_{i=1}^{n} (y_i)^{\alpha_i}\right) + \left(\prod_{i=1}^{n} (y_i)^{\alpha_i}\right)^{-1} \left(\prod_{i=1}^{n} x_i^{\alpha_i} - \prod_{i=1}^{n} (y_i)^{\alpha_i}\right).$$

Rearranging the terms we have

$$\prod_{i=1}^{n} x_i^{\alpha_i} \geq \prod_{i=1}^{n} (y_i)^{\alpha_i} \left(1 + \sum_{i=1}^{n} \alpha_i \log x_i - \sum_{i=1}^{n} \alpha_i \log y_i\right). \tag{3.42}$$

Combining (3.41) and (3.42) leads to the separable surrogate function $\widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) \triangleq \sum_{i=1}^{n} \widetilde{V}_i(x_i \,|\, \mathbf{y})$ of V, with

$$\begin{aligned}\widetilde{V}_i(x_i \,|\, \mathbf{y}) \triangleq \sum_{j:c_j>0} c_j \left(\prod_{\ell=1}^{n} (y_\ell)^{\alpha_{\ell j}}\right) \frac{|\alpha_{ij}|}{\|\boldsymbol{\alpha}_j\|_1} \left(\frac{x_i}{y_i}\right)^{\|\boldsymbol{\alpha}_j\|_1 \operatorname{sign}(\alpha_{ij})} \\ + \sum_{j:c_j<0} c_j \left(\prod_{\ell=1}^{n} (y_\ell)^{\alpha_{\ell j}}\right) \alpha_{ij} \log x_i.\end{aligned} \tag{3.43}$$

Function $\widetilde{V}(\bullet \,|\, \mathbf{y})$ is coercive since it is continuous and upperbounds V on $\mathbb{R}^n_{++}$, therefore $\mathbf{x}^{k+1} \in \mathbb{R}^n_{++}$ (recall that we assumed $\mathbf{x}^k \in \mathbb{R}^n_{++}$). Moreover, the following chain of inequalities holds:

$$V(\mathbf{x}^{k+1}) \leq \widetilde{V}(\mathbf{x}^{k+1} \,|\, \mathbf{x}^k) \leq \widetilde{V}(\mathbf{x}^k \,|\, \mathbf{x}^k) = V(\mathbf{x}^k) \leq \cdots \leq V(\mathbf{x}^0) < +\infty, \tag{3.44}$$

which implies that, given $\mathbf{x}^0 \in \mathbb{R}_{++}$, the sequence $\{\mathbf{x}^k\}$ generated by the MM algorithm always stays in the compact set $X^0 \triangleq \{\mathbf{x} \,|\, V(\mathbf{x}) \leq V(\mathbf{x}^0)\} \subseteq \mathbb{R}^n_{++}$.

As $\widetilde{V}(\mathbf{x} \,|\, \mathbf{x}^k)$ is separable, $\mathbf{x}^{k+1}$ thus can be computed in parallel. In addition, focusing on a single component x_i and performing the change of variable $x_i = e^{z_i}$, it is not difficult to check that $\widetilde{V}_i$ is convex in z_i; its global minimizer can be then computed efficiently by solving a one dimensional convex optimization problem.

Parallel Implementation

When dealing with large-scale optimization instances of Problem (3.1), solving subproblem (3.2) with respect to the entire $\mathbf{x}$ might be either computationally too costly or practically infeasible. When the feasible set X admits a Cartesian product structure, i.e., $X = X_1 \times \cdots \times X_n$, with $X_i \subseteq \mathbb{R}^{m_i}$, a natural approach to cope with the curse of dimensionally is partitioning the vector of variables $\mathbf{x}$ into n blocks according to X, i.e., $\mathbf{x} = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T\right]^T$ and each $\mathbf{x}_i \in \mathbb{R}^{m_i}$, and leveraging a multi-core computing environment to optimize the blocks *in parallel*. When using the MM algorithm, a natural way to achieve this goal is to construct a surrogate function $\widetilde{V}(\bullet \,|\, \mathbf{y})$ that is *additively separable* in the blocks, i.e., $\widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) = \sum_{i=1}^{n} \widetilde{V}_i(\mathbf{x}_i \,|\, \mathbf{y})$, so that the subproblem (3.2) can be decoupled in independent optimization problems, one per block. Some of the examples discussed above [cf. (3.26), (3.28), (3.35)] show cases where this can be readily achieved by exploring the special structure of the objective function V. However, when the objective function V does not enjoy any special structure, it seems that there is no general procedure to leverage to obtain a separable surrogate function satisfying Assumption 2.13. In such cases, a way to cope with the curse of dimensionality is to solve the subproblem (3.2) by minimizing the (nonseparable) surrogate function block-wise. Such a version of the MM algorithm, termed *Block-MM*, is discussed in details in the next section.

3.2.3 The Block Majorization-Minimization Algorithm

In this section, we introduce the Block MM (BMM) algorithm solving Problem (3.1), whose feasible set is now assumed to have a Cartesian product structure.

Assumption 2.15 *The feasible set X of Problem (3.1) admits the Cartesian product structure $X = X_1 \times \cdots \times X_n$, with each $X_i \in \mathbb{R}^{m_i}$ and $\sum_{i=1}^{n} m_i = m$.*

In the above setting, the optimization variables of (3.1) can be partitioned in (nonoverlapping) blocks $\mathbf{x}_i$. When judiciously exploited, this block-structure can lead to low-complexity algorithms that are implementable in a parallel/distributed manner. The BMM algorithm is a variant of the MM scheme (cf. Algorithm 1) whereby *only one* block of variables is updated at a time. More specifically, at iteration k, a block is selected according to some predetermined rule, say block i^k, and updated by solving the subproblem

$$\mathbf{x}_{i^k}^{k+1} \in \underset{\mathbf{x}_{i^k} \in X_{i^k}}{\operatorname{argmin}}\ \widetilde{V}_{i^k}\left(\mathbf{x}_{i^k} \,|\, \mathbf{x}^k\right), \tag{3.45}$$

whereas $\mathbf{x}_j^{k+1} = \mathbf{x}_j^k$, for all $j \neq i^k$. In (3.45), $\widetilde{V}_{i^k}(\bullet \,|\, \mathbf{x}^k)$ is a suitably chosen surrogate function of V, now function only of the block $\mathbf{x}_{i^k}$. Conditions on $\widetilde{V}_{i^k}$ are similar to those already introduced for the MM algorithm (cf. Assumption 2.13) and summarized below. We use the following notation: $O_i \subseteq \mathbb{R}^{m_i}$ is an open set containing X_i and $O = O_1 \times \cdots \times O_n$; $\mathbf{x}_{-i} \triangleq [\mathbf{x}_1^T, \ldots, \mathbf{x}_{i-1}^T, \mathbf{x}_{i+1}^T, \ldots, \mathbf{x}_n^T]^T$ denotes the vector containing all the blocks of $\mathbf{x}$ but the i-th one; $(\mathbf{x})_i \triangleq [\mathbf{0}^T, \ldots, \mathbf{0}^T, \mathbf{x}_i^T, \mathbf{0}^T, \ldots, \mathbf{0}^T]^T$ denotes the operator nulling all the blocks of $\mathbf{x}$ except for the i-th one; and with a slight abuse of notation, we use $(\mathbf{x}_i, \mathbf{y}_{-i})$ to denote the ordered tuple $(\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}, \mathbf{x}_i, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_n)$.

Assumption 2.16 *Each surrogate function $\widetilde{V}_i : O_i \times O \to \mathbb{R}$ satisfies the following conditions:*

1. *$\widetilde{V}_i(\bullet \,|\, \bullet)$ is continuous on $X_i \times X$;*
2. *$\mathbf{y}_i \in \operatorname{argmin}_{\mathbf{x}_i \in X_i} \widetilde{V}_i(\mathbf{x}_i \,|\, \mathbf{y}) - V(\mathbf{x}_i, \mathbf{y}_{-i})$, for all $\mathbf{y} \in X$;*
3. *The directional derivative of $\widetilde{V}_i'$ satisfies $\widetilde{V}_i'(\mathbf{x}_i; \mathbf{d}_i \,|\, \mathbf{x}) = V'\left(\mathbf{x}; (\mathbf{d})_i\right)$, for all $\mathbf{x}_i \in X_i$ and feasible directions $\mathbf{d}_i \in \mathbb{R}^{m_i}$ at $\mathbf{x}_i$.*

The next question is how to choose the blocks to be updated at each iteration, in order to have convergence. Any of the following rules can be adopted.

Assumption 2.17 *Blocks in (3.45) are chosen according to any of the following rules:*

1. ***Essentially cyclic rule**: There exits a finite positive T such that $\bigcup_{t=0}^{T-1}\{i^{k+t}\} = \{1,\ldots,n\}$, for all $k = 0, 1, \ldots$;*
2. ***Maximum block improvement rule**: Choose*

$$i^k \in \underset{i=1,\ldots,m}{\text{argmin}}\; V\left(\mathbf{x}_i^{k+1}, \mathbf{x}_{-i}^k\right);$$

3. ***Random-based rule**: There exists a $p_{\min} > 0$ such that $\mathbb{P}\left(i^k = j\,|\mathbf{x}^{k-1},\ldots,\mathbf{x}^0\right) = p_j^k \geq p_{\min}$, for all $j = 1,\ldots n$ and $k = 0, 1, \ldots$.*

The first two rules above are deterministic rules. Roughly speaking, the essentially cyclic rule (Assumption 2.17.1) states that all the blocks must be updated at least once within any T consecutive iterations, with some blocks possibly updated more frequently than others. Of course, a special case of this rule is the simple cyclic rule, i.e., $i^k = (k \bmod n) + 1$, whereby all the blocks are updated once every T iterations. The maximum block improvement rule (Assumption 2.17.2) is a greedy-based update: only the block that generates the largest decrease of the objective function V at $\mathbf{x}^k$ is selected and updated. Finally the random-based selection rule (Assumption 2.17.3) selects block randomly and independently, with the condition that all the blocks have a bounded away from zero probability to be chosen. The BMM algorithm is summarized in Algorithm 2.

Convergence results of Algorithm 2 consist of two major statements. Under the essential cyclic rule (Assumption 2.17.1), quasi convexity of the objective function is required [along with the uniqueness of the minimizer in (3.45)], which also guarantees the existence of the limit points. This is in the same spirit as the classical proof of convergence of Block Coordinate Descent methods; see, e.g., [14, 218, 250]. If the maximum block improvement rule (Assumption 2.17.2) or the random-based selection rule (Assumption 2.17.3) are used, then a stronger convergence result can be proved by relaxing the quasi-convexity assumption and

Algorithm 2 Block MM algorithm

```
Data : $\mathbf{x}^0 \in X$. Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Choose an index $i^k \in \{1,\ldots,n\}$;
(S.3) : Update $\mathbf{x}^k$ as
            – Set $\mathbf{x}_{i^k}^{k+1} \in \text{argmin}_{\mathbf{x}_{i^k}\in X_{i^k}} \widetilde{V}_{i^k}\left(\mathbf{x}_{i^k}\,|\,\mathbf{x}^k\right)$;
            – Set $\mathbf{x}_j^{k+1} = \mathbf{x}_j^k$, for all $j \neq i_k$;
(S.4) : $k \leftarrow k + 1$, and go to (S.1).
```

imposing the compactness of the level sets of V. In order to state the convergence result, we introduce the following additional assumptions.

Assumption 2.18 $\widetilde{V}_i\,(\bullet\,|\,\mathbf{y})$ *is quasi-convex and subproblem* (3.45) *has a unique solution, for all* $\mathbf{y} \in X$.

Assumption 2.19 *The level set* $X^0 \triangleq \{\mathbf{x} \in X \,:\, V(\mathbf{x}) \leq V\left(\mathbf{x}^0\right)\}$ *is compact and subproblem* (3.45) *has a unique solution for all* $\mathbf{x}^k \in X$ *and at least* $n-1$ *blocks.*

We are now ready to state the main convergence result of the BMM algorithm (Algorithm 2), as given in Theorem 2.20 below (the statement holds almost surely for the random-based selection rule). The proof of this theorem can be found in [194]; see also Sect. 3.2.5 for a detailed discussion on (other) existing convergence results.

Theorem 2.20 *Given Problem* (3.1) *under Assumptions 2.12 and 2.15, let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 2, with* $\widetilde{V}$ *chosen according to Assumption 2.16. Suppose that, in addition, either one of the following two conditions is satisfied:*

(a) i^k *is chosen according to the essentially cyclic rule (Assumption 2.17.1) and* $\widetilde{V}$ *further satisfies either Assumption 2.18 or 2.19;*

(b) i^k *is chosen according to the maximum block improvement rule (Assumption 2.17.2) or the random-based selection rule (Assumption 2.17.3).*

Then, every limit point of $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *is a coordinate-wise d-stationary solution of* (3.1)*. Furthermore, if* V *is regular, then every limit point is a d-stationary solution of* (3.1)*.* □

3.2.4 Applications

In this section, we show how to apply the (B)MM algorithm to solve some representative nonconvex problems arising from applications in signal processing, data analysis, and communications. More specifically, we consider the following problems: (1) Sparse least squares; (2) Nonnegative least squares; (3) Matrix factorization, including low-rank factorization and dictionary learning; and (4) the multicast beamforming problem. Our list is by no means exhaustive; it just gives a flavor of the kind of structured nonconvexity and applications which (B)MM can be successfully applied to.

3.2.4.1 Nonconvex Sparse Least Squares

Retrieving a sparse signal from its linear measurements is a fundamental problem in machine learning, signal processing, bioinformatics, physics, etc.; see [269] and [4, 99] for a recent overview and some books, respectively, on the subject. Consider a linear model, $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where $\mathbf{x} \in \mathbb{R}^m$ is the sparse signal to estimate, $\mathbf{z} \in \mathbb{R}^q$ is the vector of available measurements, $\mathbf{A} \in \mathbb{R}^{q \times m}$ is the given measurement matrix, and $\mathbf{n} \in \mathbb{R}^q$ is the observation noise. To estimate the sparse signal $\mathbf{x}$, a mainstream approach in the literature is to solve the following optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad V(\mathbf{x}) \triangleq \|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2 + \lambda G(\mathbf{x}), \tag{3.46}$$

where the first term in the objective function measures the model fitness whereas the regularization G is used to promote sparsity in the solution, and the regularization parameter $\lambda \geq 0$ is chosen to balance the trade-off between the model fitness and sparsity of the solution.

The ideal choice for G would be the cardinality of $\mathbf{x}$, also referred to as ℓ_0 "norm" of $\mathbf{x}$. However, its combinatorial nature makes the resulting optimization problem numerically intractable as the variable dimension m becomes large. Due to its favorable theoretical guarantees (under some regularity conditions on $\mathbf{A}$ [32, 33]) and the existence of efficient solution methods for convex instances of (3.46), the ℓ_1 norm has been widely adopted in the literature as convex surrogate G of the ℓ_0 function (in fact, the ℓ_1 norm is the convex envelop of the ℓ_0 function on $[-1, 1]^m$) [30, 235]. Yet there is increasing evidences supporting the use of nonconvex formulations to enhance the sparsity of the solution as well as the realism of the models [2, 36, 96, 154, 224]. For instance, it is well documented that *nonconvex* surrogates of the ℓ_0 function, such as the SCAD [82], the "transformed" ℓ_1, the logarithmic, the exponential, and the ℓ_p penalty [268], outperform the ℓ_1 norm in enhancing the sparsity of the solution. Table 3.1 summarizes these nonconvex surrogates whereas Fig. 3.5 shows their graph.

Quite interestingly, it has been recently shown that the aforementioned nonconvex surrogates of the ℓ_0 function enjoy a separable DC (Difference of Convex) structure (see, e.g., [2, 143] and references therein); specifically, we have the following

$$G(\mathbf{x}) = \sum_{i=1}^{m} g(x_i), \quad \text{with} \quad g\,(x_i) = \underbrace{\eta\,(\theta)\,|x_i|}_{\triangleq g^+(x_i)} - \underbrace{(\eta\,(\theta)\,|x_i| - g\,(x_i))}_{\triangleq g^-(x_i)}, \tag{3.47}$$

Table 3.1 Examples of nonconvex surrogates of the ℓ_0 function having a DC structure [cf. (3.47)]

Penalty function	Expression
Exp [28]	$g_{\exp}(x) = 1 - e^{-\theta\|x\|}$
$\ell_p (0 < p < 1)$ [90]	$g_{\ell_p^+}(x) = (\|x\| + \epsilon)^{1/\theta}$
$\ell_p (p < 0)$ [193]	$g_{\ell_p^-}(x) = 1 - (\theta\|x\| + 1)^p$
SCAD [82]	$g_{\text{scad}}(x) = \begin{cases} \frac{2\theta}{a+1}\|x\|, & 0 \leq \|x\| \leq \frac{1}{\theta} \\ \frac{-\theta^2\|x\|^2 + 2a\theta\|x\| - 1}{a^2-1}, & \frac{1}{\theta} < \|x\| \leq \frac{a}{\theta} \\ 1, & \|x\| > \frac{a}{\theta} \end{cases}$
Log [248]	$g_{\log}(x) = \frac{\log(1+\theta\|x\|)}{\log(1+\theta)}$

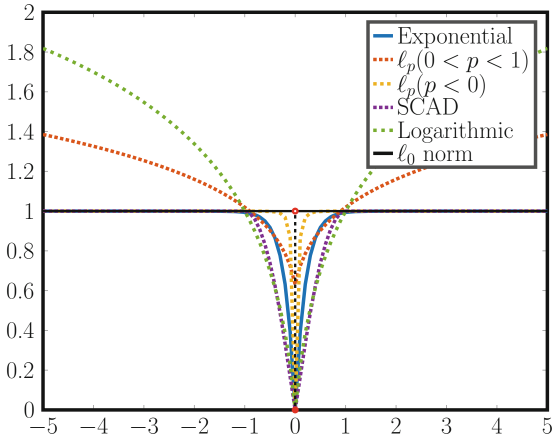


Fig. 3.5 Nonconvex surrogate functions of the ℓ_0 function given in Table 3.1

Table 3.2 Explicit expression of $\eta(\theta)$ and dg^-/dx [cf. (3.47)]

g	$\eta(\theta)$	dg_θ^-/dx
$g_{\exp}$	θ	$\text{sign}(x) \cdot \theta \cdot (1 - e^{-\theta\lvert x\rvert})$
$g_{\ell_p^+}$	$\frac{1}{\theta}\epsilon^{1/\theta - 1}$	$\frac{1}{\theta}\,\text{sign}(x) \cdot [\epsilon^{\frac{1}{\theta}-1} - (\lvert x\rvert + \epsilon)^{\frac{1}{\theta}-1}]$
$g_{\ell_p^-}$	$-p \cdot \theta$	$-\text{sign}(x) \cdot p \cdot \theta \cdot [1 - (1 + \theta\lvert x\rvert)^{p-1}]$
g_{scad}	$\frac{2\theta}{a+1}$	$\begin{cases} 0, & \lvert x\rvert \leq \frac{1}{\theta} \\ \text{sign}(x) \cdot \frac{2\theta(\theta\lvert x\rvert - 1)}{a^2-1}, & \frac{1}{\theta} < \lvert x\rvert \leq \frac{a}{\theta} \\ \text{sign}(x) \cdot \frac{2\theta}{a+1}, & \text{otherwise} \end{cases}$
$g_{\log}$	$\frac{\theta}{\log(1+\theta)}$	$\text{sign}(x) \cdot \frac{\theta^2\lvert x\rvert}{\log(1+\theta)(1+\theta\lvert x\rvert)}$

where the specific expression of $g : \mathbb{R} \to \mathbb{R}$ is given in Table 3.1; and $\eta\,(\theta)$ is a fixed given function, whose expression depends on the surrogate g under consideration, see Table 3.2. Note that the parameter θ controls the tightness of the approximation of the ℓ_0 function: in fact, it holds that $\lim_{\theta\to+\infty} g(x_i) = 1$ if $x_i \neq 0$, otherwise $\lim_{\theta\to+\infty} g(x_i) = 0$. Moreover, it can be shown that for all the functions in Table 3.1, g^- is convex and has Lipschitz continuous first derivative dg^-/dx [143], whose closed form is given in Table 3.2.

Motivated by the effectiveness of the aforementioned nonconvex surrogates of the ℓ_0 function and recent works [2, 151, 212, 261, 268], in this section, we show how to use the MM framework to design efficient algorithms for the solution of Problem (3.46), where G is assumed to have the DC structure (3.47), capturing thus in a unified way all the nonconvex ℓ_0 surrogates reported in Table 3.1. The key question is how to construct a valid surrogate function $\widetilde{V}$ of V in (3.46). We address this issue following two steps: (1) We first find a surrogate $\widetilde{G}$ for the nonconvex G in

(3.47) satisfying Assumption 2.16; and (2) then we construct the overall surrogate $\widetilde{V}$ of V, building on $\widetilde{G}$.

Step 1: Surrogate $\widetilde{G}$ of G There are two ways to construct $\widetilde{G}$, namely: (i) tailoring $\widetilde{G}$ to the specific structure of the function g under consideration (cf. Table 3.1); or (ii) leveraging the DC structure of g in (3.47) and obtain a unified expression for $\widetilde{G}$, valid for all the DC functions in Table 3.1. Few examples based on the approach (i) are shown first, followed by the general design as in (ii).

Example 8 (The Log ℓ_0 Surrogate) Let G be the "log" ℓ_0 surrogate, i.e., $G(\mathbf{x}) = \sum_{i=1}^{m} g_{\log}(x_i)$, where $g_{\log}$ is defined in Table 3.1. A valid surrogate is obtained majorizing the log function $g_{\log}$ (cf. Example 1), which leads to $\widetilde{G}(\mathbf{x} \mid \mathbf{y}) = \sum_{i=1}^{m} \widetilde{g}_{\log}(x_i \mid y_i)$, with

$$\widetilde{g}_{\log}(x_i \mid y_i) = \frac{\theta}{\log(1+\theta)} \cdot \frac{1}{1+\theta|y_i|}\,|x_i|. \tag{3.48}$$

Example 9 (The $\ell_p(0 < p < 1)$ Surrogate) Let G be the $\ell_p(0 < p < 1)$ function, i.e., $G(\mathbf{x}) = \sum_{i=1}^{m} g_{\ell_p^+}(x_i)$, where $g_{\ell_p^+}$ is defined in Table 3.1. Similar to the Log surrogate, we can derive a majorizer of such a $G(\mathbf{x})$ by exploiting the concavity of $g_{\ell_p^+}$. We have

$$\widetilde{g}_{\ell_p^+}(x_i \mid y_i) = \frac{1}{\theta}(|y_i| + \epsilon)^{1/\theta - 1}|x_i|. \tag{3.49}$$

The desired valid surrogate is then given by $\widetilde{G}(\mathbf{x} \mid \mathbf{y}) = \sum_{i=1}^{m} \widetilde{g}_{\ell_p^+}(x_i \mid y_i)$.

Example 10 (DC Surrogates) We consider now nonconvex regularizers having the DC structure (3.47). A natural approach is to keep the convex component g^+ in (3.47) while linearizing the differentiable concave part $-g^-$, which leads to the following convex majorizer:

$$\widetilde{g}(x_i \mid y_i) = \eta(\theta)\,|x_i| - \left.\frac{dg^-(x)}{dx}\right|_{x=y_i} \cdot (x_i - y_i), \tag{3.50}$$

where the expression of $dg(x)^-/dx$ is given in Table 3.2. The desired majorization then reads $\widetilde{G}(\mathbf{x} \mid \mathbf{y}) = \sum_{i=1}^{m} \widetilde{g}(x_i \mid y_i)$.

Note that although the log and ℓ_p surrogates provided in Example 8 and 9 are special cases of DC surrogates, the majorizer constructed using (3.50) is different from the ad-hoc surrogates $\widetilde{g}_{\log}$ and $\widetilde{g}_{\ell_p^+}$ in (3.48) and (3.49), respectively.

Step 2: Surrogate $\widetilde{V}$ of V We derive now the surrogate of V, when G is given by (3.47). Since the loss function $\|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$ is convex, two natural options for $\widetilde{V}$ are: 1) keeping $\|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$ unaltered while replacing G with the surrogate $\widetilde{G}$ discussed in Step 1; or 2) majorizing also $\|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$. The former approach "better" preserves the structure of the objective function but at the price of a higher cost of each iteration [cf. (3.2)]: the minimizer of the resulting $\widetilde{V}$ does not have a closed form expression; the overall algorithm is thus a double-loop scheme. The latter approach is instead motivated by the goal of obtaining low-cost iterations. We discuss next both options and establish an interesting connection between the two resulting MM algorithms.

• **Option 1** Keeping $\|\mathbf{z}-\mathbf{A}\mathbf{x}\|^2$ unaltered while using (3.50), leads to the following update in Algorithm 1:

$$\mathbf{x}^{k+1} \in \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \widetilde{V}(\mathbf{x}\,|\,\mathbf{x}^k) \triangleq \|\mathbf{A}\mathbf{x}-\mathbf{z}\|^2 + \lambda \sum_{i=1}^{m} \widetilde{g}(x_i\,|\,x_i^k) \right\}. \tag{3.51}$$

Problem (3.51) is convex but does not have a closed form solution. To solve (3.51), we develop next an ad-hoc iterative soft-thresholding-based algorithm invoking again the MM framework on $\widetilde{V}(\bullet\,|\,\mathbf{x}^k)$ in (3.51).

We denote by $\mathbf{x}^{k,r}$ the r-th iterate of the (inner loop) MM algorithm (Algorithm 1) used to solve (3.51); we initialize the inner algorithm by $\mathbf{x}^{k,0} = \mathbf{x}^k$. Since the quadratic term $\|\mathbf{A}\mathbf{x} - \mathbf{z}\|^2$ in (3.51) has Lipschitz gradient, a natural surrogate function for $\widetilde{V}(\mathbf{x}\,|\,\mathbf{x}^k)$, is [cf. (3.28)]: given $\mathbf{x}^{k,r}$,

$$\widetilde{V}^k(\mathbf{x} \mid \mathbf{x}^{k,r}) = 2\mathbf{x}^T\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k,r} - \mathbf{z}) + \frac{L}{2}\|\mathbf{x}-\mathbf{x}^{k,r}\|^2 + \lambda \sum_{i=1}^{m} \widetilde{g}(x_i\,|\,x_i^k), \tag{3.52}$$

where $L = 2\,\lambda_{\max}(\mathbf{A}^T\mathbf{A})$. Denoting

$$\mathbf{b}^{k,r} \triangleq \mathbf{x}^{k,r} - \frac{2}{L}\mathbf{A}^T(\mathbf{A}\mathbf{x}^{k,r} - \mathbf{z}) + \frac{\lambda}{L} \cdot \left(\left.\frac{dg^-(x)}{dx}\right|_{x=x_i^k} \right)_{i=1}^{m},$$

the main update of the inner MM algorithm minimizing $\widetilde{V}^k(\mathbf{x} \mid \mathbf{x}^{k,r})$ reads

$$\mathbf{x}^{k,r+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{L}{2}\|\mathbf{x}-\mathbf{b}^{k,r}\|^2 + \lambda\,\eta(\theta)\,\|\mathbf{x}\|_1. \tag{3.53}$$

Quite interestingly, the solution of Problem (3.53) can be obtained in closed form. Writing the first order optimality condition

$$\mathbf{0} \in (\mathbf{x}-\mathbf{b}^{k,r}) + \frac{\lambda\,\eta(\theta)}{L}\,\partial\|\mathbf{x}\|_1,$$

and recall that the subgradient of $\|\mathbf{x}\|_1$ takes the following form [cf. (3.10)]:

$$\partial\|\mathbf{x}\|_1 = \{\boldsymbol{\zeta} \,:\, \boldsymbol{\zeta}^T\mathbf{x} = \|\mathbf{x}\|_1,\ \|\boldsymbol{\zeta}\|_\infty \le 1\},$$

we have the following expression for $\mathbf{x}^{k,r+1}$:

$$\mathbf{x}^{k,r+1} = \mathrm{sign}(\mathbf{b}^{k,r}) \cdot \max\left\{|\mathbf{b}^{k,r}| - \frac{\lambda\, \eta(\theta)}{L}, 0\right\},$$

where the sign and max operators are applied component-wise. Introducing the *soft-thresholding operator*

$$S_\alpha(x) \triangleq \mathrm{sign}(x) \cdot \max\{|x| - \alpha, 0\}, \tag{3.54}$$

$\mathbf{x}^{k,r+1}$ can be rewritten succinctly as

$$\mathbf{x}^{k,r+1} = S_{\frac{\lambda\,\eta(\theta)}{L}}(\mathbf{b}^{k,r}), \tag{3.55}$$

where the soft-thresholding operator is applied component-wise.

Overall the double loop algorithm, based on the MM outer updates (3.51) and MM inner iterates (3.55) is summarized in Algorithm 3.

Termination Criteria As the termination criterion of Step 1 and Step 3 in Algorithm 3, one can use any valid merit function measuring the distance of the iterates from stationarity of (3.46) and optimality of (3.53), respectively. For both the inner and outer loop, it is not difficult to check that the objective functions of the associated optimization problems—(3.53) and (3.51), respectively—are strictly convex if $\lambda > 0$; therefore both optimization problem have unique minimizers. It turns out that both functions defined in (3.24) and (3.25) can be adopted as valid merit functions. The loop can be then terminated once the value of the chosen function goes below the desired threshold.

• **Option 2** Algorithm 3 is a double loop MM-based algorithm: in the outer loop, the surrogate function $\widetilde{V}(\bullet \mid \mathbf{x}^k)$ [cf. (3.51)] is iteratively minimized by means of an inner MM algorithm based on the surrogate function $\widetilde{V}^k(\mathbf{x} \mid \mathbf{x}^{k,r})$ [cf. (3.52)]. A closed look at (3.51) and (3.52) shows that the following relationship holds between

Algorithm 3 MM Algorithm for nonconvex sparse least squares

```
Data : $\mathbf{x}^0 \in X$. Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Set $r = 0$. Initialize $\mathbf{x}^{k,0}$ as $\mathbf{x}^{k,0} = \mathbf{x}^k$;
(S.3) : If $\mathbf{x}^{k,r}$ satisfies a termination criterion: STOP;
        (a) : Update $\mathbf{x}^{k,r}$ as
              $$\mathbf{x}^{k,r+1} = S_{\frac{\lambda\,\eta(\theta)}{L}}\left(\mathbf{b}^{k,r}\right);$$
        (b) : $r \leftarrow r + 1$, and go to (S.3).
(S.4) : $\mathbf{x}^{k+1} = \mathbf{x}^{k,r}$, $k \leftarrow k + 1$, and go to (S.1).
```

$\widetilde{V}$ and $\widetilde{V}^k$:

$$\widetilde{V}^k(\mathbf{x} \mid \mathbf{x}^{k,0}) \geq \widetilde{V}(\mathbf{x} \mid \mathbf{x}^k) \geq V(\mathbf{x}).$$

The above inequality shows that $\widetilde{V}^{k,0}(\mathbf{x} \mid \mathbf{x}^k)$ is in fact a valid surrogate function of V at $\mathbf{x} = \mathbf{x}^k$. This means that the inner loop of Algorithm 3 can be terminated after one iteration without affecting the convergence of the scheme. Specifically, Step 3 of Algorithm 3 can be replaced with the following iterate

$$\mathbf{x}^{k+1} = S_{\frac{\lambda\,\eta(\theta)}{L}}\left(\mathbf{b}^{k,0}\right). \tag{3.56}$$

The resulting algorithm is in fact an MM scheme minimizing $\widetilde{V}^{k,0}(\bullet \mid \mathbf{x}^k)$, whose convergence is guaranteed by Theorem 2.14.

3.2.4.2 Nonnegative Least Squares

Finding a nonnegative solution $\mathbf{x} \in \mathbb{R}^m$ of the linear model $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{n}$ has attracted significant attention in the literature. This problem arises in applications where the measured data is nonnegative; examples include image pixel intensity, economical quantities such as stock price and trading volume, biomedical records such as weight, height, blood pressure, etc. [75, 142, 213]. It is also one of the key ingredients in nonnegative matrix/tensor factorization problem for analysing structured data set. The nonnegative least squares (NNLS) problem consists in finding a nonnegative $\mathbf{x}$ that minimizes the residual error between the data and the model in least square sense:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & V(\mathbf{x}) \triangleq \|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2 \\ \text{subject to} \quad & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{3.57}$$

where $\mathbf{z} \in \mathbb{R}^q$ and $\mathbf{A}^{q \times m}$ are given.

Note that Problem (3.57) is convex. We show next how to construct a surrogate function satisfying Assumption 2.13 which is additively separable in the components of $\mathbf{x}$, so that the resulting subproblems (3.2) can be solved in parallel. To this end, we expand the square in the objective function and write

$$V(\mathbf{x}) = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{z}^T\mathbf{A}\mathbf{x} + \mathbf{z}^T\mathbf{z}. \tag{3.58}$$

Let $\mathbf{M} \succeq \mathbf{A}\mathbf{A}^T$. Using (3.29), we can majorize V by

$$\widetilde{V}(\mathbf{x} \mid \mathbf{y}) = V(\mathbf{y}) + 2(\mathbf{A}^T\mathbf{A}\mathbf{y} - \mathbf{A}^T\mathbf{z})^T(\mathbf{x} - \mathbf{y}) + (\mathbf{x} - \mathbf{y})^T\mathbf{M}\,(\mathbf{x} - \mathbf{y}), \tag{3.59}$$

for any given $\mathbf{y} \in \mathbb{R}^m$. The goal is then finding a matrix $\mathbf{M} \succeq \mathbf{A}\mathbf{A}^T$ that is diagonal, so that $\widetilde{V}(\mathbf{x} \,|\, \mathbf{y})$ becomes additively separable. We provide next two alternative expressions for $\mathbf{M}$.

• **Option 1** Since ∇V is Lipschitz continuous on $\mathbb{R}^m$, we can use the same upperbound as in Example 3. This corresponds to choose $\mathbf{M} = \lambda \mathbf{I}$, with λ such that $\lambda \geq \lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right)$. This leads overall to the following surrogate of V:

$$\widetilde{V}(\mathbf{x} \,|\, \mathbf{y}) = V(\mathbf{y}) + 2(\mathbf{A}^T\mathbf{A}\,\mathbf{y} - \mathbf{A}^T\mathbf{z})^T(\mathbf{x} - \mathbf{y}) + \lambda\|\mathbf{x} - \mathbf{y}\|^2.$$

The above choice leads to a strongly convex subproblem (3.2), whose minimizer has the following closed form:

$$\mathbf{x}^{k+1} = \left[\mathbf{x}^k - \frac{1}{\lambda}\left(\mathbf{A}^T\mathbf{A}\mathbf{x}^k - \mathbf{A}^T\mathbf{z}\right)\right]_+, \tag{3.60}$$

where $[\bullet]_+$ denotes the Euclidean projection onto the nonnegative orthant. The resulting MM algorithm (Algorithm 1) based on the update (3.60) turns out to be the renowned gradient projection algorithm with constant step-size $1/\lambda$.

• **Option 2** If Problem (3.57) has some extra structure, the surrogate $\widetilde{V}$ can be tailored to V even further. For instance, suppose that *in addition* to the structure above, there hold $\mathbf{A} \in \mathbb{R}^{q \times m}_{++}$, $\mathbf{z} \in \mathbb{R}^q_+$ and $\mathbf{z} \neq \mathbf{0}$. It has been shown in [58, 142] that the following diagonal matrix

$$\mathbf{M} \triangleq \text{Diag}\left(\frac{(\mathbf{A}^T\mathbf{A}\,\mathbf{x}^k)_1}{x_1^k}, \cdots, \frac{(\mathbf{A}^T\mathbf{A}\,\mathbf{x}^k)_m}{x_m^k}\right) \tag{3.61}$$

satisfies $\mathbf{M} \succeq \mathbf{A}^T\mathbf{A}$. Substituting (3.61) in (3.59), one obtains the following closed form solution of the resulting subproblem (3.2):

$$\mathbf{x}^{k+1} = (\mathbf{A}^T\mathbf{z}/\mathbf{A}^T\mathbf{A}\mathbf{x}^k) \cdot \mathbf{x}^k,$$

wherein both division and multiplication are intended to be applied element-wise.

3.2.4.3 Sparse Plus Low-Rank Matrix Decomposition

Another useful paradigm is to decompose a partly or fully observed data matrix into the sum of a low rank and (bilinear) sparse term; the low-rank component captures correlations and periodic trends in the data whereas the bilinear term explains parsimoniously data patterns, (co-)clusters, innovations or outliers.

Let $\mathbf{Y} \in \mathbb{R}^{m \times t} (m \leq t)$ be the data matrix. The goal is to find a low rank matrix $\mathbf{L} \in \mathbb{R}^{m \times t}$ with rank $r_0 \triangleq \text{rank}(\mathbf{L}) \ll m$, and a sparse matrix $\mathbf{S} \in \mathbb{R}^{m \times t}$ such that $\mathbf{Y} = \mathbf{L} + \mathbf{S} + \mathbf{V}$, where $\mathbf{V} \in \mathbb{R}^{m \times t}$ accounts for measurement errors. To cope with the missing data in $\mathbf{Y}$, we introduce i) the set $\Omega \subseteq M \times T$ of index pairs (i, j), where $M \triangleq \{1, \ldots, m\}$ and $T \triangleq \{1, \ldots, t\}$; and ii) the sampling operator $P_\Omega(\cdot)$, which nulls the entries of its matrix argument not in Ω, leaving the rest unchanged. In this way, one can express incomplete and (possibly noise-)corrupted data as

$$P_\Omega(\mathbf{Y}) \approx P_\Omega(\mathbf{L} + \mathbf{S}), \tag{3.62}$$

where $\approx$ is quantified by a specific loss function (and regularization) [220].

Model (3.62) subsumes a variety of statistical learning paradigms including: (robust) principal component analysis [35, 41], compressive sampling [34], dictionary learning (DL) [77, 180], non-negative matrix factorization [109, 125, 141], matrix completion, and their robust counterparts [114]. Task (3.62) also emerges in various applications such as (1) network anomaly detection [132, 156, 157]; (2) distributed acoustic signal processing [73, 74]; (3) distributed localization and sensor identification [206]; (4) distributed seismic forward modeling in geological applications [173, 271] (e.g., finding the Green's function of some model of a portion of the earth's surface); (5) topic modeling for text corpus from social media [199, 266]; (6) data and graph clustering [100, 128, 129, 236]; and (7) power grid state estimation [95, 122].

In the following, we study task (3.62) adopting the least-squares (LS) error as loss function for $\approx$. We show in detail how to design an MM algorithm for the solution of two classes of problems under (3.62), namely: (i) the low-rank matrix completion problem; and (ii) the dictionary learning problem. Similar techniques can be used also to solve other tasks modeled by (3.62).

1) Low-Rank Matrix Completion The low-rank matrix completion problem arises frequently in learning problems where the task is to fill in the blanks in the partially observed collinear data. For instance, in movie rating problems, $\mathbf{Y}$ is the rating matrix whose entries y_{ij} represent the score of movie j given by individual i if he/she has watched it, and considered missing otherwise. Despite of being highly incomplete, such a data set is also rank deficient as individuals sharing similar interests may give similar ratings (the corresponding rows of $\mathbf{Y}$ are collinear), which makes the matrix completion task possible. Considering model (3.62), the question becomes how to impose a low-rank structure on $\mathbf{L}$ and sparse structure on $\mathbf{S}$. We describe next two widely used approaches.

A first approach is enforcing a low-rank structure on $\mathbf{L}$ by promoting sparsity on the singular values of $\mathbf{L}$ [denoted by $\sigma_i(\mathbf{L})$, $i = 1, \ldots, m$] as well as on the elements of $\mathbf{S}$ via regularization. This leads to the following formulation

$$\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad V(\mathbf{L}, \mathbf{S}) \triangleq \|P_\Omega\left(\mathbf{Y} - \mathbf{L} - \mathbf{S}\right)\|_F^2 + \lambda_r \cdot G_r(\mathbf{L}) + \lambda_s \cdot G_s(\mathbf{S}), \tag{3.63}$$

where $G_r(\mathbf{L}) \triangleq \sum_{i=1}^m g_r\left(\sigma_i\left(\mathbf{L}\right)\right)$ and $G_s(\mathbf{S}) \triangleq \sum_{i=1}^r \sum_{j=1}^t g_s\left(s_{ij}\right)$ are sparsity promoting regularizers, and λ_r and λ_s positive coefficients. Since $G_r(\mathbf{L})$ promotes sparsity on the singular values of $\mathbf{L}$, this will induce a low-rank structure on $\mathbf{L}$. Note that the general formulation (3.63) contains many popular choices of low-rank inducing penalties. For instance, choosing $g_r(x) = g_s(x) = \text{card}(x)$, one gets $G_r(\mathbf{L}) = \text{rank}(\mathbf{L})$ and $G_s(\mathbf{S}) = \|\mathbf{S}\|_0 \triangleq \|\text{vec}(\mathbf{S})\|_0$, which become the exact (nonconvex) rank penalty on $\mathbf{L}$ and the cardinality penalty on $\mathbf{S}$, respectively. Another popular choice is $g_r(x) = |x|$, which leads to the convex nuclear norm penalty $G_r(\mathbf{L}) = \|\mathbf{L}\|_*$; another example is $g_r(x) = \log(x)$, which yields the nonconvex logdet penalty. To keep the analysis general, in the following we tacitly assume that g_r and g_s are any of the DC surrogate functions of the ℓ_0 function introduced in Sect. 3.2.4.1 [cf. (3.47)]. Note that, since g_r and g_s are DC and $\sigma_i(\mathbf{L})$ is a convex function of $\mathbf{L}$, they are all directionally differentiable (cf. Sect. 3.2.1). Therefore, $V(\mathbf{L}, \mathbf{S})$ in (3.63) is directionally differentiable.

A second approach to enforce a low-rank structure on $\mathbf{L}$ is to "hard-wire" a rank (at most) r into the structure of $\mathbf{L}$ by decomposing $\mathbf{L}$ as $\mathbf{L} = \mathbf{D}\mathbf{X}$, where $\mathbf{D} \in \mathbb{R}^{m\times r}$ and $\mathbf{X} \in \mathbb{R}^{r\times t}$ are two "thin" matrices. The problem then reads

$$\underset{\mathbf{D},\mathbf{X}}{\text{minimize}} \quad \|P_{\Omega}(\mathbf{Y} - \mathbf{D}\mathbf{X} - \mathbf{S})\|_F^2 + \lambda_r \cdot G_r(\mathbf{D}, \mathbf{X}) + \lambda_s \cdot G_s(\mathbf{S}), \tag{3.64}$$

where G_r and G_s promote low-rank and sparsity structures, respectively. While G_s can be chosen as in (3.63), the choice of G_r acting on the two factors $\mathbf{D}$ and $\mathbf{X}$ while imposing the low-rankness of $\mathbf{L}$ is less obvious; two alternative choices are the following. Since

$$\|\mathbf{L}\|_* = \inf_{\mathbf{D}\,\mathbf{X}=\mathbf{L}} \frac{1}{2}\left(\|\mathbf{D}\|_F^2 + \|\mathbf{X}\|_F^2\right),$$

an option is choosing

$$G_r(\mathbf{D}, \mathbf{X}) = \frac{1}{2}\left(\|\mathbf{D}\|_F^2 + \|\mathbf{X}\|_F^2\right).$$

Another low-rank inducing regularizer is the max-norm penalty

$$\|\mathbf{L}\|_{\max} \triangleq \inf_{\mathbf{D}\,\mathbf{X}=\mathbf{L}} \|\mathbf{D}\|_{2,\infty} + \|\mathbf{X}\|_{2,\infty},$$

where $\|\bullet\|_{2,\infty}$ denotes the maximum ℓ_2 row norm of a matrix; this leads to

$$G_r(\mathbf{D}, \mathbf{X}) = \|\mathbf{D}\|_{2,\infty} + \|\mathbf{X}\|_{2,\infty}.$$

Here we focus only on the first formulation, Problem (3.63); the algorithmic design for (3.64) will be addressed within the context of the dictionary learning problem, which is the subject of the next section.

To deal with Problem (3.63), the first step is to rewrite the objective function in a more convenient form, by getting rid of the projection operator P_{Ω}. Define

$\mathbf{Q} \triangleq \operatorname{diag}(\mathbf{q})$, with $q_i = 1$ if $\left(\operatorname{vec}\left(P_\Omega(\mathbf{Y})\right)\right)_i \neq 0$; and $q_i = 0$ otherwise. Then, $V(\mathbf{L}, \mathbf{S})$ in (3.63) can be rewritten as

$$V(\mathbf{L}, \mathbf{S}) = \underbrace{\|\operatorname{vec}(\mathbf{Y}) - \operatorname{vec}(\mathbf{L}+\mathbf{S})\|_{\mathbf{Q}}^2}_{Q(\mathbf{L},\mathbf{S})} + \lambda_r \cdot G_r(\mathbf{L}) + \lambda_s \cdot G_s(\mathbf{S}), \tag{3.65}$$

where $\|\mathbf{x}\|_{\mathbf{Q}}^2 \triangleq \mathbf{x}^T \mathbf{Q}\, \mathbf{x}$. Note that since q_i is equal to either 0 or 1, we have $\lambda_{\max}(\mathbf{Q}) = 1$. In the following, we derive an algorithm that alternately optimizes $\mathbf{L}$ and $\mathbf{S}$ based on the block MM algorithm described in Algorithm 2. In the following, we denote by m_{ij} the (i, j)-th entry of a generic matrix $\mathbf{M}$.

We start with the optimization of $\mathbf{S}$, given $\mathbf{L} = \mathbf{L}^k$. One can easily see that $V(\mathbf{L}^k, \mathbf{S})$ is of the same form as the objective function in Problem (3.46). Therefore, two valid surrogate functions can be readily constructed using the same technique already introduced in Sect. 3.2.4.1. Specifically, two alternative surrogates of $V(\mathbf{L}^k, \mathbf{S})$ are [cf. (3.51)]: given $\mathbf{S}^k$,

$$\widetilde{V}^{(1)}(\mathbf{L}^k, \mathbf{S} \,|\, \mathbf{S}^k) = Q(\mathbf{L}^k, \mathbf{S}) + \lambda_s \sum_{i=1}^{r} \sum_{j=1}^{t} \widetilde{g}_s(s_{ij} \,|\, s_{ij}^k) \tag{3.66}$$

and [cf. (3.52)]

$$\begin{aligned}
&\widetilde{V}^{(2)}(\mathbf{L}^k, \mathbf{S} \,|\, \mathbf{S}^k)\\
&= Q(\mathbf{L}^k, \mathbf{S}^k) + 2\left(\operatorname{vec}(\mathbf{Y} - \mathbf{L}^k) - \operatorname{vec}(\mathbf{S}^k)\right)^T \mathbf{Q}\left(\operatorname{vec}(\mathbf{Y} - \mathbf{L}^k) - \operatorname{vec}(\mathbf{S})\right)\\
&\quad + \left\|\operatorname{vec}(\mathbf{S}) - \operatorname{vec}(\mathbf{S}^k)\right\|^2 + \lambda_s \sum_{i=1}^{r} \sum_{j=1}^{t} \widetilde{g}_s(s_{ij} \,|\, s_{ij}^k)\\
&= \left\|\mathbf{S} - \widetilde{\mathbf{Y}}^k - (\lambda_s/2) \cdot \mathbf{W}^k\right\|_F^2 + \lambda_s\, \eta(\theta)\, \|\mathbf{S}\|_1 + \text{const.},
\end{aligned} \tag{3.67}$$

where $\mathbf{W}^k$ and $\widetilde{\mathbf{Y}}^k$ are matrices of the same size of $\mathbf{S}$, with (i, j)-th entries defined as

$$w_{ij}^k \triangleq \left.\frac{dg_s^-(x)}{dx}\right|_{x=s_{ij}^k} \quad \text{and} \quad \widetilde{y}_{ij}^k \triangleq \begin{cases} y_{ij} - \ell_{ij}^k, & \text{if } (i, j) \in \Omega,\\ s_{ij}^k, & \text{otherwise,} \end{cases} \tag{3.68}$$

respectively; and in const. we absorbed irrelevant constant terms.

The minimizer of $V^{(1)}(\mathbf{L}^k, \bullet \,|\, \mathbf{S}^k)$ and $V^{(2)}(\mathbf{L}^k, \bullet \,|\, \mathbf{S}^k)$ can be computed following the same steps as described in Option 1 and Option 2 in Sect. 3.2.4.2, respectively. Next, we only provide the update of $\mathbf{S}$ based on minimizing $V^{(2)}(\mathbf{L}^k, \bullet \,|\, \mathbf{S}^k)$, which is given by

$$s_{ij}^{k+1} = \operatorname{sign}\left(\widetilde{y}_{ij} + (\lambda_s/2)\cdot w_{ij}^k\right)\cdot \max\left\{\left|\widetilde{y}_{ij} + (\lambda_s/2)\cdot w_{ij}^k\right| - \eta(\theta)/2, 0\right\}. \tag{3.69}$$

Next, we fix $\mathbf{S} = \mathbf{S}^{k+1}$ and optimize $\mathbf{L}$. In order to obtain a closed form update of $\mathbf{L}$, we upperbound $Q(\bullet, \mathbf{S}^{k+1})$ and $G_r(\bullet)$ in (3.65) using (3.28) and (3.26), respectively, and following similar steps as to obtain (3.67). Specifically, a surrogate of $Q(\bullet, \mathbf{S}^{k+1})$ is: given $\mathbf{L}^k$,

$$\widetilde{Q}(\mathbf{L}, \mathbf{S}^{k+1} \,|\, \mathbf{L}^k) = \|\mathbf{L} - \mathbf{X}^k\|_F^2 + \text{const.}, \tag{3.70}$$

where $\mathbf{X}^k$ is a matrix having the same size of $\mathbf{Y}$, with entries defined as

$$x_{ij}^k \triangleq \begin{cases} y_{ij} - s_{ij}^{k+1}, & (i,j) \in \Omega; \\ \ell_{ij}^k, & \text{otherwise.} \end{cases} \tag{3.71}$$

To upperbound the nonconvex regularizer $G_r(\mathbf{L}) = \sum_{i=1}^m g\left(\sigma_i\left(\mathbf{L}\right)\right)$, we invoke (3.50) and obtain

$$\widetilde{g}_r\left(\sigma_i(\mathbf{L}) \,|\, \sigma_i(\mathbf{L}^k)\right) = \eta(\theta)|\sigma_i\left(\mathbf{L}\right)| - w_i^k \cdot \left(\sigma_i\left(\mathbf{L}\right) - \sigma_i(\mathbf{L}^k)\right), \tag{3.72}$$

with

$$w_i^k \triangleq \left.\frac{dg_r^-(x)}{dx}\right|_{x=\sigma_i(\mathbf{L}^k)}.$$

Using the directional differentiability of the singular values $\sigma_i(\mathbf{L}^k)$ (see, e.g., [97]) and the chain rule, it is not difficult to check that Assumption 2.13 (in particular the directional derivative consistency condition 2.13.3) is satisfied for $\widetilde{g}_r$; therefore $\widetilde{g}_r\left(\bullet \,|\, \sigma_i(\mathbf{L}^k)\right)$ is a valid surrogate function of g_r.

Combining (3.70) and (3.72) yields the following surrogate function of $V(\mathbf{L}, \mathbf{S}^{k+1})$:

$$\widetilde{V}\left(\mathbf{L}, \mathbf{S}^{k+1} \,|\, \mathbf{L}^k\right) = \|\mathbf{L} - \mathbf{X}^k\|_F^2 + \lambda_r \sum_{i=1}^m \left(\eta|\sigma_i\left(\mathbf{L}\right)| - w_i^k \sigma_i\left(\mathbf{L}\right)\right). \tag{3.73}$$

The final step is computing the minimizer of $\widetilde{V}(\mathbf{L} \,|\, \mathbf{L}^k)$. To this end, we first introduce the following lemma [244].

Lemma 2.21 (von Neumann's Trace Inequality) *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two* $m \times m$ *complex-valued matrices with singular values* $\sigma_1(\mathbf{A}) \geq \cdots \geq \sigma_m(\mathbf{A})$ *and* $\sigma_1(\mathbf{B}) \geq \cdots \geq \sigma_m(\mathbf{B})$*, respectively. Then,*

$$|\mathrm{Tr}(\mathbf{AB})| \leq \sum_{i=1}^{m} \sigma_i(\mathbf{A})\sigma_i(\mathbf{B}). \tag{3.74}$$

Note that Lemma 2.21 can be readily generalized to rectangular matrices. Specifically, given $\mathbf{A}, \mathbf{B}^T \in \mathbb{R}^{m\times t}$, define the augmented square matrices $\widetilde{\mathbf{A}} \triangleq [\mathbf{A}; \mathbf{0}_{(t-m)\times t}]$ and $\widetilde{\mathbf{B}} \triangleq [\mathbf{B}, \mathbf{0}_{t\times(t-m)}]$, respectively. Applying Lemma 2.21 to $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$, we get

$$\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \leq \sum_{i=1}^{t} \sigma_i(\widetilde{\mathbf{A}})\sigma_i(\widetilde{\mathbf{B}}) = \sum_{i=1}^{m} \sigma_i(\mathbf{A})\sigma_i(\mathbf{B}), \tag{3.75}$$

where equality is achieved when $\mathbf{A}$ and $\mathbf{B}^T$ share the same singular vectors. Using (3.75), we can now derive the closed form of the minimizer of (3.73).

Proposition 2.22 *Let* $\mathbf{X}^k = \mathbf{U}_X \boldsymbol{\Sigma}_X \mathbf{V}_X^T$ *be the singular value decomposition (SVD) of* $\mathbf{X}^k$*. The minimizer of* $\widetilde{V}\left(\mathbf{L}, \mathbf{S}^{k+1} \,|\, \mathbf{L}^k\right)$ *in* (3.73) *(and thus the update of* $\mathbf{L}$*) is given by*

$$\mathbf{L}^{k+1} = \mathbf{U}_X D_{\frac{\eta\lambda_r}{2}}\left(\boldsymbol{\Sigma}_X + \mathrm{Diag}(\{w_i^k \lambda_r/2\}_{i=1}^m)\right)\mathbf{V}_X^T, \tag{3.76}$$

where $D_\alpha(\boldsymbol{\Sigma})$ *denotes a diagonal matrix with the* i*-th diagonal element equal to* $(\boldsymbol{\Sigma}_{ii} - \alpha)_+$*, and* $(x)_+ \triangleq \max(0, x)$.

Proof Expanding the squares we rewrite $\widetilde{V}\left(\mathbf{L}, \mathbf{S}^{k+1} \,|\, \mathbf{L}^k\right)$ as

$$\begin{aligned}
&\widetilde{V}(\mathbf{L}, \mathbf{S}^{k+1} \,|\, \mathbf{L}^k) \\
&\quad= \mathrm{Tr}(\mathbf{L}\mathbf{L}^T) - 2\mathrm{Tr}(\mathbf{L}(\mathbf{X}^k)^T) + \mathrm{Tr}(\mathbf{X}^k(\mathbf{X}^k)^T) + \lambda_r \sum_{i=1}^{m}\left(\eta|\sigma_i(\mathbf{L})| - w_i^k \sigma_i(\mathbf{L})\right) \\
&\quad= \sum_{i=1}^{m} \sigma_i^2(\mathbf{L}) - 2\mathrm{Tr}(\mathbf{L}(\mathbf{X}^k)^T) + \lambda_r \sum_{i=1}^{m}\left(\eta|\sigma_i(\mathbf{L})| - w_i^k \sigma_i(\mathbf{L})\right) + \mathrm{Tr}(\mathbf{X}^k(\mathbf{X}^k)^T).
\end{aligned} \tag{3.77}$$

To find a minimizer of $\widetilde{V}(\bullet, \mathbf{S}^{k+1} \,|\, \mathbf{L}^k)$, we introduce the SVD of $\mathbf{L} = \mathbf{U}_L \boldsymbol{\Sigma}_L \mathbf{V}_L^T$, and optimize separately on $\mathbf{U}_L$, $\mathbf{V}_L$, and $\boldsymbol{\Sigma}_L$, with $\boldsymbol{\Sigma}_L = \mathrm{diag}(\sigma_1(\mathbf{L}), \ldots, \sigma_m(\mathbf{L}))$.

From (3.75) we have $\mathrm{Tr}(\mathbf{L}(\mathbf{X}^k)^T) \leq \sum_{i=1}^m \sigma_i(\mathbf{L})\sigma_i(\mathbf{X}^k)$, and equality is reached if $\mathbf{U}_L = \mathbf{U}_X$ and $\mathbf{V}_L = \mathbf{V}_X$, respectively; which are thus optimal. To compute the optimal $\boldsymbol{\Sigma}_L$ let us substitute $\mathbf{U}_L = \mathbf{U}_X$ and $\mathbf{V}_L = \mathbf{V}_X$ in (3.77) and solve the resulting minimization problem with respect to $\boldsymbol{\Sigma}_L$:

$$\begin{aligned}
&\underset{\{\sigma_i(\mathbf{L})\}_{i=1}^m}{\text{minimize}} && \sum_{i=1}^m \sigma_i(\mathbf{L})^2 - 2\sum_{i=1}^m \sigma_i(\mathbf{L})\sigma_i(\mathbf{X}^k) + \lambda_r \sum_{i=1}^m \Big(\eta\sigma_i(\mathbf{L}) - w_i^k \sigma_i(\mathbf{L})\Big) \\
&\text{subject to} && \sigma_i(\mathbf{L}) \geq 0, \ \forall i = 1, \ldots, m.
\end{aligned} \tag{3.78}$$

Problem (3.78) is additively separable. The optimal value of each $\sigma_i(\mathbf{L})$ is

$$\begin{aligned}
\sigma_i^\star(\mathbf{L}) &= \underset{\sigma_i(\mathbf{L})\geq 0}{\text{argmin}}\, (\sigma_i(\mathbf{L}) - \sigma_i(\mathbf{X}^k) - w_i^k\lambda_r/2 + \eta\lambda_r/2)^2 \\
&= (\sigma_i(\mathbf{X}^k) + w_i^k\lambda_r/2 - \eta\lambda_r/2)_+,
\end{aligned} \tag{3.79}$$

which completes the proof.

The block MM algorithm solving the matrix completion Problem (3.63), based on the S-updates (3.69) and L-update (3.76), is summarized in Algorithm 4.

2) Dictionary Learning Given the data matrix $\mathbf{Y} \in \mathbb{R}^{m\times t}$, the dictionary learning (DL) problem consists in finding a basis, the dictionary $\mathbf{D} \in \mathbb{R}^{m\times r}$ (with $r \ll t$), over which $\mathbf{Y}$ can be sparsely represented throughout the coefficients $\mathbf{X} \in \mathbb{R}^{r\times t}$. This problem appears in a wide range of machine learning applications, such as image denoising, video surveillance, face recognition, and unsupervised clustering. We consider the following formulation for the DL problem:

$$\underset{\mathbf{X},\, \mathbf{D}\in D}{\text{minimize}} \quad V(\mathbf{D}, \mathbf{X}) \triangleq \underbrace{\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2}_{F(\mathbf{D},\mathbf{X})} + \lambda_s\, G(\mathbf{X}), \tag{3.80}$$

where D is a convex compact set, bounding the elements of the dictionary so that the optimal solution will not go to infinity due to scaling ambiguity; and G aims at promoting sparsity on $\mathbf{X}$, with λ_s being a positive given constant. In the following, we assume that $G(\mathbf{X}) = \sum_{i=1}^r \sum_{j=1}^t g(x_{ij})$, with g being any of the DC functions introduced in (3.47).

Since $F(\mathbf{D}, \mathbf{X})$ in (3.80) is biconvex, we can derive an algorithm for Problem (3.80) based on the block MM algorithm by updating $\mathbf{D}$ and $\mathbf{X}$ alternately.

Algorithm 4 Block MM algorithm for matrix completion [cf. (3.63)]

```
Data : $\mathbf{L}^0, \mathbf{S}^0 \in \mathbb{R}^{m\times t}$. Set $k = 0$.
(S.1) : If $\mathbf{L}^k$ and $\mathbf{S}^k$ satisfy a termination criterion: STOP;
(S.2) : Alternately optimize $\mathbf{S}$ and $\mathbf{L}$:
        (a) : Update $\mathbf{S}^{k+1}$ as according to (3.69);
        (b) : Update $\mathbf{L}^{k+1}$ according to (3.76);
(S.3) : $k \leftarrow k + 1$, and go to (S.1).
```

Given $\mathbf{X} = \mathbf{X}^k$, $F(\mathbf{D}, \mathbf{X}^k)$ is convex in $\mathbf{D}$. A natural choice for a surrogate of $F(\mathbf{D}, \mathbf{X}^k)$ is $F(\mathbf{D}, \mathbf{X}^k)$ itself, that is,

$$\widetilde{F}^{(1)}(\mathbf{D} \,|\, \mathbf{X}^k) \triangleq F(\mathbf{D}, \mathbf{X}^k) = \|\mathbf{Y} - \mathbf{D}\mathbf{X}^k\|_F^2, \tag{3.81}$$

and update $\mathbf{D}$ solving

$$\mathbf{D}^{k+1} = \underset{\mathbf{D} \in D}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}^k\|_F^2. \tag{3.82}$$

Problem (3.82) is convex, but does not have a closed form solution for a general constraint set D. In some special cases, efficient iterative method can be derived to solve (3.82) by exploiting its structure (see, e.g., [195]). For instance, consider the constraint set $D = \{\mathbf{D} \in \mathbb{R}^{m \times r} \ : \ \|\mathbf{D}\|_F^2 \leq \alpha\}$. Writing the KKT conditions of (3.82) (note that Slater's constraint qualification holds), we get

$$0 \leq \alpha - \|\mathbf{D}\|_F^2 \perp \mu \geq 0, \tag{3.83a}$$

$$\nabla_{\mathbf{D}} L(\mathbf{D}, \mu) = 0, \tag{3.83b}$$

where $L(\mathbf{D}, \mu)$ is the Lagrangian function, defined as

$$L(\mathbf{D}, \mu) = \|\mathbf{Y} - \mathbf{D}\mathbf{X}^k\|_F^2 + \mu(\|\mathbf{D}\|_F^2 - \alpha). \tag{3.84}$$

For any given $\mu \geq 0$, the solution of (3.83b) is given by

$$\mathbf{D}(\mu) = \mathbf{Y}(\mathbf{X}^k)^T(\mathbf{X}^k(\mathbf{X}^k)^T + \mu\mathbf{I})^{-1}, \tag{3.85}$$

where μ needs to be chosen in order to satisfy the complementarity condition in (3.83)

$$0 \leq h(\mu) \perp \mu \geq 0, \tag{3.86}$$

with $h(\mu) \triangleq \alpha - \|\mathbf{D}(\mu)\|_F^2$. Since $h(\bullet)$ is monotone, (3.86) can be efficiently solved using bisection.

An alternative surrogate function of $F(\mathbf{D}, \mathbf{X}^k)$ leading to a closed form solution of the resulting minimization problem can be readily obtained leveraging the Lipschitz continuity of $\nabla_D F(\mathbf{D}, \mathbf{X}^k)$ and using (3.28), which yields

$$\widetilde{F}^{(2)}(\mathbf{D}, \mathbf{X}^k \,|\, \mathbf{D}^k) = 2\,\mathrm{Tr}\left\{(\mathbf{D}^k\mathbf{X}^k\mathbf{X}^{kT} - \mathbf{Y}\mathbf{X}^{kT})^T(\mathbf{D} - \mathbf{D}^k)\right\} + L\|\mathbf{D} - \mathbf{D}^k\|_F^2 + \text{const.}, \tag{3.87}$$

where $L \triangleq \lambda_{\max}(\mathbf{S}^k\mathbf{S}^{kT})$ and const. is an irrelevant constant. The update of $\mathbf{D}$ is then given by

$$\begin{aligned}\mathbf{D}^{k+1} &= P_D\left(\frac{1}{L}(\mathbf{D}^k\mathbf{X}^k\mathbf{X}^{kT} - \mathbf{Y}\mathbf{X}^{kT})\right)\\ &\triangleq \underset{\mathbf{D}\in D}{\text{argmin}}\left\|\mathbf{D} - \left(\mathbf{D}^k - \frac{1}{L}(\mathbf{D}^k\mathbf{X}^k\mathbf{X}^{kT} - \mathbf{Y}\mathbf{X}^{kT})\right)\right\|^2,\end{aligned} \tag{3.88}$$

which has a closed form expression for simple constraint sets such as $D = \mathbb{R}_+^{m\times r}$, $D = \{\mathbf{D}\,|\,\mathbf{D}\in\mathbb{R}_+^{m\times r},\ \|\mathbf{D}\|_F^2 \leq \alpha\}$, and $D = \{\mathbf{D}\,|\,\|\mathbf{d}_i\|_2^2 \leq \alpha_i,\ \forall i = 1,\ldots,m\}$.

We fix now $\mathbf{D} = \mathbf{D}^{k+1}$, and update $\mathbf{X}$. Problem (3.80) is separable in the columns of $\mathbf{X}$; the subproblem associated with the j-th column of $\mathbf{X}$, denoted by $\mathbf{x}_j$, reads

$$\mathbf{x}_j^{k+1} \in \underset{\mathbf{x}_j}{\text{argmin}}\,\frac{1}{2}\|\mathbf{y}_j - \mathbf{D}^{k+1}\mathbf{x}_j\|^2 + \lambda_s\sum_{j=1}^{t} g_s(x_{ij}). \tag{3.89}$$

Note that Problem (3.89) is of the same form of (3.46); therefore, it can be solved using the MM algorithm, based on the surrogate functions derived in Sect. 3.2.4.1 [cf. (3.51) and (3.52)].

3.2.4.4 Multicast Beamforming

We study the Max-Min Fair (MMF) beamforming problem for single group multicasting [219], where a single base station (BS) equipped with m antennas wishes to transmit common information to a group of q single-antenna users over the same frequency band. The goal of multicast beamforming is to exploit the channels and the spatial diversity offered by the multiple transmit antennas to steer transmitted power towards the group of desired users while limiting interference (leakage) to nearby co-channel users and systems.

Denoting by $\mathbf{w}\in\mathbb{C}^m$ the beamforming vector, the Max-Min beamforming problem reads

$$\begin{aligned}&\underset{\mathbf{w}\in\mathbb{C}^m}{\text{maximize}} && \min_{i=1,\ldots,q}\ \mathbf{w}^H\mathbf{R}_i\mathbf{w}\\ &\text{subject to} && \|\mathbf{w}\|^2 \leq P_T,\end{aligned} \tag{3.90}$$

where P_T is the power budget of the BS and $\mathbf{R}_i \in \mathbb{C}^{m\times m}$ is a positive semidefinite matrix modeling the channel between the BS and user i. Specifically, $\mathbf{R}_i = \mathbf{h}_i\mathbf{h}_i^H/\sigma_i^2$ if instantaneous Channel State Information (CSI) is assumed, where $\mathbf{h}_i$ is the frequency-flat quasi-static channel vector from the BS to user i and σ_i^2 is the variance of the zero-mean, wide-sense stationary additive noise at the i-th receiver;

and $\mathbf{R}_i = \mathbb{E}\{\mathbf{h}_i\mathbf{h}_i^H\}/\sigma_i^2$ represents the spatial correlation matrix if only long-term CSI is available (in the latter case, no special structure for $\mathbf{R}_i$ is assumed).

Problem (3.90) contains complex variables. One could reformulate the problem into the real domain by using separate variables for the real and imaginary parts of the complex variables, but this approach is not advisable because it does not take advantage of the structure of (real) functions of the complex variables. Following a well-established path in the signal processing and communication communities, here we work directly with complex variables by means of "Wirtinger derivatives". The main advantage of this approach is that we can use the so-called "Wirtinger calculus" to easily compute in practice derivatives of the functions in (3.90) directly in the complex domain. It can be shown that all the results in this chapter extend to the complex domain when using Wirtinger derivatives instead of classical gradients. Throughout the chapter we will freely use the Wirtinger calculus and refer the reader to [105, 126, 209] for more information on this topic.

We derive now an MM algorithm to solve Problem (3.90). To do so, we first rewrite (3.90) in the equivalent minimization form

$$\begin{array}{ll}\underset{\mathbf{w}\in\mathbb{C}^m}{\text{minimize}} & \max\limits_{i=1,\ldots,q} -\mathbf{w}^H\mathbf{R}_i\mathbf{w} \\ \text{subject to} & \|\mathbf{w}\|^2 \leq P_T.\end{array}$$

Since $f_i(\mathbf{w}) \triangleq -\mathbf{w}^H\mathbf{R}_i\mathbf{w}$ is a concave function on $\mathbb{C}^m$, it can be majorized by its first order approximation: given $\mathbf{y} \in \mathbb{C}^m$,

$$\widetilde{f}_i(\mathbf{w}\,|\,\mathbf{y}) = f_i(\mathbf{y}) + 2\,\text{Re}\left\{(\mathbf{y})^H\mathbf{R}_i(\mathbf{w}-\mathbf{y})\right\}. \tag{3.91}$$

Using (3.91) and (3.32), it is easy to check that the following convex function is a valid surrogate of $V(\mathbf{w}) \triangleq \max_{i=1,\ldots,q} -\mathbf{w}^H\mathbf{R}_i\mathbf{w}$:

$$\widetilde{V}(\mathbf{w}\,|\,\mathbf{y}) = \max_{i=1,\ldots,q} \widetilde{f}_i(\mathbf{w};\,\mathbf{y}). \tag{3.92}$$

The main iterate of the MM algorithm based on (3.92) is then given by: given $\mathbf{w}^k$,

$$\mathbf{x}^{k+1} \in \underset{\|\mathbf{w}\|^2\leq P_T}{\text{argmin}} \left\{\max_{i=1,\ldots,q} -\mathbf{w}^{kH}\mathbf{R}_i\mathbf{w}^k + 2\text{Re}\{\mathbf{w}^{kH}\mathbf{R}_i(\mathbf{w}-\mathbf{w}^k)\}\right\}. \tag{3.93}$$

Convergence to d-stationary solutions of (3.90) is guaranteed by Theorem 2.14.

3.2.5 Sources and Notes

A Bit of History The MM algorithmic framework has a long history that traces back to 1970, when the majorization principle was introduced [181] for descent-based algorithms, using line search. In 1977, the MM principle was applied to multidimensional scaling [62, 65]. Concurrently, its close relative, the (generalized) EM algorithm [66], was proposed by Dempster et al. in 1977 in the context of maximum likelihood estimation with missing observations. The authors proved the monotonicity of the EM scheme showing that the E-step generates an upperbound of the objective function. The idea of successively minimizing an upperbound of the original function appeared in subsequent works, including [63, 68, 92, 93, 135], to name a few; and it was introduced as a general algorithmic framework in [64, 101, 137]. The connection between EM and MM was clarified in [10], where the authors argued that the majorization step (also referred to as "optimization transfer") rather than the missing data is the key ingredient of EM. The EM/MM algorithm has gained significant attention and applied in various fields ever since [110, 161, 253]. A recent tutorial on the MM algorithm along with its applications to problems in signal processing, communications, and machine learning can be found in [230].

Building on the plain MM/EM, as described in Algorithm 1, several generalizations of the algorithm have been developed to improve its convergence speed, practical implementability, as well as scalability. Some representative examples are the following. Both the majorization and minimization steps can be performed inexactly: the global upperbound condition of the surrogate function can be relaxed to be just a local upperbound; and the exact minimizer of the surrogate function can be replaced by one that only decreases the value of the surrogate with respect to the current iterate [66, 133]. MM can also be coupled with line-search schemes to accelerate its convergence [6, 102, 144, 201]. Furthermore, instead of majorizing the objective function on the whole space, the "subspace MM" algorithm constructs majorizers on an iteration-dependent subspace [47–49, 131]. For structured problems whose variables are naturally partitioned in blocks, majorization can be done block-wise to reduce the scale of the subproblems and achieving tighter upperbounds [83]. Sweeping rules of the blocks such as the (essential-)cyclic rule, random-based rule, Gauss-Southwell rule, maximum improvement rule, have been studied in [83, 113, 194]. An incremental MM was proposed in [153] to minimize sum-utilities composed of a large number of cost functions.

On the Convergence of MM/EM Due to the intimacy between MM and EM, convergence results of MM cannot be summarized independently from those of EM. Therefore, in the following we will not distinguish between EM and MM. Earlier studies including the proof of monotonicity of the sequence of the objective values along with the characterization of the limit points of the sequence generated my the EM/MM algorithm were presented in [66]. Results were refined in [27, 252], under the assumption that the objective function is differentiable and the iterates lie in the interior of the constraint set: it was shown that, if the surrogate function satisfies some mild conditions, all the limit points of the sequence generated by the EM/MM algorithm are stationary points of the problem. Conditions for the convergence of the whole sequence were given in [241]. A more comprehensive study of MM convergence with extensions to constrained problems and block-wise updates, can be found in [84, 113], where the surrogate only needs to upperbound the objective function locally. Convergence of (block-)MM applied to problems with non-smooth objective functions was studied in [194] (cf. Theorems 2.14 and 2.20). All the above results considered convex constraints. Convergence of MM under non-convex constraint were only partially investigated; examples include [222] and [59, 159, 263], the latter focusing on specific problems. In many applications, the EM/MM has been observed to be slow [110, 253]. Accelerated versions of EM/MM include: (i) [6, 102, 144, 201], based on modifying the step-size; (ii) [26, 117, 133, 134, 163], based on adjusting the search direction or inexact computation of the M-step; (iii) and [152, 242, 273] based on finding a fixed-point of the algorithmic mapping. We refer the readers to [118, 161, 230] for a comprehensive overview.

On the Choice of Surrogate Function and Related Algorithms The performance of MM depends crucially on the choice of the surrogate function. On one hand, a desirable surrogate function should be sufficiently "simple" so that the resulting minimization step can be carried out efficiently; on the other hand, it should also preserve the structure of the objective function, which is expected to enhance the practical convergence. Achieving a trade-off between these two goals is often a non-trivial task. Some guidelines on how to construct valid surrogate functions along with several examples were provided in [110, 153, 253]. Quite interestingly, under specific choices of the surrogate function, the resulting (block-)MM algorithm becomes an instance of well-knowns schemes, widely studied in the literature. Examples include the EM algorithm [66], the convex-concave procedure (CCCP) [148, 192, 264], the proximal algorithms [8, 15, 50, 53, 184], the cyclic minimization algorithm [226], and block coordinate descent-based schemes [250]. Finally, the idea of approximating a function using an upperbound has also been adopted to convexify nonconvex constraint functions; examples include the inner approximation algorithm [160], the CCCP procedure, and SCA-based algorithms [81, 211, 212].

Applications of MM/EM In the last few years there has been a growing interest in using the MM framework to solve a gamut of problems in several fields, including signal/image processing, machine learning, communications, and bioinformatics, just to name a few. A non-exhaustive list of specific applications include sparse linear/logistic regression [7, 21, 23, 36, 59, 87, 88, 127, 158], sparse (generalized) principal component analysis [120, 223, 263], matrix factorization/completion [85, 86, 119, 195], phase retrieval [175, 190], edge preserving regularizations in image processing [3, 92, 93], covariance estimation [12, 227, 229, 249, 274], sequence design [222, 254, 272], nonnegative quadratic programming [138, 142, 213], signomial programming [136], and sensor network localization [8, 9, 54, 179].

In the era of big data, the desiderata of MM has steered to low computational complexity and parallel/online computing. This raises new questions and challenges, including (i) how to design MM schemes with better convergence rate;

(ii) how to extend the MM framework to stochastic/online learning problems; and (iii) how to design surrogate functions exploiting problem structure and leading to closed form solutions and/or parallel/distributed updates.

3.3 Parallel Successive Convex Approximation Methods

This lecture goes beyond MM-based methods, addressing some limitations and challenges of the MM design. The MM approach calls for the surrogate function $\widetilde{V}$ to be a *global upperbound* of the objective function V; this requirement might limit the applicability and the effectiveness of the MM method, for several reasons.

First of all, when V does not have a favorable structure to exploit, it is not easy to build a surrogate $\widetilde{V}$ (possibly convex) that is a (tight) global upper bound of V. For instance, suppose that V is twice continuously differentiable and $\widetilde{V}$ is given by (3.28). A valid choice for L in (3.28) to meet the upperbound requirement is $L \geq \sup_{\mathbf{x}\in X} \|\nabla^2 V(\mathbf{x})\|_2$. However, computing $\sup_{\mathbf{x}\in X} \|\nabla^2 V(\mathbf{x})\|_2$ for unstructured V is not in general easy. In all such cases, a natural option is leveraging some upper bound of $\sup_{\mathbf{x}\in X} \|\nabla^2 V(\mathbf{x})\|_2$ (e.g., by uniformly bounding the largest eigenvalue of $\nabla^2 V$ on X). In practice, however, these bounds can be quite loose (much larger than $\sup_{\mathbf{x}\in X} \|\nabla^2 V(\mathbf{x})\|_2$), resulting in very slow instances of the MM algorithm.

Second, an upper approximation of V might be "too conservative" and not capturing well the "global behaviour" of V; this may affect the guarantees of the resulting MM algorithm. Figure 3.6 depicts such a situation: in Fig. 3.6a an upper convex approximation is chosen for $\widetilde{V}$ whereas in Fig. 3.6b the surrogate function is not an upper bound of V but it shares with V the same gradient at the base point while preserving the "low frequency component" of V. As shown in the figure, the two surrogates have different minimizers.

Third, building upper approximations of V that are also (additively) *block separable* is in general a challenging task, making the MM method not suitable

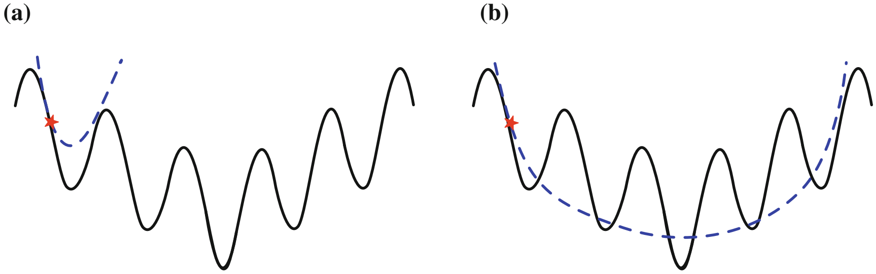


Fig. 3.6 Upper versus local approximation of the objective function. (**a**) MM approach: Upper approximation (dotted blue line) of the original function (solid black line) at the base point (red star). (**b**) SCA approach: Local approximation (dotted blue line) of the original function (solid black line) at the base point (red star)

for a parallel implementation, which instead is a desirable feature when dealing with large-scale problems. Block updates in MM schemes are possible, but in a sequential (e.g., cyclic) form, as discussed in Sect. 3.2.3 (Lecture I) for the block alternating MM algorithm (Algorithm 2). This contrasts with the intrinsic parallel nature of other algorithms for nonconvex problems, like the (proximal) gradient algorithm.

In this lecture we present a flexible SCA-based algorithmic framework that addresses the above issues. The method hinges on surrogate functions that (1) need not be a global upper bound of the objective function but only preserve locally its first order information; (2) are much easier to be constructed than upper approximations; and (3) lead to subproblems that, if constraints permit, are block separable and thus can be solved in parallel. However, the aforementioned surrogates need to be strongly convex, a property that is not required by MM algorithms. Furthermore, to guarantee convergence when the surrogate function is not a global upper bound of the objective function, a step-size is employed in the update of the variables. We begin by first describing a vanilla SCA algorithm in Sect. 3.3.2, where all the blocks are updated in parallel; several choices of the surrogate functions are discussed and convergence of the scheme under different step-size rules is provided. In Sect. 3.3.3, we extend the vanilla algorithm to the case where (1) a parallel selective update of the block variables is performed at each iteration—several deterministic and random-based selection rules will be considered—and (2) inexact solutions of the block-subproblems are used. This is motivated by applications where the parallel update of all blocks and/or the computation of the exact solutions of the subproblems at each iteration is not beneficial or affordable. In Sect. 3.3.4, "hybrid" parallel SCA methods are introduced, which combine deterministic and random-based block selection rules. These schemes have been shown to be very effective when dealing with huge-scale optimization problems. In Sect. 3.3.5, we apply the proposed (parallel) SCA methods to a variety of problems arising from applications in signal processing, machine learning, and communications, and compare their performance with those of existing MM methods. Finally, in Sect. 3.3.7 we overview the main literature and briefly discuss some extensions of the methods described in this lecture.

3.3.1 Problem Formulation

We study Problem (3.1), assuming the following structure for V:

$$\underset{\mathbf{x}\in X}{\text{minimize}} \quad V(\mathbf{x}) \triangleq F(\mathbf{x}) + G(\mathbf{x}). \tag{3.94}$$

Assumption 3.1 *Given Problem (3.94), we assume that*

1. *$X = X_1 \times \cdots X_n$, with each $\emptyset \neq X_i \subseteq \mathbb{R}^{m_i}$ closed and convex;*
2. *$F : O \to \mathbb{R}$ is C^1 on the open set $O \supseteq X$, and ∇F is L-Lipschitz on X;*

3. *$G : O \to \mathbb{R}$ is convex, possibly nonsmooth;*
4. *V is bounded from below on X.*

While Assumption 3.1 is slightly more restrictive than Assumption 2.12 (it requires G to be convex), it is general enough to cover a gamut of formulations, arising from applications in several fields; some examples are discussed in Sect. 3.3.1.1. On the other hand, under Assumption 3.1, we have more flexibility in the choice of the surrogate function and design of parallel algorithms, as it will be discussed shortly.

3.3.1.1 Some Motivating Applications

Many problems in fields as diverse as sensor networks, imaging, machine learning, data analysis, genomics, and geophysics, can be formulated as Problem (3.94) and satisfy Assumption 3.1. Some illustrative examples are documented next; see Sect. 3.3.5 for more details and some numerical results.

Example #1−LASSO Consider a linear regression model with q predictor/feature-response pairs $\{(z_i, \mathbf{a}_i)\}_{i=1}^q$, where $\mathbf{a}_i \in \mathbb{R}^m$ is a m-dimensional vector of features or predictors, and z_i is the associated response variable. Let $\mathbf{z} = (z_1, \ldots, z_q)^T$ denote the q-dimensional vector of response, and $\mathbf{A} \in \mathbb{R}^{q\times m}$ be the matrix with $\mathbf{a}_i$ in its i-th row. Then, the LASSO problem in the so-called Lagrangian form [235], aiming at finding a sparse vector of regression weights $\mathbf{x} \in \mathbb{R}^m$, is an instance of Problem (3.94), with $F(\mathbf{x}) = \|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$ and $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$, where λ is a positive given constant.

Example #2−Group LASSO There are many regression problems wherein the covariates within a group become nonzero (or zero) simultaneously. In such settings, it is natural 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 penalties. Specifically, consider the regression vector $\mathbf{x} \in \mathbb{R}^m$ possessing the group sparse pattern $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T]^T$, i.e., all the elements of $\mathbf{x}_i$ either take large values or close to zero [262]. A widely used group LASSO formulation is the instance of Problem (3.94), with $F(\mathbf{x}) = \|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2$ and $G(\mathbf{x}) = \lambda \sum_{i=1}^n \|\mathbf{x}_i\|_2$, and $\lambda > 0$.

Example #3−Sparse Logistic Regression Logistic regression has been popular in biomedical research for half a century, and has recently gained popularity to model a wider range of data. Given the training set $\{\mathbf{z}_i, w_i\}_{i=1}^q$, where $\mathbf{z}_i \in \mathbb{R}^m$ is the feature vector and $w_i \in \{-1, 1\}$ is the label of the i-th sample, the logistic regression problem based on a linear logistic model consists in minimizing the negative log likelihood [162, 214] $F(\mathbf{x}) = (1/q) \cdot \sum_{i=1}^q \log(1 + e^{-w_i \cdot \mathbf{z}_i^T \mathbf{x}})$; regularizations can be introduced, e.g., in the form $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$ (or $G(\mathbf{x}) = \lambda \sum_{i=1}^n \|\mathbf{x}_i\|_2$), with $\lambda > 0$. Clearly, this is an instance of Problem (3.94).

Example #4−Dictionary Learning Dictionary learning is an unsupervised learning problem that consists in finding a basis $\mathbf{D} \in \mathbb{R}^{q\times m}$—called dictionary—whereby data $\mathbf{z}_i \in \mathbb{R}^q$ can be sparsely represented by coefficients $\mathbf{x}_i \in \mathbb{R}^m$. Let $\mathbf{Z} \in \mathbb{R}^{q\times I}$

and $\mathbf{Z} \in \mathbb{R}^{m \times I}$ be the data and representation matrix whose columns are the data vectors $\mathbf{z}_i$ and coefficients $\mathbf{x}_i$, respectively. The DL problem is the instance of Problem (3.94), with $F(\mathbf{D}, \mathbf{X}) = \|\mathbf{Z} - \mathbf{D}\mathbf{X}\|_F^2$ and $G(\mathbf{X}) = \lambda \|\mathbf{X}\|_1$, $X = \{(\mathbf{D}, \mathbf{X}) \in \mathbb{R}^{q \times m} \times \mathbb{R}^{m \times I} : \|\mathbf{D}\,\mathbf{e}_i\|^2 \leq \alpha_i, \forall i = 1, \ldots, m\}$, where $\mathbf{e}_i$ is the i-th 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 this is an example of $F(\mathbf{D}, \mathbf{X})$ that is not jointly convex in $(\mathbf{D}, \mathbf{X})$, but bi-convex (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 parametric empirical risk minimization problem aims at finding the model $h : \mathbb{R}^m \ni X \to \mathbb{R}^q$, parameterized by $\mathbf{x}$, that minimizes the risk function $\sum_{i=1}^{I} \ell\left(h\left(\mathbf{x}; D_i\right)\right)$, where $\ell : \mathbb{R}^q \to \mathbb{R}$ is a loss function, measuring the mismatch between the model and the data. This optimization problem is a special case of Problem (3.94), with $F(\mathbf{x}) = \sum_{i=1}^{I} f_i(\mathbf{x})$ and $f_i(\mathbf{x}) \triangleq \ell\big(h\left(\mathbf{x}; D_i\right)\big)$. To promote sparsity, one can add in the objective function a regularizer G using, e.g., any of the surrogates of the ℓ_0 cardinality function listed in Table 3.1 (cf. Sect. 3.2.4.1). By absorbing the smooth G^- part in F, the resulting regularized empirical risk minimization problem is still written in the form (3.94). Note that this general problem contains the previous examples as special cases, and generalizes them by incorporating also nonconvex regularizers.

All the above examples contain separable G. Some applications involving nonseparable G are discussed next.

Example #6—Robust Linear Regression Linear least-squares estimates can behave badly when the error distribution is not normal, particularly when the errors are heavy-tailed. One remedy is to remove influential observations from the least-squares fit. Another approach, termed *robust regression*, is to use a fitting criterion that is not as vulnerable as least squares to unusual (outliers) data. Consider the system model as in Example #1; a simple example of robustification is replacing the ℓ_2 norm loss function with the ℓ_1 norm, which leads to the instance of Problem (3.94), with $F(\mathbf{x}) = 0$ and $G(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{v}\|_1$ [98].

Example #7—The Fermat-Weber Problem This problem consists in finding $\mathbf{x} \in \mathbb{R}^n$ such that the weighted sum of distances between $\mathbf{x}$ and the I anchors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_I$ is minimized [76]. It can be formulated as Problem (3.94), with $F(\mathbf{x}) = 0$ and $G(\mathbf{x}) = \sum_{i=1}^{I} \omega_i \|\mathbf{A}_i\mathbf{x} - \mathbf{v}_i\|_2$, $X = \mathbb{R}^n$, where $\mathbf{A}_i \in \mathbb{R}^{q \times n}$, $\mathbf{v}_i \in \mathbb{R}^q$, and $\omega_i > 0$ are given constants, for all i.

Example #8—The Total Variation (TV) Image Reconstruction TV minimizing models have become a successful methodology for image processing, including denoising, deconvolution, and restoration, to name a few [40]. The noise-free discrete TV reconstruction problem can be formulated as Problem (3.94) with $F(\mathbf{X}) = \|\mathbf{Z} - \mathbf{A}\mathbf{X}\|^2$ and $G(\mathbf{X}) = \lambda \cdot \mathrm{TV}(\mathbf{X})$, $X = \mathbb{R}^{m \times m}$, where $\mathbf{A} \in \mathbb{R}^{q \times m}$, $\mathbf{X} \in \mathbb{R}^{m \times m}$, $\mathbf{Z} \in \mathbb{R}^{q \times m}$, and $\mathrm{TV}(\mathbf{X}) \triangleq \sum_{i,j=1}^{m} \|\nabla_{ij}(\mathbf{X})\|_p$ is the discrete total variational semi-norm of $\mathbf{X}$, with $p = 1$ or 2 and $\nabla_{ij}(\mathbf{X})$ being the discrete gradient of $\mathbf{X}$ defined as $\nabla_{ij}(\mathbf{X}) \triangleq [(\nabla_{ij}^{(1)}(\mathbf{X})), (\nabla_{ij}^{(2)}(\mathbf{X}))]$, with

$$\nabla_{ij}^{(1)}(\mathbf{X}) \triangleq \begin{cases} X_{i+1,j} - X_{i,j}, & \text{if } i < m, \\ 0, & i = m; \end{cases} \qquad \nabla_{ij}^{(2)}(\mathbf{X})^{(2)} \triangleq \begin{cases} X_{i,j+1} - X_{i,j}, & \text{if } j < m, \\ 0, & j = m; \end{cases}$$

for all $i, j = 1, \ldots, m$.

3.3.2 Parallel SCA: Vanilla Version

We begin introducing a vanilla version of the parallel SCA framework wherein all the block variables are updated in parallel; generalizations of this scheme will be considered in Sect. 3.3.3.

The most natural parallel (Jacobi-type) solution method one can employ is solving (3.94) *blockwise* and *in parallel*: given $\mathbf{x}^k$, all the (block) variables $\mathbf{x}_i$ are updated *simultaneously* by solving the following subproblems

$$\mathbf{x}_i^{k+1} \in \underset{\mathbf{x}_i \in X_i}{\operatorname{argmin}} \left\{ F(\mathbf{x}_i, \mathbf{x}_{-i}^k) + G(\mathbf{x}_i, \mathbf{x}_{-i}^k) \right\}, \qquad \forall i \in N \triangleq \{1, \ldots, n\}. \tag{3.95}$$

Unfortunately, this method converges only under very restrictive conditions [16] that are seldom verified in practice (even in the absence of the nonsmooth function G). Furthermore, the exact computation of $\mathbf{x}_i^{k+1}$ is in general difficult, due to the nonconvexity of F. To cope with these two issues, the proposed approach consists in solving for each block instead

$$\widehat{\mathbf{x}}_i(\mathbf{x}^k) \triangleq \underset{\mathbf{x}_i \in X_i}{\operatorname{argmin}} \ \widetilde{F}_i(\mathbf{x}_i \mid \mathbf{x}^k) + G(\mathbf{x}_i, \mathbf{x}_{-i}^k), \tag{3.96}$$

and then setting

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \gamma^k \left(\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k \right). \tag{3.97}$$

In (3.96), $\widetilde{F}_i\left(\bullet \mid \mathbf{x}^k\right)$ represents a strongly convex surrogate replacing $F(\bullet, \mathbf{x}_{-i}^k)$, and in (3.97) a step-size $\gamma^k \in (0, 1]$ is introduced to control the "length" of the update along the direction $\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k$. The step-size is needed if one does not require that the surrogate $\widetilde{F}_i\left(\bullet \mid \mathbf{x}^k\right)$ is a global upper bound of $F(\bullet, \mathbf{x}_{-i}^k)$ (as in the MM algorithm).

The surrogate function $\widetilde{F}_i$ has the following properties ($\nabla \widetilde{F}_i$ denotes the partial gradient of $\widetilde{F}_i$ with respect to the first argument).

Assumption 3.2 *Each function* $\widetilde{F}_i : O_i \times O \to \mathbb{R}$ *satisfies the following conditions:*

1. $\widetilde{F}_i\,(\bullet \mid \mathbf{y})$ *is* τ_i*-strongly convex on* X_i*, for all* $\mathbf{y} \in X$;
2. $\widetilde{F}_i\,(\bullet \mid \mathbf{y})$ *is differentiable on* O_i *and* $\nabla_{\mathbf{y}_i} F\,(\mathbf{y}) = \nabla \widetilde{F}_i\,(\mathbf{y}_i \mid \mathbf{y})$*, for all* $\mathbf{y} \in X$.

Stronger convergence results can be obtained under the following additional assumptions.

Assumption 3.3 $\nabla \widetilde{F}_i\,(\mathbf{x}_i \mid \bullet)$ *is* $\widetilde{L}_i$*-Lipschitz on* X*, for all* $\mathbf{x}_i \in X_i$.

Assumption 3.3* $\widetilde{F}_i\,(\bullet \mid \bullet)$ *is continuous on* $O_i \times O$.

Assumption 3.2 states that $\widetilde{F}_i$ should be regarded as a (simple) convex approximation of F at the current iterate $\mathbf{x}^k$ that preserves the first order properties of F. Note that, as anticipated, $\widetilde{F}_i$ need not be a global upper bound of $F(\bullet, \mathbf{x}_{-i})$. Furthermore, the above assumptions guarantee that the mapping $\widehat{\mathbf{x}}(\mathbf{x}) \triangleq (\widehat{\mathbf{x}}_i(\mathbf{x}))_{i=1}^n$, with $\widehat{\mathbf{x}}_i : X \to X_i$ defined in (3.96), enjoys the following properties that are instrumental to prove convergence of the algorithm to stationary solutions of Problem (3.94).

Lemma 3.4 (Continuity of $\widehat{\mathbf{x}}$) *Consider Problem* (3.94) *under Assumption 3.1. The following hold:*

*(a) If Assumptions 3.2 and 3.3*are satisfied,* $\widehat{\mathbf{x}}(\bullet)$ *is continuous on* X;
(b) If Assumptions 3.2 and 3.3 are satisfied, and G is separable, $\widehat{\mathbf{x}}(\bullet)$ *is Lipschitz continuous on* X.

Proof See Appendix—Sect. 3.3.6.1.

Other properties of the best-response (e.g., in the presence of nonconvex constraints) can be found in [81, 211].

The described algorithm is based on solving in parallel the subproblems in (3.96), converging thus to fixed-points of the mapping $\widehat{\mathbf{x}}(\bullet)$. It is then natural to ask which relation exists between these fixed points and the (d-)stationary solutions of Problem (3.94). The following lemma addresses this question.

Lemma 3.5 (On the Fixed-Points of $\widehat{\mathbf{x}}$) *Given Problem* (3.94) *under Assumption 3.1, let each* $\widetilde{F}_i$ *in (3.96) be chosen according to Assumption 3.2. The following hold.*

(a) The set of fixed-points of $\widehat{\mathbf{x}}(\bullet)$ *coincides with that of the coordinate-wise d-stationary solutions of* (3.94);
(b) If, in addition, G is separable—$G\,(\mathbf{x}) = \sum_{i=1}^n g_i\,(\mathbf{x}_i)$*—then the set of fixed-points coincides with that of d-stationary solutions of* (3.94).

Proof The proof of statement (b) can be found in [79, Proposition 8]. The proof of statement (a) follows similar steps and thus is omitted. □

To complete the description of the algorithm, we need to specify how to choose the step-size $\gamma^k \in (0, 1]$ in (3.97). Any of the following standard rules can be used.

Assumption 3.6 *The step-size sequence* $\{\gamma^k \in (0,1]\}_{k\in\mathbb{N}_+}$ *satisfies any of the following rules:*

1. ***Bounded step-size**:* $0 < \liminf_{k\to\infty}\gamma^k \leq \limsup_{k\to\infty}\gamma^k < 2\,c_\tau/L$, *where* $c_\tau \triangleq \min_{i=1,\ldots,n}\tau_i$ *(cf. Assumption 3.2.1);*
2. ***Diminishing step-size**:* $\sum_{k=0}^{\infty}\gamma^k = +\infty$ *and* $\sum_{k=0}^{\infty}\left(\gamma^k\right)^2 < +\infty$*;*
3. ***Line search**: let* $\alpha, \delta \in (0,1)$*, choose* $\gamma^k = (\delta)^{t_k}$*, where* t_k *is the smallest nonnegative integer such that*

$$V\left(\mathbf{x}^k + \gamma^k\,\Delta\widehat{\mathbf{x}}^k\right) \leq V(\mathbf{x}^k) + \alpha\cdot\gamma^k\left(\nabla F\left(\mathbf{x}^k\right)^T\Delta\widehat{\mathbf{x}}^k + \sum_{i=1}^{n}\left(G\left(\widehat{\mathbf{x}}_i(\mathbf{x}^k),\mathbf{x}_{-i}^k\right) - G\left(\mathbf{x}^k\right)\right)\right) \tag{3.98}$$

with $\Delta\widehat{\mathbf{x}}^k \triangleq \widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k$.

The parallel SCA procedure is summarized in Algorithm 5.

Convergence of Algorithm 5 is stated below; Theorem 3.7 deals with nonseparable G while Theorem 3.8 specializes the results to the case of (block) separable G. The proof of the theorems is omitted, because they are special cases of more general results (Theorems 3.12 and 3.13) that will be introduced in Sect. 3.3.3.

Theorem 3.7 *Consider Problem (3.94) under Assumption 3.1. Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 5, with each* $\widetilde{F}_i$ *chosen according to Assumptions 3.2 and 3.3*; and let the step-size* $\gamma^k \in (0, 1/n]$*, for all* $k \in \mathbb{N}_+$*. Then, there hold:*

(a) If $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *is chosen according to Assumption 3.6.2 (diminishing rule), then*

$$\liminf_{k\to\infty}\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| = 0; \tag{3.100}$$

Algorithm 5 Parallel successive convex approximation (p-SCA)

```
Data : $\mathbf{x}^0 \in X$, $\{\gamma^k \in (0,1]\}_{k\in\mathbb{N}_+}$.
       Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : For all $i \in N$, solve in parallel

        $\widehat{\mathbf{x}}_i(\mathbf{x}^k) \triangleq \underset{\mathbf{x}_i\in X_i}{\text{argmin}}\ \widetilde{F}_i(\mathbf{x}_i \mid \mathbf{x}^k) + G(\mathbf{x}_i, \mathbf{x}_{-i}^k);$        (3.99)

(S.3) : Set $\mathbf{x}^{k+1} \triangleq \mathbf{x}^k + \gamma^k\,\left(\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\right)$;
(S.4) : $k \leftarrow k+1$, and go to (S.1).
```

(b) If $\{\gamma^k\}_{k\in\mathbb{N}_+}$ is chosen according to Assumption 3.6.1 (bounded condition) or Assumption 3.6.3 (line-search),

$$\lim_{k\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| = 0. \tag{3.101}$$

Theorem 3.8 *Consider Problem* (3.94) *under Assumption 3.1 and* $G(\mathbf{x}) = \sum_{i=1}^n g_i(\mathbf{x}_i)$*, with each* $g_i : O_i \to \mathbb{R}$ *being convex (possibly nonsmooth). Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 5, with each* $\widetilde{F}_i$ *chosen according to Assumption 3.2; and let the step-size* $\gamma^k \in (0, 1]$*, for all* $k \in \mathbb{N}_+$*. Then, there hold:*

(a) If $\widetilde{F}_i$ *satisfies Assumption 3.3*and* $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *is chosen according to Assumption 3.6.2 (diminishing rule), then (3.100) holds.*

(b) Suppose that either one of the following conditions is satisfied:

(i) Each $\widetilde{F}_i$ *satisfies Assumption 3.3* and* $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *is chosen according to Assumption 3.6.1 (bounded condition) or Assumption 3.6.3 (line-search);*

(ii) Each $\widetilde{F}_i$ *satisfies Assumption 3.3 and* $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *is chosen according to Assumption 3.6.2 (diminishing rule).*

Then, (3.101) holds.

The above theorems establish the following connection between the limit points of the sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ and the stationary points of Problem (3.94). By (3.101) and the continuity of $\widehat{\mathbf{x}}(\bullet)$ (cf. Lemma 3.4), we infer that every limit point $\mathbf{x}^\infty$ of $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ (if exists) is a fixed point of $\widehat{\mathbf{x}}(\bullet)$ and thus, by Lemma 3.5, it is a coordinate-wise d-stationary solution of Problem (3.94). If, in addition, G is separable, $\mathbf{x}^\infty$ is a d-stationary solution of (3.94). When (3.100) holds instead, there exists a subsequence $\{\mathbf{x}^{k_t}\}_{t\in\mathbb{N}_+}$ of $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ such that $\lim_{t\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^{k_t}) - \mathbf{x}^{k_t}\| = 0$, and the aforementioned connection with the (coordinate-wise) stationary solutions of (3.94) holds for every limit point of such a subsequence. The existence of a limit point of $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ is guaranteed under standard extra conditions on the feasible set X—e.g., boundedness—or on the objective function V—e.g., coercivity on X.

3.3.2.1 Discussion on Algorithm 5

Algorithm 5 represents a gamut of parallel solution methods, each of them corresponding to a specific choice of the surrogate functions $\widetilde{F}_i$ and step-size rule. Theorems above provide a unified set of conditions for the convergence of all such schemes. Some representative choices for $\widetilde{F}_i$ and γ^k are discussed next.

On the Choice of the Surrogate $\boldsymbol{\widetilde{F}_i}$ Some examples of surrogate functions satisfying Assumption 3.2 for specific F are the following.

1) Block-Wise Convexity Suppose $F(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is convex in each block $\mathbf{x}_i$ separately (but not necessarily jointly). A natural approximation for such an F exploring its "partial" convexity is: given $\mathbf{y} = (\mathbf{y}_i)_{i=1}^n \in X$,

$$\widetilde{F}(\mathbf{x} \,|\, \mathbf{y}) = \sum_{i=1}^{n} \widetilde{F}_i(\mathbf{x}_i \,|\, \mathbf{y}), \tag{3.102}$$

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

$$\widetilde{F}_i(\mathbf{x}_i \,|\, \mathbf{y}) \triangleq F(\mathbf{x}_i, \mathbf{y}_{-i}) + \frac{\tau_i}{2}(\mathbf{x}_i - \mathbf{y}_i)^T \mathbf{H}_i\, (\mathbf{x}_i - \mathbf{y}_i), \tag{3.103}$$

where τ_i is any positive constant, and $\mathbf{H}_i$ is any $m_i \times m_i$ positive definite matrix (of course, one can always choose $\mathbf{H}_i = \mathbf{I}$). The quadratic term in (3.103) can be set to zero if $F(\bullet, \mathbf{y}_{-i})$ is strongly convex on X_i, for all $\mathbf{y}_{-i} \in X_{-i} \triangleq X_1 \times \cdots \times X_{i-1} \times X_{i+1} \times \cdots \times X_n$.

2) (Proximal) Gradient-Like Approximations If no convexity is present in F, mimicking proximal-gradient methods, a valid choice of $\widetilde{F}$ is the first order approximation of F (plus a quadratic regularization), that is, $\widetilde{F}$ is given by (3.102), with each

$$\widetilde{F}_i(\mathbf{x}_i \,|\, \mathbf{y}) \triangleq \nabla_{\mathbf{x}_i} F(\mathbf{y})^T (\mathbf{x}_i - \mathbf{y}_i) + \frac{\tau_i}{2}\|\mathbf{x}_i - \mathbf{y}_i\|^2. \tag{3.104}$$

Note that the above approximation has the same form of the one used in the MM algorithm [cf. (3.28)], with the difference that in (3.104) τ_i can be any positive number (and not necessarily larger than L_i). When $\tau_i < L_i$, $\widetilde{F}_i$ in (3.104) is no longer a global upper bound of $F(\bullet, \mathbf{x}_{-i})$. In such cases, differently from the MM algorithm, step-size $\gamma^k = 1$ in Algorithm 5 may not be used at each k.

3) Sum-Utility Function Suppose that $F(\mathbf{x}) \triangleq \sum_{i=1}^{I} f_i(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. This structure arises, e.g., in multi-agent systems wherein f_i is the cost function of agent i that controls its own block variables $\mathbf{x}_i \in X_i$. In many application it is common that the cost functions f_i are convex in some agents' variables (cf. Sect. 3.3.5). To exploit this partial convexity, let us introduce the following set

$$\tilde{C}_i \triangleq \left\{ j \;:\; f_j(\bullet, \mathbf{x}_{-i}) \text{ is convex, } \forall \mathbf{x}_{-i} \in X_{-i} \right\}, \tag{3.105}$$

which represents the set of indices of all the functions f_j that are convex in $\mathbf{x}_i$, for any feasible $\mathbf{x}_{-i}$; and let $C_i \subseteq \tilde{C}_i$ be any subset of $\tilde{C}_i$. Then, the following surrogate function satisfies Assumption 3.2 while exploiting the partial convexity of F (if any): given $\mathbf{y} = (\mathbf{y}_i)_{i=1}^n \in X$,

$$\widetilde{F}(\mathbf{x} \,|\, \mathbf{y}) = \sum_{i=1}^{n} \widetilde{F}_{C_i}(\mathbf{x}_i \,|\, \mathbf{y}),$$

with each $\widetilde{F}_{C_i}$ defined as

$$\begin{aligned}\widetilde{F}_{C_i}(\mathbf{x}_i \,|\, \mathbf{y}) \triangleq \sum_{j\in C_i} f_j(\mathbf{x}_i, \mathbf{y}_{-i}) + \sum_{\ell\notin C_i} \nabla_{\mathbf{x}_i} f_\ell(\mathbf{y})^T(\mathbf{x}_i - \mathbf{y}_i) \\ + \frac{\tau_i}{2}\,(\mathbf{x}_i - \mathbf{y}_i)^T \mathbf{H}_i\,(\mathbf{x}_i - \mathbf{y}_i),\end{aligned} \tag{3.106}$$

where $\mathbf{H}_i$ is any $m_i \times m_i$ positive definite matrix. Roughly speaking, for each agent i, the above approximation function preserves the convex part of F w.r.t. $\mathbf{x}_i$ while it linearizes the nonconvex part.

4) Product of Functions Consider an F that is written as the product of functions (see [212] for some examples); without loss of generality, here we study only the case of product of two functions. Let $F(\mathbf{x}) = f_1(\mathbf{x})f_2(\mathbf{x})$, with f_1 and f_2 convex and non-negative on X; if the functions are just block-wise convex, the proposed approach can be readily extended. In view of the expression of the gradient of F and Assumption 3.2.2, $\nabla_{\mathbf{x}}F = f_2\nabla_{\mathbf{x}}f_1 + f_1\nabla_{\mathbf{x}}f_2$, it seems natural to consider the following approximation: given $\mathbf{y} \in X$,

$$\widetilde{F}(\mathbf{x}\,|\,\mathbf{y}) = f_1(\mathbf{x})f_2(\mathbf{y}) + f_1(\mathbf{y})f_2(\mathbf{x}) + \frac{\tau_i}{2}(\mathbf{x}-\mathbf{y})^T\mathbf{H}\,(\mathbf{x}-\mathbf{y}),$$

where, as usual, $\mathbf{H}$ is a positive definite matrix; this term can be omitted if f_1 and f_2 are positive on the feasible set and $f_1 + f_2$ is strongly convex (for example if one of the two functions is strongly convex). In case f_1 and f_2 are still positive but not necessarily convex, we can use the expression

$$\widetilde{F}(\mathbf{x}\,|\,\mathbf{y}) = \widetilde{f}_1(\mathbf{x}\,|\,\mathbf{y})\, f_2(\mathbf{y}) + f_1(\mathbf{y})\, \widetilde{f}_2(\mathbf{x}\,|\,\mathbf{y}),$$

where $\widetilde{f}_1$ and $\widetilde{f}_2$ are any legitimate surrogates of f_1 and f_2, respectively. Finally, if f_1 and f_2 can take nonpositive values, introducing $h_1(\mathbf{x}\,|\,\mathbf{y}) \triangleq \widetilde{f}_1(\mathbf{x}\,|\,\mathbf{y})\, f_2(\mathbf{y})$ and $h_2(\mathbf{x}\,|\,\mathbf{y}) \triangleq f_1(\mathbf{y})\, \widetilde{f}_2(\mathbf{x}\,|\,\mathbf{y})$, one can write

$$\widetilde{F}(\mathbf{x}\,|\,\mathbf{y}) = \widetilde{h}_1(\mathbf{x}\,|\,\mathbf{y}) + \widetilde{h}_2(\mathbf{x}\,|\,\mathbf{y}),$$

where $\widetilde{h}_1$ (resp. $\widetilde{f}_1$) and $\widetilde{h}_2$ (resp. $\widetilde{f}_2$) are legitimate surrogates of h_1 (resp. f_1) and h_2 (resp. f_2), respectively. Note that in this last case, we no longer need the quadratic term because it is already included in the approximations $\widetilde{f}_1$ and $\widetilde{f}_2$, and $\widetilde{h}_1$ and $\widetilde{h}_2$, respectively. As the final remark, note that the functions F discussed above belong to a class of nonconvex functions for which it does not seem possible to find a global convex upper bound; therefore, the MM techniques introduced in Lecture I are not readily applicable.

5) Composition of Functions Let $F(\mathbf{x}) = h(\mathbf{f}(\mathbf{x}))$, where $h : \mathbb{R}^q \to \mathbb{R}$ is a finite convex smooth function such that $h(u_1, \ldots, u_q)$ is nondecreasing in each component, and $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^q$ is a smooth mapping, with $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_q(\mathbf{x}))^T$

and f_i not necessarily convex. Examples of functions F belonging to such a class are those arising from nonlinear least square-based problems, that is, $F(\mathbf{x}) = \|\mathbf{f}(\mathbf{x})\|^2$, where $\mathbf{f}(\mathbf{x})$ is a smooth nonlinear (possibly) nonconvex map. A convex approximation satisfying Assumption 3.2 is: given $\mathbf{y} \in \mathbb{R}^m$,

$$\widetilde{F}(\mathbf{x} \,|\, \mathbf{y}) \triangleq h\left(\mathbf{f}(\mathbf{y}) + \nabla\mathbf{f}(\mathbf{y})(\mathbf{x} - \mathbf{y})\right) + \frac{\tau}{2}\,\|\mathbf{x} - \mathbf{y}\|^2, \tag{3.107}$$

where $\nabla\mathbf{f}(\mathbf{y})$ denotes the Jacobian of $\mathbf{f}$ at $\mathbf{y}$.

On the Choice of the Step-Size γ^k Some possible choices for the step-size satisfying Assumption 3.6 are the following.

1) Bounded Step-Size Assumption 3.6.1 requires the step-size to be eventually in the interval $[\delta, 2c_\tau/L)$, for some $\delta > 0$. A simple choice is $\gamma^k = \gamma > 0$, with $2\,c_\tau/\gamma > L$ and for all k. This simple (but conservative) condition imposes a constraint only on the ratio c_τ/γ, leaving free the choice of one of the two parameters. An interesting case is when the proximal gradient-like approximation in (3.104) is used. Setting therein each $\tau_i > L$ allows one to use step-size $\gamma = 1$, obtaining thus the MM algorithm as a special case.

2) Diminishing Step-Size In scenarios where the knowledge of system parameters, e.g., L, is not available, one can use a diminishing step-size satisfying Assumption 3.6.2. Two examples of diminishing step-size rules are:

$$\gamma^k = \gamma^{k-1}\left(1 - \epsilon\,\gamma^{k-1}\right), \quad k = 1, \ldots, \quad \gamma^0 < 1/\epsilon; \tag{3.108}$$

$$\gamma^k = \frac{\gamma^{k-1} + \alpha(k)}{1 + \beta(k)}, \qquad k = 1, \ldots, \quad \gamma^0 = 1; \tag{3.109}$$

where in (3.108) $\epsilon \in (0, 1)$ is a given constant, whereas in (3.109) $\alpha(k)$ and $\beta(k)$ are two nonnegative real functions of $k \geq 1$ such that: (i) $0 \leq \alpha(k) \leq \beta(k)$; and (ii) $\alpha(k)/\beta(k) \to 0$ as $k \to \infty$ while $\sum_k \left(\alpha(k)/\beta(k)\right) = \infty$. Examples of such $\alpha(k)$ and $\beta(k)$ are: $\alpha(k) = \alpha$ or $\alpha(k) = \log(k)^\alpha$, and $\beta(k) = \beta \cdot k$ or $\beta(k) = \beta \cdot \sqrt{k}$, where α, β are given constants satisfying $\alpha \in (0, 1)$, $\beta \in (0, 1)$, and $\alpha \leq \beta$.

3) Line Search Assumption 3.6.3 is an Armijo-like line-search that employs a backtracking procedure to find the largest γ^k generating sufficient descent of the objective function at $\mathbf{x}^k$ along the direction $\Delta\widehat{\mathbf{x}}^k$. Of course, using a step-size generated by line-search will likely be more efficient in terms of iterations than the one based on diminishing step-size rules. However, as a trade-off, performing line-search requires evaluating the objective function multiple times per iteration; resulting thus in more costly iterations. Furthermore, performing a line-search on a multicore architecture requires some shared memory and coordination among the cores/processors.

3.3.3 Parallel SCA: Selective Updates

The parallel SCA algorithm introduced in Sect. 3.3.2 consists in updating at each iteration *all* the block variables by computing the solutions of subproblems in the form (3.96). In this section, we generalize the algorithm by (i) unlocking parallel updates of a *subset* of all the blocks at a time, and (ii) allowing inexact computations of the solution of each subproblem. This is motivated by applications where computing the exact solutions of large-scale subproblems is computationally demanding and/or updating all the block variables at each iteration is not beneficial; see Sect. 3.3.5 for some examples.

Inexact Solutions Subproblems in (3.96) are solved inexactly by computing $\mathbf{z}^k$ satisfying $\|\mathbf{z}_i^k - \widehat{\mathbf{x}}_i(\mathbf{x}^k)\| \leq \varepsilon_i^k$, where ε_i^k is the desired accuracy (to be properly chosen). Some conditions on the inexact solution (and thus associated error) are needed to guarantee convergence on the resulting algorithm, as stated next.

Assumption 3.9 *Given $\widehat{\mathbf{x}}_i(\mathbf{x}^k)$ as defined in (3.96), the inexact solutions $\mathbf{z}_i^k$ satisfies: for all $i = 1, \ldots, n$,*

1. $\|\mathbf{z}_i^k - \widehat{\mathbf{x}}_i\left(\mathbf{x}^k\right)\| \leq \varepsilon_i^k$ *and* $\lim_{k\to\infty} \varepsilon_i^k = 0$*;*
2. $\widetilde{F}_i\left(\mathbf{z}_i^k \,|\, \mathbf{x}^k\right) + G\left(\mathbf{z}_i^k, \mathbf{x}_{-i}^k\right) \leq \widetilde{F}_i\left(\mathbf{x}_i^k \,|\, \mathbf{x}^k\right) + G\left(\mathbf{x}^k\right)$.

The above conditions are quite natural: Assumption 3.9.1 states that the error must asymptotically vanish [subproblems (3.96) need to be solved with increasing accuracy] while Assumption 3.9.2 requires that $\mathbf{z}_i^k$ generates a decrease in the objective function of subproblem (3.96) at iteration k [$\mathbf{z}_i^k$ need not be a minimizer of (3.96)].

Updating Only Some Blocks At each iteration k a suitable chosen subset of blocks—say $S^k \subseteq N$ [recall that $N \triangleq \{1, \ldots, n\}$]—is selected and updated by computing for each block $i \in S^k$ an inexact solution $\mathbf{z}_i^k$ of the associated subproblem (3.96): given $\mathbf{x}^k$ and S^k, let

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

Several options are possible for the block selection rule S^k. For instance, one can choose the blocks to update according to some deterministic (cyclic) or random-based rule. Greedy-like schemes—updating at each iteration only the blocks that are "far away" from the optimum—have been shown to be quite effective in some applications. Finally, one can also adopt hybrid rules that properly combine the aforementioned selection methods. For instance, one can first select a subset of blocks uniformly at random, and then within such a pool updating only the blocks resulting from a greedy rule. Of course some minimal conditions on the updating rule are necessary to guarantee convergence, as stated below.

Assumption 3.10 *The block selection satisfies one of the following rules:*

1. ***Essentially cyclic rule**: S^k is selected so that $\bigcup_{s=0}^{T-1} S^{k+s} = N$, for all $k \in \mathbb{N}_+$ and some finite $T > 0$;*
2. ***Greedy rule**: Each S^k contains at least one index i such that*

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

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

$$\underline{s}_i \cdot \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\| \leq E_i(\mathbf{x}^k) \leq \bar{s}_i \cdot \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|, \tag{3.110}$$

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

3. ***Random-based rule**: The sets S^k are realizations of independent random sets $\boldsymbol{S}^k$ taking value in the power set of N, such that $\mathbb{P}(i \in \boldsymbol{S}^k) \geq p$, for all $i = 1, \ldots, n$ and $k \in \mathbb{N}_+$, and some $p > 0$.*

The above selection rules are quite general and have a natural interpretation. The cyclic rule [Assumption 3.10.1] requires that all the blocks are updated (at least) once within T consecutive iterations, where T is an arbitrary (finite) integer. Assumption 3.10.2 is a greedy-based rule: only the blocks that are "far" from the optimum need to be updated at each iteration; $E_i\left(\mathbf{x}^k\right)$ can be viewed as a local measure of the distance of block i from optimality. The greedy rule in Assumption 3.10.2 thus calls for the updates of one block that is within a fraction ρ from the largest distance $E_i\left(\mathbf{x}^k\right)$. Some examples of valid error bound functions E_i are discussed in Sect. 3.3.3.1. Finally, Assumption 3.10.3 is a random selection rule: blocks can be selected according to any probability distribution as long as they have a positive probability to be picked. Specific rules satisfying Assumption 3.10 are discussed in Sect. 3.3.3.1.

The described parallel selective SCA method is summarized in Algorithm 6, and termed "inexact FLEXible parallel sca Algorithm" (FLEXA).

To complete the description of the algorithm, we need to specify how to choose the step-size γ^k in Step 4. Assumption 3.11 below provides some standard rules.

Algorithm 6 Inexact flexible parallel SCA algorithm (FLEXA)

```
Data : $\mathbf{x}^0 \in X$, $\{\gamma^k \in (0,1]\}_{k \in \mathbb{N}_+}$, $\varepsilon_i^k \geq 0$, for all $i \in N$ and $k \in \mathbb{N}_+$, $\rho \in (0,1]$.
       Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Choose a set $S^k$ according to any of the rules in Assumption 3.10;
(S.3) : For all $i \in S^k$, solve (3.96) with accuracy $\varepsilon_i^k$ :
                 find $\mathbf{z}_i^k \in X_i$ s.t. $\|\mathbf{z}_i^k - \widehat{\mathbf{x}}_i\left(\mathbf{x}^k\right)\| \leq \varepsilon_i^k$;
        Set $\widehat{\mathbf{z}}_i^k = \mathbf{z}_i^k$ for $i \in S^k$, and $\widehat{\mathbf{z}}_i^k = \mathbf{x}_i^k$ for $i \notin S^k$;
(S.4) : Set $\mathbf{x}^{k+1} \triangleq \mathbf{x}^k + \gamma^k\,(\widehat{\mathbf{z}}^k - \mathbf{x}^k)$;
(S.5) : $k \leftarrow k+1$, and go to (S.1).
```

Assumption 3.11 *The step-size sequence* $\{\gamma^k \in (0,1]\}_{k\in\mathbb{N}_+}$ *satisfies any of the following rules:*

1. ***Bounded step-size:*** $0 < \liminf_{k\to\infty} \gamma^k \le \limsup_{k\to\infty} \gamma^k < c_\tau/L$, *where* $c_\tau \triangleq \min_{i=1,\ldots n} \tau_i$*;*
2. ***Diminishing step-size:*** $\sum_{k=0}^{\infty} \gamma^k = +\infty$ *and* $\sum_{k=0}^{\infty} (\gamma^k)^2 < +\infty$*. In addition, if* S^k *is chosen according to the cyclic rule [Assumption 3.10.1],* γ^k *further satisfies* $0 < \eta_1 \le \gamma^{k+1}/\gamma^k \le \eta_2 < +\infty$*, for sufficiently large k, and some* $\eta_1 \in (0,1)$ *and* $\eta_2 \ge 1$*;*
3. ***Line-search:*** *Let* $\alpha, \delta \in (0,1)$*, choose* $\gamma^k = (\delta)^{t_k}$*, where* t_k *is the smallest nonnegative integer such that*

$$V(\mathbf{x}^k + \gamma^k\, \Delta\widehat{\mathbf{x}}^k) \le V(\mathbf{x}^k) + \alpha \cdot \gamma^k \left(\nabla F(\mathbf{x}^k)^T\, \Delta\widehat{\mathbf{x}}^k + \sum_{i\in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right) \right), \tag{3.111}$$

where $\Delta\widehat{\mathbf{x}}^k \triangleq (\widehat{\mathbf{z}}^k - \mathbf{x}^k)$.

Convergence of Algorithm 6 is stated below and summarized in the flow chart in Fig. 3.7. Theorem 3.12 applies to settings where the step-size is chosen according to the bounded rule or line-search while Theorem 3.13 states convergence under the diminishing step-size rule.

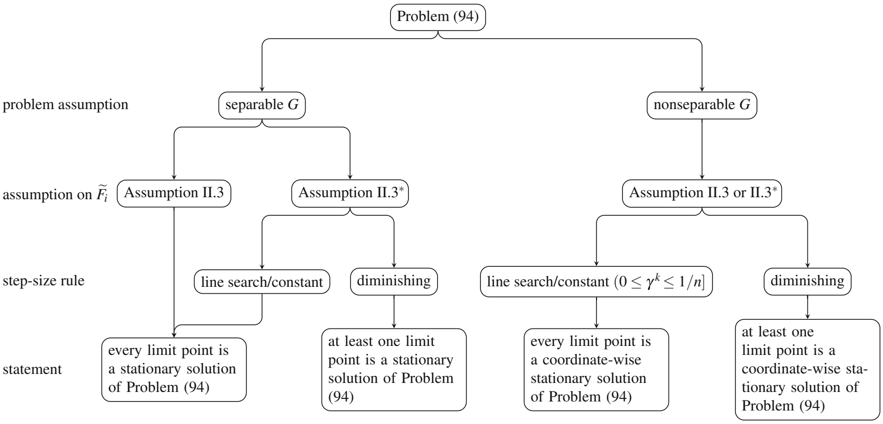

Problem (94)
problem assumption
separable G
nonseparable G
assumption on $\widetilde{F}_i$
Assumption II.3
Assumption II.3*
Assumption II.3 or II.3*
step-size rule
line search/constant
diminishing
line search/constant $(0 \le \gamma^k \le 1/n]$
diminishing
statement
every limit point is a stationary solution of Problem (94)
at least one limit point is a stationary solution of Problem (94)
every limit point is a coordinate-wise stationary solution of Problem (94)
at least one limit point is a coordinate-wise stationary solution of Problem (94)

Fig. 3.7 Convergence of FLEXA (Algorithm 6)

Theorem 3.12 *Consider Problem* (3.94) *under Assumption 3.1. Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 6, under the following conditions:*

(i) Each surrogate function $\widetilde{F}_i$ *satisfies Assumptions 3.2–3.3 or 3.2–3.3*;*
(ii) S^k *is chosen according to any of the rules in Assumption 3.10;*
(iii) Each inexact solution $\mathbf{z}_i^k$ *satisfies Assumption 3.9;*
(iv) $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *is chosen according to either Assumption 3.11.1 (bounded rule) or Assumption 3.11.3 (line-search); in addition, if G is nonseparable,* $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *also satisfies* $\gamma^k \in (0, 1/n]$*, for all* $k \in \mathbb{N}_+$.

Then (3.101) holds [almost surely if S^k *is chosen according to Assumption 3.10.3 (random-based rule)].*

Theorem 3.13 *Consider Problem* (3.94) *under Assumption 3.1. Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 6, under conditions (i), (ii) and (iii) of Theorem 3.12. Suppose that* $\{\gamma^k\}_{k\in\mathbb{N}_+}$ *is chosen according to Assumption 3.11.2 (diminishing rule).*

Then, (3.100) *holds [almost surely if* S^k *is chosen according to Assumption 3.10.3 (random-based rule)]. Furthermore, if G is separable and the surrogate functions* $\widetilde{F}_i$ *satisfy Assumption 3.3, then also (3.101) holds (almost surely under Assumption 3.10.3).*

3.3.3.1 Discussion on Algorithm 6

The framework described in Algorithm 6 can give rise to very different schemes. We cannot discuss here the entire spectrum of choices; we provide just a few examples of error bound functions E_i and block selection rules S^k.

On the Choice of the Error Bound Function E_i Any function satisfying (3.110) is a valid candidate for E_i. Of course, one can always choose $E_i(\mathbf{x}) = \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|$, corresponding to $\underline{s}_i = \bar{s}_i = 1$ in (3.110). This is a valuable choice if the computation of $\widehat{\mathbf{x}}_i(\mathbf{x}^k)$ can be easily accomplished. For instance, this is the case in the LASSO problem when the block variables are scalars: $\widehat{x}_i(\mathbf{x}^k)$ can be computed in closed form using the soft-thresholding operator [7]; see Sect. 3.3.5 for details.

In situations where the computation of $\|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|$ is not possible or advisable (e.g., when a closed form expression is lacking and the blocks have a large size), one can resort to alternative less expensive metrics satisfying (3.110). For example, assume momentarily that $G \equiv 0$. Then, it is known [78, Proposition 6.3.1] that, under the stated assumptions, $\|\Pi_{X_i}(\mathbf{x}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k)) - \mathbf{x}_i^k\|$ is an error bound for the minimization problem in (3.96) and therefore it satisfies (3.110), where $\Pi_{X_i}(\mathbf{y})$ denotes the Euclidean projection of $\mathbf{y}$ onto the closed and convex set X_i. In this case, one can choose $E_i(\mathbf{x}^k) = \|\Pi_{X_i}(\mathbf{x}_i^k - \nabla_{\mathbf{x}_i} F(\mathbf{x}^k)) - \mathbf{x}_i^k\|$. If $G(\mathbf{x}) \not\equiv 0$ things become more involved. In several cases of practical interest, adequate error bounds can be derived using [238, Lemma 7].

It is interesting to note that the computation of E_i is only needed if a *partial* update of the (block) variables is performed; otherwise (when $S^k = N$) one can dispense with the computation of E_i.

On the Block Selection Rule S^k The selection rules satisfying Assumption 3.10 are extremely flexible, ranging from deterministic to random-based selection rules. For instance, one can always choose $S^k = N$, resulting in the simultaneous deterministic update of all the (block) variables at each iteration (Algorithm 5). At the other extreme, one can update a single (block) variable per time, thus obtaining a Gauss-Southwell kind of method. Virtually, one can explore all the possibilities "in between", e.g., by choosing properly S^k and leveraging the parameter ρ in (3.110) to control the desired degree of parallelism or using suitably chosen cyclic-based as well as random-based rules. This flexibility can be coupled with the possibility of computing at each iteration only inexact solutions (Step 3), without affecting the convergence of the resulting scheme (provided that Assumption 3.9 is satisfied).

The selection of the most suitable updating rule depends on the specific problem, including the problem scale, computational environment, data acquisition process, as well as the communication among the processors. For instance, versions of Algorithm 6 where all (or most of) the variables are updated at each iteration are particularly amenable to implementation in *distributed* environments (e.g., multi-user communications systems, ad-hoc networks, etc.). In fact, in these settings, not only the calculation of the inexact solutions $\mathbf{z}_i^k$ can be carried out in parallel, but the information that "the i-th subproblem" has to exchange with the "other subproblems" in order to compute the next iteration is very limited. A full appreciation of the potentialities of this approach in distributed settings depends however on the specific application under consideration; we discuss some examples in Sect. 3.3.5. The cyclic order has the advantage of being extremely simple to implement. Random selection-based rules are essentially as cheap as cyclic selections while alleviating some of the pitfalls of cyclic updates. They are also relevant in distributed environments wherein data are not available in their entirety, but are acquired either in batches or over a network. In such scenarios, one might be interested in running the optimization at a certain instant even with the limited, randomly available information. A main limitation of random/cyclic selection rules is that they remain disconnected from the status of the optimization process, which instead is exactly the kind of behavior that greedy-based updates try to avoid, in favor of faster convergence, but at the cost of more intensive computation.

We conclude the discussion on the block selection rules providing some specific deterministic and random-based rules that we found effective in our experiments.

- **Deterministic selection**: In addition to the selection rules discussed above, a specific (albeit general) approach is to define first a finite cover $\{S_i\}_{i=1}^M$ of N and then update the blocks by selecting the S_i's cyclically. It is also admissible to randomly shuffle the order of the sets S_i before one update cycle.

- **Random-based selection**: The sampling rule $\boldsymbol{S}$ (for notational simplicity the iteration index k will be omitted) is uniquely characterized by the probability mass function

$$\mathbb{P}(S) \triangleq \mathbb{P}\left(\boldsymbol{S} = S\right), \quad S \subseteq N,$$

which assign probabilities to the subsets S of N. Associated with $\boldsymbol{S}$, define the probabilities $q_j \triangleq \mathbb{P}(|\boldsymbol{S}| = j)$, for $j = 1, \ldots, n$. The following proper sampling rules, proposed in [198] for convex problems with separable G, are instances of rules satisfying Assumption 3.10.3.

1. ***Uniform (U) sampling***: All blocks are selected with the same (non zero) probability:

$$\mathbb{P}(i \in \boldsymbol{S}) = \mathbb{P}(j \in \boldsymbol{S}) = \frac{\mathbb{E}\left[|\boldsymbol{S}|\right]}{n}, \quad \forall i \neq j \in N.$$

2. ***Doubly Uniform (DU) sampling***: All sets S of equal cardinality are generated with equal probability, i.e., $\mathbb{P}(S) = \mathbb{P}(S^{'})$, for all $S, S^{'} \subseteq N$ such that $|S| = |S^{'}|$. The density function is then

$$\mathbb{P}(S) = \frac{q_{|S|}}{\begin{pmatrix} n \\ |S| \end{pmatrix}}.$$

3. ***Nonoverlapping Uniform (NU) sampling***: It is a uniform sampling assigning positive probabilities only to sets forming a partition of N. Let $S^1, \ldots, S^p$ be a partition of N, with each $\left|S^i\right| > 0$, the density function of the NU sampling is:

$$\mathbb{P}(S) = \begin{cases} \dfrac{1}{p}, \text{ if } S \in \left\{S^1, \ldots, S^p\right\}; \\ 0 \quad \text{otherwise}; \end{cases}$$

which corresponds to $\mathbb{P}(i \in \boldsymbol{S}) = 1/p$, for all $i \in N$.

4. ***Nice Sampling (NS)***: Given an integer $0 \leq \tau \leq n$, a τ-nice sampling is a DU sampling with $q_\tau = 1$ (i.e., each subset of τ blocks is chosen with the same probability).

   Using the NS one can control the degree of parallelism of the algorithm by tuning the cardinality τ of the random sets generated at each iteration, which makes this rule particularly appealing in a multi-core environment. Indeed, one can set τ equal to the number of available cores/processors, and assign each block coming out from the greedy selection (if implemented) to a dedicated processor/core.

As a final remark, note that the DU/NU rules contain as special cases sequential and fully parallel updates wherein at each iteration a *single* block is updated uniformly at random, or *all* blocks are updated.

5. ***Sequential sampling***: It is a DU sampling with $q_1 = 1$, or a NU sampling with $p = n$ and $S^j = j$, for $j = 1, \ldots, p$.
6. ***Fully parallel sampling***: It is a DU sampling with $q_n = 1$, or a NU sampling with $p = 1$ and $S^1 = N$.

Other interesting uniform and nonuniform practical rules (still satisfying Assumption 3.10) can be found in [197, 198]. Furthermore, see [55, 56] for extensive numerical results comparing the different sampling schemes.

3.3.3.2 Convergence Analysis of Algorithm 6

In this subsection, we prove convergence of Algorithm 6 (Theorems 3.12 and 3.13). We consider only deterministic block selection rules (namely Assumptions 3.10.1 and 3.10.2); the proof under random-based block selection rules follows similar steps and thus is omitted.

Preliminaries

We first introduce some preliminary technical results that will be used to prove the aforementioned theorems.

Lemma 3.14 (Descent Lemma[14]) *Let $F : \mathbb{R}^m \to \mathbb{R}$ be continuously differentiable, with L-Lipschitz gradient. Then, there holds:*

$$\left| F(\mathbf{y}) - F(\mathbf{x}) - \nabla F(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) \right| \leq \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m. \tag{3.112}$$

Lemma 3.15 *Let $(a_i)_{i=1}^n$ be a n-tuple of nonnegative numbers such that $\sum_{i=1}^n a_i \geq \delta$, with $\delta > 0$. Then, it holds $\sum_{i=1}^n a_i \leq \frac{n}{\delta} \sum_{i=1}^n a_i^2$.*

Proof Define $\mathbf{a} \triangleq [a_1, \ldots, a_n]^T$. The desired result follows readily from $\|\mathbf{a}\|_2^2 \overset{(a)}{\geq} \frac{1}{n} \|\mathbf{a}\|_1^2 \overset{(b)}{\geq} \frac{\delta}{n} \|\mathbf{a}\|_1$, where in (a) we used the Jensen's inequality and $\mathbf{a} \geq 0$ while (b) is due to $\|\mathbf{a}\|_1 \geq \delta$. □

Lemma 3.16 ([17, Lemma 1]) *Let $\{Y^k\}_{k \in \mathbb{N}_+}$, $\{W^k\}_{k \in \mathbb{N}_+}$, and $\{Z^k\}_{k \in \mathbb{N}_+}$ be three sequences such that W^k is nonnegative for all k. Assume that*

$$Y^{k+1} \leq Y^k - W^k + Z^k, \quad k = 0, 1, \ldots, \tag{3.113}$$

and that the series $\sum_{k=0}^{T} Z^k$ *converges as* $T \to \infty$. *Then either* $Y^k \to -\infty$, *or else* Y^k *converges to a finite value and* $\sum_{k=0}^{\infty} W^k < \infty$.

Lemma 3.17 *Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 6, with each* $\gamma^k \in (0, 1/n)$. *For every* $k \in \mathbb{N}_+$ *and* $S^k \subseteq N$, *there holds:*

$$G(\mathbf{x}^{k+1}) - G(\mathbf{x}^k) \le \gamma^k \sum_{i\in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right), \tag{3.114}$$

where $\mathbf{z}_i^k$ *is the inexact solution defined in Step 3 of the algorithm.*

Furthermore, if G is separable, we have: for $\gamma^k \in (0, 1)$,

$$G(\mathbf{x}^{k+1}) - G(\mathbf{x}^k) \le \gamma^k \sum_{i\in S^k} \left(g_i(\mathbf{z}_i^k) - g_i(\mathbf{x}_i^k) \right). \tag{3.115}$$

Proof See Appendix—Sect. 3.3.6.2. □

Lemma 3.18 *Under Assumptions 3.1, 3.2, and 3.9, the inexact solution* $\mathbf{z}_i^k$ *satisfies*

$$\nabla_{\mathbf{x}_i} F(\mathbf{x}^k)^T (\mathbf{z}_i^k - \mathbf{x}_i^k) + G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \le -\frac{\tau_i}{2} \|\mathbf{z}_i^k - \mathbf{x}_i^k\|^2. \tag{3.116}$$

Proof The proof follows readily from the strong convexity of $\widetilde{F}_i$ and Assumption 3.9.2. □

Lemma 3.19 *Let* S^k *be selected according to the greedy rule (cf. Assumption 3.10.2). Then, there exists a constant* $0 < \tilde{c} \le 1$ *such that*

$$\left\| \left(\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k \right)_{S^k} \right\| \ge \tilde{c} \left\| \widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k \right\|. \tag{3.117}$$

Proof See Appendix—Sect. 3.3.6.3. □

Proposition 3.20 *Let* $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 6, in the setting of Theorem 3.12 or Theorem 3.13. The following hold [almost surely, if* S^k *is chosen according to Assumption 3.10.3 (random-based rule)]:*

(a)

$$\sum_{k=0}^{\infty} \gamma^k \left\| \widehat{\mathbf{z}}^k - \mathbf{x}^k \right\|^2 < +\infty; \tag{3.118}$$

(b)

$$\lim_{k\to\infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| = 0. \tag{3.119}$$

Proof Without loss of generality, we consider next only the case of nonseparable G. By the descent lemma (cf. Lemma 3.114) and Steps 3-4 of the algorithm, we have

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) + \gamma^k \nabla F(\mathbf{x}^k)^T (\widehat{\mathbf{z}}^k - \mathbf{x}^k) + \frac{\left(\gamma^k\right)^2 L}{2} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2.$$

Consider the case of G nonseparable. We have:

$$\begin{aligned} V(\mathbf{x}^{k+1}) &= F(\mathbf{x}^{k+1}) + G(\mathbf{x}^{k+1}) \\ &\overset{(3.114)}{\leq} V(\mathbf{x}^k) + \gamma^k \, \nabla F(\mathbf{x}^k)^T (\widehat{\mathbf{z}}^k - \mathbf{x}^k) + \frac{\left(\gamma^k\right)^2 L}{2} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2 \\ &\quad + \gamma^k \sum_{i \in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right) \qquad (3.120) \\ &\overset{(3.116)}{\leq} V(\mathbf{x}^k) - \frac{\gamma^k}{2} (c_\tau - \gamma^k L) \, \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2. \qquad (3.121) \end{aligned}$$

If γ^k satisfies either Assumption 3.11.1 (bounded rule) or Assumption 3.11.2 (diminishing rule), statement (a) of the proposition comes readily from Lemma 3.16 and Assumption 3.1.4.

Consider now the case where γ^k is chosen according to Assumption 3.11.3 (line search). First of all, we prove that there exists a suitable $\gamma^k \in (0, \gamma^0]$, with $\gamma^0 \in (0, 1/n]$ (if G is separable $\gamma^0 \in (0, 1]$), such that the Armijo-like condition (3.111) holds. By (3.120), the line-search condition (3.111) is satisfied if

$$\begin{aligned} &\gamma^k \left(\nabla F(\mathbf{x}^k)^T (\widehat{\mathbf{z}}^k - \mathbf{x}^k) + \sum_{i \in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right) \right) + \frac{\left(\gamma^k\right)^2 L}{2} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2 \\ &\quad \leq \alpha \cdot \gamma^k \left(\nabla F(\mathbf{x}^k)^T \, (\widehat{\mathbf{z}}^k - \mathbf{x}^k) + \sum_{i \in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right) \right), \end{aligned}$$

which, rearranging the terms, yields

$$\begin{aligned} &\frac{\gamma^k \cdot L}{2} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2 \\ &\qquad \leq -(1-\alpha) \left(\nabla F(\mathbf{x}^k)^T (\widehat{\mathbf{z}}^k - \mathbf{x}^k) + \sum_{i \in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right) \right). \end{aligned} \qquad (3.122)$$

Since (cf. Lemma 3.18)

$$\|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2 \leq -\frac{2}{c_\tau} \cdot \left(\nabla F(\mathbf{x}^k)^T (\widehat{\mathbf{z}}^k - \mathbf{x}^k) + \sum_{i \in S^k} \left(G(\mathbf{z}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k) \right) \right),$$

inequality (3.122) [and thus (3.111)] is satisfied by any $\gamma^k \leq \min\left\{\gamma^0, \frac{c_\tau(1-\alpha)}{L}\right\} < +\infty$.

We show next that γ^k obtained by (3.111) is uniformly bounded away from zero. This is equivalent to show $t_k < +\infty$. Without loss of generality we consider $t_k \geq 1$ [otherwise (3.111) is satisfied by $\gamma^k = \gamma^0$]. Since t_k is the smallest positive integer such that (3.111) holds with $\gamma^k = (\delta)^{t_k}$, it must be that the same inequality is not satisfied by $\gamma^k = (\delta)^{t_k - 1}$. Consequently, it must be $(\delta)^{t_k-1} > \frac{c_\tau(1-\alpha)}{L}$, and thus

$$\gamma^k \geq \min\left\{\gamma^0, \frac{c_\tau\,(1-\alpha)}{L} \cdot \delta\right\}. \tag{3.123}$$

Using (3.123) in (3.121), we obtain

$$V(\mathbf{x}^{k+1}) \leq V(\mathbf{x}^k) - \beta_2\, \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2, \qquad \forall k \in \mathbb{N}_+, \tag{3.124}$$

where $\beta_2 > 0$ is some finite constant. The rest of the proof follows the same arguments used to prove the statements of the proposition from (3.121).

We prove now statement (b). By Step 4 of Algorithm 6, it suffices to show that

$$\lim_{k\to\infty} \gamma^k\, \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\| = 0.$$

Using (3.118), we have

$$\lim_{k\to\infty} \gamma^k \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2 = 0.$$

Since $\gamma^k \in (0, 1]$, it holds

$$\lim_{k\to\infty} \left(\gamma^k\, \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|\right)^2 \leq \lim_{k\to\infty} \gamma^k \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\|^2 = 0. \tag{3.125}$$

This completes the proof. □

Proof of Theorem 3.12

We prove (3.101) for each of the block selection rules in Assumption 3.10 separately.

• **Essentially Cyclic Rule [Assumption 3.10.1]** We start bounding $\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|$ as follows:

$$\begin{aligned}\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| &\leq \sum_{i=1}^{n} \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\| \\ &\leq \sum_{i=1}^{n} \left(\|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \widehat{\mathbf{x}}_i(\mathbf{x}^{k+s_i^k})\| + \|\widehat{\mathbf{x}}_i(\mathbf{x}^{k+s_i^k}) - \mathbf{x}_i^{k+s_i^k}\| + \|\mathbf{x}_i^{k+s_i^k} - \mathbf{x}_i^k\| \right)\end{aligned} \tag{3.126}$$

where $s_i^k \triangleq \min\{t \in \{1, \ldots, T\} \,|\, i \in S^{k+t}\}$, so that $k + s_i^k$ is the first time that block i is selected (updated) since iteration k. Note that $1 \leq s_i^k \leq T$, for all $i \in N$ and $k \in \mathbb{N}_+$ (due to Assumption 3.10.1). We show next that the three terms on the RHS of (3.126) are asymptotically vanishing, which proves (3.101).

Since $1 \leq s_i^k \leq T$, we can write

$$\lim_{k\to\infty} \|\mathbf{x}^{k+s_i^k} - \mathbf{x}^k\| \leq \lim_{k\to\infty} \sum_{j=1}^{T} \|\mathbf{x}^{k+j} - \mathbf{x}^{k+j-1}\| \overset{(3.119)}{=} 0, \tag{3.127}$$

which, by the continuity of $\widehat{\mathbf{x}}(\bullet)$ (cf. Lemma 3.4), leads also to

$$\lim_{k\to\infty} \|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \widehat{\mathbf{x}}_i(\mathbf{x}^{k+s_i^k})\| = 0. \tag{3.128}$$

Let $T_i \subseteq \mathbb{N}_+$ be the set of iterations at which block i is updated. It follows from Assumption 3.10.1 that $|T_i| = +\infty$, for all $i \in N$. This together with (3.118) implies

$$\sum_{k\in T_i} \|\mathbf{z}_i^k - \mathbf{x}_i^k\|^2 < +\infty, \quad \forall i \in N,$$

and thus

$$\lim_{T_i \ni k\to\infty} \left\|\mathbf{z}_i^k - \mathbf{x}_i^k\right\| = 0 \quad \Rightarrow \quad \lim_{k\to\infty} \left\|\mathbf{z}_i^{k+s_i^k} - \mathbf{x}_i^{k+s_i^k}\right\| = 0, \quad \forall i \in N.$$

Therefore,

$$\begin{aligned}&\lim_{k\to\infty} \left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{k+s_i^k}) - \mathbf{x}_i^{k+s_i^k}\right\| \\ &\quad\leq \lim_{k\to\infty} \left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{k+s_i^k}) - \mathbf{z}_i^{k+s_i^k}\right\| + \lim_{k\to\infty} \left\|\mathbf{z}_i^{k+s_i^k} - \mathbf{x}_i^{k+s_i^k}\right\| = 0, \quad \forall i \in N.\end{aligned} \tag{3.129}$$

Combining (3.126) with (3.127)-(3.129) and invoking again the continuity of $\widehat{\mathbf{x}}(\bullet)$, we conclude that $\lim_{k\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| = 0$.

• **Greedy Rule [Assumption 3.10.2]** We have

$$\begin{aligned} \tilde{c}\,\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| &\overset{(3.117)}{\leq} \|(\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k)_{S^k}\| \\ &\leq \|(\widehat{\mathbf{z}}^k - \mathbf{x}^k)_{S^k}\| + \|(\widehat{\mathbf{x}}(\mathbf{x}^k) - \widehat{\mathbf{z}}^k)_{S^k}\| \\ &\overset{A.3.9.1}{\leq} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\| + \sum_{i\in S^k} \epsilon_i^k \underset{k\to\infty}{\longrightarrow} 0, \end{aligned} \tag{3.130}$$

where the last implication comes from $\lim_{k\to\infty} \epsilon_i^k = 0$, for all $i \in N$—cf. Assumption 3.9.1—and $\lim_{k\to\infty} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\| = 0$—due to Proposition 3.20 and the fact that γ^k is bounded away from zero, when the step-size is chosen according to Assumption 3.11.1 or Assumption 3.11.3 [cf. (3.123)]. This proves (3.101).

Proof of Theorem 3.13

Consider now the diminishing step-size rule [Assumption 3.11.2].

1) Proof of (3.100): $\liminf_{k\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| = 0$ By Proposition 3.20 and the step-size rule, we have $\liminf_{k\to\infty} \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\| = 0$, for all choices of S^k. We proceed considering each of the block selection rules in Assumption 3.10 separately.

• **Essentially Cyclic Rule [Assumption 3.10.1]** For notational simplicity, let us assume that S^k is a singleton, that is, $S^k = \{i^k\}$, where i^k denotes the index of the block selected at iteration k. The proof can be readily extended to the general case $|S^k| > 1$.

We have

$$\begin{aligned} \liminf_{k\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| &\overset{(a)}{\leq} \liminf_{r\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^{rT}) - \mathbf{x}^{rT}\| \leq \liminf_{r\to\infty} \sum_{i=1}^{n} \left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{rT}) - \mathbf{x}_i^{rT}\right\| \\ &\leq \liminf_{r\to\infty} \sum_{i=1}^{n} \left(\left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{rT+s_i^{rT}}) - \mathbf{x}_i^{rT+s_i^{rT}}\right\| + \left\|\mathbf{x}_i^{rT+s_i^{rT}} - \mathbf{x}_i^{rT}\right\| \right. \\ &\quad \left. + \left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{rT}) - \widehat{\mathbf{x}}_i(\mathbf{x}^{rT+s_i^{rT}})\right\| \right) \\ &= \liminf_{r\to\infty} \sum_{i=1}^{n} \left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{rT+s_i^{rT}}) - \mathbf{x}_i^{rT+s_i^{rT}}\right\| \end{aligned}$$

$$
\begin{aligned}
&+\underbrace{\lim_{r\to\infty}\sum_{i=1}^{n}\left\|\mathbf{x}_i^{rT+s_i^{rT}}-\mathbf{x}_i^{rT}\right\|}_{\overset{(3.127)}{=}0}+\underbrace{\lim_{r\to\infty}\sum_{i=1}^{n}\left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{rT})-\widehat{\mathbf{x}}_i(\mathbf{x}^{rT+s_i^{rT}})\right\|}_{\overset{(3.128)}{=}0}\\
&\le\liminf_{r\to\infty}\sum_{i=1}^{n}\left\|\mathbf{z}_i^{rT+s_i^{rT}}-\mathbf{x}_i^{rT+s_i^{rT}}\right\|+\underbrace{\lim_{r\to\infty}\sum_{i=1}^{n}\left\|\widehat{\mathbf{x}}_i(\mathbf{x}^{rT+s_i^{rT}})-\mathbf{z}_i^{rT+s_i^{rT}}\right\|}_{\overset{(A.3.9.1)}{\le}\lim_{r\to\infty}\sum_{i=1}^{n}\varepsilon_i^{rT+s_i^{rT}}\overset{(A.3.9.1)}{=}0}\\
&\le\liminf_{r\to\infty}\sum_{k=rT+1}^{(r+1)T}\left\|\mathbf{z}_{i^k}^{k}-\mathbf{x}_{i^k}^{k}\right\|,
\end{aligned}
\tag{3.131}
$$

where (a) follows from the fact that the infimum of a subsequence is larger than that of the original sequence.

To complete the proof, we show next that the term on the RHS of (3.131) is zero. Recalling that if the cyclic block selection rule is implemented, the diminishing step-size γ^k is assumed to further satisfy $\eta_1 \le \gamma^{k+1}/\gamma^k \le \eta_2$, with $\eta_1 \in (0, 1)$ and $\eta_2 \ge 1$ (cf. Assumption 3.11.2), we have

$$
\begin{aligned}
+\infty &\overset{(3.118)}{\ge} \lim_{k\to\infty}\sum_{t=1}^{k}\gamma^t\,\|\widehat{\mathbf{z}}^t-\mathbf{x}^t\|^2=\lim_{k\to\infty}\sum_{t=1}^{k}\gamma^t\,\|\mathbf{z}_{i^t}^t-\mathbf{x}_{i^t}^t\|^2\\
&=\lim_{k\to\infty}\sum_{r=0}^{k}\sum_{t=rT+1}^{(r+1)T}\gamma^t\,\|\mathbf{z}_{i^t}^t-\mathbf{x}_{i^t}^t\|^2\\
&\ge(\eta_1)^{T-1}\lim_{k\to\infty}\sum_{r=0}^{k}\gamma^{rT+1}\sum_{t=rT+1}^{(r+1)T}\|\mathbf{z}_{i^t}^t-\mathbf{x}_{i^t}^t\|^2,
\end{aligned}
\tag{3.132}
$$

where in the last inequality we used $\gamma^{k+1}/\gamma^k \ge \eta_1$. Since

$$
+\infty=\lim_{k\to\infty}\sum_{t=1}^{k}\gamma^t=\lim_{k\to\infty}\sum_{r=0}^{k}\sum_{t=rT+1}^{(r+1)T}\gamma^t\le T\cdot(\eta_2)^{T-1}\lim_{k\to\infty}\sum_{r=0}^{k}\gamma^{rT+1},
$$

it follows from (3.132) that

$$
\liminf_{r\to\infty}\sum_{t=rT+1}^{(r+1)T}\|\mathbf{z}_{i^t}^t-\mathbf{x}_{i^t}^t\|^2=0,
\tag{3.133}
$$

which, combined with (3.131), proves the desired result.

• **Greedy Rule [Assumption 3.10.2]** Taking the liminf on both sides of (3.130) leads to the desired result.

2) Proof of (3.101) $\limsup_{k\to\infty} \|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| = 0$. Recall that in this setting, $\widehat{\mathbf{x}}(\bullet)$ is Lipschitz continuous with constant $\hat{L}$. We prove the result for the essentially cyclic rule and greedy rule separately.

• **Essentially Cyclic Rule [Assumption 3.10.1]** As in the proof of Theorem 3.13, let us assume w.l.o.g. that $S^k = \{i^k\}$, where i^k denotes the index of the block selected at iteration k.

By (3.126)–(3.129), it is sufficient to prove $\limsup_{k\to\infty} \|\mathbf{z}_i^{k+s_i^k} - \mathbf{x}_i^{k+s_i^k}\| = 0$, for all $i \in N$. Since

$$\limsup_{k\to\infty} \|\mathbf{z}_i^{k+s_i^k} - \mathbf{x}_i^{k+s_i^k}\| \le \limsup_{k\to\infty} \underbrace{\sum_{t=k\,T+1}^{(k+1)T} \|\mathbf{z}_{i^t}^t - \mathbf{x}_{i^t}^t\|}_{\triangleq \Delta^k} \tag{3.134}$$

we prove next $\limsup_{k\to\infty} \Delta^k = 0$.

Assume on the contrary that $\limsup_{k\to\infty} \Delta^k > 0$. Since $\liminf_{k\to\infty} \Delta^k = 0$ [cf. (3.133)], there exists a $\delta > 0$ such that $\Delta^k < \delta$ for infinitely many k and also $\Delta^k > 2 \cdot T\delta$ for infinitely many k. Therefore, there exists a set $K \subseteq \mathbb{N}_+$, with $|K| = \infty$, such that for each $k \in K$, one can find an integer $j_k > k$ such that

$$\Delta^k \ge 2 \cdot \delta \cdot T, \qquad \Delta^{j_k} \le \delta \tag{3.135}$$

$$\delta < \Delta^\ell < 2 \cdot \delta \cdot T, \qquad \text{if } \; k < \ell < j_k. \tag{3.136}$$

Define the following quantities: for any $k \in K$, let

$$T_i^k \triangleq \left\{ r \in \{k\,T+1, \ldots, (k+1)\,T\} \mid i^r = i \right\} \quad \text{and} \quad t_i^k \triangleq \min T_i^k. \tag{3.137}$$

Note that T_i^k (resp. t_i^k) is the set of (iteration) indices (resp. the smallest index) within $[k\,T+1, (k+1)\,T]$ at which the block index i is selected. Because of Assumption 3.10.1, it must be $1 \le |T_i^k| \le T$, for all k, where $|T_i^k|$ is the number of times block i has been selected in the iteration window $[k\,T+1, (k+1)\,T]$. Then we have

$$\begin{aligned}
\delta \cdot T &= 2 \cdot \delta \cdot T - \delta \cdot T \\
&\le \Delta^k - T \cdot \Delta^{j_k} = \sum_{r=k\,T+1}^{(k+1)T} \|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\| - T \sum_{r=j_k\cdot T+1}^{(j_k+1)\,T} \|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\| \\
&\le \sum_{i=1}^n \sum_{r\in T_i^k} \|\mathbf{z}_i^r - \mathbf{x}_i^r\| - \sum_{i=1}^n T \cdot \left\|\mathbf{z}_i^{t_i^{j_k}} - \mathbf{x}_i^{t_i^{j_k}}\right\|
\end{aligned}$$

$$
\begin{aligned}
&\le \sum_{i=1}^{n} \sum_{r\in T_i^k} \left\|\mathbf{z}_i^r - \mathbf{x}_i^r\right\| - \sum_{i=1}^{n} |T_i^k| \cdot \left\|\mathbf{z}_i^{t_i^{j_k}} - \mathbf{x}_i^{t_i^{j_k}}\right\| \\
&= \sum_{i=1}^{n} \sum_{r\in T_i^k} \left(\left\|\mathbf{z}_i^r - \mathbf{x}_i^r\right\| - \left\|\mathbf{z}_i^{t_i^{j_k}} - \mathbf{x}_i^{t_i^{j_k}}\right\| \right) \\
&\overset{(a)}{\le} \sum_{i=1}^{n} \sum_{r\in T_i^k} \left(\left\|\widehat{\mathbf{x}}_i(\mathbf{x}^r) - \widehat{\mathbf{x}}_i(\mathbf{x}^{t_i^{j_k}})\right\| + \left\|\mathbf{x}_i^r - \mathbf{x}_i^{t_i^{j_k}}\right\| \right) + \underbrace{\sum_{i=1}^{n} \sum_{r\in T_i^k} (\epsilon_i^{t_i^{j_k}} + \epsilon_i^r)}_{\tilde{\epsilon}_1^k} \\
&\overset{(b)}{\le} \left(1+\hat{L}\right) \sum_{i=1}^{n} \sum_{r\in T_i^k} \left\|\mathbf{x}^r - \mathbf{x}^{t_i^{j_k}}\right\| + \tilde{\epsilon}_1^k \\
&\le \left(1+\hat{L}\right) \sum_{i=1}^{n} \sum_{r\in T_i^k} \sum_{s=r}^{t_i^{j_k}-1} \gamma^s \left\|\mathbf{z}_{i^s}^s - \mathbf{x}_{i^s}^s\right\| + \tilde{\epsilon}_1^k \\
&\le \left(1+\hat{L}\right) \sum_{i=1}^{n} |T_i^k| \cdot \sum_{r=k\,T+1}^{t_i^{j_k}-1} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\| + \tilde{\epsilon}_1^k \\
&\le \left(1+\hat{L}\right) \sum_{i=1}^{n} |T_i^k| \cdot \left(\sum_{r=k\,T+1}^{j_k\cdot T} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\| + \sum_{r=j_k\cdot T+1}^{t_i^{j_k}-1} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\| \right) + \tilde{\epsilon}_1^k \\
&\le \left(1+\hat{L}\right) \cdot (n\cdot T) \sum_{r=k\,T+1}^{j_k\cdot T} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\| + \underbrace{\left(1+\hat{L}\right) \cdot T \sum_{r=j_k\cdot T+1}^{t_i^{j_k}-1} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\|}_{\tilde{\epsilon}_2^k} + \tilde{\epsilon}_1^k \\
&\le \left(1+\hat{L}\right) \cdot (n\cdot T) \sum_{s=k}^{j_k-1} \sum_{r=s\,T+1}^{(s+1)\,T} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\| + \tilde{\epsilon}_1^k + \tilde{\epsilon}_2^k \\
&\overset{(c)}{\le} \left(1+\hat{L}\right) \cdot (n\cdot T) \sum_{s=k}^{j_k-1} (\eta_2)^{T-1} \gamma^{s\,T+1} \underbrace{\sum_{r=s\,T+1}^{(s+1)\,T} \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\|}_{=\Delta^s} + \tilde{\epsilon}_1^k + \tilde{\epsilon}_2^k \\
&\overset{(3.135)–(3.136)}{=} \left(1+\hat{L}\right) \cdot (n\cdot T) \cdot (\eta_2)^{T-1} \sum_{s=k}^{j_k-1} \gamma^{s\,T+1} \sum_{r=s\,T+1}^{(s+1)T} \frac{\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\|^2}{\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\|} + \tilde{\epsilon}_1^k + \tilde{\epsilon}_2^k \\
&\overset{(d)}{\le} \frac{T}{\delta} \cdot \left(1+\hat{L}\right) \cdot (n\cdot T) \cdot (\eta_2)^{T-1} \underbrace{\sum_{s=k}^{j_k-1} \gamma^{s\,T+1} \sum_{r=s\,T+1}^{(s+1)T} \|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\|^2}_{\tilde{\epsilon}_3^k} + \tilde{\epsilon}_1^k + \tilde{\epsilon}_2^k,
\end{aligned}
\tag{3.138}
$$

where in (a) we used the reverse triangle inequality and Assumption 3.9.1; (b) is due to the Lipschitz continuity of $\widehat{\mathbf{x}}_i$; in (c) we used $\gamma^{k+1}/\gamma^k \leq \eta_2$, with $\eta_2 \geq 1$; and (d) is due to Lemma 3.15.

We prove now that $\tilde{\epsilon}_1^k \downarrow 0$, $\tilde{\epsilon}_2^k \downarrow 0$, and $\tilde{\epsilon}_3^k \downarrow 0$. Since $\epsilon_i^k \downarrow 0$ for all $i \in N$ (cf. Assumption 3.9.1), it is not difficult to check that $\tilde{\epsilon}_1^k \downarrow 0$. The same result for $\tilde{\epsilon}_2^k$ comes from the following bound:

$$\tilde{\epsilon}_2^k \leq \left(1+\hat{L}\right)\cdot T \sum_{r=j_k\cdot T+1}^{(j_k+1)\,T} \gamma^r \left\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\right\| \leq \left(1+\hat{L}\right)\cdot T\cdot(\eta_2)^{T-1}\cdot\gamma^{j_k\cdot T+1}\cdot\Delta^{j_k}$$

$$\overset{(3.135)}{\leq} \left(\left(1+\hat{L}\right)\cdot T\cdot(\eta_2)^{T-1}\cdot\delta\right)\gamma^{j_k\cdot T+1} \underset{k\to\infty}{\longrightarrow} 0,$$

Finally, it follows from (3.132) that the series $\sum_{s=0}^{\infty}\gamma^{\,s\,T+1}\sum_{r=s\,T+1}^{(s+1)T}\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\|^2$ is convergent; the Cauchy convergence criterion implies that

$$\tilde{\epsilon}_3^k = \sum_{s=k}^{j_k-1}\gamma^{\,s\,T+1}\sum_{r=s\,T+1}^{(s+1)T}\|\mathbf{z}_{i^r}^r - \mathbf{x}_{i^r}^r\|^2 \underset{k\to\infty}{\longrightarrow} 0.$$

By the vanishing properties of $\tilde{\epsilon}_1^k$, $\tilde{\epsilon}_2^k$, and $\tilde{\epsilon}_3^k$, there exists a sufficiently large $k \in K$, say $\bar{k}$, such that

$$\tilde{\epsilon}_1^k \leq \frac{T\,\delta}{4}, \quad \tilde{\epsilon}_2^k \leq \frac{T\,\delta}{4}, \quad \tilde{\epsilon}_3^k \leq \frac{\delta^2}{4\left(1+\hat{L}\right)\cdot n\cdot T\cdot(\eta_2)^{T-1}}, \qquad \forall k \in K,\ k \geq \bar{k}, \tag{3.139}$$

which contradicts (3.138). Therefore, it must be $\limsup_{k\to\infty}\Delta^k = 0$.

- **Greedy Rule Assumption 3.10.2** By (3.130), it is sufficient to prove $\limsup_{k\to+\infty}\widehat{\Delta}^k \triangleq \|\widehat{\mathbf{z}}^k - \mathbf{x}^k\| = 0$. Assume the contrary, that is, $\limsup_{k\to+\infty}\widehat{\Delta}^k > 0$. Since $\liminf_{k\to+\infty}\widehat{\Delta}^k = 0$ (cf. Proposition 3.20), there exists a $\delta > 0$ such that $\widehat{\Delta}^k < \delta$ for infinitely many k and also $\widehat{\Delta}^k > 2\cdot(\delta/\tilde{c})$ for infinitely many k, where $0 < \tilde{c} \leq 1$ is the constant defined in Lemma 3.19. Therefore, there is a set $\widehat{K} \subseteq \mathbb{N}_+$, with $|\widehat{K}| = \infty$, such that for each $k \in \widehat{K}$, there exists an integer $j_k > k$, such that

$$\widehat{\Delta}^k \geq 2\,\frac{\delta}{\tilde{c}}, \qquad \widehat{\Delta}^{j_k} \leq \delta, \tag{3.140}$$

$$\delta < \widehat{\Delta}^t < 2\,\frac{\delta}{\tilde{c}} \qquad \text{if} \quad k < t < j_k. \tag{3.141}$$

We have

$$\begin{aligned}
\frac{\delta}{\tilde{c}} &= \frac{2\delta}{\tilde{c}} - \frac{\delta}{\tilde{c}} \\
&\leq \left\| \widehat{\mathbf{z}}^k - \mathbf{x}^k \right\| - \frac{1}{\tilde{c}} \left\| \widehat{\mathbf{z}}^{j_k} - \mathbf{x}^{j_k} \right\| \\
&\overset{(a)}{\leq} \| (\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k)_{S^k} \| - \frac{1}{\tilde{c}} \| (\widehat{\mathbf{x}}(\mathbf{x}^{j_k}) - \mathbf{x}^{j_k})_{S^{j_k}} \| + \underbrace{\sum_{i \in S^k} \epsilon_i^k + \frac{1}{\tilde{c}} \sum_{i \in S^{j_k}} \epsilon_i^{j_k}}_{\triangleq \tilde{\epsilon}^k} \\
&\overset{(3.117)}{\leq} \| \widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k \| - \| \widehat{\mathbf{x}}(\mathbf{x}^{j_k}) - \mathbf{x}^{j_k} \| + \tilde{\epsilon}^k \\
&\leq \| \widehat{\mathbf{x}}(\mathbf{x}^k) - \widehat{\mathbf{x}}(\mathbf{x}^{j_k}) \| + \| \mathbf{x}^k - \mathbf{x}^{j_k} \| + \tilde{\epsilon}^k \\
&\overset{(b)}{\leq} (1 + \hat{L}) \, \| \mathbf{x}^k - \mathbf{x}^{j_k} \| + \tilde{\epsilon}^k \\
&\leq (1 + \hat{L}) \sum_{t=k}^{j_k - 1} \gamma^t \, \| \widehat{\mathbf{z}}^t - \mathbf{x}^t \| + \tilde{\epsilon}^k \\
&= (1 + \hat{L}) \sum_{t=k}^{j_k - 1} \gamma^t \, \frac{\| \widehat{\mathbf{z}}^t - \mathbf{x}^t \|^2}{\| \widehat{\mathbf{z}}^t - \mathbf{x}^t \|} + \tilde{\epsilon}^k \\
&\overset{(3.140)-(3.141)}{<} \frac{1 + \hat{L}}{\delta} \sum_{t=k}^{j_k - 1} \gamma^t \, \| \widehat{\mathbf{z}}^t - \mathbf{x}^t \|^2 + \tilde{\epsilon}^k,
\end{aligned} \tag{3.142}$$

where in (a) we used the triangle inequality and Assumption 3.9.1; and in (b) we used the Lipschitz continuity of $\widehat{\mathbf{x}}$.

Since $\tilde{\epsilon}^k \downarrow 0$ (due to $\epsilon_i^k \downarrow 0$, for all i; see Assumption 3.9.1) and $\sum_{t=k}^{j_k-1} \gamma^t \, \| \widehat{\mathbf{z}}^t - \mathbf{x}^t \|^2 \underset{k \to \infty}{\longrightarrow} 0$ (due to $\sum_{k=0}^{\infty} \gamma^k \left\| \widehat{\mathbf{z}}^k - \mathbf{x}^k \right\|^2 < +\infty$; see Proposition 3.20), there exists a sufficiently large $k \in \widehat{K}$, say $\widehat{k}$, such that

$$\tilde{\epsilon}^k \leq \frac{\delta}{3 \cdot \tilde{c}} \quad \text{and} \quad \sum_{t=k}^{j_k-1} \gamma^t \, \| \widehat{\mathbf{z}}^t - \mathbf{x}^t \|^2 \leq \frac{\delta^2}{3 \cdot \tilde{c} \cdot (1 + \hat{L})}, \quad \forall k \in \widehat{K}, \quad k \geq \widehat{k}, \tag{3.143}$$

which contradicts (3.142). Therefore it must be $\lim_{k \to \infty} \| \widehat{\mathbf{z}}^k - \mathbf{x}^k \| = 0$.

3.3.4 Parallel SCA: Hybrid Schemes

FLEXA (Algorithm 6) and its convergence theory cover fully parallel Jacobi as well as Gauss-Southwell-type methods, and many of their variants. Of course, every block selection rule S^k has advantages and disadvantages, as discussed in Sect. 3.3.3.1. A natural question is whether it is possible to design *hybrid* schemes that inherit the best of the aforementioned plain selection schemes. In the following, we introduce three hybrid strategies that are particularly well suited to parallel optimization on multi-core/processor architectures.

3.3.4.1 Random-Greedy Schemes

In huge-scale optimization problems, the number of variables can be so large that computing the error bound functions E_i for each block i is not computationally affordable. To reduce the computational burden, one can think of a hybrid random/greedy block selection rule that combines random and greedy updates in the following form. First, a random selection is performed—the set $\widehat{S}^k \subseteq N$ is generated. Second, a greedy procedure is invoked to select *within the pool* $\widehat{S}^k$ only the subset of blocks, say $S^k \subseteq \widehat{S}^k$, that are "promising" according to the value of $E_i(\mathbf{x}^k)$. Finally all the blocks in S^k are updated in parallel. This procedure captures both the advantages of random and greedy schemes: the random selection drops off a large proportion of the blocks allowing one to save computation while the greedy selection finds the best candidates that generate the "maximum improvement" on the objective function. The procedure is summarized in Algorithm 7. Convergence of the algorithm was studied in [56] when $\{\gamma^k\}_{k\in\mathbb{N}_+}$ is chosen according to the diminishing step-size rule [Assumption 3.11.2]. The proof can be extended to deal with the other step-size rules (Assumption 3.11) following similar steps as those in Sect. 3.3.3.2, and thus is omitted.

Algorithm 7 Hybrid random-greedy FLEXA

```
Data : $\mathbf{x}^0 \in X$, $\{\gamma^k \in (0,1]\}_{k\in\mathbb{N}_+}$, $\varepsilon_i^k \geq 0$, for all $i \in N$ and $k \in \mathbb{N}_+$, $\rho \in (0,1]$.
       Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Randomly generate a set of blocks $\widehat{S}^k \subseteq N$;
(S.3) : Set $M^k \triangleq \max_{i\in\widehat{S}^k}\{E_i(\mathbf{x}^k)\}$;
          Choose a subset $S^k \subseteq \widehat{S}^k$ that contains at least one index $i$
          for which $E_i(\mathbf{x}^k) \geq \rho M^k$;
(S.4) : For all $i \in S^k$, solve (3.96) with accuracy $\varepsilon_i^k$ :
              find $\mathbf{z}_i^k \in X_i$ s.t. $\|\mathbf{z}_i^k - \widehat{\mathbf{x}}_i\left(\mathbf{x}^k\right)\| \leq \varepsilon_i^k$;
          Set $\widehat{\mathbf{z}}_i^k = \mathbf{z}_i^k$ for $i \in S^k$, and $\widehat{\mathbf{z}}_i^k = \mathbf{x}_i^k$ for $i \notin S^k$;
(S.5) : Set $\mathbf{x}^{k+1} \triangleq \mathbf{x}^k + \gamma^k\,(\widehat{\mathbf{z}}^k - \mathbf{x}^k)$;
(S.6) : $k \leftarrow k+1$, and go to (S.1).
```

3.3.4.2 Parallel-Cyclic Schemes

In a multi-core/processor architecture, another strategy to tackle huge-scale optimization problems is to adopt a hybrid parallel-cyclic strategy whereby the blocks of variables are partitioned among the workers (e.g., cores, processors) and updated *in parallel*, with each worker processing *sequentially* one block at a time.

Specifically, suppose that there are P workers. Let $\{I_p\}_{p=1}^{P}$ be a partition of N (the indices in I_p follow the natural order). Assign the blocks $\mathbf{x}_i$, with $i \in I_p$, to worker p, and write $\mathbf{x}_p \triangleq (\mathbf{x}_{p\,i})_{i\in I_p}$, where $\mathbf{x}_{p\,i}$ denotes the i-th block assigned to worker p; $\mathbf{x}_{-p} \triangleq (\mathbf{x}_i)_{i\notin I_p}$ is the vector of remaining variables, assigned to the other workers. Finally, given I_p, we partition $\mathbf{x}_p$ as $\mathbf{x}_p = (\mathbf{x}_{p,\,i<}, \mathbf{x}_{p,\,i\geq})$, where $\mathbf{x}_{p,\,i<} \triangleq (\mathbf{x}_{p\,j})_{j\in I_p,\, j<i}$ is the vector containing all the variables in $\mathbf{x}_p$ that appear before i (according to the order in I_p) whereas $\mathbf{x}_{p,\,i\geq} \triangleq (\mathbf{x}_{p\,j})_{j\in I_p,\, j\geq i}$ contains the remaining variables.

Once the optimization variables have been assigned to the P workers, one could in principle apply the plain parallel algorithm, described in Algorithm 5, which would lead to the following scheme. All the workers update in parallel their variables $(\mathbf{x}_{pi})_{i\in I_p}$, but each worker can process only one block $\mathbf{x}_{p\,i}$ at a time. This means that, at iteration k, every worker p computes a suitable $\mathbf{z}_{p\,i}^k$, for each block $i \in I_p$ (one at a time), by keeping all variables but $\mathbf{x}_{p\,i}$ fixed to $(\mathbf{x}_{p\,j}^k)_{i\neq j\in I_p}$ and $\mathbf{x}_{-p}^k$. Since we are solving the problems for each group of variables assigned to a worker sequentially but without using the most recent information for the updates, this approach seems a waste of resources. It is much more efficient to use, within each worker, a Gauss-Seidel scheme whereby the *most recent* iterates are used in all subsequent calculations. More specifically, at each iteration k, each worker p computes sequentially an inexact version $\mathbf{z}_{p\,i}^k$ of the solution $\widehat{\mathbf{x}}_{p\,i}\left(\mathbf{x}_{p,\,i<}^{k+1}, \mathbf{x}_{p,\,i\geq}^{k}, \mathbf{x}_{-p}^{k}\right)$, with $i \in I_p$, and updates $\mathbf{x}_{p\,i}$ as $\mathbf{x}_{pi}^{k+1} = \mathbf{x}_{p\,i}^k + \gamma^k\left(\mathbf{z}_{p\,i}^k - \mathbf{x}_{p\,i}^k\right)$. The scheme is summarized in Algorithm 8. Convergence

Algorithm 8 Hybrid parallel-cyclic FLEXA

```
Data : $\mathbf{x}^0 \in X$, $\{\gamma^k \in (0,1]\}_{k\in\mathbb{N}_+}$, $\varepsilon_{p\,i}^k \geq 0$, for all $i \in I_p$, $p \in \{1,\ldots,P\}$, and $k \in \mathbb{N}_+$, $\rho \in (0,1]$.
        Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : For all $p \in \{1,\ldots,P\}$ do (in parallel),
            For all $i \in I_p$ do (sequentially)
                a) Find $\mathbf{z}_{p\,i}^k \in X_i$ s.t.
                    $\left\|\mathbf{z}_{p\,i}^k - \widehat{\mathbf{x}}_{p\,i}\left(\mathbf{x}_{p,\,i<}^{k+1}, \mathbf{x}_{p,\,i\geq}^{k}, \mathbf{x}_{-p}^{k}\right)\right\| \leq \varepsilon_{p\,i}^k$;
                b) Set $\mathbf{x}_{pi}^{k+1} \triangleq \mathbf{x}_{p\,i}^k + \gamma^k\left(\mathbf{z}_{p\,i}^k - \mathbf{x}_{p\,i}^k\right)$;
(S.3) : $k \leftarrow k+1$, and go to (S.1).
```

of the algorithm was studied in [79] when $\{\gamma^k\}_{k\in\mathbb{N}_+}$ is chosen according to the diminishing step-size rule [Assumption 3.11.2]. The proof can be extended to deal with the other step-size rules (Assumption 3.11) following similar steps as those in Sect. 3.3.3.2, and thus is omitted.

3.3.4.3 Parallel-Greedy-Cyclic Schemes

In Algorithm 8, at each iteration k, the workers update all their blocks, sequentially over $(\mathbf{x}_{pi})_{i\in I_p}$. However, in some large-scale instances of Problem (3.94), updating all variables might not always be beneficial or doable. Furthermore, using the latest information in the updates of each worker may require extra calculations (e.g., computing all block gradients) and communication overhead (these aspects are discussed on specific examples in Sect. 3.3.5). It may be of interest then to consider a hybrid parallel-greedy-cyclic scheme, where Algorithm 8 is equipped with a greedy selection rule. More specifically, at each iteration k, each worker proceeds as in Algorithm 8 but performing the cyclic sweeping only on a subset S_p^k of its own variables I_p, where the subset S_p^k is chosen according to the greedy rule in Assumption 3.10. To introduce formally the scheme, we extend the notation used in Algorithm 8 as follows. We reorder the components of $\mathbf{x}_p$ so that the first $|S_p^k|$ variables are those in S_p^k and the remaining variables are those in $I_p \setminus S_p^k$; we write $\mathbf{x}_p = (\mathbf{x}_{S_p^k}, \mathbf{x}_{I_p\setminus S_p^k})$, where $\mathbf{x}_{S_p^k}$ denotes the vector containing the variables indexed in S_p^k while $\mathbf{x}_{I_p\setminus S_p^k}$ contains the remaining variables in I_p but S_p^k. Given an index $i \in S_p^k$, we partition $\mathbf{x}_{S_p^k}$ as $\mathbf{x}_{S_p^k} = (\mathbf{x}_{S_p^k i<}, \mathbf{x}_{S_p^k i\geq})$, where $\mathbf{x}_{S_p^k i<}$ is the vector containing all variables in S_p^k that appear before i (in the order assumed in S_p^k), while $\mathbf{x}_{S_p^k i\geq}$ are the remaining variables in S_p^k. With a slight abuse of notation, we will write $\mathbf{x} = (\mathbf{x}_{S_p^k i<}, \mathbf{x}_{S_p^k i\geq}, \mathbf{x}_{I_p\setminus S_p^k}, \mathbf{x}_{-p})$. The parallel-greedy-cyclic FLEXA is formally described in Algorithm 9.

3.3.5 *Applications*

In this section we apply (Hybrid) FLEXA to some representative convex and nonconvex problems, arising from applications in communications and signal processing/machine learning. More specifically, we consider the following problems: (i) the transceiver design in Single-Input-Single-Output (SISO) and Multiple-Input-Multiple-Output (MIMO) multiuser interference systems (cf. Sect. 3.3.5.1); (ii) the LASSO problem (cf. Sect. 3.3.5.2); and (iii) the logistic regression problem (cf. Sect. 3.3.5.3). For each of the problems above, we show how to choose ad-hoc surrogate functions and provide extensive numerical results.

Algorithm 9 Hybrid parallel-greedy-cyclic FLEXA

```
Data : $\mathbf{x}^0 \in X$, $\{\gamma^k \in (0,1]\}_{k \in \mathbb{N}_+}$, $\varepsilon^k_{p\,i} \geq 0$, for all $i \in I_p$, $p \in \{1, \ldots, P\}$, and $k \in \mathbb{N}_+$, $\rho \in (0,1]$.
        Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Set $M^k \triangleq \max_i \{E_i(\mathbf{x}^k)\}$.
          Choose sets $S^k_p \subseteq I_p$ so that $\cup_{p=1}^P S^k_p$ contains at least one index $i$
          for which $E_i(\mathbf{x}^k) \geq \rho M^k$.
(S.3) : For all $\ell \in P$ do (in parallel),
            For all $i \in S^k_\ell$ do (sequentially)
                a) Find $\mathbf{z}^k_{\ell i} \in X_i$ s.t.
                    $\left\| \mathbf{z}^k_{\ell i} - \widehat{\mathbf{x}}_{\ell i} \left( \mathbf{x}^{k+1}_{S^k_\ell i <}, \mathbf{x}^k_{S^k_\ell i \geq}, \mathbf{x}_{I_\ell \setminus S^k_\ell}, \mathbf{x}^k_{-\ell} \right) \right\| \leq \varepsilon^k_{\ell i};$
                b) Set $\mathbf{x}^{k+1}_{\ell i} \triangleq \mathbf{x}^k_{\ell i} + \gamma^k \left( \mathbf{z}^k_{\ell i} - \mathbf{x}^k_{\ell i} \right)$
(S.4) : Set $\mathbf{x}^{k+1}_{\ell i} = \mathbf{x}^k_{\ell i}$ for all $i \notin S^k$,
          $k \leftarrow k + 1$, and go to (S.1).
```

3.3.5.1 Resource Allocation in SISO/MIMO Multiuser Systems

Consider a multiuser system, composed of I transmitter-receiver pairs (users). Each transmitter (resp. receiver) is equipped with n_T (resp. n_R) transmit (resp. receive) antennas (w.l.o.g. we assumed that all the transceivers have the same number of antennas). Each transmitter is interested in transmitting its own data stream to its own receiver. No multiple access scheme is fixed a-priori (like OFDMA, TDMA, or CDMA); hence, Multi-User Interference (MUI) is experienced at each receiver. Let $\mathbf{x}_i \in \mathbb{C}^{n_T}$ be the signal transmitted by user i. Assuming a linear channel model, the received signal from user i reads

$$\mathbf{y}_i = \underbrace{\mathbf{H}_{ii}\,\mathbf{x}_i}_{\text{desired signal}} + \underbrace{\sum_{j \neq i} \mathbf{H}_{ij}\,\mathbf{x}_j}_{\text{multiuser interference}} + \underbrace{\mathbf{n}_i}_{\text{noise}}, \tag{3.144}$$

where $\mathbf{H}_{ij} \in \mathbb{C}^{n_R \times n_T}$ is the channel matrix between the transmitter j and the receiver i, and $\mathbf{n}_i$ is the additive Gaussian zero-mean noise at the receiver i, with variance $\sigma_i^2 > 0$ (the noise is assumed to be white without loss of generality, otherwise one can always pre-whiten the channel matrices). The first term on the RHS of (3.144) represents the useful signal for user i while the second one is the MUI due to the other users' concurrent transmissions. Note that the system model in (3.144) captures a fairly general MIMO setup, describing multiuser transmissions over multiple channels, which may represent frequency channels (as in multicarrier systems), time slots (as in time-division multiplexing systems), or spatial channels (as in transmit/receive beamforming systems); each of the aforementioned cases corresponds to a specific structure of the channel matrices $\mathbf{H}_{ij}$.

Denoting by $\mathbf{Q}_i \triangleq \mathbb{E}(\mathbf{x}_i\mathbf{x}_i^H) \succeq \mathbf{0}$ the covariance matrix of the symbols transmitted by agent i, each transmitter i is subject to the following general power constraints

$$\mathcal{Q}_i \triangleq \left\{\mathbf{Q}_i \in \mathbb{C}^{n_T \times n_T} \,:\, \mathbf{Q}_i \succeq \mathbf{0}, \quad \mathrm{tr}(\mathbf{Q}_i) \leq P_i^{\mathrm{ave}}, \quad \mathbf{Q}_i \in \mathcal{Z}_i\right\}, \tag{3.145}$$

where $\mathrm{tr}(\mathbf{Q}_i) \leq P_i^{\mathrm{ave}}$ is a constraint on the maximum average transmit power, with P_i^{ave} being the transmit power in unit of energy per transmission; and $\mathcal{Z}_i \subseteq \mathbb{C}^{n_T \times n_T}$ is an arbitrary closed and convex set, which can capture additional power/interference constraints (if any), such as: (i) null constraints $\mathbf{U}_i^H\mathbf{Q}_i = \mathbf{0}$, where $\mathbf{U}_i \in \mathbb{C}^{n_T \times r_i}$ is a full rank matrix with $r_i < n_T$, whose columns represent the spatial and/or "frequency" directions along with user i is not allowed to transmit; (ii) soft-shaping constraints $\mathrm{tr}\left(\mathbf{G}_i^H\mathbf{Q}_i\mathbf{G}_i\right) \leq I_i^{\mathrm{ave}}$, which permit to control the power radiated (and thus the interference generated) onto the range space of $\mathbf{G}_i \in \mathbb{C}^{n_T \times n_T}$; (iii) peak-power constraints $\lambda_{\max}\left(\mathbf{T}_i^H\mathbf{Q}_i\mathbf{T}_i\right) \leq I_i^{\mathrm{peak}}$, which limit the average peak power of transmitter i along the direction spanned by the range space of $\mathbf{T}_i \in \mathbb{C}^{n_T \times n_T}$, with $\lambda_{\max}$ denoting the maximum eigenvalue of the argument matrix; and (iv) per-antenna constraints $[\mathbf{Q}_i]_{nn} \leq \alpha_{in}$, which control the maximum average power radiated by each antenna.

Under standard information theoretical assumptions, the maximum achievable rate on each link i can be written as follows: given $\mathbf{Q} \triangleq (\mathbf{Q}_i)_{i=1}^I$,

$$R_i(\mathbf{Q}_i, \mathbf{Q}_{-i}) \triangleq \log\det\left(\mathbf{I} + \mathbf{H}_{ii}^H\mathbf{R}_i(\mathbf{Q}_{-i})^{-1}\mathbf{H}_{ii}\mathbf{Q}_i\right), \tag{3.146}$$

where $\det(\bullet)$ is the determinant of the argument matrix; $\mathbf{Q}_{-i} \triangleq (\mathbf{Q}_j)_{j\neq i}$ denotes the tuple of the (complex-valued) covariance matrices of all the transmitters except the i-th one; and $\mathbf{R}_i(\mathbf{Q}_{-i}) \triangleq \mathbf{R}_{n_i} + \sum_{j\neq i}\mathbf{H}_{ij}\mathbf{Q}_j\mathbf{H}_{ij}^H$ is the covariance matrix of the multiuser interference plus the thermal noise $\mathbf{R}_{n_i}$ (assumed to be full-rank).

As system design, we consider the maximization of the users' (weighted) sum rate, subject to the power constraint (3.145), which reads

$$\begin{aligned} &\underset{\mathbf{Q}_1,\ldots,\mathbf{Q}_I}{\text{maximize}} && \sum_{i=1}^{I}\alpha_i\, R_i(\mathbf{Q}_i, \mathbf{Q}_{-i}) \\ &\text{subject to} && \mathbf{Q}_i \in \mathcal{Q}_i, \quad \forall i = 1,\ldots,I, \end{aligned} \tag{3.147}$$

where $(\alpha_i)_{i=1}^I$ are given positive weights, which one can use to prioritize some user with respect to another. We remark that the proposed algorithmic framework can be applied also to other objective functions involving the rate functions, see [81, 212].

Clearly (3.147) is an instance of (3.94) (with $G = 0$ and involving complex variables) and thus we can apply the algorithmic framework described in this lecture. We begin considering the sum-rate maximization problem (3.147) over

SISO frequency selective channels; we then extend the analysis to the more general MIMO case.

Sum-Rate Maximization Over SISO Interference Channels

Given the system model (3.144), consider SISO frequency selective channels: the channel matrices $\mathbf{H}_{ij}$ are $m \times m$ Toeplitz circulant matrices and $\mathbf{R}_{n_i}$ are $m \times m$ diagonal matrices, with diagonal entries $\sigma_{i\,1}^2, \ldots, \sigma_{i\,m}^2$ ($\sigma_{i\,\ell}^2$ is the variance of the noise on channel ℓ); and m is the length of the transmitted block [note that in (3.144) it becomes $n_T = n_R = m$]; see, e.g., [246]. The eigendecomposition of each $\mathbf{H}_{ij}$ reads: $\mathbf{H}_{ij} = \mathbf{F}\,\mathbf{D}_{ij}\,\mathbf{F}^H$, where $\mathbf{F}$ is the IFFT matrix, i.e., $[\mathbf{F}]_{\ell'\,\ell} = \exp(j\,2\pi\;(\ell'-1)(\ell-1)/N)/\sqrt{N}$, for $\ell', \ell = 1, \ldots N$; and $\mathbf{D}_{ij}$ is the diagonal matrix whose diagonal entries $H_{ij}(1), \ldots, H_{ij}(N)$ are the coefficients of the frequency-response of the channel between the transmitter j and the receiver i.

Orthogonal Frequency Division Multiplexing (OFDM) transmissions correspond to the following structure for the covariance matrices: $\mathbf{Q}_i = \mathbf{F}\,\mathrm{diag}(\mathbf{p}_i)\,\mathbf{F}^H$, where $\mathbf{p}_i \triangleq (p_{i\,\ell})_{\ell=1}^m$ is the transmit power profile of user i over the m frequency channels. The power constraints read: given $\mathbf{I}_i^{\max} \in \mathbb{R}_+^{q_i}$ and $\mathbf{W}_i \in \mathbb{R}_+^{q_i \times m}$,

$$P_i \triangleq \left\{\mathbf{p}_i \in \mathbb{R}_+^N \;:\; \mathbf{W}_i\,\mathbf{p}_i \leq \mathbf{I}_i^{\max}\right\}, \tag{3.148}$$

where the inequality has to be intended component-wise. To avoid redundant constraints, we assume w.l.o.g. that all the columns of $\mathbf{W}_i$ are linearly independent.

The maximum achievable rate on each link i becomes [cf. (3.146)]

$$r_i(\mathbf{p}_i, \mathbf{p}_{-i}) \triangleq \sum_{\ell=1}^{m} \log\left(1 + \frac{|H_{ii}\,(\ell)|^2\,p_{i\,\ell}}{\sigma_{i\,\ell}^2 + \sum_{j\neq i}\left|H_{ij}\,(\ell)\right|^2\,p_{j\,\ell}}\right), \tag{3.149}$$

where $\mathbf{p}_{-i} \triangleq (\mathbf{p}_j)_{j\neq i}$ is the power profile of all the users $j \neq i$.

The system design (3.147) reduces to the following nonconvex optimization problem

$$\begin{aligned} &\underset{\mathbf{p}_1,\ldots,\mathbf{p}_I}{\text{maximize}} && \sum_{i=1}^{I} \alpha_i\,r_i(\mathbf{p}_i, \mathbf{p}_{-i}) \\ &\text{subject to} && \mathbf{p}_i \in P_i, \quad \forall i = 1, \ldots, I. \end{aligned} \tag{3.150}$$

We apply next FLEXA (Algorithm 5) to (3.150); we describe two alternative SCA-decompositions, corresponding to two different choices of the surrogate functions.

Decomposition #1—Pricing Algorithms Since the sum-rate maximization problem (3.150) is an instance of the problem considered in Example 3 in Sect. 3.3.2.1, a first approach is to use the surrogate (3.106). Since the rate $r_i(\mathbf{p}_i, \mathbf{p}_{-i})$ is concave in $\mathbf{p}_i$, for any given $\mathbf{p}_{-i} \geq \mathbf{0}$, we have $\tilde{C}_i = \{i\}$ [cf. (3.105)] and thus $C_i \equiv \tilde{C}_i$, which leads to the following surrogate function: given $\mathbf{p}^k \geq \mathbf{0}$ at iteration k,

$$\widetilde{F}(\mathbf{p} \,|\, \mathbf{p}^k) = \sum_{i=1}^{I} \widetilde{F}_i(\mathbf{p}_i \,|\, \mathbf{p}^k),$$

where

$$\widetilde{F}_i(\mathbf{p}_i \,|\, \mathbf{p}^k) \triangleq \alpha_i \cdot r_i(\mathbf{p}_i, \mathbf{p}_{-i}^k) - \boldsymbol{\pi}_i(\mathbf{p}^k)^T(\mathbf{p}_i - \mathbf{p}_i^k) - \frac{\tau_i}{2}\left\|\mathbf{p}_i - \mathbf{p}_i^k\right\|^2,$$

τ_i is an arbitrary nonnegative constant, and $\boldsymbol{\pi}_i(\mathbf{p}^k) \triangleq (\pi_{i\,\ell}(\mathbf{p}^k))_{\ell=1}^m$ is defined as

$$\pi_{i\,\ell}(\mathbf{p}^k) \triangleq -\sum_{j \in N_i} \alpha_j \,|H_{ji}(\ell)|^2 \frac{\mathtt{snr}_{j\,\ell}^k}{(1 + \mathtt{snr}_{j\,\ell}^k) \cdot \mathtt{mui}_{j\,\ell}^k};$$

N_i denotes the set of (out) neighbors of user i, i.e., the set of users j's which user i interferers with; and $\mathtt{snr}_{j\,\ell}^k$ and $\mathtt{mui}_{j\,\ell}^k$ are the Signal-to-Interference-plus-Noise (SINR) and the multiuser interference-plus-noise power ratios experienced by user j on the frequency ℓ, generated by the power profile $\mathbf{p}^k$:

$$\mathtt{snr}_{j\,\ell}^k \triangleq \frac{|H_{jj}(\ell)|^2 p_{j\,\ell}^k}{\mathtt{mui}_{j\,\ell}^k}, \quad \text{and} \quad \mathtt{mui}_{j\,\ell}^k \triangleq \sigma_{j\,\ell}^2 + \sum_{i \neq j} |H_{ji}(\ell)|^2 p_{i\,\ell}^k.$$

All the users in parallel will then solve the following strongly concave subproblems: given $\mathbf{p}^k = (\mathbf{p}_i^k)_{i=1}^I$,

$$\hat{\mathbf{p}}_i(\mathbf{p}^k) \triangleq \underset{\mathbf{p}_i \in P_i}{\operatorname{argmax}} \left\{\alpha_i \cdot r_i(\mathbf{p}_i, \mathbf{p}_{-i}^k) - \boldsymbol{\pi}_i(\mathbf{p}^k)^T(\mathbf{p}_i - \mathbf{p}_i^k) - \frac{\tau_i}{2}\left\|\mathbf{p}_i - \mathbf{p}_i^k\right\|^2\right\}.$$

Note that the best-response $\hat{\mathbf{p}}_i(\mathbf{p}^k)$ can be computed in closed form (up to the multiplies associated with the inequality constraints in P_i) according to the following multi-level waterfilling-like expression [209]: setting each $\tau_i > 0$,

$$\begin{aligned}\hat{\mathbf{p}}_i(\mathbf{p}^k) \triangleq \Bigg[&\frac{1}{2}\mathbf{p}_i^k \circ \left(\mathbf{1} - (\mathbf{snr}_i^k)^{-1}\right) + \\ &-\frac{1}{2\,\tau_i}\left(\tilde{\boldsymbol{\mu}}_i - \sqrt{\left[\tilde{\boldsymbol{\mu}}_i - \tau_i\, \mathbf{p}_i^k \circ \left(\mathbf{1} + (\mathbf{snr}_i^k)^{-1}\right)\right]^2 + 4\tau_i w_i \mathbf{1}}\right)\Bigg]_+\end{aligned} \tag{3.151}$$

where $\circ$ denotes the Hadamard product and $[\bullet]_+$ denotes the projection onto the nonnegative orthant $\mathbb{R}_+^m$; $(\mathbf{snr}_i^k)^{-1} \triangleq (1/\mathtt{snr}_{i\,\ell}^k)_{\ell=1}^m$ and $\tilde{\boldsymbol{\mu}}_i \triangleq \boldsymbol{\pi}_i(\mathbf{p}^k) + \mathbf{W}_i^T \boldsymbol{\mu}_i$, with the multiplier vector $\boldsymbol{\mu}_i$ chosen to satisfy the nonlinear complementarity condition (CC)

$$\mathbf{0} \leq \boldsymbol{\mu}_i \perp \mathbf{I}_i^{\max} - \mathbf{W}_i\, \hat{\mathbf{p}}_i(\mathbf{p}^k) \geq \mathbf{0}.$$

The optimal $\boldsymbol{\mu}_i$ satisfying the CC can be efficiently computed (in a finite number of steps) using the nested bisection method described in [209, Algorithm 6]; we omit further details here. Note that, in the presence of power budget constraints only, $\boldsymbol{\mu}_i$ reduces to a scalar quantity μ_i such that $0 \leq \mu_i \perp p_i - \mathbf{1}^T \hat{\mathbf{p}}_i(\mathbf{p}^k) \geq 0$, whose solution can be obtained using the classical bisection algorithm (or the methods in [182]).

Given $\hat{\mathbf{p}}_i(\mathbf{p}^k)$, one can now use, e.g., Algorithm 5, with any of the valid choices for the step-size $\{\gamma^k\}$ [cf. Assumption 3.6]. Since there is no coordination among the users as well as no centralized control in network, one is interested in designing distributed algorithms. This naturally suggests the use of a diminishing step-size rule in Algorithm 5. For instance, good candidates are the rules in (3.108) or (3.109). Note that the resulting algorithm is fairly distributed. Indeed, given the interference generated by the other users [and thus the MUI coefficients $\mathtt{mui}_{jn}^k$] and the current interference price $\boldsymbol{\pi}_i(\mathbf{p}^k)$, each user can efficiently and locally compute the optimal power allocation $\hat{\mathbf{p}}_i(\mathbf{p}^k)$ via the waterfilling-like expression (3.151). The estimation of the prices $\pi_{i\ell}(\mathbf{p}^k)$ requires however some signaling among nearby users.

Decomposition #2—DC Algorithms An alternative class of algorithms for the sum-rate maximization problem (3.150) can be obtained exploring the DC structure of the rate functions (3.149). By doing so, the sum-rate can be decomposed as the sum of a concave and convex function, namely $U(\mathbf{p}) = f_1(\mathbf{p}) + f_2(\mathbf{p})$, with

$$f_1(\mathbf{p}) \triangleq \sum_{i=1}^{I} \alpha_i \sum_{\ell=1}^{m} \log\left(\sigma_{i\,\ell}^2 + \sum_{j=1}^{I} \left|H_{ij}(\ell)\right|^2 p_{j\,\ell}\right),$$

$$f_2(\mathbf{p}) \triangleq -\sum_{i=1}^{I} \alpha_i \sum_{\ell=1}^{m} \log\left(\sigma_{i\ell}^2 + \sum_{j=1, j\neq i}^{I} \left|H_{ij}(\ell)\right|^2 p_{j\,\ell}\right).$$

A concave surrogate can be readily obtained from $U(\mathbf{p})$ by linearizing $f_2(\mathbf{p})$ and keeping $f_1(\mathbf{p})$ unaltered. This leads to the following strongly concave subproblem for each agent i: given $\mathbf{p}^k \geq \mathbf{0}$,

$$\widetilde{\mathbf{p}}_i(\mathbf{p}^k) \triangleq \underset{\mathbf{p}_i \in P_i}{\operatorname{argmax}} \left\{ f_1(\mathbf{p}_i, \mathbf{p}_{-i}^k) - \boldsymbol{\pi}_i(\mathbf{p}^k)^T (\mathbf{p}_i - \mathbf{p}_i^k) - \frac{\tau_i}{2} \left\| \mathbf{p}_i - \mathbf{p}_i^k \right\|^2 \right\}$$

where $\boldsymbol{\pi}_i(\mathbf{p}^k) \triangleq (\pi_{i\,\ell}(\mathbf{p}^k))_{\ell=1}^m$, with

$$\pi_{i\,\ell}(\mathbf{p}^k) \triangleq -\sum_{j\in N_i} \alpha_j\, |H_{ji}\,(\ell)\,|^2\, \frac{1}{\mathtt{mui}_{j\,\ell}^k}. \tag{3.152}$$

The best-response $\widetilde{\mathbf{p}}_i(\mathbf{p}^k)$ can be efficiently computed using a fixed-point-based procedure, in the same spirit of [183]; we omit further details. Note that the communication overhead to compute the prices (3.151) and (3.152) is the same, but the computation of $\widetilde{\mathbf{p}}_i(\mathbf{p}^k)$ requires more (channel state) information exchange than that of $\hat{\mathbf{p}}_i(\mathbf{p}^k)$, since each user i also needs to estimate the cross-channels $\{|H_{ji}\,(\ell)\,|^2\}_{j\in N_i}$.

Numerical Example We compare now Algorithm 5 based on the best-response $\hat{\mathbf{p}}_i(\mathbf{p}^k)$ in (3.151) (termed `SR-FLEXA`, `SR` stands for Sum-Rate), with those proposed in [183] [termed `SCALE` and `SCALE one-step`, the latter being a simplified version of `SCALE` where instead of solving the fixed-point equation (16) in [183], only one iteration of (16) is performed], [207] (which is an instance of the block MM algorithm described in Algorithm 2, and is termed `Block-MM`), and [215] (termed `WMMSE`). Since the algorithms in [183, 207, 215] can only deal with power budget constraints, to run the comparison, we simplified the sum-rate maximization problem (3.150) considering only power budget constraints (and all $\alpha_i = 1$). We assume the same power budget $P_i^{\text{ave}} = p$, noise variances $\sigma_{i\,\ell}^2 = \sigma^2$, and $\mathtt{snr} = p/\sigma^2 = 3$dB for all the users. We simulated SISO frequency-selective channels with $m = 64$ subcarriers; the channels are generated as FIR filters of order $L = 10$, whose taps are i.i.d. Gaussian random variables with zero mean and variance $1/(d_{ij}^3(L+1)^2)$, where d_{ij} is the distance between the transmitter j and the receiver i. All the algorithms are initialized by choosing the uniform power allocation, and are terminated when (the absolute value of) the sum-utility error in two consecutive rounds becomes smaller than $1e$-3. The accuracy in the bisection loops (required by all methods) is set to $1e$-6. In `SR-FLEXA`, we used the rule (3.108) with $\epsilon = 1e$-2. In Fig. 3.8, we plot the average number of iterations required by the aforementioned algorithms to converge (under the same termination criterion) versus the number of users; the average is taken over 100 independent channel realizations; in Fig. 3.8a we set $d_{ij}/d_{ii} = 3$ whereas in Fig. 3.8b we have $d_{ij}/d_{ii} = 1$ while in both figures $d_{ij} = d_{ji}$ and $d_{ii} = d_{jj}$, for all i and $j \neq i$; the setting in Fig. 3.8a emulates a "low" MUI environment whereas the one in Fig. 3.8b a "high" MUI scenario. All the algorithms reach the same average sum-rate. The figures clearly show that the proposed `SR-FLEXA` outperforms all the others (note that `SCALE` and `WMMSE` are also simultaneous-based schemes). For instance, in Fig. 3.8a, the gap with `WMMSE` (in terms of number of iterations needed to reach convergence) is about one order of magnitude, for all the network sizes considered in the experiment, which reduces to two times in the "high" interference scenario considered in Fig. 3.8b. Such a behavior (requiring less iterations than other methods, with gaps ranging from few times to one order of magnitude) has

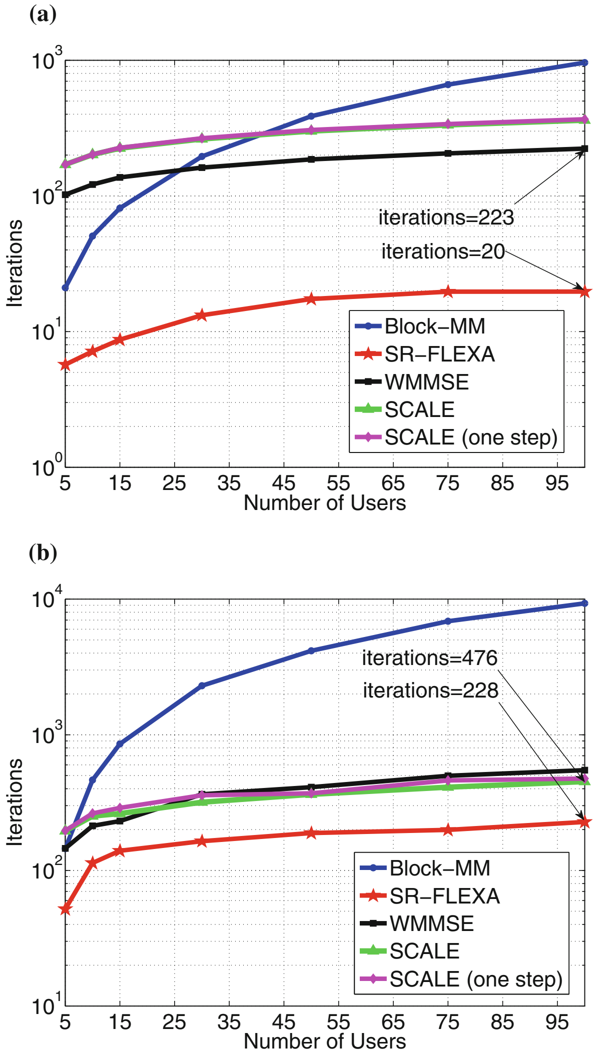


Fig. 3.8 Sum-rate maximization problem (3.150) (SISO frequency-selective channels): Average number of iterations versus number of users. Note that all algorithms are simultaneous except `Block-MM`, which is sequential. Also, all the algorithms are observed to converge to the same stationary solution of Problem (3.150). The figures are taken from [210]. (**a**) Low MUI: The proposed method, `SR-FLEXA`, is one order of magnitude faster than the WMMSE algorithm. (**b**) High MUI: The proposed method, `SR-FLEXA`, is two times faster than the WMMSE algorithm

been observed also for other choices of d_{ij}/d_{ii}, termination tolerances, and step-size rules; more experiments can be found in [210, 221]. Note that `SR-FLEXA`, `SCALE one-step`, `WMMSE`, and `Block-MM` have similar per-user computational complexity, whereas `SCALE` is much more demanding and is not appealing for a real-time implementation. Therefore, Fig. 3.8 provides also a roughly indication of the per-user cpu time of `SR-FLEXA`, `SCALE one-step`, and `WMMSE`.

Sum-Rate Maximization Over MIMO Interference Channels

Let us focus now on the general MIMO formulation (3.147).

Similarly to the SISO case, we can invoke the surrogate (3.106) with $C_i = \{i\}$, corresponding to keeping R_i in (3.146) unaltered and linearizing the rest of the sum, that is, $\sum_{j \neq i} R_j$. Invoking the Wirtinger calculus (see, e.g., [105, 126, 209]), the subproblem solved by each agent i at iteration k reads: given $\mathbf{Q}^k = (\mathbf{Q}_i^k)_{i=1}^I$, with each $\mathbf{Q}^k \succeq \mathbf{0}$,

$$\hat{\mathbf{Q}}_i(\mathbf{Q}^k) \triangleq \underset{\mathbf{Q}_i \in \mathcal{Q}_i}{\operatorname{argmax}} \left\{ \alpha_i \, r_i(\mathbf{Q}_i, \mathbf{Q}_{-i}^k) - \left\langle \boldsymbol{\Pi}_i(\mathbf{X}^k), \mathbf{Q}_i \right\rangle - \tau_i \left\| \mathbf{Q}_i - \mathbf{Q}_i^k \right\|_F^2 \right\} \tag{3.153}$$

where $\langle \mathbf{A}, \mathbf{B} \rangle \triangleq \operatorname{Re}\{\operatorname{tr}(\mathbf{A}^H \mathbf{B})\}$; $\tau_i > 0$,

$$\boldsymbol{\Pi}_i(\mathbf{Q}^k) \triangleq \sum_{j \in N_i} \alpha_j \, \mathbf{H}_{ji}^H \, \widetilde{\mathbf{R}}_j(\mathbf{Q}_{-j}^k) \, \mathbf{H}_{ji},$$

with N_i defined as in the SISO case; and

$$\widetilde{\mathbf{R}}_j(\mathbf{Q}_{-j}^k) \triangleq \mathbf{R}_j(\mathbf{Q}_{-j}^k)^{-1} - (\mathbf{R}_j(\mathbf{Q}_{-j}^k) + \mathbf{H}_{jj}\mathbf{Q}_j^k\mathbf{H}_{jj}^H)^{-1}.$$

Note that, once the price matrix $\boldsymbol{\Pi}_i(\mathbf{Q}^k)$ is given, the best-response $\hat{\mathbf{Q}}_i(\mathbf{Q}^k)$ can be computed locally by each user solving a convex optimization problem. Moreover, for some specific structures of the feasible sets $\mathcal{Q}_i$, the case of full-column rank channel matrices $\mathbf{H}_i$, and $\tau_i = 0$, a solution in closed form (up to the multipliers associated with the power budget constraints) is also available [124]; see also [260] for other examples. Given $\hat{\mathbf{Q}}_i(\mathbf{Q}^k)$, one can now use Algorithm 5 (adapted to the complex case), with any of the valid choices for the step-size $\{\gamma^k\}$.

Complexity Analysis and Message Exchange We compare here the computational complexity and signaling (i.e., message exchange) of Algorithm 5 based on the best-response $\hat{\mathbf{Q}}_i(\mathbf{Q}^k)$ (termed `MIMO-SR-FLEXA`) with those of the schemes proposed in the literature for a similar problem, namely the `MIMO-Block-MM` [124, 207], and the `MIMO-WMMSE` [215]. For the purpose of complexity analysis, since all algorithms include a similar bisection step which generally takes few iterations, we will ignore this step in the computation of the complexity. Also, MIMO-WMMSE and MIMO-SR-FLEXA are simultaneous schemes, while MIMO-Block-MM is sequential; we then compare the algorithms by given the *per-round complexity*, where one round means one update from all the users. Recalling that n_T (resp. n_R) denotes the number of antennas at each transmitter (resp. receiver), the computational complexity of the algorithms is [210]:

- MIMO-Block-MM: $O\left(I^2(n_T n_R^2 + n_T^2 n_R + n_R^3) + I\, n_T^3\right)$;
- MIMO-WMMSE: $O\left(I^2(n_T n_R^2 + n_T^2 n_R + n_T^3) + I\, n_R^3\right)$ [215];
- MIMO-SR-FLEXA: $O\left(I^2(n_T n_R^2 + n_T^2 n_R) + I(n_T^3 + n_R^3)\right)$.

The complexity of the three algorithms is very similar, and equivalent in the case in which $n_T = n_R (\triangleq m)$, given by $O(I^2 m^3)$.

In a real system, the MUI covariance matrices $\mathbf{R}_i(\mathbf{Q}_{-i})$ come from an estimation process. It is thus interesting to understand how the complexity changes when the computation of $\mathbf{R}_i(\mathbf{Q}_{-i})$ from $\mathbf{R}_{n_i} + \sum_{j\neq i} \mathbf{H}_{ij}\mathbf{Q}_j\mathbf{H}_{ij}^H$ is not included in the analysis. We obtain the following [210]:

- MIMO-Block-MM: $O\left(I^2(n_T n_R^2 + n_T^2 n_R + n_R^3) + I\, n_T^3\right)$;
- MIMO-WMMSE: $O\left(I^2(n_T^2 n_R + n_T^3) + I(n_R^3 + n_T n_R^2)\right)$;
- MIMO-SR-FLEXA: $O\left(I^2(n_T n_R^2 + n_T^2 n_R) + I(n_T^3 + n_R^3)\right)$.

Finally, if one is interested in the time necessary to complete one iteration, it can be shown that it is proportional to the above complexity divided by I.

As far as the communication overhead is concerned, the same remarks we made about the schemes described in the SISO setting, apply also here for the MIMO case. The only difference is that now the users need to exchange a (pricing) matrix rather than a vector, resulting in $O(I^2\, n_R^2)$ amount of message exchange per-iteration for all the algorithms.

Numerical Example #1 In Tables 3.3 and 3.4 we compare the MIMO-SR-FLEXA, the MIMO-Block-MM [124, 207], and the MIMO-WMMSE [215], in terms of average number of iterations required to reach convergence, for different number of users, normalized distances $d \triangleq d_{ij}/d_{ii}$ (with $d_{ij} = d_{ji}$ and $d_{ii} = d_{jj}$ for all i and $j \neq i$), and termination accuracy (namely: $1e$-3 and $1e$-6). All the transmitters/receivers are equipped with four antenna; we simulated uncorrelated fading channels, whose coefficients are Gaussian distributed with zero mean and variance $1/d_{ij}^3$ (all the

Table 3.3 Sum-rate maximization problem (3.147) (MIMO frequency-selective channels): average number of iterations (termination accuracy=$1e$-6)

	# of users = 10			# of users = 50			# of users = 100		
	d=1	d=2	d=3	d=1	d=2	d=3	d=1	d=2	d=3
MIMO-Block-MM	1370.5	187	54.4	4148.5	1148	348	8818	1904	704
MIMO-WMMSE	169.2	68.8	53.3	138.5	115.2	76.7	154.3	126.9	103.2
MIMO-SR-FLEXA	169.2	24.3	6.9	115.2	34.3	9.3	114.3	28.4	9.7

Table 3.4 Sum-rate maximization problem (3.147) (MIMO frequency-selective channels): average number of iterations (termination accuracy=1e-3)

	# of users = 10			# of users = 50			# of users = 100		
	d=1	d=2	d=3	d=1	d=2	d=3	d=1	d=2	d=3
MIMO-Block-MM	429.4	74.3	32.8	1739.5	465.5	202	3733	882	442.6
MIMO-WMMSE	51.6	19.2	14.7	59.6	24.9	16.3	69.8	26.0	19.2
MIMO-SR-FLEXA	48.6	9.4	4.0	46.9	12.6	5.1	49.7	12	5.5

channel matrices are full-column rank); and we set $\mathbf{R}_{n_i} = \sigma^2\mathbf{I}$ for all i, and $\texttt{snr} \triangleq p/\sigma^2 = 3$ dB. In MIMO-SR-FLEXA, we used the step-size rule (3.108), with $\epsilon = 1e$-5; in (3.153) we set $\tau_i = 0$ and computed $\hat{\mathbf{Q}}_i(\mathbf{Q}^k)$ using the closed form solution in [124]. All the algorithms reach the same average sum-rate.

Given the results in Tables 3.3 and 3.4, the following comments are in order. MIMO-SR-FLEXA outperforms the other schemes in terms of iterations, while having similar (or even better) computational complexity. Interestingly, the iteration gap with the other schemes reduces with the distance and the termination accuracy. More specifically: MIMO-SR-FLEXA (i) seems to be much faster than the other schemes (about one order of magnitude) when $d_{ij}/d_{ii} = 3$ [say low interference scenarios], and just a bit faster (or comparable to MIMO-WMMSE) when $d_{ij}/d_{ii} = 1$ [say high interference scenarios]; and (ii) it is much faster than the others, if an high termination accuracy is set (see Table 3.3). Also, the convergence speed of MIMO-SR-FLEXA is not affected too much by the number of users. Finally, in our experiments, we also observed that the performance of MIMO-SR-FLEXA is not affected too much by the choice of the parameter ϵ in the (3.108): a change of ϵ of many orders of magnitude leads to a difference in the average number of iterations which is within 5%; we refer the reader to [221] for details, where one can also find a comparison of several other step-size rules. We must stress however that MIMO-Block-MM and MIMO-WMMSE do not need any tuning, which is an advantage with respect to MIMO-SR-FLEXA.

Numerical Example #2 We compare now the MIMO-WMMSE [215] and the MIMO-SR-FLEXA in a MIMO broadcast cellular system composed of multiple cells, with one Base Station (BS) and multiple randomly generated Mobile Terminals (MTs) in each cell. Each MT experiences both intra-cell and inter-cell interference. We refer to [215] for a detailed description of the system model, the explicit expressions of the BS-MT downlink rates, and the corresponding sum-rate maximization problem.

The setup of our experiments is the following [210]. We simulated seven cells with multiple randomly generated MTs; each BS and MT is equipped with four transmit and receive antennas. Channels are Rayleigh fading, whose path-loss are generated using the 3 GPP(TR 36.814) methodology [1]. We assume white zero-mean Gaussian noise at each mobile receiver, with variance σ^2, and same power budget p for all the BSs; the SNR is set to $\texttt{snr} \triangleq p/\sigma^2 = 3$dB. Both algorithms MIMO-WMMSE and MIMO-SR-FLEXA are initialized by choosing the same feasible randomly generated point, and are terminated when (the absolute value of) the sum-rate error in two consecutive rounds becomes smaller than 1*e*-2. In MIMO-SR-FLEXA, the step-size rule (3.108) is used, with $\epsilon = 1e\text{-}3$ and $\gamma^0 = 1$; the unique solution $\hat{\mathbf{Q}}_i(\mathbf{Q}^k)$ of users' subproblems is computed in closed form adapting the procedure in [124]. The experiments were run using Matlab R2012a on a 12 × 2.40 GHz Intel Xeon E5645 Processor Cores machine, equipped with 48 GB of memory and 24,576 Kbytes of data cache; the operation system is Linux (RedHat Enterprise Linux 6.1 2.6.32 Kernel). In Fig. 3.9a we plot the average cpu time versus the total number of MTs for the two algorithms under the same termination criterion, whereas in Fig. 3.9b we reported the final achieved average sum-rate. The curves are averaged over 1500 channel/topology realizations. It can be observed that MIMO-SR-FLEXA significantly outperforms MIMO-WMMSE in terms of cpu time when the number of active users is large; moreover MIMO-SR-FLEXA also yields better sum-rates. We observed similar results also under different settings (e.g., SNR, number of cells/BSs, etc.); see [221] for more details.

3.3.5.2 LASSO Problem

Consider the LASSO problem in the following form [235] (cf. Sect. 3.3.1.1):

$$\underset{\mathbf{x}}{\text{minimize}}\ V(\mathbf{x}) \triangleq \frac{1}{2}\,\|\mathbf{z} - \mathbf{A}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1, \tag{3.154}$$

where $\mathbf{A} \in \mathbb{R}^{q\times m}$ is the matrix whose columns $\mathbf{a}_i$ are the prediction or feature vectors; z_i is the response variable associated to $\mathbf{a}_i$; and $\lambda > 0$ is the regularization weight.

FLEXA for LASSO Observing that the univariate instance of (3.154) has a closed form solution, it is convenient to decompose $\mathbf{x}$ in *scalar* components ($m_i = 1$, for all $i \in N$) and update them in parallel. In order to exploit the quadratic structure of V in (3.154) a natural choice for the surrogate function is (3.102). Therefore, the subproblem associated with the scalar x_i reads: given $\mathbf{x}^k$,

$$\widehat{x}_i(\mathbf{x}^k) \triangleq \underset{x_i\in\mathbb{R}}{\operatorname{argmin}} \left\{ \frac{1}{2}\,\left\|\mathbf{r}_i^k - \mathbf{a}_i\,x_i\right\|^2 + \frac{\tau_i}{2}\cdot(x_i - x_i^k)^2 + \lambda\cdot|x_i|\right\},$$

where the residual $\mathbf{r}_i^k$ is defined as

$$\mathbf{r}_i^k \triangleq \mathbf{z} - \sum_{j\neq i}\mathbf{a}_j\,x_j^k.$$

Invoking the first order optimality conditions (c.f. Definition 2.9) [we write $\widehat{x}_i$ for $\widehat{x}_i(\mathbf{x}^k)$]:

$$-\left(\mathbf{a}_i^T\,\mathbf{r}_i^k + \tau_i\,x_i^k\right) + \left(\tau_i + \|\mathbf{a}_i\|^2\right)\widehat{x}_i + \lambda\,\partial|\widehat{x}_i| \ni 0,$$

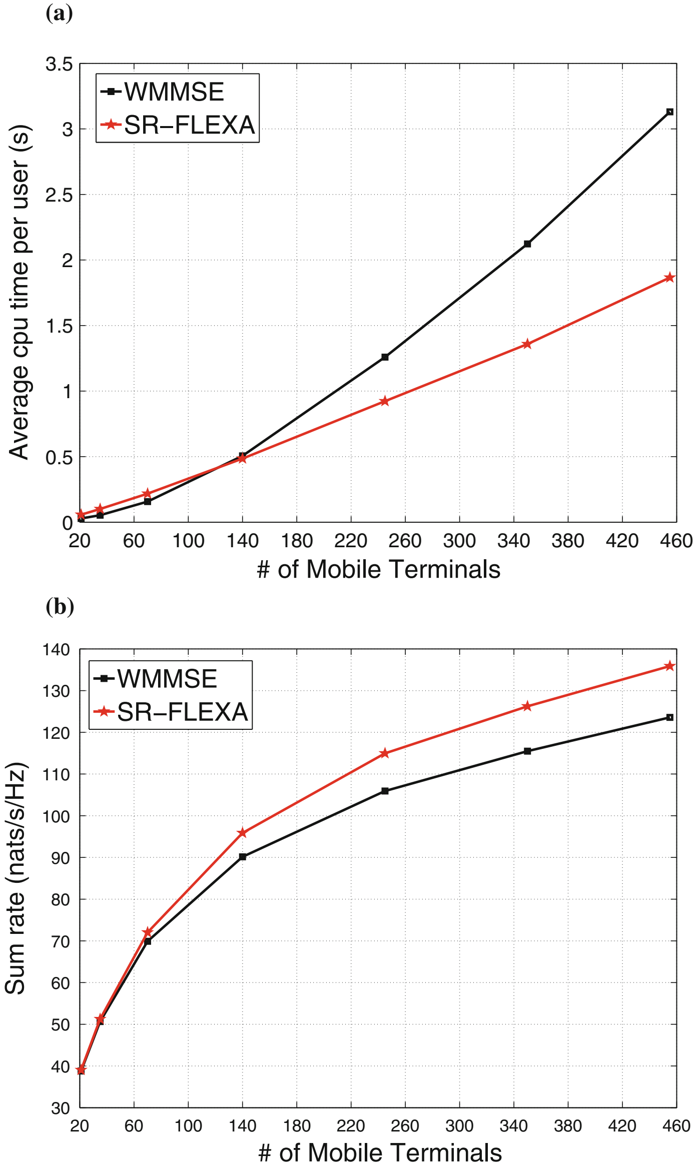


Fig. 3.9 Sum-rate maximization problem over Interference Broadcast Channels: `MIMO-SR-FLEXA` versus `MIMO-WMMSE`. The figures are taken from [210]. (**a**) Average cpu time versus the number of mobile terminals. (**b**) Average sum-rate versus the number of mobile terminals

and the expression of $\partial|x|$ [cf. (3.9)], one can readily obtain the closed form expression of $\widehat{x}_i(\mathbf{x}^k)$, that is,

$$\widehat{x}_i(\mathbf{x}^k) = \frac{1}{\tau_i + \|\mathbf{a}_i\|^2} \cdot \mathcal{S}_\lambda\left(\mathbf{a}_i^T\, \mathbf{r}_i^k + \tau_i\, x_i^k\right), \tag{3.155}$$

where $\mathcal{S}_\lambda(\bullet)$ is the soft-thresholding operator, defined in (3.54).

We consider the instance of Algorithm 6, with the following choice of the free parameters:

- *Exact solution* $\widehat{x}_i(\mathbf{x}^k)$: In Step 3 we use the best-response $\widehat{x}_i(\mathbf{x}^k)$ as defined in (3.155), that is, $z_i^k = \widehat{x}_i(\mathbf{x}^k)$ (exact solution).
- *Proximal weights* τ_i: While in the proposed algorithmic framework we considered fixed values of τ_i, varying τ_i a finite number of times does not affect the theoretical convergence properties of the algorithms. We found that the following choices work well in practice: (i) τ_i are initially all set to $\tau_i = \text{tr}(\mathbf{A}^T\mathbf{A})/(2m)$, i.e., to half of the mean of the eigenvalues of $\nabla^2 F$; (ii) all τ_i are doubled if at a certain iteration the objective function does not decrease; and (iii) they are all halved if the objective function decreases for ten consecutive iterations or the relative error on the objective function $\texttt{re}(\mathbf{x})$ is sufficiently small, specifically if

$$\texttt{re}(\mathbf{x}) \triangleq \frac{V(\mathbf{x}) - V^*}{V^*} \leq 10^{-2}, \tag{3.156}$$

  where V^* is the optimal value of the objective function V (in our experiments on LASSO, V^* is known). In order to avoid increments in the objective function, whenever all τ_i are doubled, the associated iteration is discarded, and in Step 4 of Algorithm 6 it is set $\mathbf{x}^{k+1} = \mathbf{x}^k$. In any case we limited the number of possible updates of the values of τ_i to 100.
- *Step-size* γ^k: The step-size γ^k is updated according to the following rule:

$$\gamma^k = \gamma^{k-1}\left(1 - \min\left\{1, \frac{10^{-4}}{\texttt{re}(\mathbf{x}^k)}\right\}\theta\,\gamma^{k-1}\right), \quad k = 1, \ldots, \tag{3.157}$$

  with $\gamma^0 = 0.9$ and $\theta = 1e - 7$. The above diminishing rule is based on (3.108) while guaranteeing that γ^k does not become too close to zero before the relative error is sufficiently small.
- *Greedy selection rule* S^k: In Step 2, we use the following greedy selection rule (satisfying Assumption 3.10.2):

$$S^k = \{i : E_i(\mathbf{x}^k) \geq \sigma \cdot M^k\}, \quad \text{with} \quad E_i(\mathbf{x}^k) = |\widehat{x}_i(\mathbf{x}^k) - x_i^k|.$$

  In our tests we consider two options for σ, namely: (i) $\sigma = 0$, which leads to a *fully parallel* scheme wherein at each iteration *all* variables are updated; and (ii) $\sigma = 0.5$, which corresponds to updating only a subset of all the variables at each iteration. Note that for both choices of σ, the resulting set S^k satisfies the requirement in Step 2 of Algorithm 6; indeed, S^k always contains the index i corresponding to the largest $E_i(\mathbf{x}^k)$. We will refer to these two instances of the algorithm as FLEXA $\sigma = 0$ and FLEXA $\sigma = 0.5$.

Algorithms in the Literature We compared the above versions of FLEXA with the most competitive parallel and sequential (Block MM) algorithms proposed in the literature to solve the LASSO problem. More specifically, we consider the following schemes.

- FISTA: The Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) proposed in [7] 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 V, 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 ∇F [7].
- SpaRSA: This is the first order method proposed in [251]; 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 all the experiments, we set the parameters of SpaRSA as in [251]: $M = 5$, $\sigma = 0.01$, $\alpha_{\max} = 1e30$, and $\alpha_{\min} = 1e - 30$.
- GRock & Greedy-1BCD: GRock is a parallel algorithm proposed in [187] that performs well on sparse LASSO 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 increases; roughly speaking, GRock is guaranteed to converge if the columns of the data matrix $\mathbf{A}$ in the LASSO problem are “almost” orthogonal, a feature that in general is not satisfied by real data. A special instance with convergence guaranteed is the one where only one block per time (chosen in a greedy fashion) is updated; we refer to this special case as greedy-1BCD.
- Parallel ADMM: This is a classical Alternating Method of Multipliers (ADMM). We implemented the parallel version proposed in [67].

In the implementation of the parallel algorithms, the data matrix $\mathbf{A}$ of the LASSO problem is generated as follows. Each processor generates a slice of the matrix itself such that $\mathbf{A} = [\mathbf{A}_1\, \mathbf{A}_2 \cdots \mathbf{A}_P]$, where P is the number of parallel processors, and each $\mathbf{A}_i$ has m/P columns. Thus the computation of each product $\mathbf{Ax}$ (which is required to evaluate ∇F) and the norm $\|\mathbf{x}\|_1$ (that is G) is divided into the parallel jobs of computing $\mathbf{A}_i\,\mathbf{x}_i$ and $\|\mathbf{x}_i\|_1$, followed by a reducing operation.

Numerical Examples We generated six groups of LASSO problems using the random generator proposed by Nesterov [174], which permits to control the sparsity of the solution. For the first five groups, we considered problems with 10,000 variables and matrices $\mathbf{A}$ with 9000 rows. The five groups differ in the degree of sparsity of the solution, namely: the percentage of non zeros in the solution is 1%, 10%, 20%, 30%, and 40%, respectively. The last group is formed by instances with 100,000 variables and 5000 rows for **A**, and solutions having 1% of non zero variables. In all experiments and for all the algorithms, the initial point was set to the zero vector.

Results of the experiments for the 10,000 variables groups are reported in Fig. 3.10, where we plot the relative error as defined in (3.156) versus the CPU time; all the curves are obtained using (up to) 40 cores, and averaged over ten independent random realizations. Note that the CPU time includes communication times (for parallel algorithms) and the initial time needed by the methods to perform all pre-iteration computations (this explains why the curves of ADMM start after the others; in fact ADMM requires some nontrivial initializations). For one instance, the one corresponding to 1% of the sparsity of the solution, we plot also the relative error versus iterations [Fig. 3.10(a2)]; similar behaviors of the algorithms have been observed also for the other instances, and thus are not reported. Results for the LASSO instance with 100,000 variables are plotted in Fig. 3.11. The curves are averaged over five random realizations.

The following comments are in order. On all the tested problems, FLEXA $\sigma = 0.5$ outperforms in a consistent manner all the other implemented algorithms. In particular, as the sparsity of the solution decreases, the problems become harder and the selective update operated by FLEXA $\sigma = 0.5$ improves over FLEXA $\sigma = 0$, where instead all variables are updated at each iteration. FISTA is capable to approach relatively fast low accuracy when the solution is not too sparse, but has difficulties in reaching high accuracy. SpaRSA seems to be very insensitive to the degree of sparsity of the solution; it behaves well on 10,000 variables problems and not too sparse solutions, but is much less effective on very large-scale problems. The version of GRock with $P = 40$ is the closest match to FLEXA, but only when the problems are very sparse (but it is not supported by a convergence theory on our test problems). This is consistent with the fact that its convergence properties are at stake when the problems are quite dense. Furthermore, if the problem is very large, updating only 40 variables at each iteration, as GRock does, could slow down the convergence, especially when the optimal solution is not very sparse. From this point of view, FLEXA $\sigma = 0.5$ seems to strike a good balance between not updating variables that are probably zero at the optimum and nevertheless update a sizeable amount of variables when needed in order to enhance convergence.

Remark 3.21 (On the Parallelism) Fig. 3.11 shows that FLEXA seems to exploit well parallelism on LASSO problems. Indeed, when passing from 8 to 20 cores, the running time approximately halves. This kind of behavior has been consistently observed also for smaller problems and different number of cores (not reported here). Note that practical speed-up due to the use of a parallel architecture is given by several factor that are not easily predictable, including communication times among the cores, the data format, etc. Here we do not pursue a theoretical study of the speed-up but refer to [37] for some initial study. We finally observe that GRock appears to improve greatly with the number of cores. This is due to the fact that

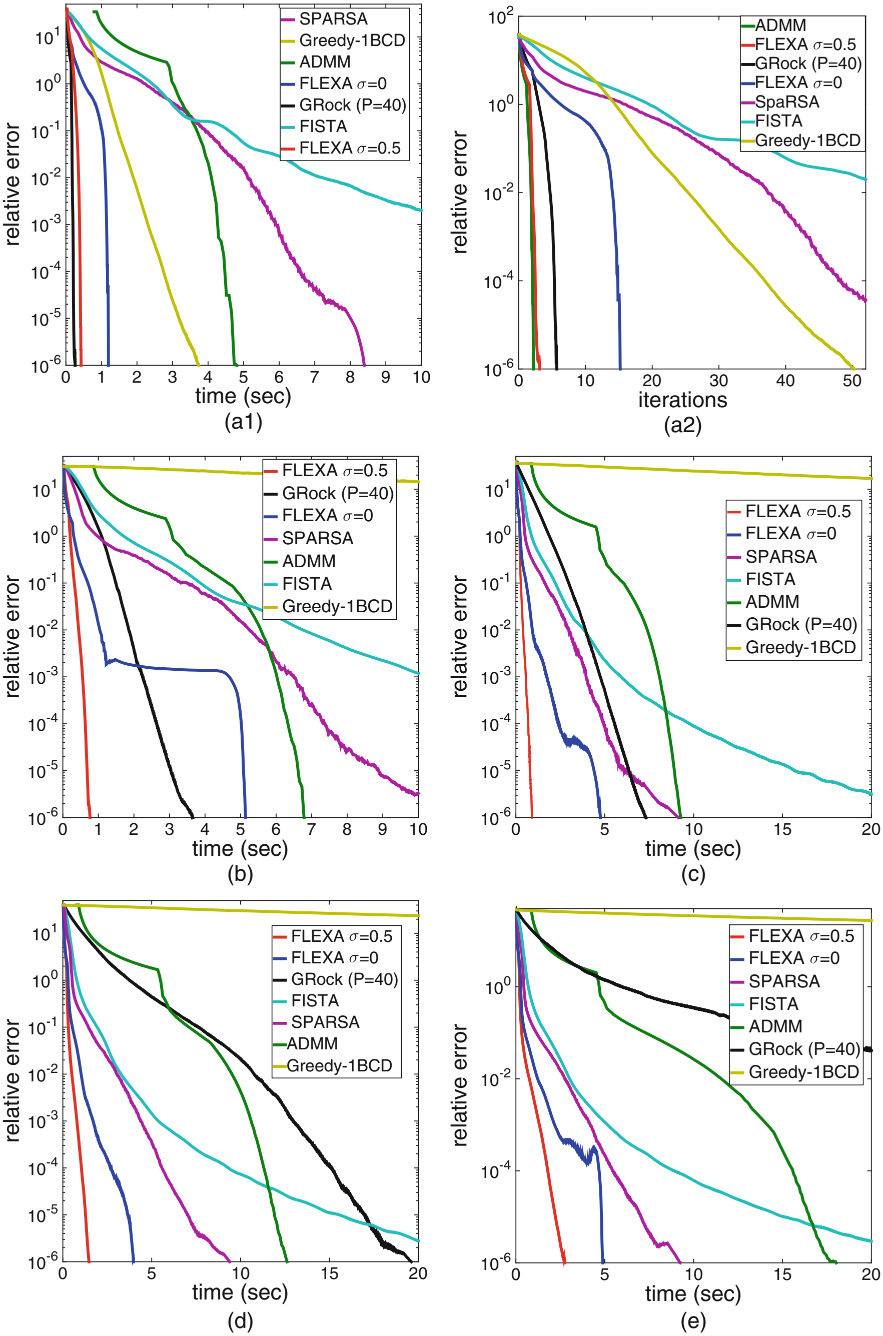


Fig. 3.10 LASSO problem (3.154) with 10,000 variables; relative error vs. time (in seconds) for: (**a1**) 1% non zeros, (**b**) 10% non zeros, (**c**) 20% non zeros, (**d**) 30% non zeros, (**e**) 40% non zeros; (**a2**) relative error vs. iterations for 1% non zeros. The figures are taken from [79]

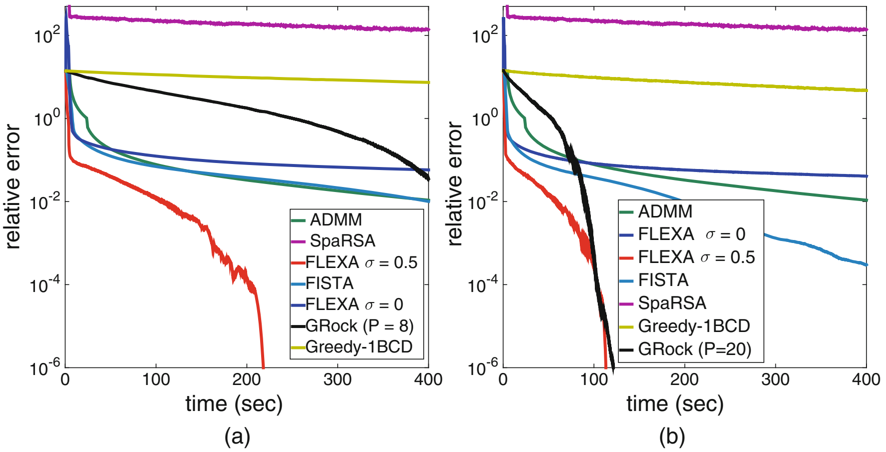


Fig. 3.11 LASSO problem (3.154) with 10^5 variables; Relative error vs. time for: (**a**) 8 cores, (**b**) 20 cores. The figures are taken from [79]

in GRock the maximum number of variables that is updated in parallel is exactly equal to the number of cores (i.e., the degree of parallelism), and this might become a serious drawback on very large problems (on top of the fact that convergence is in jeopardy). On the contrary, the theory presented in this chapter permits the parallel update of *any* number of variables while guaranteeing convergence.

Remark 3.22 (On Selective Updates) It is interesting to comment why FLEXA $\sigma = 0.5$ behaves better than FLEXA $\sigma = 0$. To understand the reason behind this phenomenon, we first note that Algorithm 6 has the remarkable capability to *identify those variables that will be zero at a solution*; we do not provide here the proof of this statement but only an informal argument. Roughly speaking, it can be shown that, for k large enough, those variables that are zero in $\widehat{\mathbf{x}}(\mathbf{x}^k) = (\widehat{x}_i(\mathbf{x}^k))_{i=1}^m$ will be zero also in a limiting solution $\bar{\mathbf{x}}$. Therefore, suppose that k is large enough so that this identification property already takes place (we will say that "we are in the identification phase") and consider an index i such that $\bar{x}_i = 0$. Then, if x_i^k is zero, it is clear, by Steps 3 and 4, that $x_i^{k'}$ will be zero for all indices $k' > k$, independently of whether i belongs to S^k or not. In other words, if a variable that is zero at the solution is already zero when the algorithm enters the identification phase, *that variable will be zero in all subsequent iterations*; this fact, intuitively, should enhance the convergence speed of the algorithm. Conversely, if when we enter the identification phase x_i^k is not zero, the algorithm will have to bring it back to zero iteratively. This explains why updating only variables that we have "strong" reason to believe will be non zero at a solution is a better strategy than updating them all. Of course, there may be a problem dependence and the best value of σ can vary from problem to problem. But the above explanation supports the idea that it might be wise to "waste" some calculations and perform only a partial ad-hoc update of the variables.

3.3.5.3 The Logistic Regression Problem

Consider the logistic regression problem in the following form [235]:

$$\underset{\mathbf{x}}{\text{minimize}} \quad V(\mathbf{x}) = \sum_{i=1}^{q} \log(1 + e^{-w_i \cdot \mathbf{z}_i^T \mathbf{x}}) + \lambda \|\mathbf{x}\|_1, \tag{3.158}$$

where $\mathbf{z}_i \in \mathbb{R}^m$ is the feature vector of sample i, with the associated label $w_i \in \{-1, 1\}$.

FLEXA for Logistic Regression Problem (3.158) is a highly nonlinear problem involving many exponentials that, notoriously, gives rise to numerical difficulties. Because of these high nonlinearities, a Gauss-Seidel (sequential) approach is expected to be more effective than a pure Jacobi (parallel) method, a fact that was confirmed by the experiments in [79]. For this reason, for the logistic regression problem we tested both `FLEXA` and the hybrid scheme in Algorithm 9, which will term `GJ-FLEXA`. The setting of the free parameters in GJ-FLEXA is essentially the same as the one described for LASSO (cf. Sect. 3.3.5.2), but with the following differences:

- *Exact solution* $\widehat{x}_i(\mathbf{x}^k)$: The surrogate function $\widetilde{F}_i$ is chosen as the second order approximation of the original function F: given the current iterate $\mathbf{x}^k$,

  $$\widetilde{F}_i(x_i \mid \mathbf{x}^k) = F(\mathbf{x}^k) + \nabla_{x_i} F(\mathbf{x}^k) \cdot (x_i - x_i^k) + \frac{1}{2}(\nabla^2_{x_i x_i} F(\mathbf{x}^k) + \tau_i) \cdot (x_i - x_i^k)^2 + \lambda \cdot |x_i|,$$

  which leads to the following closed form solution for $\widehat{x}_i(\mathbf{x}^k)$:

  $$\widehat{x}_i(\mathbf{x}^k) = S_{\lambda \cdot t_i^k}\left(x_i^k - t_i^k \cdot \nabla_{x_i} F(\mathbf{x}^k)\right), \quad \text{with} \quad t_i^k \triangleq \left(\tau_i + \nabla^2_{x_i x_i} F(\mathbf{x}^k)\right)^{-1},$$

  where $S_\lambda(\bullet)$ is the soft-thresholding operator, defined in (3.54).
- *Proximal weights* τ_i: The initial τ_i are set to $\mathrm{tr}(\mathbf{Z}^T \mathbf{Z})/(2\,m)$, for all i, where m is the total number of variables and $\mathbf{Z} = [\mathbf{z}_1\, \mathbf{z}_2\, \cdots\, \mathbf{z}_q]^T$.
- *Step-size* γ^k: We use the step-size rule (3.157). However, since the optimal value V^* is not known for the logistic regression problem, $\mathtt{re}(\mathbf{x})$ can no longer be computed. We replace $\mathtt{re}(\mathbf{x})$ with the merit function $\|\mathbf{M}(\mathbf{x})\|_\infty$, with

  $$\mathbf{M}(\mathbf{x}) \triangleq \nabla F(\mathbf{x}) - \Pi_{[-\lambda,\lambda]^m}\left(\nabla F(\mathbf{x}) - \mathbf{x}\right).$$

  Here the projection $\Pi_{[-\lambda,\lambda]^m}(\mathbf{y})$ can be efficiently computed; it acts component-wise on $\mathbf{y}$, since $[-\lambda, \lambda]^m = [-\lambda, \lambda] \times \cdots \times [-\lambda, \lambda]$. Note that $\mathbf{M}(\mathbf{x})$ is a valid optimality measure function; indeed, it is continuous and $\mathbf{M}(\mathbf{x}) = \mathbf{0}$ is equivalent to the standard necessary optimality condition for Problem (3.94), see [31].

Table 3.5 Data sets for the logistic regression tests [Problem (3.158)]

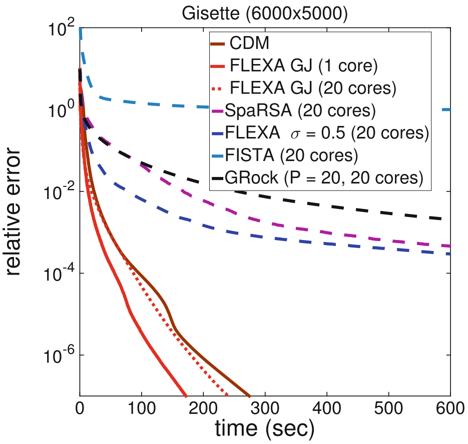
Data set	q	m	λ
gisette (scaled)	6000	5000	0.25
real-sim	72309	20958	4
rcv1	677399	47236	4

Algorithms in the Literature We compared FLEXA ($\sigma = 0.5$) (cf. Sect. 3.3.5.2) and GJ-FLEXA with the other parallel algorithms introduced in Sect. 3.3.5.2 for the LASSO problem (whose tuning of the free parameters is the same as in Figs. 3.10 and 3.11), namely: FISTA, SpaRSA, and GRock. For the logistic regression problem, we also tested one more algorithm, that we call CDM. This Coordinate Descent Method is an extremely efficient Gauss-Seidel-type method (customized for logistic regression), and is part of the LIBLINEAR package available at http://www.csie.ntu.edu.tw/~cjlin/.

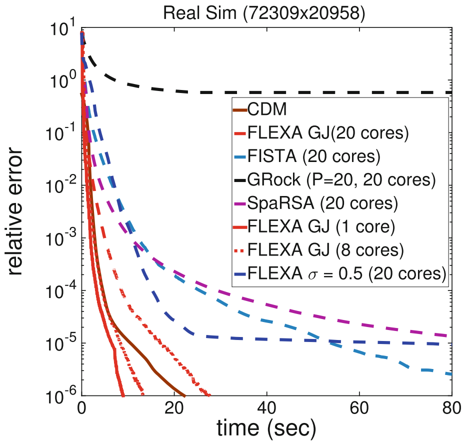
We tested the aforementioned algorithms on three instances of the logistic regression problem that are widely used in the literature, and whose essential data features are given in Table 3.5; we downloaded the data from the LIBSVM repository http://www.csie.ntu.edu.tw/~cjlin/libsvm/, which we refer to for a detailed description of the test problems. In our implementation, the matrix $\mathbf{Z}$ is column-wise partitioned according to $\mathbf{Z} = \left[\tilde{\mathbf{Z}}_1 \, \tilde{\mathbf{Z}}_2 \, \cdots \, \tilde{\mathbf{Z}}_P\right]$ and distributively stored across P processors, where $\tilde{\mathbf{Z}}_i$ is the set of columns of $\mathbf{Z}$ owned by processor i.

In Fig. 3.12, we plotted the relative error vs. the CPU time (the latter defined as in Figs. 3.10 and 3.11) achieved by the aforementioned algorithms for the three datasets, and using a different number of cores, namely: 8, 16, 20, 40; for each algorithm but GJ-FLEXA we report only the best performance over the aforementioned numbers of cores. Note that in order to plot the relative error, we had to preliminary estimate V^* (which is not known for logistic regression problems). To do so, we ran GJ-FLEXA until the merit function value $\|\mathbf{M}(\mathbf{x}^k)\|_\infty$ went below 10^{-7}, and used the corresponding value of the objective function as estimate of V^*. We remark that we used this value only to plot the curves. Next to each plot, we also reported the overall FLOPS counted up till reaching the relative errors as indicated in the table. Note that the FLOPS of GRock on real-sim and rcv1 are those counted in 24 h simulation time; when terminated, the algorithm achieved a relative error that was still very far from the reference values set in our experiment. Specifically, GRock reached 1.16 (instead of $1e-4$) on real-sim and 0.58 (instead of $1e-3$) on rcv1; the counted FLOPS up till those error values are still reported in the tables.

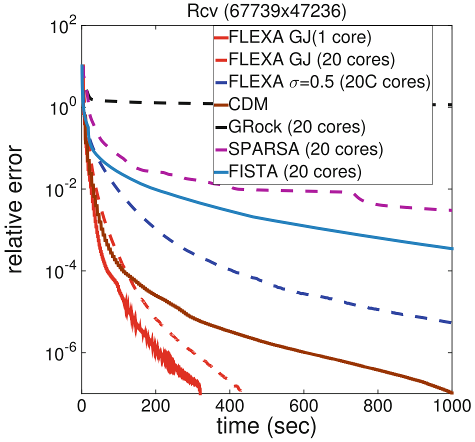
The analysis of the figures shows that, due to the high nonlinearities of the objective function, Gauss-Seidel-type methods outperform the other schemes. In spite of this, FLEXA still behaves quite well. But GJ-FLEXA with one core, thus a non parallel method, clearly outperforms all other algorithms. The explanation can be the following. GJ-FLEXA with one core is essentially a Gauss-Seidel-type method but with two key differences: the use of a stepsize and more importantly a (greedy) selection rule by which only some variables are updated at each round. As the number of cores increases, the algorithm gets "closer and closer" to a Jacobi-type



Algorithms	FLOPS (1e-2/1e-6)
GJ-FLEXA (1C)	1.30e+10/1.23e+11
GJ-FLEXA (20C)	5.18e+11/5.07e+12
FLEXA $\sigma = 0.5$ (20C)	1.88e+12/4.06e+13
CDM	2.15e+10/1.68e+11
SpaRSA (20C)	2.20e+12/5.37e+13
FISTA (20C)	3.99e+12/5.66e+13
GRock (20C)	7.18e+12/1.81e+14



Algorithms	FLOPS (1e-4/1e-6)
GJ-FLEXA (1C)	2.76e+9/6.60e+9
GJ-FLEXA (20C)	9.83e+10/2.85e+11
FLEXA $\sigma = 0.5$ (20C)	3.54e+10/4.69e+11
CDM	4.43e+9/2.18e+10
SpaRSA (20C)	7.18e+9/1.94e+11
FISTA (20C)	3.91e+10/1.56e+11
GRock (20C)	8.30e+14 (after 24h)



Algorithms	FLOPS (1e-3/1e-6)
GJ-FLEXA (1C)	3.61e+10/2.43e+11
GJ-FLEXA (20C)	1.35e+12/6.22e+12
FLEXA $\sigma = 0.5$ (20C)	8.53e+11/7.19e+12
CDM	5.60e+10/6.00e+11
SpaRSA (20C)	9.38e+12/7.20e+13
FISTA (20C)	2.58e+12/2.76e+13
GRock (20C)	1.72e+15 (after 24h)

Fig. 3.12 Logistic Regression problem (3.158): relative error vs. time (in seconds) and FLOPS for (i) gisette, (ii) real-sim, and (iii) rcv. The figures are taken from [79]

method, and because of the high nonlinearities, moving along a “Jacobi direction” does not bring improvements. In conclusion, for logistic regression problems, our experiments suggests that while the (opportunistic) selection of variables to update seems useful and brings to improvements even in comparison to the extremely efficient, dedicated `CDM` algorithm/software, parallelism (at least, in the form embedded in our scheme), does not appear to be beneficial as instead observed for LASSO problems.

3.3.6 Appendix

3.3.6.1 Proof of Lemma 3.4

The continuity of $\widehat{\mathbf{x}}(\bullet)$ follows readily from [200]; see also [106].

We prove next the Lipschitz continuity of $\widehat{\mathbf{x}}(\bullet)$, under the additional assumption that G is separable. Let $\mathbf{x}_i, \mathbf{z}_i \in X_i$. Invoking the optimality conditions of $\widehat{\mathbf{x}}_i(\mathbf{x})$ and $\widehat{\mathbf{x}}_i(\mathbf{z})$, we have

$$(\mathbf{y}_1 - \widehat{\mathbf{x}}_i(\mathbf{x}))^T(\nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{x})\,|\,\mathbf{x})) + g_i(\mathbf{y}_1) - g_i(\widehat{\mathbf{x}}_i(\mathbf{x})) \geq 0, \quad \forall \mathbf{y}_1 \in X_i,$$
$$(\mathbf{y}_2 - \widehat{\mathbf{x}}_i(\mathbf{z}))^T(\nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{z})\,|\,\mathbf{z})) + g_i(\mathbf{y}_2) - g_i(\widehat{\mathbf{x}}_i(\mathbf{z})) \geq 0, \quad \forall \mathbf{y}_2 \in X_i.$$

Letting $\mathbf{y}_1 = \widehat{\mathbf{x}}_i(\mathbf{z})$ and $\mathbf{y}_2 = \widehat{\mathbf{x}}_i(\mathbf{x})$ and summing the two inequalities above, we obtain

$$(\widehat{\mathbf{x}}_i(\mathbf{z}) - \widehat{\mathbf{x}}_i(\mathbf{x}))^T \left(\nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{x})\,|\,\mathbf{x}) - \nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{z})\,|\,\mathbf{z})\right) \geq 0.$$

Adding and subtracting $\nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{z})\,|\,\mathbf{x})$ and using the uniform strongly convexity of $\widetilde{F}_i$ with respect to its first argument (cf. Assumption 3.2.1) and the Lipschitz continuity of $\nabla\widetilde{F}_i$ with respect to its second argument (cf. Assumption 3.3) yield

$$\begin{aligned}\tau_i\,\|\widehat{\mathbf{x}}_i(\mathbf{z}) - \widehat{\mathbf{x}}_i(\mathbf{x})\|^2 &\leq (\widehat{\mathbf{x}}_i(\mathbf{z}) - \widehat{\mathbf{x}}_i(\mathbf{x}))^T(\nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{z})\,|\,\mathbf{x}) - \nabla\widetilde{F}_i(\widehat{\mathbf{x}}_i(\mathbf{z})\,|\,\mathbf{z}))\\ &\leq \widetilde{L}_i\,\|\widehat{\mathbf{x}}_i(\mathbf{z}) - \widehat{\mathbf{x}}_i(\mathbf{x})\| \cdot \|\mathbf{x} - \mathbf{z}\|.\end{aligned}$$

Therefore, $\widehat{\mathbf{x}}_i(\bullet)$ is Lipschitz continuous on X with constant $\hat{L}_i \triangleq \widetilde{L}_i/\tau_i$. □

3.3.6.2 Proof of Lemma 3.17

The proof is adapted by [56, Lemma 10] and reported here for completeness.

With a slight abuse of notation, we will use $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_{-(i,j)})$, with $i < j$, to denote the ordered tuple $(\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}, \mathbf{x}_i, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_{j-1}, \mathbf{x}_j, \mathbf{y}_{j+1}, \ldots, \mathbf{y}_n)$.

Given $k \geq 0$, $S^k \subseteq N$, and $\gamma^k \leq 1/n$, let $\bar{\gamma}^k = \gamma^k n \leq 1$. Define $\check{\mathbf{x}}^k \triangleq (\check{\mathbf{x}}_i^k)_{i \in N}$, with $\check{\mathbf{x}}_i^k = \mathbf{x}_i^k$ if $i \notin S^k$, and

$$\check{\mathbf{x}}_i^k \triangleq \bar{\gamma}^k \, \widehat{\mathbf{z}}_i^k + (1 - \bar{\gamma}^k) \, \mathbf{x}_i^k, \tag{3.159}$$

otherwise. Then $\mathbf{x}^{k+1}$ in Step 4 of the algorithm can be written as

$$\mathbf{x}^{k+1} = \frac{n-1}{n} \, \mathbf{x}^k + \frac{1}{n} \, \check{\mathbf{x}}^k. \tag{3.160}$$

Using (3.160) and invoking the convexity of G, the following recursion holds for all k:

$$\begin{aligned}
G(\mathbf{x}^{k+1}) &= G\left(\frac{1}{n}(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k) + \frac{1}{n}(\mathbf{x}_1^k, \check{\mathbf{x}}_{-1}^k) + \frac{n-2}{n}\mathbf{x}^k\right) \\
&= G\left(\frac{1}{n}\,(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k) + \frac{n-1}{n}\,\left(\mathbf{x}_1^k, \frac{1}{n-1}\,\check{\mathbf{x}}_{-1}^k + \frac{n-2}{n-1}\,\mathbf{x}_{-1}^k\right)\right) \\
&\leq \frac{1}{n}\,G\left(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k\right) + \frac{n-1}{n}\,G\left(\mathbf{x}_1^k, \frac{1}{n-1}\,\check{\mathbf{x}}_{-1}^k + \frac{n-2}{n-1}\,\mathbf{x}_{-1}^k\right) \\
&= \frac{1}{n}\,G\left(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k\right) + \frac{n-1}{n}\,G\left(\frac{1}{n-1}\left(\mathbf{x}_1^k, \check{\mathbf{x}}_{-1}^k\right) + \frac{n-2}{n-1}\mathbf{x}^k\right) \\
&= \frac{1}{n}G\left(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k\right) + \frac{n-1}{n}\,G\left(\frac{1}{n-1}\,\left(\check{\mathbf{x}}_2^k, \mathbf{x}_{-2}^k\right)\right. \\
&\quad \left. + \frac{1}{n-1}\left(\mathbf{x}_1^k, \mathbf{x}_2^k, \check{\mathbf{x}}_{-(1,2)}^k\right) + \frac{n-3}{n-1}\,\mathbf{x}^k\right) \qquad (3.161) \\
&= \frac{1}{n}G\left(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k\right) + \frac{n-1}{n}\,G\left(\frac{1}{n-1}\,\left(\check{\mathbf{x}}_2^k, \mathbf{x}_{-2}^k\right)\right. \\
&\quad \left. + \frac{n-2}{n-1}\left(\mathbf{x}_1^k, \mathbf{x}_2^k, \frac{1}{n-2}\,\check{\mathbf{x}}_{-(1,2)}^k + \frac{n-3}{n-2}\,\mathbf{x}_{-(1,2)}^k\right)\right) \\
&\leq \frac{1}{n}G\left(\check{\mathbf{x}}_1^k, \mathbf{x}_{-1}^k\right) + \frac{1}{n}\,G\left(\check{\mathbf{x}}_2^k, \mathbf{x}_{-2}^k\right) \\
&\quad + \frac{n-2}{n}\,G\left(\mathbf{x}_1^k, \mathbf{x}_2^k, \frac{1}{n-2}\,\check{\mathbf{x}}_{-(1,2)}^k + \frac{n-3}{n-2}\,\mathbf{x}_{-(1,2)}^k\right) \\
&\leq \quad \cdots \quad \leq \frac{1}{n}\sum_{i \in N} G(\check{\mathbf{x}}_i^k, \mathbf{x}_{-i}^k).
\end{aligned}$$

Using (3.161), the difference of $G(\mathbf{x}^{k+1})$ and $G(\mathbf{x}^k)$ can be bounded as

$$\begin{aligned} G(\mathbf{x}^{k+1}) - G(\mathbf{x}^k) &\leq \frac{1}{n}\sum_{i\in N}\left(G(\check{\mathbf{x}}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k)\right) \\ &= \frac{1}{n}\sum_{i\in S^k}\left(G(\check{\mathbf{x}}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k)\right) \\ &\leq \frac{1}{n}\sum_{i\in S^k}\left(\bar{\gamma}^k G(\widehat{\mathbf{z}}_i^k, \mathbf{x}_{-i}^k) + (1-\bar{\gamma}^k)G(\mathbf{x}^k) - G(\mathbf{x}^k)\right) \\ &= \gamma^k \sum_{i\in S^k}\left(G(\widehat{\mathbf{z}}_i^k, \mathbf{x}_{-i}^k) - G(\mathbf{x}^k)\right). \end{aligned} \tag{3.162}$$

□

3.3.6.3 Proof of Lemma 3.19

The proof can be found in [79, Lemma 10], and reported here for completeness.

For notational simplicity, we will write $\mathbf{x}_{S^k}$ for $(\mathbf{x})_{S^k}$ [recall that $(\mathbf{x})_{S^k}$ denotes the vector whose block component i is equal to $\mathbf{x}_i$ if $i \in S^k$, and zero otherwise]. Let j_k be an index in S^k such that $E_{j_k}(\mathbf{x}^k) \geq \rho \max_i E_i(\mathbf{x}^k)$ (cf. Assumption 3.10.2). Then, by the error bound condition (3.110) it is easy to check that the following chain of inequalities holds:

$$\begin{aligned} \bar{s}_{j_k}\|\widehat{\mathbf{x}}_{S^k}(\mathbf{x}^k) - \mathbf{x}_{S^k}^k\| &\geq \bar{s}_{j_k}\|\widehat{\mathbf{x}}_{j_k}(\mathbf{x}^k) - \mathbf{x}_{j_k}^k\| \\ &\geq E_{j_k}(\mathbf{x}^k) \\ &\geq \rho \max_i E_i(\mathbf{x}^k) \\ &\geq \left(\rho \min_i \underline{s}_i\right)\left(\max_i\{\|\widehat{\mathbf{x}}_i(\mathbf{x}^k) - \mathbf{x}_i^k\|\}\right) \\ &\geq \left(\frac{\rho \min_i \underline{s}_i}{n}\right)\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|. \end{aligned}$$

Hence we have for any k,

$$\|\widehat{\mathbf{x}}_{S^k}(\mathbf{x}^k) - \mathbf{x}_{S^k}^k\| \geq \left(\frac{\rho \min_i \underline{s}_i}{n\bar{s}_{j_k}}\right)\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\| \geq \left(\frac{\rho \min_i \underline{s}_i}{n \max_j \bar{s}_j}\right)\|\widehat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|. \quad \square$$

3.3.7 Sources and Notes

Although parallel (deterministic and stochastic) block-methods have a long history (mainly for convex problems), recent years have witnessed a revival of such methods and their (probabilistic) analysis; 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, fast, and resilient algorithms. The literature is vast and a comprehensive overview of current methods goes beyond the scope of this commentary. Here we only focus on SCA-related methods and refer to [25, 218, 250] (and references therein) as entry point to other numerical optimization algorithms.

Parallel SCA-Related Methods The roots of parallel deterministic SCA schemes (wherein *all* the variables are updated simultaneously) can be traced back at least to the work of Cohen on the so-called auxiliary principle [51, 52] and its related developments, see e.g. [7, 29, 91, 164, 167, 174, 185, 187, 198, 210, 238, 251]. Roughly speaking, these works can be divided in two groups, namely: parallel solution methods for *convex* objective functions [7, 29, 51, 52, 167, 187, 198] and *nonconvex* ones [91, 164, 174, 185, 210, 238, 251]. All methods in the former group (and [91, 164, 174, 238, 251]) are (proximal) gradient schemes; they thus share the classical drawbacks of gradient-like schemes; moreover, by replacing the convex function F with its first order approximation, they do not take any advantage of any structure of F beyond mere differentiability. Exploiting some available structural properties of F, instead, has been shown to enhance (practical) convergence speed, see e.g. [210]. Comparing with the second group of works [91, 164, 174, 185, 210, 238, 251], the parallel SCA algorithmic framework introduced in this lecture improves on their convergence properties while adding great flexibility in the selection of the variables to update at each iteration. For instance, with the exception of [69, 145, 187, 205, 238], all the aforementioned works do not allow parallel updates of a *subset* of all variables, a feature that instead, fully explored as we do, can dramatically improve the convergence speed of the algorithm, as shown in Sect. 3.3.5. Moreover, with the exception of [185], they all require an Armijo-type line-search, whereas the scheme in [185] is based on diminishing step-size-rules, but its convergence properties are quite weak: not all the limit points of the sequence generated by this scheme are guaranteed to be stationary solutions of (3.94).

The SCA-based algorithmic framework introduced in this lecture builds on and generalizes the schemes proposed in [56, 79, 210]. More specifically, Algorithm 5 was proposed in [210] for smooth instances of Problem (3.94) (i.e., $G = 0$); convergence was established when constant (Assumption 3.11.1) or diminishing (Assumption 3.11.2) step-sizes are employed (special case of Theorem 3.8). In [79], this algorithm was extended to deal with nonsmooth *separable* functions G while incorporating inexact updates (Assumption 3.9.1) and the greedy selection rule in Assumption 3.10.2 (cf. Algorithm 6) as well as hybrid Jacobi/Gauss-Seidel updates (as described in Algorithm 8); convergence was established under Assumption 3.11.2 (diminishing step-size) (special case of Theorem 3.12). Finally, in [56], an instance of Algorithm 6 was proposed, to deal with *nonseparable* convex functions G, and using random block selection rules (Assumption 3.10.4) or hybrid random-greedy selection rules (Algorithm 7); convergence was established when a diminishing step-size is employed (special case of Theorem 3.13).

While [56, 79, 210] studied some instances of parallel SCA methods in isolation (and only for some block selection rules and step-size rules) the contribution of this lecture is to provide a broader and unified view and analysis of such methods.

SCA Methods for Nonconvex Constrained Optimization (Parallel) SCA methods have been recently extended to deal with nonconvex constraints; state-of-the-art developments can be found in [80, 81, 211] along with their applications to some representative problems in Signal Processing, Communications, and Machine Learning [212]. More specifically, consider the following generalization of Problem (3.94):

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & F(\mathbf{x}) + H(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in X, \\ & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, J, \end{aligned} \tag{3.163}$$

where $g_j(\mathbf{x}) \leq 0$, $j = 1, \ldots, J$, represent *nonconvex nonsmooth* constraints; and H is now a *nonsmooth*, possibly *nonconvex* function.

A natural extension of the SCA idea introduced in this lecture to the general class of nonconvex problems (3.163) is replacing all the nonconvex functions with suitably chosen convex surrogates, and solve instead the following convexified problems: given $\mathbf{x}^k$,

$$\begin{aligned} \widehat{\mathbf{x}}(\mathbf{x}^k) \triangleq \underset{\mathbf{x}}{\text{argmin}} \quad & \widetilde{F}(\mathbf{x} \,|\, \mathbf{x}^k) + \widetilde{H}(\mathbf{x} \,|\, \mathbf{x}^k) \\ \text{s.t.} \quad & \mathbf{x} \in X, \\ & \tilde{g}_j(\mathbf{x} \,|\, \mathbf{y}) \leq 0, \quad j = 1, \ldots, J; \end{aligned} \tag{3.164}$$

where $\widetilde{F}$, $\widetilde{H}$ and $\widetilde{g}_j$ are (strongly) convex surrogates for F, H, and g_j, respectively. The update of $\mathbf{x}^k$ is then given by

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

Conditions on the surrogates in (3.164) and convergence of the resulting SCA algorithms can be found in [211] for the case of smooth H and g_j (or nonsmooth DC), and in [81] for the more general setting of nonsmooth functions; *parallel* and *distributed* implementations are also discussed. Here we only mention that the surrogates $\widetilde{g}_j$ must be a global convex upper bound of the associated nonconvex g_j (as for the MM algorithms—see Lecture I). This condition was removed in [80] where a different structure for the subproblems (3.164) was proposed. The work [80] also provides a complexity analysis of the SCA-based algorithms.

Other SCA-related methods for nonconvex constrained problem are discussed in [80, 81, 211, 212], which we refer to for details.

Asynchronous SCA Methods In the era of data deluge, data-intensive applications give rise to extremely large-scale problems, which naturally call for *asynchronous*, parallel solution methods. In fact, well suited to modern computational architectures (e.g., shared memory systems, message passing-based systems, cluster computers, cloud federations), asynchronous methods reduce the idle times of workers, mitigate communication and/or memory-access congestion, and make algorithms more fault-tolerant. Although asynchronous block-methods have a long history (see, e.g., [5, 16, 45, 89, 237]), in the past few years, the study of asynchronous parallel optimization methods has witnessed a revival of interest. Indeed, asynchronous parallelism has been applied to many state-of-the-art optimization algorithms (mainly for convex objective functions and constraints), including stochastic gradient methods [111, 139, 147, 155, 171, 186, 196] and ADMM-like schemes [107, 112, 247]. The asynchronous counterpart of BCD methods has been introduced and studied in the seminal work [149], which motivated and oriented much of subsequent research in the field, see e.g. [60, 61, 150, 188, 189].

Asynchronous parallel SCA methods were recently proposed and analyzed in [37, 38] for nonconvex problems in the form (3.94), with G separable and X_i possibly nonconvex [we refer to such a class of problems as Problem (3.94)$'$]. In the asynchronous parallel SCA method [37, 38], workers (e.g., cores, cpus, or machines) continuously and without coordination with each other, update a block-variable by solving a strongly convex block-model of Problem (3.94)$'$. 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 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 $\mathbf{x}^{k-\mathbf{d}^k} \triangleq (x_1^{k-d_1^k}, x_2^{k-d_2^k}, \ldots, x_n^{k-d_n^k})$, where $\mathbf{d}^k \triangleq (d_1^k, d_2^k, \ldots, d_n^k)$ 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; see [37]. 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 (3.94):

$$\hat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k}) \triangleq \underset{\mathbf{x}_{i^k} \in \tilde{X}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k})}{\operatorname{argmin}} \quad \tilde{F}_{i^k}(\mathbf{x}_{i^k} \mid \mathbf{x}^{k-\mathbf{d}^k}) + g_{i^k}(\mathbf{x}_{i^k}), \tag{3.165}$$

and then setting

$$\mathbf{x}_{i^k}^{k+1} = \mathbf{x}_{i^k}^k + \gamma \left(\hat{\mathbf{x}}_{i^k}(\mathbf{x}^{k-\mathbf{d}^k}) - \mathbf{x}_{i^k}^k \right). \tag{3.166}$$

In (3.165), $\tilde{F}_{i^k}(\bullet \mid \mathbf{y})$ represents a strongly convex surrogate of F, and $\tilde{X}_{i^k}$ is a convex set obtained replacing the nonconvex functions defining X_{i^k} by suitably chosen upper convex approximations, respectively; both $\tilde{F}_{i^k}$ and $\tilde{X}_{i^k}$ are built using the out-of-sync information $\mathbf{x}^{k-\mathbf{d}^k}$. If the set X_{i^k} is convex, then $\tilde{X}_{i^k} = X_{i^k}$. More details on the choices of $\tilde{F}_{i^k}$ and $\tilde{X}_{i^k}$ can be found in [37].

Almost all modern asynchronous algorithms for convex and nonconvex problems are modeled in a probabilistic way. 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; see [37] for a detailed discussion on this issue and several practical examples. Another unrealistic assumption often made in the literature [60, 149, 150, 196] 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, e.g., Examples 4 and 5 in [37]). In a nutshell, this assumption may be satisfied only if all workers have the same computational power and have access to all variables. In [37] a more general, and sophisticated probabilistic model describing the statistics of $(i^k; \mathbf{d}^k)$ was introduced, and convergence of the asynchronous parallel SCA method (3.165)–(3.166) established; theoretical complexity results were also provided, showing nearly ideal linear speedup when the number of workers is not too large. The new model in [37] neither postulates the independence between i^k and $\mathbf{d}^k$ nor requires artificial changes in the algorithm to enforce it (like those recently proposed in the probabilistic models [139, 155, 186] used in stochastic gradient methods); it handles instead the potential dependency among variables directly, fixing thus the theoretical issues that mar most of the aforementioned papers. It also lets one analyze for the first time in a sound way several practically used and effective computing settings and new models of asynchrony. 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; [37] is the first work proving convergence of such practically effective methods in an asynchronous setting.

Another important feature of the asynchronous algorithm (3.165)–(3.166) is its SCA nature, that is, the ability to handle nonconvex objective functions and nonconvex constraints by solving, 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 [147, 265] and [60, 61] that studied unconstrained and constrained nonconvex optimization problems, respectively. However, [60, 61] proposed 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. On the other hand, the SCA method [37] needs a feasible initial point and the ability to build approximations $\tilde{X}_i$ satisfying some technical conditions, as given in [37, Assumption D]. The two approaches thus complement each other and may cover different applications.

3.4 Distributed Successive Convex Approximation Methods

This lecture complements the first two, extending the SCA algorithmic framework developed therein to distributed (nonconvex, multi-agent) optimization over networks with *arbitrary*, possibly time-varying, topology.

The SCA methods introduced in Lecture II (cf. Sect. 3.3) unlock parallel updates from the workers; however, to perform its update, each worker must have the knowledge of some global information on the optimization problem, such as (part of) the objective function V, its gradient, and the current value of the optimization variable of the other agents. This clearly limits the applicability of these methods to network architectures wherein such information can be efficiently acquired (e.g., through suitably defined message-passing protocols and node coordination). Examples of such systems include the so-called multi-layer *hierarchical networks* (HNet); see Fig. 3.13. A *HNet* consists of distributed nodes (DNs), cluster heads (CHs) and a master node, each having some local information on the optimization problem. Each CH can communicate with a (possibly dynamically formed) cluster of DNs as well as a higher layer CH, through either deterministic or randomly activated links. The HNet arises in many important applications including sensor networks, cloud-based software defined networks, and shared-memory systems. The HNet is also a generalization of the so-called "star network" (a two-layer HNet) that is commonly adopted in several parallel computing environments; see e.g. the Parameter Server [146] or the popular DiSCO [267] algorithm, just to name a few.

On the other hand, there are networks that lack of a hierarchical structure or "special" nodes; an example is the class of general *mesh networks* (MNet), which consists solely of DNs, and each of them is connected with a subset of neighbors, via possibly time-varying and directional communication links; see Fig. 3.13. When the directional links are present, the MNet is referred to as a *digraph*. The MNet has been very popular to model applications such as *ad-hoc* (telecommunication)

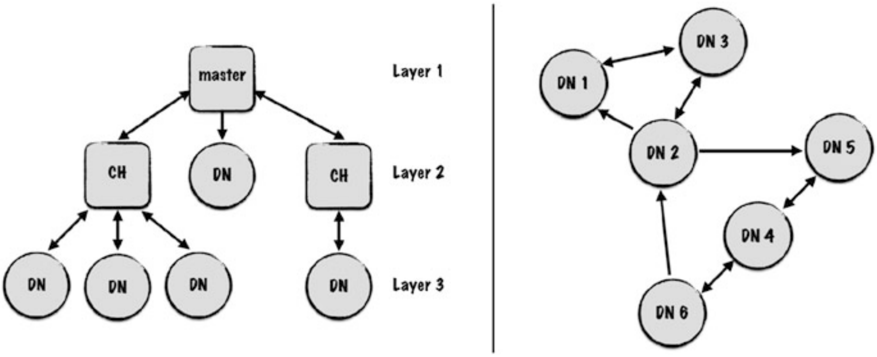


Fig. 3.13 Left: a three-layer **hierarchical network**, with one master node, 2 cluster heads, and 5 distributed nodes. Right: A six-node **mesh network**. The double arrowed (resp. single arrowed) links represent bi-directional (resp. directional) communication links

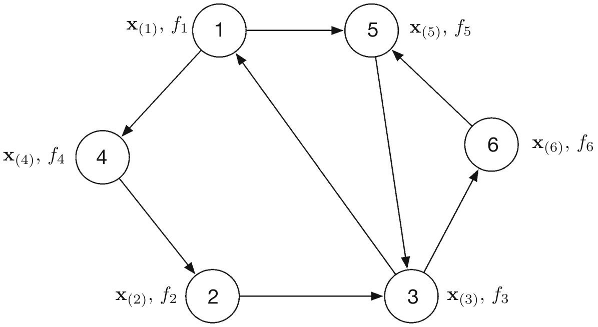


Fig. 3.14 An example of Problem (3.167) over a directed communication network. Each agent i knows only its own function f_i. To solve cooperatively (3.167), the agents create a local copy $\mathbf{x}_{(i)}$ of the common set of variables $\mathbf{x}$. These local copies are iteratively updated by the owners using only local (neighbor) information, so that asymptotically a consensus among them on a stationary solution of (3.167) is achieved

networks and social networks, where there are no obvious central controllers. Performing the SCA methods introduced in the previous lectures on such networks might incur in a computation/communication inefficient implementation.

The objective of this lecture is to devise distributed algorithms based on SCA techniques that are implementable efficiently on such general network architectures. More specifically, we consider a system of I DNs (we will use interchangeably also the words "workers" or "agents") that can communicate through a network, modeled as a directed graph, possibly time-varying; see Fig. 3.14. Agents want to cooperatively solve the following *networked* instance of Problem (3.94):

$$\underset{\mathbf{x}\in X}{\text{minimize}} \quad V(\mathbf{x}) \triangleq \underbrace{\sum_{i=1}^{I} f_i(\mathbf{x})}_{F(\mathbf{x})} + G(\mathbf{x}), \tag{3.167}$$

where the objective function F is now the sum of the local cost functions $f_i : O \to \mathbb{R}$ of the agents, assumed to be smooth but possibly nonconvex whereas $G : O \to \mathbb{R}$ is a nonsmooth convex function; $O \supseteq X$ is an open set and $X \subseteq \mathbb{R}^m$ is a convex, closed set. In this networked setting, each agent i knows only its own functions f_i (and G and X as well). The problem and network settings are described in more details in Sect. 3.4.1, along with some motivating applications.

The design of distributed algorithms for Problem (3.167) faces two challenges, namely: the nonconvexity of F and the lack of full knowledge of F from each agent. To cope with these issues, this lecture builds on the idea of SCA techniques coupled with suitably designed message passing protocols (compatible with the local agent knowledge of the network) aiming at disseminating information among the nodes as well as locally estimating ∇F from each agent. More specifically, for each agent

i, a *local copy* $\mathbf{x}_{(i)}$ of the global variable $\mathbf{x}$ is introduced (cf. Fig. 3.14). We say that a *consensus* is reached if $\mathbf{x}_{(i)} = \mathbf{x}_{(j)}$, for all $i \neq j$. To solve (3.167) over a network, *two major steps* are performed iteratively: local computation (to enhance the local solution quality), and local communication (to reach global consensus). In the first step, all the agents in parallel optimize their own variables $\mathbf{x}_{(i)}$ by solving a suitably chosen convex approximation of (3.167), built using the available local information.

In the second step, agents communicate with their neighbors to acquire some new information instrumental to align users' local copies (and thus enforce consensus asymptotically) and update the surrogate function used in their local optimization subproblems. These two steps will be detailed in the rest of the sections of this lecture, as briefly outlined next. Section 3.4.2 introduces distributed weighted-averaging algorithms to solve the (unconstrained) consensus problem over both static and time-varying (di-)graphs; a perturbed version of these consensus protocols is also introduced to unlock tracking of time-varying signals over networks. These message-passing protocols constitute the core of the distributed SCA-based algorithms that are discussed in this lecture: they will be used as an underlying mechanism for diffusing the information from one agent to every other agent in the network as well as track locally the gradient of the sum-utility F. In Sect. 3.4.3, we build the proposed distributed algorithmic framework combining SCA techniques with the consensus/tracking protocols introduced in Sect. 3.4.2, and study its convergence; a connection with existing (special case) schemes is also discussed. Some numerical results are presented in Sect. 3.4.4. Finally, the main literature on related works is discussed in Sect. 3.4.5 along with some extensions and open problems.

3.4.1 Problem Formulation

We study Problem (3.167) under the following assumptions.

Assumption 4.1 *Given Problem* (3.167)*, assume that*

1. $\emptyset \neq X \subseteq \mathbb{R}^m$ *is closed and convex;*
2. $f_i : O \to \mathbb{R}$ *is* C^1 *on the open set* $O \supseteq X$*, and* ∇f_i *is* L_i*-Lipschitz on* X*;*
3. $G : O \to \mathbb{R}$ *is convex, possibly nonsmooth;*
4. V *is bounded from below on* X.

Assumption 4.1 can be viewed as the distributed counterpart of Assumption 3.1 (Lecture II, cf. Sect. 3.3). Furthermore, we make the blanket assumption that each agent i knows only its local function f_i, the common regularizer G, and the feasible set X; therefore, agents must communicate over a network to solve (3.167). We consider the following network setup.

Network Model Agents are connected through a (communication) network, which is modeled as a graph; the set of agents are the nodes of the graph while the set of edges represents the communication links. We will consider both *static* and *time-varying* graphs, as well as *undirected* and *directed* graphs. We will use the following notation: $G^k = \left(V, E^k\right)$ denotes the directed graph that connects the agent at (the discrete) time k, where $V \triangleq \{1, \ldots, I\}$ is the set of nodes and E^k is the set of edges (agents' communication links); we use $(i, j) \in E^k$ to indicate that the link is directed from node i to node j. The *in*-neighborhood of agent i at time k (including node i itself) is defined as $N_i^{\text{in}, k} \triangleq \{j \mid (j, i) \in E^k\} \cup \{i\}$ whereas its *out*-neighborhood (including node i itself) is defined as $N_i^{\text{out}, k} \triangleq \{j \mid (i, j) \in E^k\} \cup \{i\}$. Of course, if the graph is undirected, the set of in-neighbors and out-neighbors are identical. These neighbors capture the local view of the network from agent i at time k: At the time the communication is performed, agent i can receive information from its current in-neighbors and send information to its current out neighbors. Note that we implicitly assumed that only inter-node (intermittent) communications between single-hop neighbors can be performed. The out-degree of agent i at time k is defined as the cardinality of $N_i^{\text{out}, k}$, and is denoted by $d_i^k \triangleq |N_i^{\text{out}, k}|$. We will treat static and/or undirected graphs as special cases of the above time-varying directed setting.

An important aspect of graphs is their *connectivity* properties. An *undirected* (static) graph is connected if there is a path connecting every pair of two distinct nodes. A *directed* (static) graph is strongly connected if there is a directed path from any node to any other node in the graph. For time-varying (di-)graphs we will invoke the following "long-term" connectivity property.

Assumption 4.2 (*B*-Strongly Connectivity) *The digraph sequence* $\{G^k\}_{k \in \mathbb{N}_+}$ *is* B*-strongly connected, i.e., there exists an integer* $B > 0$ *(possibly unknown to the agents) such that the digraph with edge set* $\cup_{t=k}^{k+B-1} E^t$ *is strongly connected, for all* $k \geq 0$.

Generally speaking, the above assumption permits strong connectivity to occur over a time window of length B: the graph obtained by taking the union of any B consecutive graphs is strongly connected. Intuitively, this lets information propagate throughout the network. Assumption 4.2 is standard and well-accepted in the literature.

3.4.1.1 Some Motivating Applications

Problems in the form (3.167), under Assumptions 4.1 and 4.2, have found a wide range of applications in several areas, including network information processing, telecommunications, multi-agent control, and machine learning. In particular, they are a key enabler of many nonconvex in-network "big data" analytic tasks, including nonlinear least squares, dictionary learning, principal/canonical component analysis, low-rank approximation, and matrix completion, just to name a few. Time-varying communications arise, for instance, in mobile wireless networks (e.g., ad-hoc networks) wherein nodes are mobile and/or communicate throughout fading channels. Moreover, since nodes generally transmit at different power and/or communication channels are not symmetric, directed links is a natural assumption. Some illustrative examples are briefly discussed next; see Sect. 3.4.4 for more details and some numerical results.

Example #1−(Sparse) Empirical Risk Minimization In Example #5 in Sect. 3.3.1.1 (Lecture II), we introduced the empirical risk minimization (ERM) problem, which consists in estimating a parameter $\mathbf{x}$ from a given data set $\{D_i\}_{i=1}^I$ by minimizing the risk function $F(\mathbf{x}) \triangleq \sum_{i=1}^I \ell\big(h(\mathbf{x}, D_i)\big)$. Consider now the scenario where the data set is not centrally available but split among I agents, connected through a network; agent i only owns the portion D_i. All the agents want to collaboratively estimate $\mathbf{x}$, still minimizing $F(\mathbf{x})$. This distributed counterpart of the ERM problem is an instance of (3.167), with $f_i(\mathbf{x}) \triangleq \ell\big(h(\mathbf{x}, D_i)\big)$. Many distributed statistical learning problems fall under this umbrella. Examples include: the least squares problem with $f_i(\mathbf{x}) \triangleq \|\mathbf{y}_i - \mathbf{A}_i\mathbf{x}\|^2$, where $D_i \triangleq (\mathbf{y}_i, \mathbf{A}_i)$; the sparse logistic regression with $f_i(\mathbf{x}) \triangleq \sum_{j=1}^{n_i} \log(1 + e^{-w_{ij}\mathbf{y}_{ij}^T\mathbf{x}})$, where $D_i \triangleq \{(w_{ij}, \mathbf{y}_{ij})\}_{j=1}^{n_i}$, and their sparse counterpart with suitable choices of the regularizer $G(\mathbf{x})$ (cf. Table 3.1 in Lecture I, Sect. 3.2.4.1).

Example #2−Sparse Principal Component Analysis Consider an m-dimensional data set $\{\mathbf{d}_i\}_{i=1}^n$ with zero mean stored distributively among I agents, each agent i owns $\{\mathbf{d}_j\}_{j\in N_i}$, where $\{N_i\}_{i=1}^I$ forms a partition of $\{1, \ldots, n\}$. The problem of sparse principal component analysis is to find a sparse direction $\mathbf{x}$ along which the variance of the data points, measured by $\sum_{i=1}^n \|\mathbf{d}_i^T\mathbf{x}\|^2$, is maximized. Construct the matrix $\mathbf{D}_i \in \mathbb{R}^{|N_i|\times m}$ by stacking $\{\mathbf{d}_j\}_{j\in N_i}$ row-wise, the problem can be formulated as an instance of (3.167) with $f_i(\mathbf{x}) \triangleq -\|\mathbf{D}_i\mathbf{x}\|^2$, $X \triangleq \{\mathbf{x} \mid \|\mathbf{x}\|_2 \le 1\}$, and $G(\mathbf{x})$ being some sparsity promoting regularizer (cf. Table 3.1 in Lecture I, Sect. 3.2.4.1).

Example #3−Target Localization Consider the problem of locating n targets using measurements from I sensors, embedded in a network. Each sensor i knows its own position $\mathbf{s}_i$ and $d_{i\,t}$, the latter representing the squared Euclidean distance between the target t and the node. Given the position $\mathbf{x}_t$ of each target t, an error measurement of agent i is $e_i(\mathbf{x}_t) \triangleq \sum_{t=1}^n p_{it}\,(d_{it} - \|\mathbf{x}_t - \mathbf{s}_i\|^2)^2$, where $p_{it} \in \{0, 1\}$ is a given binary variable taking value zero if the ith agent does not have any measurement related to target t. The problem of estimating the locations $\{\mathbf{x}_t\}_{t=1}^n$ can be thus formulated as an instance of (3.167), with $\mathbf{x} \triangleq \{\mathbf{x}_t\}_{t=1}^n$, $f_i(\mathbf{x}) \triangleq e_i(\mathbf{x}_t)$, and $X \triangleq \prod_{t=1}^n X_t$, where X_t characterizes the region where target t belongs to.

3.4.2 Preliminaries: Average Consensus and Tracking

In this section, we introduce some of the building blocks of the distributed algorithmic framework that will be presented in Sect. 3.4.3, namely: (i) a consensus algorithm implementable on undirected (Sect. 3.4.2.1) and directed (Sect. 3.4.2.2) time-varying graphs; (ii) a dynamic consensus protocol to track the average of time-varying signals over time-varying (directed) graphs (Sect. 3.4.2.3); and (iii) a perturbed consensus protocol unifying and generalizing the schemes in (i) and (ii) (Sect. 3.4.2.4).

3.4.2.1 Average Consensus Over Undirected Graphs

The consensus problem (also termed agreement problem) is one of the basic problems arising in decentralized coordination and control. Here we are interested in the so-called *average* consensus problem, as introduced next.

Consider a network of I agents, each of which having some initial (vector) variable $\mathbf{u}_i \in \mathbb{R}^m$. The agents are interconnected over a (time-varying) network; the graph modeling the network at time k is denoted by G^k (cf. Sect. 3.4.1). Each agent i controls a local variable $\mathbf{x}_{(i)}$ that is updated at each iteration k using the information of its immediate neighbors $N_i^{\text{in},\,k}$; we denote by $\mathbf{x}_{(i)}^k$ the value of $\mathbf{x}_{(i)}$ at iteration k. The average consensus problem consists in designing a distributed algorithm obeying the communication structure of each graph G^k, and enforcing

$$\lim_{k\to\infty} \|\mathbf{x}_{(i)}^k - \bar{\mathbf{u}}\| = 0, \quad \forall i = 1,\ldots,I, \quad \text{with} \quad \bar{\mathbf{u}} \triangleq \frac{1}{I}\sum_{i=1}^{I} \mathbf{u}_i.$$

One can construct a weighted-averaging protocol that solves the consensus problem as follows. Let each $\mathbf{x}_{(i)}^0 = \mathbf{u}_{(i)}^0$; given the iterate $\mathbf{x}_{(i)}^k$, at time $k+1$, each agent receives values $\mathbf{x}_{(j)}^k$ from its current (in-)neighbors, and updates its variable by setting

$$\mathbf{x}_{(i)}^{k+1} = \sum_{j\in N_i^{\text{in},k}} w_{ij}^k\, \mathbf{x}_{(j)}^k, \tag{3.168}$$

where w_{ij}^k are some positive weights, to be properly chosen. For a more compact representation, we define the nonnegative weight-matrix[1] $\mathbf{W}^k \triangleq (w_{ij}^k)_{i,j=1}^I$, whose nonzero pattern is compliant with the topology of the graph G^k (in the sense of Assumption 4.3 below). Recall that the set of neighbors $N_i^{\text{in},\,k}$ contains also agent i (cf. Sect. 3.4.1).

Assumption 4.3 *Given the graph sequence* $\{G^k\}_{k\in\mathbb{N}_+}$, *each matrix* $\mathbf{W}^k \triangleq (w_{ij})_{i,j=1}^I$ *satisfies:*

1. $w_{ij}^k = 0$, *if* $(j,i) \notin E^k$; *and* $w_{ij}^k \geq \kappa$, *if* $(j,i) \in E^k$;

[1] Note that, for notational simplicity, here we use reverse links for the weight assignment, that is, each weight w_{ij} is assigned to the directed edge $(j,i) \in E^k$.

2. $w_{ii}^k \geq \kappa$*, for all* $i = 1, \ldots, I$*;*

for some given $\kappa > 0$.

Using Assumption 4.3.1, we can write the consensus protocol (3.168) in the following equivalent form

$$\mathbf{x}_{(i)}^{k+1} = \sum_{j=1}^{I} w_{ij}^k \, \mathbf{x}_{(j)}^k. \tag{3.169}$$

Note that (3.169) is compliant with the graph topology: the agents can only exchange information (according to the direction of the edge) if there exists a communication link between them. Also, Assumption 4.3.2 states that each agent should include in the update (3.169) its own current information.

Convergence of $\{\mathbf{x}_{(i)}^k\}_{k\in\mathbb{N}_+}$ in (3.169) to the average $\bar{\mathbf{u}}$ calls for the following extra assumption.

Assumption 4.4 *Each* $\mathbf{W}^k$ *is doubly-stochastic, i.e.,* $\mathbf{1}^T\mathbf{W}^k = \mathbf{1}^T$ *and* $\mathbf{W}^k\mathbf{1} = \mathbf{1}$.

Assumption 4.4 requires **1** being both the left and right eigenvector of $\mathbf{W}^k$ associated to the eigenvalue 1; intuitively, the column stochasticity plays the role of preserving the total sum of the $\mathbf{x}_{(i)}$'s (and thus the average) in the network while the row stochasticity locks consensus.

When the graphs G^k are undirected (or are directed and admits a compliant doubly-stochastic matrix), several rules have been proposed in the literature to build a weight matrix satisfying Assumptions 4.3 and 4.4. Examples include the Laplacian weight rule [204]; the maximum degree weight, the Metropolis-Hastings, and the least-mean square consensus weight rules [258]. Table 3.6 summarizes the aforementioned rules [203], where in the Laplacian weight rule, λ is a positive

Table 3.6 Examples of rules for doubly-stochastic weight matrices compliant to an undirected graph $G = (V, E)$ (or a digraph admitting a double-stochastic matrix)

Rule name	Weight expression
Metropolis-Hastings	$w_{ij} = \begin{cases} \dfrac{1}{\max\{d_i, d_j\}}, & \text{if } (i,j) \in E, \\ 1 - \sum_{j\neq i} w_{ij}^k, & \text{if } i = j, \\ 0, & \text{if } (i,j) \notin E \text{ and } i \neq j; \end{cases}$
Laplacian	$\mathbf{W} = \mathbf{I} - \lambda\mathbf{L}, \lambda > 0$
Maximum-degree	$w_{ij} = \begin{cases} 1/I, & \text{if } (i,j) \in E, \\ 1 - (d_i - 1)/I, & \text{if } i = j, \\ 0, & \text{if } (i,j) \notin E \text{ and } i \neq j. \end{cases}$

In the Laplacian weight rule, λ is a positive constant and **L** is the Laplacian of the graph constant and $\mathbf{L}$ is the graph Laplacian, whose the ij-th entry L_{ij} is defined as

$$L_{ij} \triangleq \begin{cases} d_i - 1, & \text{if } i = j; \\ -1, & \text{if } (i, j) \in E \text{ and } i \neq j; \\ 0, & \text{otherwise}; \end{cases} \tag{3.170}$$

where d_i is the degree of node i.

Convergence of the average-consensus protocol (3.169) is stated in the next theorem, whose proof is omitted because it is a special case of that of Theorem 4.11 (cf. Sect. 3.4.2.4).

Theorem 4.5 *Let* $\{G^k\}_{k\in\mathbb{N}_+}$ *be a sequence of graphs satisfying Assumption 4.2. Consider the average-consensus protocol* (3.169)*, where each* $\{\mathbf{W}^k\}_{k\in\mathbb{N}_+}$ *is chosen according to Assumptions 4.3 and 4.4. Then, the sequence* $\{\mathbf{x}^k \triangleq (\mathbf{x}^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$ *generated by* (3.169) *satisfies: for all* $k \in \mathbb{N}_+$*,*

(a) `Invariance of the average:`

$$\sum_{i=1}^{I} \mathbf{x}^{k+1}_{(i)} = \sum_{i=1}^{I} \mathbf{x}^{k}_{(i)} = \cdots = \sum_{i=1}^{I} \mathbf{x}^{0}_{(i)}; \tag{3.171}$$

(b) `Geometric decay of the consensus error:`

$$\left\| \mathbf{x}^k_{(i)} - \frac{1}{I}\sum_{j=1}^{I} \mathbf{x}^k_{(j)} \right\| \leq c_u \cdot (\rho_u)^k \cdot \|\mathbf{x}^0\|, \quad \forall i = 1, \ldots, I, \tag{3.172}$$

where $\mathbf{x}^0 \triangleq (\mathbf{x}^0_{(i)})_{i=1}^I$*, and* $c_u > 0$ *and* $\rho_u \in (0, 1)$ *are constants defined as*

$$c_u \triangleq \frac{2I}{\rho_u} \cdot \frac{2(1 + \kappa^{-(I-1)B})}{1 - \kappa^{(I-1)B}} \quad \textit{and} \quad \rho_u \triangleq \left(1 - \kappa^{(I-1)B}\right)^{\frac{1}{(I-1)B}}, \tag{3.173}$$

with B *and* κ *defined in Assumption 4.2 and Assumption 4.3, respectively.*

In words, Theorem 4.5 states that each $\mathbf{x}^k_{(i)}$ converges to the initial average $(1/I)\cdot\sum_{i=1}^I \mathbf{x}^0_{(i)}$ at a geometric rate. Since $\mathbf{x}^0_{(i)}$ is initialized as $\mathbf{x}^0_{(i)} \triangleq \mathbf{u}_i$, each $\{\mathbf{x}^k_{(i)}\}_{k\in\mathbb{N}_+}$ converges to $\bar{\mathbf{u}}$ geometrically.

Remark 4.6 While Theorem 4.5 has been stated under Assumption 4.4 (because we are mainly interested in the convergence to the average $\bar{\mathbf{u}}$), it is important to remark that the row-stochasticity of each $\mathbf{W}^k$ (rather than doubly-stochasticity) is enough for the sequence $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ generated by the protocol (3.169) to geometrically reach a consensus, that is, $\lim_{k\to\infty} \|\mathbf{x}^k_{(i)} - \mathbf{x}^k_{(j)}\| = 0$, for all $i, j = 1, \ldots I$ and $i \neq j$. However, the limit point of $\{\mathbf{x}^k\}_{k\in\mathbb{N}_+}$ is no longer the average of the $\mathbf{x}^0_{(i)}$'s.

3.4.2.2 Average Consensus Over Directed Graphs

A key assumption for the distributed protocol (3.169) to reach the *average* consensus is that each matrix $\mathbf{W}^k$, compliant with the graph G^k, is doubly-stochastic.

While such constructions exist for networks with bi-directional (possibly time-varying) communication links, they become computationally prohibitive or infeasible for networks with directed links, for several reasons. First of all, not all digraphs admit a compliant (in the sense of Assumption 4.3) doubly-stochastic weight matrix; some form of balancedness in the graph is needed [94], which limits the class of networks over which the consensus protocol (3.169) can be applied. Furthermore, conditions for a digraph to admit such a doubly-stochastic matrix are not easy to be checked in practice; and, even when possible, constructing a doubly-stochastic weight matrix compliant to the digraph calls for computationally intense, generally centralized, algorithms.

To solve the average consensus problem over digraphs that do not admit a doubly-stochastic matrix, a further assumption is needed [103] along with a modification of the protocol (3.169). Specifically, a standard assumption in the literature is that every agent i knows its out-degree d_i^k at each time k. This means that, while broadcasting its own message, every agent knows how many nodes will receive it. The problem of computing the out-degree using only local information has been considered in a number of works (see, e.g., [121, 243] and the references therein). Various algorithms have been proposed, mainly based on flooding, which, however, requires significant communication overhead and storage. A less demanding consensus-based approach can be found in [44].

Under the above assumption, the average consensus can be achieved on digraphs using the so-called push-sum protocol [123]. Each agent i controls two local variables, $\mathbf{z}_{(i)} \in \mathbb{R}^m$ and $\phi_{(i)} \in \mathbb{R}$, which are updated at each iteration k still using only the information of its immediate neighbors. The push-sum protocols reads: for all $i = 1, \ldots, I$,

$$\begin{aligned} \mathbf{z}_{(i)}^{k+1} &= \sum_{j=1}^{I} a_{ij}^k \, \mathbf{z}_{(j)}^k, \\ \phi_{(i)}^{k+1} &= \sum_{j=1}^{I} a_{ij}^k \, \phi_{(j)}^k, \end{aligned} \tag{3.174}$$

where $\mathbf{z}_{(i)}$ and $\phi_{(i)}$ are initialized as $\mathbf{z}_{(i)}^0 = \mathbf{u}_i$ and $\phi_{(i)}^0 = 1$, respectively; and the coefficient a_{ij}^k are defined as

$$a_{ij}^k \triangleq \begin{cases} \dfrac{1}{d_j^k}, & \text{if } j \in \mathcal{N}_i^{\text{in}, k}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.175}$$

Note that the scheme (3.174) is a broadcast (i.e., one-way) communication protocol: each agent i broadcasts ("pushes out") the values $\mathbf{z}_{(j)}^k/d_j^k$ and $\phi_{(j)}^k/d_j^k$, which are received by its out-neighbors; at the receiver side, every node aggregates the received information according to (3.174) (i.e., summing the pushed values, which explains the name "push-sum").

Introducing the weight matrices $\mathbf{A}^k \triangleq (a_{ij}^k)_{i,j=1}^I$, it is easy to check that, for general digraphs, $\mathbf{A}^k$ may no longer be row-stochastic (i.e., $\mathbf{A}^k\mathbf{1} = \mathbf{1}$). This means that the z- and ϕ-updates in (3.174) do not reach a consensus. However, because $\mathbf{1}^T\,\mathbf{A}^k = \mathbf{1}^T$, the sums of the z- and ϕ-variables are preserved: at every iteration $k \in \mathbb{N}_+$,

$$\begin{aligned}\sum_{i=1}^I \mathbf{z}_{(i)}^{k+1} &= \sum_{i=1}^I \mathbf{z}_{(i)}^{k} = \cdots = \sum_{i=1}^I \mathbf{z}_{(i)}^{0} = \sum_{i=1}^I \mathbf{u}_i,\\ \sum_{i=1}^I \phi_{(i)}^{k+1} &= \sum_{i=1}^I \phi_{(i)}^{k} = \cdots = \sum_{i=1}^I \phi_{(i)}^{0} = I.\end{aligned} \tag{3.176}$$

This implies that, if the iterates $(\mathbf{z}_{(i)}^k/\phi_{(i)}^k)_{i=1}^I$ converge to a consensus (note that each $\mathbf{z}_{(i)}^k/\phi_{(i)}^k$ is well-defined because the weights ϕ_i^k are all positive), then the consensus value must be the average $\bar{\mathbf{u}}$, as shown next. Let $\mathbf{c}^\infty$ be the consensus value, that is, $\lim_{k\to\infty} \mathbf{z}_{(i)}^k/\phi_{(i)}^k = \mathbf{c}^\infty$, for all $i = 1, \ldots, I$. Then, it must be

$$\left\|\sum_{i=1}^I \mathbf{u}_i - I\cdot\mathbf{c}^\infty\right\| \overset{(3.176)}{=} \left\|\sum_{i=1}^I \left(\mathbf{z}_{(i)}^k - \phi_{(i)}^k\cdot\mathbf{c}^\infty\right)\right\| \le I\cdot\sum_{i=1}^I \left\|\mathbf{z}_{(i)}^k/\phi_{(i)}^k - \mathbf{c}^\infty\right\| \underset{k\to\infty}{\longrightarrow} 0, \tag{3.177}$$

which shows that $\mathbf{c}^\infty = (1/I)\cdot\sum_{i=1}^I \mathbf{u}_i$. Convergence of $(\mathbf{z}_{(i)}^k/\phi_{(i)}^k)_{i=1}^I$ to the consensus is proved in [11, 123], which we refer to the interested reader.

Here, we study instead an equivalent reformulation of the push-sum algorithm, as given in [208, 232], which is more suitable for the integration with optimization (cf. Sect. 3.4.3). Eliminating the z-variables in (3.174), and considering arbitrary *column-stochastic* weight matrices $\mathbf{A}^k \triangleq (a_{ij}^k)_{i,j=1}^I$, compliant to the graph G^k [not necessarily given by (3.175)], we have [208, 232]:

$$\begin{aligned}\phi_{(i)}^{k+1} &= \sum_{j=1}^I a_{ij}^k\,\phi_{(j)}^k,\\ \mathbf{x}_{(i)}^{k+1} &= \frac{1}{\phi_{(i)}^{k+1}}\sum_{j=1}^I a_{ij}^k\,\phi_{(j)}^k\,\mathbf{x}_{(j)}^k,\end{aligned} \tag{3.178}$$

where $\mathbf{x}_{(i)}^0$ is set to $\mathbf{x}_{(i)}^0 = \mathbf{u}_i/\phi_{(i)}^0$, for all $i = 1, \ldots, I$; and $\phi_{(i)}^0$ are arbitrary positive scalars such that $\sum_{i=1}^I \phi_{(i)}^0 = I$. For simplicity, hereafter, we tacitly set $\phi_{(i)}^0 = 1$ (implying $\mathbf{x}_{(i)}^0 = \mathbf{u}_i$), for all $i = 1, \ldots, I$. Similarly to (3.174), in the protocol (3.178), every agent i controls and updates the variables $\mathbf{x}_{(i)}$ and $\phi_{(i)}$, based on the information $\phi_{(j)}^k$ and $\phi_{(j)}^k\,\mathbf{x}_{(j)}^k$ received from its current in-neighbors. We will refer to (3.178) as *condensed push-sum* algorithm.

Combining the weights a_{ij}^k and the $\phi_{(i)}^k$ in the update of $\mathbf{x}_{(i)}^k$ in a single coefficient

$$w_{ij}^k \triangleq \frac{a_{ij}^k\,\phi_{(j)}^k}{\sum_{j=1}^I a_{ij}^k\,\phi_{(j)}^k}, \tag{3.179}$$

it is not difficult to check that the matrices $\mathbf{W}^k \triangleq (w_{ij}^k)_{i,j=1}^I$ are row-stochastic, that is, $\mathbf{W}^k\mathbf{1} = \mathbf{1}$, and compliant to G^k. This means that each $\mathbf{x}_{(i)}^{k+1}$ in (3.178) is updated performing a convex combination of the variables $(\mathbf{x}_{(j)}^k)_{j\in N_i^{\text{in},k}}$. This is a key property that will be leveraged in Sect. 3.4.3 to build a distributed optimization algorithm for constrained optimization problems wherein the feasibility of the iterates is preserved at each iteration. The above equivalent formulation also sheds light on the role of the ϕ-variables: they rebuild dynamically the missing row stochasticity of the weights a_{ij}^k, thus enforcing the consensus on the x-variables.

Since the following quantities are invariants of the dynamics (3.178) (recall that $\mathbf{A}^k$ are column stochastic), that is, for all $k \in \mathbb{N}_+$,

$$\sum_{i=1}^I \phi_{(i)}^{k+1}\,\mathbf{x}_{(i)}^{k+1} = \sum_{i=1}^I \phi_{(i)}^k\,\mathbf{x}_{(i)}^k = \cdots = \sum_{i=1}^I \phi_{(i)}^0\,\mathbf{x}_{(i)}^0 = \sum_{i=1}^I \mathbf{u}_i,$$

$$\sum_{i=1}^I \phi_{(i)}^{k+1} = \sum_{i=1}^I \phi_{(i)}^k = \cdots = \sum_{i=1}^I \phi_{(i)}^0 = I,$$

by a similar argument used in (3.177), one can show that, if the $\mathbf{x}_{(i)}^k$ are consensual—$\lim_{k\to\infty}\mathbf{x}_{(i)}^k = \mathbf{x}^\infty$, for all $i = 1, \ldots, I$—it must be $\mathbf{x}^\infty = (1/I)\cdot\sum_{i=1}^I \mathbf{u}_i$. Convergence to the consensus at geometric rate is stated in Theorem 4.8 below, under the following assumption on the weight matrices $\mathbf{A}^k$ (the proof of the theorem is omitted because it is a special case of that of Theorem 4.11, cf. Sect. 3.4.2.4).

Assumption 4.7 *Each $\mathbf{A}^k$ is compliant with G^k (i.e., it satisfies Assumption 4.3) and it is column stochastic, i.e., $\mathbf{1}^T\mathbf{A}^k = \mathbf{1}^T$.*

Theorem 4.8 *Let $\{G^k\}_{k\in\mathbb{N}_+}$ be a sequence of graphs satisfying Assumption 4.2. Consider the condensed push-sum protocol (3.178), where each $\{\mathbf{A}^k\}_{k\in\mathbb{N}}$ is chosen according to Assumption 4.7. Then, the sequence $\{\mathbf{x}^k \triangleq (\mathbf{x}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ generated* *by* (3.178) *satisfies: for all* $k \in \mathbb{N}_+$,

(a) `Invariance of the weighted-sum:`

$$\sum_{i=1}^{I} \phi_{(i)}^{k+1}\, \mathbf{x}_{(i)}^{k+1} = \sum_{i=1}^{I} \phi_{(i)}^{k}\, \mathbf{x}_{(i)}^{k} = \cdots = \sum_{i=1}^{I} \phi_{(i)}^{0}\, \mathbf{x}_{(i)}^{0}; \tag{3.180}$$

(b) `Geometric decay of the consensus error:`

$$\left\| \mathbf{x}_{(i)}^{k} - \frac{1}{I} \sum_{j=1}^{I} \phi_{(j)}^{k}\, \mathbf{x}_{(j)}^{k} \right\| \le c_d \cdot (\rho_d)^k \cdot \|\mathbf{x}^0\|, \quad \forall i = 1, \ldots, I, \tag{3.181}$$

where $\mathbf{x}^0 \triangleq (\mathbf{x}_{(i)}^0)_{i=1}^I$, *and* $c_d > 0$ *and* $\rho_d \in (0,1)$ *are defined as*

$$c_d \triangleq \frac{2I}{\rho} \cdot \frac{2(1+\tilde{\kappa}_d^{-(I-1)B})}{1-\kappa_d^{(I-1)B}} \quad \text{and} \quad \rho_d \triangleq \left(1 - \tilde{\kappa}_d^{(I-1)B}\right)^{\frac{1}{(I-1)B}},$$

and $\tilde{\kappa}_d \triangleq \kappa^{2(I-1)B+1}/I$, *with* B *and* κ *defined in Assumption 4.2 and Assumption 4.3, respectively.*

Note that to reach consensus, the condensed push-sum protocol requires that the weight matrices $\mathbf{A}^k$ are column stochastic but not doubly-stochastic. Of course, one can always use the classical push-sum weights as in (3.175), which is an example of rule satisfying Assumption 4.7. Moreover, if the graph G^k is undirected or is directed but admits a compliant doubly-stochastic matrix, one can also choose in (3.178) a doubly-stochastic $\mathbf{A}^k$. In such a case, the ϕ-update in (3.178) indicates that $\phi_{(i)}^k = 1$, for all $i = 1, \ldots, I$ and $k \in \mathbb{N}_+$. Therefore, (3.178) [and also (3.174)] reduces to the plain consensus scheme (3.169).

3.4.2.3 Distributed Tracking of Time-Varying Signals' Average

In this section, we extend the condensed push-sum protocol to the case where the signals $\mathbf{u}_i$ are no longer constant but time-varying; the value of the signal at time k owned by agent i is denoted by $\mathbf{u}_i^k$. The goal becomes designing a distributed algorithm obeying the communication structure of each graph G^k that tracks the average of $(\mathbf{u}_i^k)_{i=1}^I$, i.e.,

$$\lim_{k\to\infty} \|\mathbf{x}_{(i)}^k - \bar{\mathbf{u}}^k\| = 0, \quad \forall i = 1, \ldots, I, \quad \text{with} \quad \bar{\mathbf{u}}^k \triangleq \frac{1}{I}\sum_{i=1}^{I} \mathbf{u}_i^k.$$

We first introduce the algorithm for undirected graphs (or directed ones that admit a doubly-stochastic matrix); we then extend the scheme to the more general setting of arbitrary directed graphs.

Distributed Tracking Over Undirected Graphs

Consider a (possibly) time-varying network, modeled by a sequence of undirected graphs (or directed graphs admitting a doubly-stochastic matrix) $\{G^k\}_{k\in\mathbb{N}_+}$. As for the average consensus problem, we let each agent i maintain and update a variable $\mathbf{x}_{(i)}^k$ that represents a local estimate of $\bar{\mathbf{u}}^k$; we set $\mathbf{x}_{(i)}^0 \triangleq \mathbf{u}_i^0$. Since $\mathbf{u}_i^k$ is time-varying, a direct application of the protocol (3.169), developed for the plain average consensus problem, cannot work because it would drive all $\mathbf{x}_{(i)}^k$ to converge to $\bar{\mathbf{u}}^0$. Therefore, we need to modify the vanilla scheme (3.169) to account for the variability of $\mathbf{u}_i^k$'s. We construct next the distributed tracking algorithm inductively, building on (3.169).

Recall that, in the average consensus scheme (3.169), a key property to set the consensus value to the average of the initial values $\mathbf{x}_{(i)}^0$ is the invariance of the average $\sum_{i=1}^I \mathbf{x}_{(i)}^k$ throughout the dynamics (3.169):

$$\sum_{i=1}^I \mathbf{x}_{(i)}^{k+1} = \sum_{i=1}^I \sum_{j=1}^I w_{ij}^k \, \mathbf{x}_{(j)}^k = \sum_{i=1}^I \mathbf{x}_{(i)}^k = \cdots = \sum_{i=1}^I \mathbf{x}_{(i)}^0, \tag{3.182}$$

which is met if the weight matrices $\mathbf{W}^k$ are column stochastic. The row-stochasticity of $\mathbf{W}^k$ enforces asymptotically a consensus. When it comes to solve the tracking problem, it seems then natural to require such an invariance of the (time-varying) average, that is, $\sum_{i=1}^I \mathbf{x}_{(i)}^k = \sum_{i=1}^I \mathbf{u}_i^k$, for all $k \in \mathbb{N}_+$, while enforcing a consensus on $\mathbf{x}_{(i)}^k$'s. Since $\mathbf{x}_{(i)}^0 \triangleq \mathbf{u}_i^0$ we have $\sum_{i=1}^I \mathbf{x}_{(i)}^0 \triangleq \sum_{i=1}^I \mathbf{u}_i^0$. Suppose now that, at iteration k, we have $\sum_{i=1}^I \mathbf{x}_{(i)}^k = \sum_{i=1}^I \mathbf{u}_i^k$. In order to satisfy $\sum_{i=1}^I \mathbf{x}_{(i)}^{k+1} \triangleq \sum_{i=1}^I \mathbf{u}_i^{k+1}$, it must be

$$\begin{aligned}\sum_{i=1}^I \mathbf{x}_{(i)}^{k+1} &= \sum_{i=1}^I \mathbf{u}_i^{k+1}\\ &= \sum_{i=1}^I \mathbf{u}_i^k + \sum_{i=1}^I \left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^k\right)\\ &\overset{(a)}{=} \sum_{i=1}^I \mathbf{x}_{(i)}^k + \sum_{i=1}^I \left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^k\right)\\ &\overset{(b)}{=} \sum_{i=1}^I \left(\sum_{j=1}^I w_{ij}^k \, \mathbf{x}_{(j)}^k + \left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^k\right)\right),\end{aligned} \tag{3.183}$$

where in (a) we used $\sum_{i=1}^I \mathbf{x}_{(i)}^k = \sum_{i=1}^I \mathbf{u}_i^k$; and (b) follows from the column stochasticity of $\mathbf{W}^k$ [cf. (3.182)]. This naturally suggests the following modification of the protocol (3.169): for all $i = 1, \ldots, I$,

$$\mathbf{x}_{(i)}^{k+1} = \sum_{j=1}^{I} w_{ij}^{k}\, \mathbf{x}_{(j)}^{k} + \left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^{k}\right), \tag{3.184}$$

with $\mathbf{x}_{(i)}^0 = \mathbf{u}_i^0$, $i = 1, \ldots, I$. Generally speaking, (3.184) has the following interpretation: by averaging neighbors' information, the first term $\sum_{j=1}^{I} w_{ij}^{k}\, \mathbf{x}_{(j)}^{k}$ aims at enforcing a consensus among the $\mathbf{x}_{(i)}^{k}$'s while the second term is a perturbation that bias the current sum $\sum_{i=1}^{I} \mathbf{x}_{(i)}^{k}$ towards $\sum_{i=1}^{I} \mathbf{u}_i^k$. By (3.183) and a similar argument as in (3.177), it is not difficult to check that if the $\mathbf{x}_{(i)}^{k}$ are consensual it must be $\lim_{k\to\infty} \|\mathbf{x}_{(i)}^{k} - \bar{\mathbf{u}}^k\| = 0$, for all $i = 1, \ldots, I$. Theorem 4.10 proves this result (as a special case of the tracking scheme over arbitrary directed graphs).

Distributed Tracking Over Arbitrary Directed Graphs

Consider now the case of digraphs $\{G^k\}_{k\in\mathbb{N}_+}$ with arbitrary topology. With the results established in Sect. 3.4.2.2, the tracking mechanism (3.184) can be naturally generalized to this more general setting by "perturbing" the condensed push-sum scheme (3.178) as follows: for $i = 1, \ldots, I$,

$$\begin{aligned} \phi_{(i)}^{k+1} &= \sum_{j=1}^{I} a_{ij}^{k}\, \phi_{(j)}^{k}, \\ \mathbf{x}_{(i)}^{k+1} &= \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^{k}\, \phi_{(j)}^{k}\, \mathbf{x}_{(j)}^{k} + \frac{1}{\phi_{(i)}^{k+1}} \cdot \left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^{k}\right), \end{aligned} \tag{3.185}$$

where $\mathbf{x}_{(i)}^0 = \mathbf{u}_i^0/\phi_{(i)}^0$, $i = 1, \ldots, I$; and $(\phi_{(i)}^0)_{i=1}^I$ are arbitrary positive scalars such that $\sum_{i=1}^{I} \phi_{(i)}^{k} = I$. For simplicity, hereafter, we tacitly set $\phi_{(i)}^0 = 1$, for all $i = 1, \ldots, I$. Note that, differently from (3.184), in (3.185), we have scaled the perturbation by $(\phi_{(i)}^{k+1})^{-1}$ so that the weighted average is preserved, that is, $\sum_{i=1}^{I} \phi_{(i)}^{k}\, \mathbf{x}_{(i)}^{k} = \sum_{i=1}^{I} \mathbf{u}_i^k$, for all $k \in \mathbb{N}_+$.

Convergence of the tracking scheme (3.185) is stated in Theorem 4.10 below, whose proof is omitted because is a special case of the more general result stated in Theorem 4.11 (cf. Sect. 3.4.2.4).

Assumption 4.9 *Let* $\{\mathbf{u}_i^k\}_{k\in\mathbb{N}_+}$ *be such that*

$$\lim_{k\to\infty} \|\mathbf{u}_i^{k+1} - \mathbf{u}_i^{k}\| = 0, \quad \forall i = 1, \ldots, I.$$

Theorem 4.10 *Let $\{G^k\}_{k\in\mathbb{N}_+}$ be a sequence of graphs satisfying Assumption 4.2. Consider the distributed tracking protocol (3.185), where each $\mathbf{A}^k$ is chosen according to Assumption 4.7. Then, the sequence $\{\mathbf{x}^k \triangleq (\mathbf{x}^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$ generated by (3.185) satisfies:*

(a) `Invariance of the weighted-sum:`

$$\sum_{i=1}^{I} \phi^k_{(i)}\, \mathbf{x}^k_{(i)} = \sum_{i=1}^{I} \mathbf{u}^k_i, \quad \forall k \in \mathbb{N}_+; \tag{3.186}$$

(b) `Asymptotic consensus:` *if, in addition, the sequence of signals $\{(\mathbf{u}^k_i)_{i=1}^I\}_{k\in\mathbb{N}_+}$ satisfies Assumption 4.9, then*

$$\lim_{k\to\infty} \left\| \mathbf{x}^k_{(i)} - \bar{\mathbf{u}}^k \right\| = 0, \quad \forall i = 1, \ldots, I. \tag{3.187}$$

3.4.2.4 Perturbed Condensed Push-Sum

In this section, we provide a unified proof of Theorems 4.5, 4.8, and 4.10 by interpreting the consensus and tracking schemes introduced in the previous sections [namely: (3.169), (3.178) and (3.185)] as special instances of the following *perturbed condensed push-sum* protocol: for all $i = 1, \ldots, I$, and $k \in \mathbb{N}_+$,

$$\begin{aligned} \phi^{k+1}_{(i)} &= \sum_{j=1}^{I} a^k_{ij}\, \phi^k_{(j)}, \\ \mathbf{x}^{k+1}_{(i)} &= \frac{1}{\phi^{k+1}_{(i)}} \sum_{j=1}^{I} a^k_{ij}\, \phi^k_{(j)}\, \mathbf{x}^k_{(j)} + \boldsymbol{\varepsilon}^k_i, \end{aligned} \tag{3.188}$$

with given $\mathbf{x}^0 \triangleq (\mathbf{x}^0_{(i)})_{i=1}^I$, and $\phi^0_{(i)} = 1$, for all $i = 1, \ldots, I$; where $\boldsymbol{\varepsilon}^k_i$ models a perturbation locally injected into the system by agent i at iteration k. Clearly, the schemes (3.178) and (3.185) are special case of (3.188), obtained setting $\boldsymbol{\varepsilon}^k_i \triangleq \mathbf{0}$ and $\boldsymbol{\varepsilon}^k_i \triangleq (\phi^{k+1}_{(i)})^{-1}(\mathbf{u}^{k+1}_i - \mathbf{u}^k_i)$, respectively. Furthermore, if the matrices $\mathbf{A}^k$ are doubly-stochastic, we obtain as special cases the schemes (3.169) and (3.184), respectively.

Convergence of the scheme (3.188) is given in the theorem below.

Theorem 4.11 *Let $\{G^k\}_{k\in\mathbb{N}_+}$ be a sequence of graphs satisfying Assumption 4.2. Consider the perturbed condensed push-sum protocol (3.188), where each $\mathbf{A}^k$ is chosen according to Assumption 4.7. Then, the sequences $\{\mathbf{x}^k \triangleq (\mathbf{x}^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$ and*

$\{\boldsymbol{\phi}^k \triangleq (\phi^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$ *generated by* (3.188) *satisfy:*

(a) `Bounded` $\{\boldsymbol{\phi}^k\}_{k\in\mathbb{N}_+}$*:*

$$\begin{aligned}\phi_{lb} &\triangleq \inf_{k\in\mathbb{N}_+}\min_{1\le i\le I}\phi^k_{(i)} \ge \kappa^{2(I-1)B},\\ \phi_{ub} &\triangleq \sup_{k\in\mathbb{N}_+}\max_{1\le i\le I}\phi^k_{(i)} \le I-\kappa^{2(I-1)B};\end{aligned} \tag{3.189}$$

(b) `Bounded consensus error`*: for all* $k\in\mathbb{N}_+$*,*

$$\left\|\mathbf{x}^k_{(i)} - \frac{1}{I}\sum_{j=1}^I \phi^k_{(j)}\,\mathbf{x}^k_{(j)}\right\| \le c\cdot\left((\rho)^k\,\|\mathbf{x}^0\| + \sum_{t=0}^{k-1}(\rho)^{k-1-t}\,\|\boldsymbol{\varepsilon}^t\|\right),\quad \forall i=i,\ldots,I, \tag{3.190}$$

where $\boldsymbol{\varepsilon}^k \triangleq (\boldsymbol{\varepsilon}^k_i)_{i=1}^I$*, and* $c>0$ *and* $\rho\in(0,1)$ *are constants defined as*

$$c \triangleq \frac{2I}{\rho}\cdot\frac{2(1+\tilde{\kappa}^{-(I-1)B})}{1-\tilde{\kappa}^{(I-1)B}} \quad \textit{and} \quad \rho \triangleq \left(1-\tilde{\kappa}^{(I-1)B}\right)^{\frac{1}{(I-1)B}},$$

with $\tilde{\kappa} \triangleq \kappa\cdot\left(\phi_{lb}/\phi_{ub}\right)$*, and* B *and* κ *defined in Assumption 4.2 and Assumption 4.3, respectively.*

Proof See Sect. 3.4.2.5. □

Discussion

Let us apply now Theorem 4.11 to study the impact on the consensus value of the following three different perturbation errors:

1. **Error free**: $\boldsymbol{\varepsilon}^k_i = \mathbf{0}$, for all $i=1,\ldots,I$, and $k\in\mathbb{N}_+$;
2. **Vanishing error**: $\lim_{k\to\infty}\|\boldsymbol{\varepsilon}^k_i\| = 0$, for all $i=1,\ldots,I$, and $k\in\mathbb{N}_+$;
3. **Bounded error**: There exists a constant $0\le M<+\infty$ such that $\|\boldsymbol{\varepsilon}^k_i\|\le M$, for all $i=1,\ldots,I$, and $k\in\mathbb{N}_+$.

Case 1: Error Free Since $\boldsymbol{\varepsilon}^k_i=\mathbf{0}$, $i=1,\ldots,I$, the perturbed consensus protocol (3.188) reduces to the condensed push-sum consensus scheme (3.178). According to Theorem 4.11, each $\mathbf{x}^k_{(i)}$ converges to the weighted average $(1/I)\cdot\sum_{i=1}^I\phi^k_{(i)}\mathbf{x}^k_{(i)}$ geometrically. Since $(1/I)\cdot\sum_{i=1}^I\phi^k_{(i)}\mathbf{x}^k_{(i)} = (1/I)\cdot\sum_{i=1}^I\phi^0_{(i)}\mathbf{x}^0_{(i)} = \bar{\mathbf{u}}$, for all $k\in\mathbb{N}_+$, each $\|\mathbf{x}^k_{(i)}-\bar{\mathbf{u}}\|$ vanishes geometrically, which proves Theorem 4.8.

If, in addition, the weighted matrices $\mathbf{A}^k$ are also row-stochastic (and thus doubly-stochastic), we have $\phi^k_{(i)}=1$, for all $k\in\mathbb{N}_+$. Therefore, (3.188) reduces to the vanilla average consensus protocol (3.169); and in Theorem 4.11 we have $\tilde{\kappa} = \kappa$, which proves Theorem 4.5 as a special case.

Case 2: Vanishing Error To study the consensus value in this setting, we need the following lemma on the convergence of product of sequences.

Lemma 4.12 ([72, Lemma 7]) *Let* $0 < \lambda < 1$*, and let* $\{\beta^k\}_{k\in\mathbb{N}_+}$ *and* $\{\nu^k\}_{k\in\mathbb{N}_+}$ *be two positive scalar sequences. Then, the following hold:*

(a) If $\lim\limits_{k\to\infty} \beta^k = 0$*, then*

$$\lim_{k\to\infty} \sum_{t=1}^{k} (\lambda)^{k-t} \beta^t = 0.$$

(b) If $\sum\limits_{k=1}^{\infty} (\beta^k)^2 < \infty$ *and* $\sum\limits_{k=1}^{\infty} (\nu^k)^2 < \infty$*, then*

1) $\lim\limits_{k\to\infty} \sum\limits_{t=1}^{k} \sum\limits_{l=1}^{t} (\lambda)^{t-l} (\beta^l)^2 < \infty$;

2) $\lim\limits_{k\to\infty} \sum\limits_{t=1}^{k} \sum\limits_{l=1}^{t} (\lambda)^{t-l} \beta^t \nu^l < \infty$.

□

Consider (3.190): Since $\rho \in (0, 1)$, invoking Lemma 4.12(a) and $\lim_{k\to\infty} \|\boldsymbol{\varepsilon}^k\| = 0$, we have

$$\lim_{k\to\infty} \sum_{t=0}^{k-1} (\rho)^{k-1-t} \|\boldsymbol{\varepsilon}^t\| = 0; \tag{3.191}$$

hence $\lim_{k\to\infty} \|\mathbf{x}_{(i)}^k - (1/I) \cdot \sum_{j=1}^{I} \phi_{(j)}^k \mathbf{x}_{(j)}^k\| = 0$ Note that, in this case, the rate of convergence to the weighted average may not be geometric.

Consider, as special case, the tracking algorithm (3.185). i.e., set in (3.188) $\boldsymbol{\varepsilon}_i^k = (\phi_{(i)}^{k+1})^{-1}(\mathbf{u}_i^{k+1} - \mathbf{u}_i^k)$. We have $(1/I) \cdot \sum_{i=1}^{I} \phi_{(i)}^k \mathbf{x}_{(i)}^k = \bar{\mathbf{u}}^k$, which proves Theorem 4.10.

Case 3: Bounded Error Since $\|\boldsymbol{\varepsilon}_i^k\| \le M$, $i = 1, \ldots, I$, the consensus error in (3.190) can be bounded as

$$\begin{aligned}\left\| \mathbf{x}_{(i)}^k - \frac{1}{I} \sum_{j=1}^{I} \phi_{(j)}^k \mathbf{x}_{(j)}^k \right\| &\le c \left((\rho)^k \|\mathbf{x}^0\| + M \cdot \sum_{t=0}^{k-1} (\rho)^{k-1-t} \right) \\ &\le c \left((\rho)^k \|\mathbf{x}^0\| + M \cdot \frac{1 - (\rho)^k}{1 - \rho} \right).\end{aligned} \tag{3.192}$$

3.4.2.5 Proof of Theorem 4.11

We prove now Theorem 4.11, following the analysis in [208]. We first introduce some intermediate results and useful notation.

Preliminaries

It is convenient to rewrite the perturbed consensus protocol (3.188) in a more compact form. To do so, let us introduce the following notation: given the matrix $\mathbf{A}^k$ compliant to the graph G^k (cf. Assumption 4.3) and $\mathbf{W}^k$ defined in (3.179), let

$$\mathbf{x}^k \triangleq [\mathbf{x}_{(1)}^{kT}, \ldots, \mathbf{x}_{(I)}^{kT}]^T, \tag{3.193}$$

$$\boldsymbol{\varepsilon}^k \triangleq [\boldsymbol{\varepsilon}_1^{kT}, \ldots, \boldsymbol{\varepsilon}_I^{kT}]^T, \tag{3.194}$$

$$\boldsymbol{\phi}^k \triangleq [\phi_{(1)}^k, \ldots, \phi_{(I)}^k]^T, \tag{3.195}$$

$$\boldsymbol{\Phi}^k \triangleq \mathrm{Diag}(\boldsymbol{\phi}^k), \tag{3.196}$$

$$\widehat{\boldsymbol{\Phi}}^k \triangleq \boldsymbol{\Phi}^k \otimes \mathbf{I}_m, \tag{3.197}$$

$$\widehat{\mathbf{A}}^k \triangleq \mathbf{A}^k \otimes \mathbf{I}_m, \tag{3.198}$$

$$\widehat{\mathbf{W}}^k \triangleq \mathbf{W}^k \otimes \mathbf{I}_m, \tag{3.199}$$

where Diag(•) denotes a diagonal matrix whose diagonal entries are the elements of the vector argument. Under the column stochasticity of $\mathbf{A}^k$ (Assumption 4.7), it is not difficult to check that the following relationship exists between $\mathbf{W}^k$ and $\mathbf{A}^k$ (and $\widehat{\mathbf{W}}^k$ and $\widehat{\mathbf{A}}^k$):

$$\mathbf{W}^k = \left(\boldsymbol{\Phi}^{k+1}\right)^{-1} \mathbf{A}^k\, \boldsymbol{\Phi}^k \quad \text{and} \quad \widehat{\mathbf{W}}^k = \left(\widehat{\boldsymbol{\Phi}}^{k+1}\right)^{-1} \widehat{\mathbf{A}}^k\, \widehat{\boldsymbol{\Phi}}^k. \tag{3.200}$$

Using the above notation, the perturbed consensus protocol (3.188) can be rewritten in matrix form as

$$\boldsymbol{\phi}^{k+1} = \mathbf{A}^k \boldsymbol{\phi}^k \quad \text{and} \quad \mathbf{x}^{k+1} = \widehat{\mathbf{W}}^k\, \mathbf{x}^k + \boldsymbol{\varepsilon}^k. \tag{3.201}$$

To study the dynamics of the consensus error in (3.201), let us introduce the matrix products: given $k, t \in \mathbb{N}_+$, with $k \geq t$,

$$\mathbf{A}^{k:t} \triangleq \begin{cases} \mathbf{A}^k \mathbf{A}^{k-1} \cdots \mathbf{A}^t, & \text{if } k > t \geq 0, \\ \mathbf{A}^k, & \text{if } k = t \geq 0, \end{cases} \tag{3.202}$$

$$\mathbf{W}^{k:t} \triangleq \begin{cases} \mathbf{W}^k \mathbf{W}^{k-1} \cdots \mathbf{W}^t, & \text{if } k > t \geq 0, \\ \mathbf{W}^k, & \text{if } k = t \geq 0, \end{cases} \tag{3.203}$$

and

$$\widehat{\mathbf{A}}^{k:t} \triangleq \mathbf{A}^{k:t} \otimes \mathbf{I}_m, \tag{3.204}$$

$$\widehat{\mathbf{W}}^{k:t} \triangleq \mathbf{W}^{k:t} \otimes \mathbf{I}_m. \tag{3.205}$$

Define the weight-averaging matrix as

$$\mathbf{J}_{\boldsymbol{\phi}^k} \triangleq \frac{1}{I}\left(\mathbf{1}\,(\boldsymbol{\phi}^k)^T\right) \otimes \mathbf{I}_m, \tag{3.206}$$

so that

$$\mathbf{J}_{\boldsymbol{\phi}^k}\,\mathbf{x}^k = \mathbf{1} \otimes \frac{1}{I}\sum_{i=1}^{I} \phi^k_{(i)}\mathbf{x}^k_{(i)},$$

i.e., $\mathbf{J}_{\boldsymbol{\phi}^k}\,\mathbf{x}^k$ computes the average of $\mathbf{x}^k_{(i)}$ weighted by $(\phi^k_{(i)})_{i=1}^I$ and stacks it I times in a column vector. Under the column stochasticity of $\mathbf{A}^k$ (Assumption 4.7), it is not difficult to check that the following property holds between $\mathbf{J}_{\boldsymbol{\phi}^k}$ and $\widehat{\mathbf{W}}^{k:t}$: for all $k, t \in \mathbb{N}_+$, with $k \geq t$,

$$\mathbf{J}_{\boldsymbol{\phi}^{k+1}}\,\widehat{\mathbf{W}}^{k:t} = \mathbf{J}_{\boldsymbol{\phi}^t} = \widehat{\mathbf{W}}^{k:t}\,\mathbf{J}_{\boldsymbol{\phi}^t}. \tag{3.207}$$

Convergence of the perturbed consensus protocol boils down to studying the dynamics of $\|\widehat{\mathbf{W}}^{k:t} - \mathbf{J}_{\boldsymbol{\phi}^t}\|$ (this will be more clear in the next subsection). The following lemma shows that, in the setting of Theorem 4.11, $\widehat{\mathbf{W}}^{k:t}$ converges geometrically to $\mathbf{J}_{\boldsymbol{\phi}^t}$; the proof of the lemma is omitted and can be found in [208, Lemma 2].

Lemma 4.13 *Let $\{G^k\}_{k\in\mathbb{N}_+}$ be a sequence of graphs satisfying Assumption 4.2; let $\{\mathbf{A}^k\}_{k\in\mathbb{N}_+}$ be a sequence of weight matrices satisfying Assumption 4.7; and let $\{\mathbf{W}^k\}_{k\in\mathbb{N}_+}$ be the sequence of (row-stochastic) matrices with $\mathbf{W}^k$ related to $\mathbf{A}^k$ by (3.200). Then, the following holds: for all $k,\ t \in \mathbb{N}_+$, with $k \geq t$,*

$$\left\|\widehat{\mathbf{W}}^{k:t} - \mathbf{J}_{\boldsymbol{\phi}^t}\right\|_2 \leq c \cdot (\rho)^{k-t+1}, \tag{3.208}$$

where $c > 0$ and $\rho \in (0, 1)$ are defined in Theorem 4.11.

Proof of Theorem 4.11

We are now ready to prove the theorem. We start rewriting the dynamics of the consensus error $\mathbf{x}^k - \mathbf{J}_{\boldsymbol{\phi}^k}\mathbf{x}^k$ in a form that permits the application of Lemma 4.13.

Applying the x-update in (3.201) recursively, we have

$$\begin{aligned}\mathbf{x}^k &= \widehat{\mathbf{W}}^{k-1}\mathbf{x}^{k-1} + \boldsymbol{\varepsilon}^{k-1}\\ &= \widehat{\mathbf{W}}^{k-1}\left(\widehat{\mathbf{W}}^{k-2}\mathbf{x}^{k-2} + \boldsymbol{\varepsilon}^{k-2}\right) + \boldsymbol{\varepsilon}^{k-1}\\ &= \cdots = \widehat{\mathbf{W}}^{k-1:0}\,\mathbf{x}^0 + \sum_{t=1}^{k-1}\widehat{\mathbf{W}}^{k-1:t}\boldsymbol{\varepsilon}^{t-1} + \boldsymbol{\varepsilon}^{k-1}.\end{aligned} \tag{3.209}$$

Therefore, the weighted average $\mathbf{J}_{\boldsymbol{\phi}^k}\mathbf{x}^k$ can be written as

$$\begin{aligned}\mathbf{J}_{\boldsymbol{\phi}^k}\mathbf{x}^k &\overset{(3.209)}{=} \mathbf{J}_{\boldsymbol{\phi}^k}\,\widehat{\mathbf{W}}^{k-1:0}\,\mathbf{x}^0 + \mathbf{J}_{\boldsymbol{\phi}^k}\sum_{t=1}^{k-1}\widehat{\mathbf{W}}^{k-1:t}\,\boldsymbol{\varepsilon}^{t-1} + \mathbf{J}_{\boldsymbol{\phi}^k}\,\boldsymbol{\varepsilon}^{k-1}\\ &\overset{(3.207)}{=} \mathbf{J}_{\boldsymbol{\phi}^0}\,\mathbf{x}^0 + \sum_{t=1}^{k-1}\mathbf{J}_{\boldsymbol{\phi}^t}\,\boldsymbol{\varepsilon}^{t-1} + \mathbf{J}_{\boldsymbol{\phi}^k}\,\boldsymbol{\varepsilon}^{k-1}.\end{aligned} \tag{3.210}$$

Using (3.209) and (3.210) and Lemma 4.13, the consensus error can be bounded as

$$\begin{aligned}&\left\|\mathbf{x}^k - \mathbf{J}_{\boldsymbol{\phi}^k}\mathbf{x}^k\right\|\\ &= \left\|\left(\widehat{\mathbf{W}}^{k-1:0} - \mathbf{J}_{\boldsymbol{\phi}^0}\right)\mathbf{x}^0 + \sum_{t=1}^{k-1}\left(\widehat{\mathbf{W}}^{k-1:t} - \mathbf{J}_{\boldsymbol{\phi}^t}\right)\boldsymbol{\varepsilon}^{t-1} + \left(\mathbf{I} - \mathbf{J}_{\boldsymbol{\phi}^k}\right)\boldsymbol{\varepsilon}^{k-1}\right\|\\ &\overset{(3.208)}{\leq} c\cdot(\rho)^k\|\mathbf{x}^0\| + c\cdot\sum_{t=1}^{k-1}(\rho)^{k-t}\,\|\boldsymbol{\varepsilon}^{t-1}\| + \underbrace{\left\|\mathbf{I} - \mathbf{J}_{\boldsymbol{\phi}^k}\right\|_2}_{\leq\sqrt{2\,I}}\|\boldsymbol{\varepsilon}^{k-1}\|\\ &\leq c\cdot\left((\rho)^k\|\mathbf{x}^0\| + \sum_{t=0}^{k-1}(\rho)^{k-t-1}\,\|\boldsymbol{\varepsilon}^t\|\right),\end{aligned} \tag{3.211}$$

where in the last inequality we used $c > \sqrt{2\,I}$. The inequality $\|\mathbf{I} - \mathbf{J}_{\boldsymbol{\phi}^k}\|_2 \leq \sqrt{2\,I}$ can be proved as follows. Let $\mathbf{z} \in \mathbb{R}^{I\cdot m}$ be an arbitrary vector; partition $\mathbf{z}$ as $\mathbf{z} = (\mathbf{z}_i)_{i=1}^I$, with each $\mathbf{z}_i \in \mathbb{R}^m$. Then,

$$\begin{aligned}\left\|\mathbf{z} - \mathbf{J}_{\boldsymbol{\phi}^k}\,\mathbf{z}\right\| &\leq \|\mathbf{z} - \mathbf{J}_{\mathbf{1}}\,\mathbf{z}\| + \left\|\mathbf{J}_{\mathbf{1}}\mathbf{z} - \mathbf{J}_{\boldsymbol{\phi}^k}\mathbf{z}\right\| \overset{(a)}{\leq} \|\mathbf{z}\| + \frac{\sqrt{I}}{I}\left\|\sum_{i=1}^I\mathbf{z}_i - \sum_{i=1}^I\phi_i^k\mathbf{z}_i\right\|\\ &\leq \|\mathbf{z}\| + \frac{\sqrt{I}}{I}\sqrt{I^2 - I}\,\|\mathbf{z}\| \leq \sqrt{2\,I}\,\|\mathbf{z}\|,\end{aligned}$$

where in (a) we used $\|\mathbf{I}-\mathbf{J}_{\mathbf{1}}\| = 1$ (note that $\mathbf{I}-\mathbf{J}_{\mathbf{1}}$ is a Toeplitz matrix, with diagonal entries equal to $1 - 1/I$ and off-diagonal entries all equal to $-1/I$; therefore, its eigenspectrum is given by $\{0, 1, \ldots, 1\}$).

The inequality (3.211) proves the theorem. □

3.4.3 Distributed SCA Over Time-Varying Digraphs

We are now ready to introduce the proposed distributed algorithmic framework to solve Problem (3.167), which combines SCA techniques (introduced in the previous lectures) with the consensus/tracking protocols described in Sect. 3.4.2. We consider the optimization over time-varying (B-strongly connected) digraphs (cf. Assumption 4.2); distributed algorithms for undirected or time invariant networks can be obtained as special cases.

As already anticipated, each agent i maintains and updates iteratively a *local copy* $\mathbf{x}_{(i)}$ of the global variable $\mathbf{x}$, along with an auxiliary variable $\mathbf{y}_{(i)} \in \mathbb{R}^m$, whose goal is to track locally the average of the gradients $(1/I) \cdot \sum_{i=1}^{I} \nabla f_i$ (the importance of this extra variable will be clarified shortly), an information that is not available at the agent's side; let $\mathbf{x}_{(i)}^k$ and $\mathbf{y}_{(i)}^k$ denote the values of $\mathbf{x}_{(i)}$ and $\mathbf{y}_{(i)}$ at iteration k, respectively. To update these variables, *two major steps* are performed iteratively, namely:

Step 1–Local SCA (optimization): Given $\mathbf{x}_{(i)}^k$ and $\mathbf{y}_{(i)}^k$, each agent i solves a convexification of Problem (3.167), wherein V is replaced by a suitably chosen strongly convex surrogate, which is built using only the available local information $\mathbf{x}_{(i)}^k$ and $\mathbf{y}_{(i)}^k$;

Step 2–Communication: All the agents broadcast the solutions computed in Step 1, and update their own variables $\mathbf{x}_{(i)}^k \to \mathbf{x}_{(i)}^{k+1}$ and $\mathbf{y}_{(i)}^k \to \mathbf{y}_{(i)}^{k+1}$, based on the information received from their neighbors.

The two steps above need to be designed so that: i) all the $\mathbf{x}_{(i)}^k$ will be asymptotically consensual, that is, $\lim_{k\to\infty} \|\mathbf{x}_{(i)}^k - (1/I) \cdot \sum_{j=1}^{I} \mathbf{x}_{(j)}^k\| = 0$, for all i; and ii) every limit point of $(1/I) \cdot \sum_{j=1}^{I} \mathbf{x}_{(j)}^k$ is a stationary solution of Problem (3.167). We describe next the above two steps in detail.

Step 1: Local SCA (Optimization) Each agent i faces two issues to solve Problem (3.167), namely: f_i is not convex and $\sum_{j\neq i} f_j$ is not known. To cope with the first issue, we leverage the SCA techniques introduced in the previous lectures. More specifically, at iteration k, agent i solves a convexification of V in (3.167) having the following form

$$\widehat{\mathbf{x}}_i\left(\mathbf{x}_{(i)}^k\right) \triangleq \underset{\mathbf{x}_{(i)}\in X}{\operatorname{argmin}}\ \widetilde{F}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) + G\left(\mathbf{x}_{(i)}\right), \tag{3.212}$$

where $\widetilde{F}_i : O \times O \to \mathbb{R}$ is a suitably chosen surrogate of F. To guarantee that a fixed point of $\widehat{\mathbf{x}}_i(\bullet)$ is a stationary solution of (3.167), a naive application of the SCA theory developed in Lecture II, would call $\widetilde{F}_i$ to satisfy the following gradient consistency condition (cf. Assumption 3.2, Sect. 3.3.2): $\widetilde{F}_i$ is C^1 on O and

$$\nabla \widetilde{F}_i\left(\mathbf{x}_{(i)}^k \mid \mathbf{x}_{(i)}^k\right) = \nabla F\left(\mathbf{x}_{(i)}^k\right) = \nabla f_i\left(\mathbf{x}_{(i)}^k\right) + \sum_{j \neq i} \nabla f_j(\mathbf{x}_{(i)}^k), \quad k \in \mathbb{N}_+. \tag{3.213}$$

For example, a surrogate function satisfying the above condition would be:

$$\widetilde{F}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) = \widetilde{f}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) + \left(\sum_{j \neq i} \nabla f_j\left(\mathbf{x}_{(i)}^k\right)\right)^T \left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right), \tag{3.214}$$

where $\widetilde{f}_i(\bullet \mid \mathbf{x}_{(i)}^k) : O \to \mathbb{R}$ is a strongly convex surrogate of f_i on X, consistent to f_i, in the following sense (cf. Assumption 3.2).

Assumption 4.14 *Each function $\widetilde{f}_i : O \times O \to \mathbb{R}$ satisfies the following conditions:*

1. $\widetilde{f}_i\,(\bullet \mid \mathbf{x})$ *is τ_i-strongly convex on X, for all* $\mathbf{x} \in X$*;*
2. $\widetilde{f}_i\,(\bullet \mid \mathbf{x})$ *is C^1 on O and $\nabla \widetilde{f}_i\,(\mathbf{x} \mid \mathbf{x}) = \nabla f_i\,(\mathbf{x})$, for all* $\mathbf{x} \in X$*;*
3. $\nabla \widetilde{f}_i\,(\mathbf{x} \mid \bullet)$ *is $\widetilde{L}_i$-Lipschitz on X, for all* $\mathbf{x} \in X$.

Unfortunately, the surrogate function in (3.214) cannot be used by agent i, because of the lack of knowledge of $\sum_{j \neq i} \nabla f_j(\mathbf{x}_{(i)}^k)$; hence a gradient consistency condition in the form (3.213) cannot be enforced in such a distributed setting.

To cope with this issue, the idea, first proposed in [72] and further developed in [208], is to replace $\sum_{j \neq i} \nabla f_j(\mathbf{x}_{(i)}^k)$ in (3.214) [and in (3.213)] with a *local, asymptotically consistent*, approximation, so that condition (3.213) will be satisfied in the limit, as $k \to \infty$. This can be accomplished, e.g., using the following surrogate function:

$$\widetilde{F}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k, \mathbf{y}_{(i)}^k\right) = \widetilde{f}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) + \left(I \cdot \mathbf{y}_{(i)}^k - \nabla f_i(\mathbf{x}_{(i)}^k)\right)^T \left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right), \tag{3.215}$$

where $\widetilde{f}_i$ is defined as in (3.214); and $\mathbf{y}_{(i)}^k$ in (3.215) is an auxiliary variable controlled by agent i, aiming at tracking locally the average of the gradients $(1/I) \cdot \sum_{j=1}^I \nabla f_j(\mathbf{x}_{(i)}^k)$, that is, $\lim_{k\to\infty} \|\mathbf{y}_{(i)}^k - (1/I) \cdot \sum_{j=1}^I \nabla f_j(\mathbf{x}_{(i)}^k)\| = 0$. This explains the role of the linear term in (3.215): under the claimed tracking property of $\mathbf{y}_{(i)}^k$, we have

$$\lim_{k\to\infty} \left\| \left(I \cdot \mathbf{y}_{(i)}^k - \nabla f_i(\mathbf{x}_{(i)}^k)\right) - \sum_{j \neq i} \nabla f_j(\mathbf{x}_{(i)}^k) \right\| = 0, \tag{3.216}$$

which would guarantee that the gradient consistency condition (3.213), with now $\widetilde{F}_i(\bullet \mid \mathbf{x}_{(i)}^k)$ replaced by $\widetilde{F}_i(\bullet \mid \mathbf{x}_{(i)}^k, \mathbf{y}_{(i)}^k)$ in (3.215), will be *asymptotically* satisfied, that is,

$$\lim_{k\to\infty} \left\| \nabla \widetilde{F}_i(\mathbf{x}_{(i)}^k \mid \mathbf{x}_{(i)}^k, \mathbf{y}_{(i)}^k) - \nabla F(\mathbf{x}_{(i)}^k) \right\| = 0.$$

As it will be shown later, this relaxed condition is in fact enough to prove that, if convergence and consensus are asymptotically achieved, the limit point of all the local variables $\mathbf{x}_{(i)}^k$ is a stationary solution of Problem (3.167). Leveraging the distributed tracking protocol (3.185) (cf. Sect. 3.4.2.3), in Step 2 below, we show how to devise an update for the $\mathbf{y}_{(i)}^k$ variables that uses only local information and such that (3.216) asymptotically holds.

Using $\widetilde{F}_i(\bullet \mid \mathbf{x}_{(i)}^k, \mathbf{y}_{(i)}^k)$ defined in (3.215), the local optimization step performing by each agent i consists then in solving the following strongly convex problem:

$$\widetilde{\mathbf{x}}_i^k \triangleq \underset{\mathbf{x}_{(i)}\in X}{\operatorname{argmin}} \; \widetilde{F}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k, \mathbf{y}_{(i)}^k\right) + G\left(\mathbf{x}_{(i)}\right), \tag{3.217}$$

followed by the step-size update

$$\mathbf{x}_{(i)}^{k+1/2} = \mathbf{x}_{(i)}^k + \gamma^k \left(\widetilde{\mathbf{x}}_i^k - \mathbf{x}_{(i)}^k\right), \tag{3.218}$$

where $\gamma^k \in (0, 1]$ is the step-size, to be properly chosen.

Step 2–Communication Given $\mathbf{x}_{(i)}^{k+1/2}$ and $\mathbf{y}_{(i)}^k$, each agent i communicates with its current neighbors in order to achieve asymptotic consensus on $\mathbf{x}_{(i)}$'s as well as track $(1/I)\cdot\sum_{j=1}^I \nabla f_j(\mathbf{x}_{(i)}^k)$ by $\mathbf{y}_{(i)}^k$. Both goals can be accomplished using (two instances of) the condensed perturbed push-sum protocol (3.188), introduced in Sect. 3.4.2.4.

More specifically, after obtaining $\mathbf{x}_{(j)}^{k+1/2}$ from its neighbors, each agent i updates its own local estimate $\mathbf{x}_{(i)}$ employing:

$$\phi_{(i)}^{k+1} = \sum_{j=1}^I a_{ij}^k \, \phi_{(j)}^k; \tag{3.219}$$

$$\mathbf{x}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^I a_{ij}^k \, \phi_{(j)}^k \, \mathbf{x}_{(j)}^{k+1/2}, \tag{3.220}$$

where the weights $\mathbf{A}^k \triangleq (a_{ij}^k)_{i=1}^I$ are chosen according to Assumption 4.7; and the $\phi_{(i)}^k$ variables are initialized to $\phi_{(i)}^0 = 1$. This update can be clearly implemented locally: All agents i) send their local variables $\phi_{(j)}^k$ and $\phi_{(j)}^k \, \mathbf{x}_{(j)}^{k+1/2}$ to their out-neighbors; and ii) linearly combine with coefficients a_{ij}^k the information coming from their in-neighbors.

A local update for the $\mathbf{y}_{(i)}^k$ variable aiming at tracking $(1/I) \cdot \sum_{j=1}^{I} \nabla f_j(\mathbf{x}_{(i)}^k)$ can be readily obtained invoking the distributed tracking protocol (3.185), and setting therein $\mathbf{u}_i^k \triangleq \nabla f_i(\mathbf{x}_{(i)}^k)$. This leads to

$$\mathbf{y}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^k\, \phi_{(j)}^k\, \mathbf{y}_{(j)}^k + \frac{1}{\phi_{(i)}^{k+1}} \left(\nabla f_i\left(\mathbf{x}_{(i)}^{k+1}\right) - \nabla f_i\left(\mathbf{x}_{(i)}^k\right)\right), \tag{3.221}$$

where $\phi_{(i)}^{k+1}$ is defined in (3.219), and $\mathbf{y}_{(i)}^0 = \nabla f_i(\mathbf{x}_{(i)}^0)$. Note that, as for $\mathbf{x}_{(i)}^{k+1}$, the update of $\mathbf{y}_{(i)}^k$ is performed by agent i using only the information coming from its neighbors, with the same signaling as for (3.219)–(3.220).

The described distributed SCA method (Step 1 and Step 2) is summarized in Algorithm 10, and termed *distributed Successive cONvex Approximation algorithm over Time-varying digrAphs* (SONATA).

Algorithm 10 SCA over time-varying digraphs (SONATA)

```
Data : $\mathbf{x}_{(i)}^0 \in X$, $\phi_{(i)}^0 = 1$, $\mathbf{y}_{(i)}^0 = \nabla f_i(\mathbf{x}_{(i)}^0)$, for all $i = 1, \ldots, I$; $\{\gamma^k \in (0, 1]\}_{k \in \mathbb{N}_+}$.
       Set $k = 0$.
(S.1) : If $\mathbf{x}^k$ satisfies a termination criterion: STOP;
(S.2) : Local SCA. Each agent $i$ computes

    $\widetilde{\mathbf{x}}_i^k \triangleq \underset{\mathbf{x}_{(i)} \in X}{\text{argmin}}\ \widetilde{f}_i\left(\mathbf{x}_{(i)} \,|\, \mathbf{x}_{(i)}^k\right) + \left(I \cdot \mathbf{y}_{(i)}^k - \nabla f_i(\mathbf{x}_{(i)}^k)\right)^T \left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right) + G\left(\mathbf{x}_{(i)}\right),$

    $\mathbf{x}_{(i)}^{k+1/2} = \mathbf{x}_{(i)}^k + \gamma^k \left(\widetilde{\mathbf{x}}_i^k - \mathbf{x}_{(i)}^k\right);$

(S.3) : Averaging and gradient tracking.
        Each agent $i$ sends out its local variables $\phi_{(i)}^k$, $\mathbf{x}_{(i)}^{k+1/2}$ and $\mathbf{y}_{(i)}^k$, and receives
        $\phi_{(j)}^k$, $\mathbf{x}_{(j)}^{k+1/2}$ and $\mathbf{y}_{(j)}^k$, with $j \in N_i^{\text{in}, k} \backslash \{i\}$. Then, it updates:

    $\phi_{(i)}^{k+1} = \sum_{j=1}^{I} a_{ij}^k\, \phi_{(j)}^k,$

    $\mathbf{x}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^k\, \phi_{(j)}^k\, \mathbf{x}_{(j)}^{k+1/2},$

    $\mathbf{y}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^k\, \phi_{(j)}^k\, \mathbf{y}_{(j)}^k + \frac{1}{\phi_{(i)}^{k+1}} \left(\nabla f_i\left(\mathbf{x}_{(i)}^{k+1}\right) - \nabla f_i\left(\mathbf{x}_{(i)}^k\right)\right);$

(S.4) : $k \leftarrow k + 1$, and go to (S.1).
```

Convergence of Algorithm 10 is stated in Theorem 4.16 below. We first introduce a standard condition on the steps-size γ^k and a proper merit function assessing the convergence of the algorithm.

Assumption 4.15 *The step-size* $\gamma^k \in (0, 1]$ *satisfies the standard diminishing rule:* $\lim_{k\to\infty} \gamma^k = 0$ *and* $\sum_{k=0}^{\infty} \gamma^k = +\infty$.

Given $\{\mathbf{x}^k \triangleq (\mathbf{x}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ generated by Algorithm 10, convergence of the algorithm is stated measuring the distance of the average sequence $\bar{\mathbf{x}}^k \triangleq (1/I) \cdot \sum_{i=1}^I \mathbf{x}_{(i)}^k$ from optimality and well as the consensus disagreement among the local variables $\mathbf{x}_{(i)}^k$'s. More specifically, let us introduce the following function as a measure of optimality:

$$J(\bar{\mathbf{x}}^k) \triangleq \left\| \bar{\mathbf{x}}^k - \underset{\mathbf{y}\in X}{\operatorname{argmin}} \left\{ \nabla F(\bar{\mathbf{x}}^k)^T (\mathbf{y} - \bar{\mathbf{x}}^k) + \frac{1}{2}\|\mathbf{y} - \bar{\mathbf{x}}^k\|_2^2 + G(\mathbf{y}) \right\} \right\|. \tag{3.222}$$

Note that J is a valid measure of stationarity because it is continuous and $J(\bar{\mathbf{x}}^\infty) = 0$ if and only if $\bar{\mathbf{x}}^\infty$ is a d-stationary solution of Problem (3.167). The consensus disagreement at iteration k is defined as

$$D(\mathbf{x}^k) \triangleq \|\mathbf{x}^k - \mathbf{1}_I \otimes \bar{\mathbf{x}}^k\|.$$

Note that D is equal to 0 if and only if all the $\mathbf{x}_{(i)}^k$'s are consensual. We combine the metrics J and D in a single merit function, defined as

$$M(\mathbf{x}^k) \triangleq \max\left\{J(\bar{\mathbf{x}}^k)^2,\ D(\mathbf{x}^k)^2\right\},$$

which captures the progresses of the algorithm towards optimality and consensus.

We are now ready to state the main convergence results for Algorithm 10.

Theorem 4.16 *Consider Problem* (3.167) *under Assumption 4.1; and let* $\{G^k\}_{k\in\mathbb{N}_+}$ *be a sequence of graphs satisfying Assumption 4.2. Let* $\{\mathbf{x}^k \triangleq (\mathbf{x}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 10 under Assumptions 4.7 and 4.14; and let* $\bar{\mathbf{x}}^k = (1/I) \cdot \sum_{i=1}^I \mathbf{x}_{(i)}^k$ *be the average sequence. Furthermore, suppose that either one of the following is satisfied.*

(a) `(diminishing step-size)`*: The step-size* γ^k *satisfies Assumption 4.15;*
(b) `(constant step-size)`*: The step-size* γ^k *is fixed—*$\gamma^k = \gamma$*, for all* $k \in \mathbb{N}_+$*—and it is sufficiently small (see [208, Theorem 5] for the specific expression of the upper bound on* γ*).*

Then, there holds

$$\lim_{k\to\infty} M(\mathbf{x}^k) = 0. \tag{3.223}$$

Proof See Sect. 3.4.3.2. □

Theorem 4.17 below provides an upper bound on the number of iterations needed to decrease $M(\mathbf{x}^k)$ below a given accuracy $\epsilon > 0$; we omit the proof and refer to [208, Theorem 6] for more details.

Theorem 4.17 *Consider Problem* (3.167) *and Algorithm 10 in the setting of Theorem 4.16. Given $\epsilon > 0$, let T_ϵ be the first iteration k such that $M(\mathbf{x}^k) \leq \epsilon$.*

(a) `(diminishing step-size)`*: Suppose that the step-size γ^k satisfies Assumption 4.15. Then,*

$$T_\epsilon \leq \inf\left\{k \in \mathbb{N}_+ : \sum_{t=0}^{k} \gamma^t \geq \frac{B_0}{\epsilon}\right\},$$

where $B_0 > 0$ is a constant independent on ϵ [208];

(b) `(constant step-size)`*: Suppose that the step-size γ^k is fixed, $\gamma^k = \gamma$, for all $k \in \mathbb{N}_+$. Then, there exists a sufficiently small $\bar{\gamma} \in (0, 1]$—independent of ϵ (see [208, Theorem 6] for the specific expression of $\bar{\gamma}$)—such that, if $\gamma \in (0, \bar{\gamma}]$, then it holds*

$$T_\epsilon = O\left(\frac{1}{\epsilon}\right).$$

Discussion and Generalizations

On the Convergence Stating convergence for constrained (nonsmooth) optimization problems in the form (3.167), Theorems 4.16 and 4.17 (proved in our work [208]) significantly enlarge the class of convex and nonconvex problems which distributed algorithms can be applied to with convergence guarantees. We remark that convergence is established without requiring that the (sub)gradients of F or G is bounded; this is a major improvement with respect to current distributed methods for nonconvex problems [20, 72, 233, 245] and nonsmooth convex ones [169].

We remark that convergence (as stated in the above theorems) can also be established weakening the assumption on the strongly convexity of the surrogates $\widetilde{f}_i$ (Assumption 4.14) to just convexity, as long as the feasible set X is compact. Also, with mild additional assumptions on G—see Lecture II—convergence can be also proved in the case wherein agents solve their subproblems (3.217) inexactly.

ATC- Versus CAA Updates As a final remark, we note that variants of SONATA wherein the order of the consensus, tracking, and local updates are differently combined, are still convergent, in the sense of Theorems 4.16 and 4.17. We briefly elaborate on this issue next.

Using a jargon well established in the literature [203], the update of the x-variables in Step 3 of Algorithm 10 is in the form of so-called *Adapt-Then-Combine (ATC)* strategy: eliminating the intermediate variable $\mathbf{x}_{(i)}^{k+1/2}$, each $\mathbf{x}_i^k$ follows the dynamic

$$\mathbf{x}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^{k}\, \phi_{(j)}^{k} \left(\mathbf{x}_{(j)}^{k} + \gamma^{k} \left(\widetilde{\mathbf{x}}_{j}^{k} - \mathbf{x}_{(j)}^{k}\right)\right). \tag{3.224}$$

The name ATC comes from the form of (3.224): each agent i first "adapts" its local copy $\mathbf{x}_{(i)}^{k}$ moving along the direction $\widetilde{\mathbf{x}}_{i}^{k} - \mathbf{x}_{(i)}^{k}$, that is, $\mathbf{x}_{(i)}^{k} \to \mathbf{x}_{(i)}^{k} + \gamma^{k}\, (\widetilde{\mathbf{x}}_{i}^{k} - \mathbf{x}_{(i)}^{k})$; and then it "combines" its new update with that of its in-neighbors.

As an alternative to (3.224), one can employ the so-called *Combine-And-Adapt (CAA)* update (also termed "consensus strategy" in [203]), which reads

$$\mathbf{x}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^{k}\, \phi_{(j)}^{k}\, \mathbf{x}_{(j)}^{k} + \frac{\phi_{(i)}^{k}}{\phi_{(i)}^{k+1}} \cdot \gamma^{k} \left(\widetilde{\mathbf{x}}_{i}^{k} - \mathbf{x}_{(i)}^{k}\right). \tag{3.225}$$

According to this protocol, each agent i first "combines" (by weighted-averaging) its current $\mathbf{x}_{(i)}^{k}$ with those of its neighbors, and then "adapt" the resulting update moving along the direction $\widetilde{\mathbf{x}}_{i}^{k} - \mathbf{x}_{(i)}^{k}$. Note that, when dealing with constraint optimization, the CAA update in general does not preserve the feasibility of the iterates while the ATC protocol does.

The ATC and CAA protocols can be interchangeably used also in the update of the tracking variables $\mathbf{y}_{(i)}^{k}$ in Step 3 of SONATA. While the y-update as stated in the algorithm [cf. (3.221)] is in the CAA form, one can also use the ATC-based update, which reads

$$\mathbf{y}_{(i)}^{k+1} = \frac{1}{\phi_{(i)}^{k+1}} \sum_{j=1}^{I} a_{ij}^{k} \left(\phi_{(j)}^{k}\mathbf{y}_{(j)}^{k} + \nabla f_j\big(\mathbf{x}_{(j)}^{k+1}\big) - \nabla f_j\big(\mathbf{x}_{(j)}^{k}\big)\right).$$

One can show that the above versions of SONATA are all convergent.

3.4.3.1 SONATA (Algorithm 10) and Special Cases

The SONATA framework represents a gamut of algorithms, each of them corresponding to a specific choice of the surrogate functions, step-size, and weight matrices. In this section, we focus on recent proposals in the literature that built on the idea of distributed gradient tracking [70–72, 172, 191, 257, 259], and we show that all of them are in fact special cases of SONATA. A more detailed analysis of the state of the art can be found in Sect. 3.4.5.

The idea of tracking the gradient averages through the use of consensus coupled with distributed optimization was independently introduced in the NEXT & SONATA framework [70–72] and [208, 232] for the general class of (convex) constrained nonsmooth nonconvex problems (3.167) and in [259] for the special case of strongly convex unconstrained smooth optimization. The algorithmic framework in [70–72] is applicable to optimization problems over time-varying graphs, but requires the use of doubly stochastic matrices. This assumption was removed in SONATA [208, 232] by using column-stochastic matrices as in the push-sum based methods. The scheme in [259] is implementable only over undirected fixed graphs. A convergence rate analysis of the scheme in [259] in the case of strongly convex smooth unconstrained optimization problems was later developed in [172, 191] for undirected graphs and in [172] for time-varying directed graphs. Complexity results for NEXT and SONATA for (strongly) convex and nonconvex constrained optimization problems over (time-varying) digraphs can be found in [208, 231]; differently from [172, 191], the analysis in [208, 231] applies to general surrogate functions (satisfying Assumption 4.14).

We establish next a formal connection between SONATA and all these schemes.

Preliminaries

Since all the aforementioned works but [70–72] are applicable only to the special instance of Problem (3.167) wherein $X = \mathbb{R}^m$ (unconstrained), $G = 0$ (only smooth objectives), and F is strongly convex, throughout this section, for a fair comparison, we only consider this setting. We begin customizing Algorithm 10 to this special instance of (3.167) as follows. Choose each surrogate function $\widetilde{f}_i$ in (3.215) as

$$\widetilde{f}_i(\mathbf{x}_{(i)} \,|\, \mathbf{x}_{(i)}^k) = f_i(\mathbf{x}_{(i)}^k) + \nabla f_i(\mathbf{x}_{(i)}^k)^T(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k) + \frac{I}{2}\left\|\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right\|^2.$$

This leads to the following closed form expression for $\widetilde{\mathbf{x}}_i^k$ in (3.217) (recall that $X = \mathbb{R}^m$ and $G = 0$):

$$\begin{aligned}\widetilde{\mathbf{x}}_i^k &= \underset{\mathbf{x}_i}{\operatorname{argmin}}\left(I \cdot \mathbf{y}_{(i)}^k\right)^\top\left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right) + \frac{I}{2}\left\|\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right\|^2 \\ &= \mathbf{x}_{(i)}^k - \mathbf{y}_{(i)}^k.\end{aligned} \tag{3.226}$$

Define now $\mathbf{g}_i^k \triangleq \nabla f_i(\mathbf{x}_{(i)}^k)$, $\mathbf{g}^k = [\mathbf{g}_1^{k\,T}, \ldots, \mathbf{g}_I^{k\,T}]^T$, and $\mathbf{y}^k = [\mathbf{y}_{(1)}^{k\,T}, \ldots, \mathbf{y}_{(I)}^{k\,T}]^T$; and recall the definitions of $\boldsymbol{\phi}^k$, $\boldsymbol{\Phi}^k$, $\mathbf{A}^k$, $\mathbf{W}^k$ and $\widehat{\mathbf{W}}^k$ introduced in Sect. 3.4.2.5. Using (3.226), Algorithm 10 under the ATC or CAA updates can be rewritten in vector/matrix form as: for all $k \in \mathbb{N}_+$,

$$\begin{aligned}
\boldsymbol{\phi}^{k+1} &= \mathbf{A}^k\, \boldsymbol{\phi}^k, \\
\mathbf{W}^k &= (\boldsymbol{\Phi}^{k+1})^{-1}\mathbf{A}^k\, \boldsymbol{\Phi}^k, \\
\mathbf{x}^{k+1} &= \begin{cases} \widehat{\mathbf{W}}^k \left(\mathbf{x}^k - \gamma^k\, \mathbf{y}^k\right), & \text{if ATC is employed;} \\ \widehat{\mathbf{W}}^k \mathbf{x}^k - \gamma^k\, (\widehat{\boldsymbol{\Phi}}^{k+1})^{-1}\widehat{\boldsymbol{\Phi}}^k \mathbf{y}^k, & \text{if CAA is employed;} \end{cases} \\
\mathbf{y}^{k+1} &= \begin{cases} \widehat{\mathbf{W}}^k\, \left(\mathbf{y}^k + (\widehat{\boldsymbol{\Phi}}^k)^{-1}(\mathbf{g}^{k+1} - \mathbf{g}^k)\right), & \text{if ATC is employed;} \\ \widehat{\mathbf{W}}^k\, \mathbf{y}^k + (\widehat{\boldsymbol{\Phi}}^{k+1})^{-1}\left(\mathbf{g}^{k+1} - \mathbf{g}^k\right), & \text{if CAA is employed;} \end{cases}
\end{aligned} \tag{3.227}$$

which we will refer to as *ATC/CAA-SONATA-L* (L stands for “linearized”).

In the special case where all $\mathbf{A}^k$ are doubly-stochastic matrices, we have $\mathbf{W}^k = \mathbf{A}^k$ and $\widehat{\mathbf{W}}^k \triangleq \mathbf{A}^k \otimes \mathbf{I}_m$; ATC/CAA-SONATA-L reduces to

$$\begin{aligned}
\mathbf{x}^{k+1} &= \begin{cases} \widehat{\mathbf{W}}^k \left(\mathbf{x}^k - \gamma^k\, \mathbf{y}^k\right), & \text{for the ATC update;} \\ \widehat{\mathbf{W}}^k \mathbf{x}^k - \gamma^k\, \mathbf{y}^k, & \text{for the CAA update;} \end{cases} \\
\mathbf{y}^{k+1} &= \begin{cases} \widehat{\mathbf{W}}^k\, \left(\mathbf{y}^k + \mathbf{g}^{k+1} - \mathbf{g}^k\right), & \text{for the ATC update;} \\ \widehat{\mathbf{W}}^k\, \mathbf{y}^k + \left(\mathbf{g}^{k+1} - \mathbf{g}^k\right), & \text{for the CAA update;} \end{cases}
\end{aligned} \tag{3.228}$$

which is referred to as *ATC/CAA-NEXT-L* (because the algorithm becomes an instance of NEXT [70–72]).

Connection with Current Algorithms

We are now in the position to show that the algorithms in [172, 191, 257, 259] are all special cases of SONATA and the earlier proposal NEXT [70–72].

Aug-DGM [259] and Algorithm in [191] Introduced in [259] for *undirected, time-invariant* graphs, the Aug-DGM algorithm reads

$$\begin{aligned}
\mathbf{x}^{k+1} &= \widehat{\mathbf{W}} \left(\mathbf{x}^k - \mathrm{Diag}\,(\boldsymbol{\gamma} \otimes \mathbf{1}_m)\, \mathbf{y}^k\right), \\
\mathbf{y}^{k+1} &= \widehat{\mathbf{W}} \left(\mathbf{y}^k + \mathbf{g}^{k+1} - \mathbf{g}^k\right),
\end{aligned} \tag{3.229}$$

where $\widehat{\mathbf{W}} \triangleq \mathbf{W} \otimes \mathbf{I}_m$, $\mathbf{W}$ is a doubly-stochastic matrix compliant with the graph G (cf. Assumption 4.3), and $\boldsymbol{\gamma} \triangleq (\gamma_i)_{i=1}^I$ is the vector of agents’ step-sizes.

A similar algorithm was proposed independently in [191] (in the same network setting of [259]), which reads

$$\begin{aligned} \mathbf{x}^{k+1} &= \widehat{\mathbf{W}}\left(\mathbf{x}^k - \gamma\,\mathbf{y}^k\right), \\ \mathbf{y}^{k+1} &= \widehat{\mathbf{W}}\mathbf{y}^k + \mathbf{g}^{k+1} - \mathbf{g}^k. \end{aligned} \tag{3.230}$$

Clearly Aug-DGM [259] in (3.229) and Algorithm [191] in (3.230) coincide with ATC-NEXT-L [cf. (3.228)].

(Push-)DIGing [172] Appeared in [172] and applicable to *time-varying undirected graphs*, the DIGing Algorithm reads

$$\begin{aligned} \mathbf{x}^{k+1} &= \widehat{\mathbf{W}}^k\,\mathbf{x}^k - \gamma\,\mathbf{y}^k, \\ \mathbf{y}^{k+1} &= \widehat{\mathbf{W}}^k\,\mathbf{y}^k + \mathbf{g}^{k+1} - \mathbf{g}^k, \end{aligned} \tag{3.231}$$

where $\widehat{\mathbf{W}}^k \triangleq \mathbf{W}^k \otimes \mathbf{I}_m$ and $\mathbf{W}^k$ is a doubly-stochastic matrix compliant with the graph G^k. Clearly, DIGing coincides with CAA-NEXT-L [cf. (3.228)], earlier proposed in [70–72].

The push-DIGing algorithm [172] extends DIGing to time-varying *digraphs*, and it is an instance of ATC-SONATA-L [cf. (3.227)], with $a_{ij}^k = 1/d_j^k$, $i, j = 1, \ldots I$.

ADD-OPT [257] Finally, we mention the ADD-OPT algorithm, proposed in [257] for *static digraphs*, which takes the following form:

$$\begin{aligned} \mathbf{z}^{k+1} &= \widehat{\mathbf{A}}\,\mathbf{z}^k - \gamma\,\widetilde{\mathbf{y}}^k, \\ \boldsymbol{\phi}^{k+1} &= \mathbf{A}\,\boldsymbol{\phi}^k, \\ \mathbf{x}^{k+1} &= (\widehat{\boldsymbol{\Phi}}^{k+1})^{-1}\,\mathbf{z}^{k+1}, \\ \widetilde{\mathbf{y}}^{k+1} &= \widehat{\mathbf{A}}\,\widetilde{\mathbf{y}}^k + \mathbf{g}^{k+1} - \mathbf{g}^k. \end{aligned} \tag{3.232}$$

Introducing the transformation $\mathbf{y}^k = (\widehat{\boldsymbol{\Phi}}^k)^{-1}\widetilde{\mathbf{y}}^k$, it is not difficult to check that (3.232) can be rewritten as

$$\begin{aligned} \boldsymbol{\phi}^{k+1} &= \mathbf{A}\,\boldsymbol{\phi}^k, \\ \mathbf{W}^k &= (\boldsymbol{\Phi}^{k+1})^{-1}\,\mathbf{A}\,\boldsymbol{\Phi}^k, \\ \mathbf{x}^{k+1} &= \widehat{\mathbf{W}}^k\,\mathbf{x}^k - \gamma\,(\widehat{\boldsymbol{\Phi}}^{k+1})^{-1}\,\widehat{\boldsymbol{\Phi}}^k\,\mathbf{y}^k, \\ \mathbf{y}^{k+1} &= \widehat{\mathbf{W}}^k\,\mathbf{y}^k + (\widehat{\boldsymbol{\Phi}}^{k+1})^{-1}\left(\mathbf{g}^{k+1} - \mathbf{g}^k\right), \end{aligned} \tag{3.233}$$

where $\widehat{\mathbf{W}}^k \triangleq \mathbf{W}^k \otimes \mathbf{I}_m$. Comparing (3.227) with (3.233), one can readily see that ADD-OPT coincides with CAA-SONATA-L.

Table 3.7 Connection of SONATA [208, 232] with current algorithms employing gradient tracking

Algorithms	Connection with SONATA	Instance of problem (3.167)	Graph topology/Weight matrix
NEXT [70, 72]	Special case of SONATA	F nonconvex $G \neq 0$ $X \subseteq \mathbb{R}^m$	Time-varying doubly-stochasticable digraph
Aug-DGM [191, 259]	ATC-NEXT-L ($\boldsymbol{\gamma} = \gamma \mathbf{1}_I$) (3.228)	F convex $G = 0$ $X = \mathbb{R}^m$	Static undirected graph
DIGing [172]	CAA-NEXT-L (3.228)	F convex $G = 0$ $X = \mathbb{R}^m$	Time-varying doubly-stochasticable digraph
push-DIGing [172]	ATC-SONATA-L (3.227)	F convex $G = 0$ $X = \mathbb{R}^m$	Time-varying digraph
ADD-OPT [257]	CAA-SONATA-L (3.227)	F convex $G = 0$ $X = \mathbb{R}^m$	Static digraph

We summarize the connections between the different versions of SONATA(-NEXT) and its special cases in Table 3.7.

3.4.3.2 Proof of Theorem 4.16

The proof of Theorem 4.16 is quite involved and can be found in [208]. Here we provide a simplified version, under the extra Assumption 4.18 on Problem (3.167) (stated below) and the use of a square-summable (and thus diminishing) step-size in the algorithm.

Assumption 4.18 *Given Problem* (3.167)*, in addition to Assumption 4.1, suppose that*

1. *The gradient of F is bounded on X, i.e., there exists a constant $0 < L_F < +\infty$ such that $\|\nabla F(\mathbf{x})\| \leq L_F$, $\forall \mathbf{x} \in X$;*
2. *The subgradient of G is bounded on X, i.e., there exists a constant $0 < L_G < +\infty$ such that $\|\partial G(\mathbf{x})\| \leq L_G$, $\forall \mathbf{x} \in X$.*

Assumption 4.19 *The step size $\gamma^k \in (0, 1]$ satisfies the diminishing rule: $\sum_{k=0}^{\infty} \gamma^k = +\infty$ and $\sum_{k=0}^{\infty} (\gamma^k)^2 < +\infty$.*

Next, we prove separately

$$\lim_{k \to \infty} D(\mathbf{x}^k) = 0, \tag{3.234}$$

and

$$\lim_{k\to\infty} J(\mathbf{x}^k) = 0, \tag{3.235}$$

which imply (3.223).

Technical Preliminaries and Sketch of the Proof
We introduce here some preliminary definitions and results along with a sketch of the proof of the theorem.

Weighted Averages $\bar{\mathbf{x}}_{\phi}^k$ and $\bar{\mathbf{y}}_{\phi}^k$ Define the weighted averages for the local copies $\mathbf{x}_{(i)}$ and the tracking variables $\mathbf{y}_{(i)}$:

$$\bar{\mathbf{x}}_{\phi}^k \triangleq \frac{1}{I}\sum_{i=1}^{I}\phi_{(i)}^k \mathbf{x}_{(i)}^k \quad \text{and} \quad \bar{\mathbf{y}}_{\phi}^k \triangleq \frac{1}{I}\sum_{i=1}^{I}\phi_{(i)}^k \mathbf{y}_{(i)}^k. \tag{3.236}$$

Using (3.218), (3.220) and (3.221), the dynamics of $\{\bar{\mathbf{x}}_{\phi}^k\}_{k\in\mathbb{N}_+}$ and $\{\bar{\mathbf{y}}_{\phi}^k\}_{k\in\mathbb{N}_+}$ generated by Algorithm 10 read: for all $k \in \mathbb{N}_+$,

$$\bar{\mathbf{x}}_{\phi}^{k+1} = \bar{\mathbf{x}}_{\phi}^k + \frac{\gamma^k}{I}\sum_{i=1}^{I}\phi_{(i)}^k\left(\widetilde{\mathbf{x}}_i^k - \bar{\mathbf{x}}_{\phi}^k\right), \tag{3.237}$$

and

$$\bar{\mathbf{y}}_{\phi}^{k+1} = \bar{\mathbf{y}}_{\phi}^k + \frac{1}{I}\sum_{i=1}^{I}\left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^k\right), \quad \text{with} \quad \mathbf{u}_i^k \triangleq \nabla f_i(\mathbf{x}_{(i)}^k), \tag{3.238}$$

respectively. Let $\mathbf{u}^k \triangleq (\mathbf{u}_i^k)_{i=1}^I$. Note that, since each $\mathbf{y}_{(i)}^0 = \mathbf{u}_i^0 = \nabla f_i(\mathbf{x}_{(i)}^0)$ and $\phi_i^0 = 1$, we have [cf. Theorem 4.10(a)]: for all $k \in \mathbb{N}_+$,

$$\bar{\mathbf{y}}_{\phi}^k = \frac{1}{I}\sum_{i=1}^{I}\mathbf{u}_i^k. \tag{3.239}$$

The average quantities $\bar{\mathbf{x}}_{\phi}^k$ and $\bar{\mathbf{y}}_{\phi}^k$ will play a key role in proving asymptotic consensus and tracking. In fact, by (3.239), tracking is asymptotically achieved if $\lim_{k\to\infty}\|\mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_{\phi}^k\| = 0$, for all $i = 1, \ldots, I$; and by

$$\left\|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}^k\right\| \leq \left\|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_{\phi}^k\right\| + \left\|\frac{1}{I}\sum_{j=1}^{I}\left(\mathbf{x}_{(j)}^k - \bar{\mathbf{x}}_{\phi}^k\right)\right\| \leq B_1\sum_{j=1}^{I}\left\|\mathbf{x}_{(j)}^k - \bar{\mathbf{x}}_{\phi}^k\right\|, \quad i = 1, \ldots I, \tag{3.240}$$

with B_1 being a finite positive constant, it follows that consensus is asymptotically reached if $\lim_{k\to\infty}\|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_\phi^k\| = 0$, for all $i = 1,\ldots,I$. These facts will will be proved in Step 1 of the proof, implying (3.234).

Properties of the Best-Response $\widetilde{\mathbf{x}}_i^k$ and Associated Quantities We study here the connection between agents' best-responses $\widetilde{\mathbf{x}}_i^k$, defined in (3.217), and the "ideal" best-response $\widehat{\mathbf{x}}_i(\mathbf{x}_{(i)}^k)$ defined in (3.212) (not computable locally by the agents), with $\widetilde{F}_i$ given by (3.214), which we rewrite here for convenience: given $\mathbf{z} \in X$,

$$\widehat{\mathbf{x}}_i(\mathbf{z}) \triangleq \underset{\mathbf{x}_{(i)}\in X}{\operatorname{argmin}}\ \widetilde{f}_i\big(\mathbf{x}_{(i)} \mid \mathbf{z}\big) + \big(\nabla F(\mathbf{z}) - \nabla f_i(\mathbf{z})\big)^T\big(\mathbf{x}_{(i)} - \mathbf{z}\big) + G\left(\mathbf{x}_{(i)}\right). \tag{3.241}$$

As observed in Sect. 3.4.3, $\widetilde{\mathbf{x}}_i^k$ can be interpreted as a locally computable proxy of $\widehat{\mathbf{x}}_i(\mathbf{x}_{(i)}^k)$. We establish next the following connection among $\widetilde{\mathbf{x}}_i^k$, $\widehat{\mathbf{x}}_i(\bullet)$, and the stationary solutions of Problem (3.167): i) every fixed point of $\widehat{\mathbf{x}}_i(\bullet)$ is a stationary solution of Problem (3.167) (cf. Lemma 4.20); and ii) the distance between these two mappings, $\|\widetilde{\mathbf{x}}_i^k - \widehat{\mathbf{x}}_i(\mathbf{x}_{(i)}^k)\|$, asymptotically vanishes, if consensus and tracking are achieved (cf. Lemma 4.21). This also establishes the desired link between the limit points of $\mathbf{x}_i^k$ and the fixed points of $\widehat{\mathbf{x}}_i(\bullet)$ [and thus the stationary solutions of (3.167)].

Lemma 4.20 *In the setting of Theorem 4.16, the best-response map $X \ni \mathbf{z} \mapsto \widehat{\mathbf{x}}_i(\mathbf{z})$, defined in (3.241), with $i = 1,\ldots I$, enjoys the following properties:*

(a) $\widehat{\mathbf{x}}_i(\bullet)$ is $\hat{L}_i$-Lipschitz continuous on X;
(b) The set of fixed points of $\widehat{\mathbf{x}}_i(\bullet)$ coincides with the set of stationary solutions of Problem (3.167). Therefore, $\widehat{\mathbf{x}}_i(\bullet)$ has a fixed point.

Proof The proof follows the same steps of that of Lemma 3.4 and Lemma 3.5 (Lecture II), and thus is omitted. □

Lemma 4.21 *Let $\{\mathbf{x}^k = (\mathbf{x}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ and $\{\mathbf{y}^k = (\mathbf{y}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$, be the sequence generated by Algorithm 10. Given $\widehat{\mathbf{x}}_i(\bullet)$ and $\widetilde{\mathbf{x}}_i^k$ in the setting of Theorem 4.16, with $i = 1,\ldots,I$, the following holds: for every $k \in \mathbb{N}_+$ and $i = 1,\ldots,I$,*

$$\left\|\widetilde{\mathbf{x}}_i^k - \widehat{\mathbf{x}}_i\big(\mathbf{x}_{(i)}^k\big)\right\| \le B_2\left(\left\|\mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k\right\| + \sum_{j=1}^I\left\|\mathbf{x}_{(j)}^k - \bar{\mathbf{x}}_\phi^k\right\|\right), \tag{3.242}$$

where B_2 is some positive, finite, constant.

Proof For notational simplicity, let us define $\widehat{\mathbf{x}}_i^k \triangleq \widehat{\mathbf{x}}_i\big(\mathbf{x}_{(i)}^k\big)$. We will also use the following shorthand: $\pm a \triangleq +a - a$, with $a \in \mathbb{R}$. Invoking the first order optimality condition for $\widetilde{\mathbf{x}}_i^k$ and $\widehat{\mathbf{x}}_i^k$, we have

$$\big(\widehat{\mathbf{x}}_i^k - \widetilde{\mathbf{x}}_i^k\big)^T\left(\nabla\widetilde{f}_i(\widetilde{\mathbf{x}}_i^k \mid \mathbf{x}_{(i)}^k) + I\cdot\mathbf{y}_{(i)}^k - \nabla f_i(\mathbf{x}_{(i)}^k)\right) + G\big(\widehat{\mathbf{x}}_i^k\big) - G\big(\widetilde{\mathbf{x}}_i^k\big) \ge 0$$

and

$$\left(\widetilde{\mathbf{x}}_i^k - \widehat{\mathbf{x}}_i^k\right)^T \left(\nabla \widetilde{f}_i(\widehat{\mathbf{x}}_i^k \mid \mathbf{x}_{(i)}^k) + \nabla F\big(\mathbf{x}_{(i)}^k\big) - \nabla f_i\big(\mathbf{x}_{(i)}^k\big)\right) + G\big(\widetilde{\mathbf{x}}_i^k\big) - G\big(\widehat{\mathbf{x}}_i\big) \geq 0,$$

respectively. Summing the two inequalities and using the strongly convexity of $\widetilde{f}_i(\bullet \mid \mathbf{x}_{(i)}^k)$ [cf. Assumption 4.14.1], leads to the desired result

$$\begin{aligned}\tau_i \left\|\widetilde{\mathbf{x}}_i^k - \widehat{\mathbf{x}}_i^k\right\| &\leq \left\|\nabla F\big(\mathbf{x}_{(i)}^k\big) - I \cdot \mathbf{y}_{(i)}^k \pm I \cdot \bar{\mathbf{y}}_\phi^k\right\| \\ &\overset{(3.239),\, A.4.1.2}{\leq} I \cdot \left\|\mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k\right\| + \sum_{j=1}^{I} L_j \left\|\mathbf{x}_{(i)}^k - \mathbf{x}_{(j)}^k\right\|.\end{aligned}$$

□

Structure of the Proof The rest of the proof is organized in the following steps:

- **Step 1:** We first study the dynamics of the consensus and tracking errors, proving, among other results, that asymptotic consensus and tracking are achieved, that is, $\lim_{k\to\infty} \|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_\phi^k\| = 0$ and $\lim_{k\to\infty} \|\mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k\| = 0$, for all $i = 1, \ldots, I$. By (3.240), this also proves (3.234);
- **Step 2:** We proceed studying the descent properties of $\{V(\bar{\mathbf{x}}_\phi^k)\}_{k\in\mathbb{N}_+}$, from which we will infer $\lim_{k\to\infty} \|\widetilde{\mathbf{x}}_i^k - \bar{\mathbf{x}}_\phi^k\| = 0$, for all $i = 1, \ldots, I$;
- **Step 3:** Finally, using the above results, we prove (3.235).

Step 1: Asymptotic Consensus and Tracking

We begin observing that the dynamics $\{\mathbf{x}^k \triangleq (\mathbf{x}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ [cf. (3.220)] and $\{\mathbf{y}^k \triangleq (\mathbf{y}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ [cf. (3.221)] generated by Algorithm 10 are instances of the perturbed condensed push-sum protocol (3.188), with errors

$$\boldsymbol{\varepsilon}_i^k = \gamma^k (\phi_{(i)}^{k+1})^{-1} \sum_{j=1}^{I} a_{ij}^k\, \phi_{(j)}^k \left(\widetilde{\mathbf{x}}_j^k - \mathbf{x}_{(j)}^k\right) \tag{3.243}$$

and

$$\boldsymbol{\varepsilon}_i^k = (\phi_{(i)}^{k+1})^{-1} \left(\mathbf{u}_i^{k+1} - \mathbf{u}_i^k\right), \tag{3.244}$$

respectively. We can then leverage the convergence results introduced in Sect. 3.4.2.4 to prove the desired asymptotic consensus and tracking. To do so, we first show that some related quantities are bounded.

Lemma 4.22 *Let* $\{\mathbf{x}^k = (\mathbf{x}^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$, $\{\mathbf{y}^k = (\mathbf{y}^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$, *and* $\{\boldsymbol{\phi}^k = (\phi^k_{(i)})_{i=1}^I\}_{k\in\mathbb{N}_+}$ *be the sequence generated by Algorithm 10, in the setting of Theorem 4.16 and under the extra Assumptions 4.18–4.19. Then, the following hold: for all* $i = 1, \ldots, I$, *(a)* $\{\boldsymbol{\phi}^k\}_{k\in\mathbb{N}_+}$ *is uniformly bounded:*

$$\phi_{lb} \cdot \mathbf{1} \leq \boldsymbol{\phi}^k \leq \phi_{ub} \cdot \mathbf{1}, \quad \forall k \in \mathbb{N}_+, \tag{3.245}$$

where ϕ_{lb} *and* ϕ_{ub} *are finite, positive constants, defined in (3.189);*
(b)

$$\sup_{k\in\mathbb{N}_+} \left\| \mathbf{y}^k_{(i)} - \bar{\mathbf{y}}^k_{\phi} \right\| < \infty; \tag{3.246}$$

(c)

$$\sup_{k\in\mathbb{N}_+} \left\| \mathbf{x}^k_{(i)} - \widetilde{\mathbf{x}}^k_i \right\| < \infty. \tag{3.247}$$

Proof Statement (a) is a consequence of Theorem 4.11(a). Statement (b) follows readily from (3.192), observing that the errors $\boldsymbol{\varepsilon}^k_i = (\phi^{k+1}_{(i)})^{-1}(\mathbf{u}^{k+1}_i - \mathbf{u}^k_i)$ are all uniformly bounded, due to Assumption 4.18.1 and (3.245).

We prove now statement (c). Since $\widetilde{\mathbf{x}}^k_{(i)}$ is the unique optimal solution of Problem (3.217), invoking the first order optimality condition, we have

$$\left(\mathbf{x}^k_{(i)} - \widetilde{\mathbf{x}}^k_i\right)^T \left(\nabla \widetilde{f}_i(\widetilde{\mathbf{x}}^k_i \,|\, \mathbf{x}^k_{(i)}) + I \cdot \mathbf{y}^k_{(i)} - \nabla f_i(\mathbf{x}^k_{(i)}) + \boldsymbol{\xi}^k\right) \geq 0,$$

where $\boldsymbol{\xi}^k \in \partial G(\widetilde{\mathbf{x}}^k_i)$, and $\|\boldsymbol{\xi}^k\| \leq L_G$ (cf. Assumption 4.18.2). Since $\widetilde{f}_i\left(\bullet \,|\, \mathbf{x}^k_i\right)$ is τ_i-strongly convex (cf. Assumption 4.14), we have

$$\begin{aligned} \left\| \mathbf{x}^k_{(i)} - \widetilde{\mathbf{x}}^k_i \right\| &\leq \frac{I}{\tau_i} \left\| \mathbf{y}^k_{(i)} \right\| + \frac{L_G}{\tau_i} \\ &\leq \frac{I}{\tau_i} \left\| \mathbf{y}^k_{(i)} - \bar{\mathbf{y}}^k_{\phi} \right\| + \left\| \bar{\mathbf{y}}^k_{\phi} \right\| + \frac{L_G}{\tau_i} \\ &\overset{(3.239),(3.246)}{\leq} B_3 + \frac{1}{I} \sum_{i=1}^I \left\| \mathbf{u}^k_i \right\| \leq B_4, \end{aligned} \tag{3.248}$$

for all $k \in \mathbb{N}_+$, where the last inequality follows from Assumption 4.18.1, and B_3 and B_4 are some positive, finite, constants. □

We can now study the dynamics of the consensus error.

Proposition 4.23 *Let $\{\mathbf{x}^k = (\mathbf{x}_{(i)}^k)_{i=1}^I\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm 10, in the setting of Theorem 4.16, and under the extra Assumptions 4.18–4.19; and let $\{\bar{\mathbf{x}}_\phi^k\}_{k\in\mathbb{N}_+}$, with $\bar{\mathbf{x}}_\phi^k$ defined in (3.237). Then, all the $\mathbf{x}_{(i)}^k$ are asymptotically consensual, that is,*

$$\lim_{k\to\infty} \left\| \mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_\phi^k \right\| = 0, \quad \forall i = 1, \ldots, I. \tag{3.249}$$

Furthermore, there hold: for all $i = 1, \ldots, I$,

$$\sum_{k=0}^{\infty} \gamma^k \left\| \mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_\phi^k \right\| < \infty, \tag{3.250}$$

$$\sum_{k=0}^{\infty} \left\| \mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_\phi^k \right\|^2 < \infty. \tag{3.251}$$

Proof We use again the connection between (3.220) and the perturbed condensed push-sum protocol (3.188), with error $\boldsymbol{\varepsilon}_i^k$ given by (3.243). Invoking Theorem 4.11 and Lemma 4.12(a) [cf. (3.191)], to prove (3.249), it is then sufficient to show that all the errors $\boldsymbol{\varepsilon}_i^k$ are asymptotically vanishing. This follows readily from the following facts: $\gamma^k \downarrow 0$ (Assumption 4.19); $0 < \phi_{lb} \leq \phi_{(i)}^k \leq \phi_{ub} < \infty$, for all $k \in \mathbb{N}_+$ and $i = 1, \ldots, I$ [cf. Theorem 4.11(a)]; and $\sup_{k\in\mathbb{N}_+} \|\mathbf{x}_{(i)}^k - \widetilde{\mathbf{x}}_i^k\| < \infty$ [Lemma 4.22(c)].

We prove now (3.250). Invoking Theorem 4.11(b), we can write

$$\begin{aligned}\lim_{k\to\infty} \sum_{t=1}^{k} \gamma^t \left\| \mathbf{x}_{(i)}^t - \bar{\mathbf{x}}_\phi^t \right\| &\overset{(3.190)}{\leq} c \cdot \lim_{k\to\infty} \sum_{t=1}^{k} \gamma^t \left((\rho)^t \|\mathbf{x}^0\| + \sum_{s=0}^{t-1} (\rho)^{t-1-s} \|\boldsymbol{\varepsilon}^s\| \right) \\ &\overset{(a)}{\leq} B_5 \cdot \lim_{k\to\infty} \sum_{t=1}^{k} (\rho)^t + B_6 \cdot \underbrace{\lim_{k\to\infty} \sum_{t=1}^{k} \sum_{s=0}^{t-1} (\rho)^{t-1-s} \gamma^s \gamma^t}_{<\infty \quad [\text{Lemma } 4.12(b2)]} < \infty,\end{aligned}$$

where B_5 and B_6 are some positive, finite, constants, and in (a) we used $\|\boldsymbol{\varepsilon}^t\| \leq B_7\, \gamma^t$, for some positive, finite B_7 [which follows from the same arguments used to prove (3.249)].

To prove (3.251), we can use similar steps and write:

$$\lim_{k\to\infty} \sum_{t=1}^{k} \left\| \mathbf{x}_{(i)}^t - \bar{\mathbf{x}}_\phi^t \right\|^2 \leq \lim_{k\to\infty} \sum_{t=1}^{k} \left(B_5\, (\rho)^t + B_6 \cdot \sum_{s=0}^{t-1} (\rho)^{t-1-s} \gamma^s \right)^2$$

$$\leq 2 \cdot B_5^2 \lim_{k\to\infty} \sum_{t=0}^{k} (\rho)^{2\cdot t} + 2\, B_6^2 \cdot \lim_{k\to\infty} \sum_{t=0}^{k} \sum_{s=0}^{t-1} \sum_{s'=0}^{t-1} (\rho)^{t-1-s} (\rho)^{t-1-s'} \gamma^s \gamma^{s'} \overset{(a)}{<} \infty,$$

where in (a) we used

$$
\begin{aligned}
&\lim_{k\to\infty}\sum_{t=0}^{k}\sum_{s=0}^{t-1}\sum_{s'=0}^{t-1}(\rho)^{t-1-s}\,(\rho)^{t-1-s'}\,\gamma^{s}\,\gamma^{s'}\\
&\quad\le\lim_{k\to\infty}\sum_{t=0}^{k}\sum_{s=0}^{t-1}\sum_{s'=0}^{t-1}(\rho)^{t-1-s}\,(\rho)^{t-1-s'}\,(\gamma^{s})^{2}\\
&\quad=\lim_{k\to\infty}\sum_{t=0}^{k}\underbrace{\left(\sum_{s'=0}^{t-1}(\rho)^{t-1-s'}\right)}_{\le\frac{1}{1-\rho}}\sum_{s=0}^{t-1}(\rho)^{t-1-s}\,(\gamma^{s})^{2}<\infty,
\end{aligned}
$$

where the last implication is a consequence of Lemma 4.12(b1). □

We prove next a similar result for the tracking error.

Proposition 4.24 *Let $\{\mathbf{y}^k=(\mathbf{y}^k_{(i)})_{i=1}^{I}\}_{k\in\mathbb{N}_+}$ be the sequence generated by Algorithm 10, in the setting of Theorem 4.16, and under the extra Assumptions 4.18–4.19; and let $\{\bar{\mathbf{y}}^k_{\phi}\}_{k\in\mathbb{N}_+}$, with $\bar{\mathbf{y}}^k_{\phi}$ defined in (3.238). Then, all the $\mathbf{y}^k_{(i)}$ asymptotically track the gradient average $(1/I)\cdot\sum_{i=1}^{I}\nabla f_i(\mathbf{x}^k_{(i)})$, that is,*

$$
\lim_{k\to\infty}\left\|\mathbf{y}^k_{(i)}-\bar{\mathbf{y}}^k_{\phi}\right\|=0,\quad\forall i=1,\ldots,I.\tag{3.252}
$$

Furthermore, there holds

$$
\sum_{k=0}^{\infty}\gamma^k\left\|\mathbf{y}^k_{(i)}-\bar{\mathbf{y}}^k_{\phi}\right\|<\infty,\quad\forall i=1,\ldots,I.\tag{3.253}
$$

Proof Invoking Theorem 4.10 and using (3.239), to prove (3.252), it is sufficient to show that Assumption 4.9 holds. By Assumption 4.1.2, this reduces to prove $\lim_{k\to\infty}\|\mathbf{x}^{k+1}_{(i)}-\mathbf{x}^k_{(i)}\|=0$, for all $i=1,\ldots,I$. We write:

$$
\begin{aligned}
\lim_{k\to\infty}\left\|\mathbf{x}^{k+1}_{(i)}-\mathbf{x}^k_{(i)}\right\|\le{}&\underbrace{\lim_{k\to\infty}\left\|\mathbf{x}^{k+1}_{(i)}-\bar{\mathbf{x}}^{k+1}_{\phi}\right\|}_{\overset{(3.249)}{=}0}+\underbrace{\lim_{k\to\infty}\left\|\mathbf{x}^{k}_{(i)}-\bar{\mathbf{x}}^{k}_{\phi}\right\|}_{\overset{(3.249)}{=}0}\\
&+\lim_{k\to\infty}\underbrace{\left\|\frac{\gamma^k}{I}\sum_{i=1}^{I}\phi^k_{(i)}\left(\widetilde{\mathbf{x}}^k_i-\bar{\mathbf{x}}^k_{\phi}\right)\right\|}_{\overset{(3.247)}{\le}\gamma^k\cdot B_8}=0,
\end{aligned}\tag{3.254}
$$

where B_8 is some positive, finite, constant.

We prove next (3.253). Invoking Theorem 4.11(b), with $\boldsymbol{\varepsilon}_i^t$ defined in (3.244), we have

$$\begin{aligned}
&\lim_{k\to\infty}\sum_{t=1}^{k}\gamma^t\left\|\mathbf{y}_{(i)}^t-\bar{\mathbf{y}}_\phi^t\right\| \\
&\overset{(3.190)}{\leq} c\cdot\lim_{k\to\infty}\sum_{t=1}^{k}\gamma^t\left((\rho)^t\,\|\mathbf{u}^0\|+\sum_{s=0}^{t-1}(\rho)^{t-1-s}\,\|\boldsymbol{\varepsilon}^s\|\right) \\
&\overset{(a)}{\leq} B_9\cdot\lim_{k\to\infty}\sum_{t=1}^{k}(\rho)^t+B_{10}\cdot\lim_{k\to\infty}\sum_{i=1}^{I}\sum_{t=1}^{k}\sum_{s=0}^{t-1}(\rho)^{t-1-s}\,\gamma^t\left\|\mathbf{x}_{(i)}^{s+1}-\mathbf{x}_{(i)}^{s}\right\| \\
&\overset{(b)}{\leq} B_9\cdot\lim_{k\to\infty}\sum_{t=1}^{k}(\rho)^t+B_{10}\cdot B_8\cdot I\cdot\underbrace{\lim_{k\to\infty}\sum_{t=1}^{k}\sum_{s=0}^{t-1}(\rho)^{t-1-s}\,\gamma^t\,\gamma^s}_{<\infty\quad[\text{Lemma 4.12}(b2)]} \\
&\quad+B_{10}\cdot\underbrace{\lim_{k\to\infty}\sum_{i=1}^{I}\sum_{t=1}^{k}\sum_{s=0}^{t-1}(\rho)^{t-1-s}\,\gamma^t\left(\left\|\mathbf{x}_{(i)}^{s+1}-\bar{\mathbf{x}}_\phi^{s+1}\right\|+\left\|\mathbf{x}_{(i)}^{s}-\bar{\mathbf{x}}_\phi^{s}\right\|\right)}_{<\infty\quad[(3.251)\ \&\ \text{Lemma 4.12}(b2)]}<\infty,
\end{aligned}$$

where B_9 and B_{10} are some positive, finite, constants; in (a) we used the definition of $\boldsymbol{\varepsilon}_i^t$ [cf. (3.244)] along with Assumption 4.1.2; and in (b) we invoked the same upper bound used in (3.254).

□

Step 2: Descent of $\{V(\bar{\mathbf{x}}_\phi^k)\}_{k\in\mathbb{N}_+}$

We study now the descent of V along $\{\bar{\mathbf{x}}_\phi^k\}_{k\in\mathbb{N}_+}$. For notational simplicity, define

$$\Delta\mathbf{x}_{(i)}^k\triangleq\mathbf{x}_{(i)}^k-\bar{\mathbf{x}}_\phi^k\quad\text{and}\quad\Delta\widetilde{\mathbf{x}}_{(i)}^k\triangleq\widetilde{\mathbf{x}}_i^k-\bar{\mathbf{x}}_\phi^k,\quad i=1,\ldots,I,\tag{3.255}$$

where we recall that $\widetilde{\mathbf{x}}_i^k$ is defined in (3.217). Invoking the first order optimality condition of $\widetilde{\mathbf{x}}_i^k$, it is not difficult to check that the following property holds for $\Delta\widetilde{\mathbf{x}}_{(i)}^k$:

$$\left(\Delta\widetilde{\mathbf{x}}_{(i)}^k\right)^T\left(\nabla\widetilde{f}_i\big(\bar{\mathbf{x}}_\phi^k\mid\mathbf{x}_{(i)}^k\big)-\nabla f_i\big(\mathbf{x}_{(i)}^k\big)+I\cdot\mathbf{y}_{(i)}^k\right)+G\big(\widetilde{\mathbf{x}}_i^k\big)-G\big(\bar{\mathbf{x}}_\phi^k\big)\leq-\tau_i\left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|^2.\tag{3.256}$$

We begin studying the dynamics of F along $\{\bar{\mathbf{x}}_\phi^k\}_{k\in\mathbb{N}_+}$. Using (3.237) and invoking the descent lemma [cf. Lemma 3.14], with $L \triangleq \sum_{i=1}^I L_i$, we have:

$$\begin{aligned}
&F\big(\bar{\mathbf{x}}_\phi^{k+1}\big)\\
&\le F\big(\bar{\mathbf{x}}_\phi^{k}\big) + \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \nabla F\big(\bar{\mathbf{x}}_\phi^{k}\big)^T \Delta\widetilde{\mathbf{x}}_{(i)}^k + \frac{L}{2}\frac{(\gamma^k)^2}{I}\sum_{i=1}^I (\phi_{(i)}^k)^2 \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|^2\\
&\overset{(3.256)}{\le} F\big(\bar{\mathbf{x}}_\phi^{k}\big) - \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(\tau_i \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|^2 + G\big(\widetilde{\mathbf{x}}_{(i)}^k\big) - G\big(\bar{\mathbf{x}}_\phi^{k}\big)\right)\\
&\qquad + \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(\nabla f_i\big(\mathbf{x}_{(i)}^k\big) - \nabla \tilde{f}_i\big(\bar{\mathbf{x}}_\phi^{k} \mid \mathbf{x}_{(i)}^k\big) \pm \nabla \tilde{f}_i\big(\bar{\mathbf{x}}_\phi^{k} \mid \bar{\mathbf{x}}_\phi^{k}\big)\right)^T \Delta\widetilde{\mathbf{x}}_{(i)}^k\\
&\qquad + \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(\nabla F\big(\bar{\mathbf{x}}_\phi^{k}\big) - I\cdot \mathbf{y}_{(i)}^k \pm \big(I\cdot \bar{\mathbf{y}}_\phi^k\big)\right)^T \Delta\widetilde{\mathbf{x}}_{(i)}^k + \frac{L}{2}\frac{(\gamma^k)^2}{I}\sum_{i=1}^I (\phi_{(i)}^k)^2 \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|^2\\
&\overset{(a)}{\le} F\big(\bar{\mathbf{x}}_\phi^{k}\big) - \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(\tau_i \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|^2 + G\big(\widetilde{\mathbf{x}}_{(i)}^k\big) - G\big(\bar{\mathbf{x}}_\phi^{k}\big)\right)\\
&\qquad + \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(L_i + \tilde{L}_i\right) \left\|\Delta\mathbf{x}_{(i)}^k\right\| \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|\\
&\qquad + \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(\sum_{j=1}^I L_j \left\|\Delta\mathbf{x}_{(j)}^k\right\|\right) \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\| + \gamma^k \sum_{i=1}^I \phi_{(i)}^k \left\|\mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k\right\| \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|\\
&\qquad + \frac{L}{2}\frac{(\gamma^k)^2}{I}\sum_{i=1}^I (\phi_{(i)}^k)^2 \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\|^2,
\end{aligned} \tag{3.257}$$

where in (a) we used (3.239), Assumption 4.14.2, and Assumption 4.1.2.

Invoking the convexity of G and (3.237), we can write

$$G\big(\bar{\mathbf{x}}_\phi^{k+1}\big) \le G\big(\bar{\mathbf{x}}_\phi^{k}\big) - \frac{\gamma^k}{I}\sum_{i=1}^I \phi_{(i)}^k \left(G\big(\bar{\mathbf{x}}_\phi^{k}\big) - G\big(\widetilde{\mathbf{x}}_i^k\big)\right). \tag{3.258}$$

Substituting (3.258) in (3.257) and using (3.245) [cf. Lemma 4.22], $\tau_{\min} \triangleq \min_i \tau_i > 0$, and (3.249), we can write: for sufficiently large $k \in \mathbb{N}_+$,

$$\begin{aligned} V\big(\bar{\mathbf{x}}_\phi^{k+1}\big) &\le V\big(\bar{\mathbf{x}}_\phi^{k}\big) - \gamma^k \left(\frac{\tau_{\min}\,\phi_{lb}}{I} - \frac{L\,\phi_{ub}^2\,\gamma^k}{2\,I}\right) \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\|^2 \\ &\quad + B_{11}\cdot\gamma^k \sum_{i=1}^{I} \left(\left\| \Delta\mathbf{x}_{(i)}^k \right\| + \sum_{j=1}^{I} \left\| \Delta\mathbf{x}_{(j)}^k \right\| + \left\| \mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k \right\| \right) \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| \\ &\le V\big(\bar{\mathbf{x}}_\phi^{k}\big) - B_{12}\,\gamma^k \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\|^2 + \underbrace{B_{13}\cdot\gamma^k \sum_{i=1}^{I} \left(\left\| \mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k \right\| + \left\| \Delta\mathbf{x}_{(i)}^k \right\| \right)}_{\triangleq T^k} \end{aligned} \tag{3.259}$$

where B_{11}, B_{12}, B_{13} are some finite, positive, constants, and in the last inequality we used the fact that $\left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| \le B_{14}$, with $i = 1, \ldots, I$, for sufficiently large $k \in \mathbb{N}_+$ and some finite, positive, constant B_{14}, which is a direct consequence of (3.247) (cf. Lemma 4.22) and (3.249) (cf. Proposition 4.23).

Note that, $\sum_{k=0}^{\infty} T^k < \infty$, due to Proposition 4.23 [cf. (3.250)] and Proposition 4.24 [cf (3.253)]. We can then apply Lemma 3.16 to (3.259), with the following identifications:

$$Y^k = V\big(\bar{\mathbf{x}}_\phi^k\big), \quad Z^k = T^k, \quad \text{and} \quad W^k = B_{12}\,\gamma^k \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\|^2,$$

which, using Assumption 3.1.4, yields (i) $\lim_{k\to\infty} V\big(\bar{\mathbf{x}}_\phi^k\big) = V^\infty$, with V^∞ finite; and (ii)

$$\sum_{k=0}^{\infty} \gamma^k \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\|^2 < \infty. \tag{3.260}$$

Since $\sum_{k=0}^{\infty} \gamma^k = \infty$, (3.260) implies $\liminf_{k\to\infty} \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| = 0$.

We prove next that $\limsup_{k\to\infty} \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| = 0$, for all $i = 1, \ldots, I$. Assume on the contrary that $\limsup_{k\to\infty} \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| > 0$. Since $\liminf_{k\to\infty} \sum_{i=1}^{I} \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| = 0$, there exists an infinite set $K \subseteq \mathbb{N}_+$, such that for all $k \in K$, one can find an integer $t_k > k$ such that

$$\sum_{i=1}^{I} \|\Delta\widetilde{\mathbf{x}}_{(i)}^k\| < \delta, \qquad \sum_{i=1}^{I} \|\Delta\widetilde{\mathbf{x}}_{(i)}^{t_k}\| > 2\delta, \tag{3.261}$$

$$\delta \le \sum_{i=1}^{I} \|\Delta\widetilde{\mathbf{x}}_{(i)}^j\| \le 2\delta, \qquad \text{if} \quad k < j < t_k. \tag{3.262}$$

Therefore, for all $k \in K$,

$$\begin{aligned}
\delta &< \sum_{i=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{t_k} \right\| - \sum_{i=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{k} \right\| \leq \sum_{i=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{t_k} - \varDelta\widetilde{\mathbf{x}}_{(i)}^{k} \pm \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^{k}\right) \pm \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^{t_k}\right) \right\| \\
&\overset{(a)}{\leq} \sum_{i=1}^{I} \left(1+\hat{L}_i\right) \left\| \bar{\mathbf{x}}_\phi^{t_k} - \bar{\mathbf{x}}_\phi^{k} \right\| + \underbrace{\sum_{i=1}^{I} \left\| \widetilde{\mathbf{x}}_{(i)}^{t_k} - \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^{t_k}\right) \right\| + \sum_{i=1}^{I} \left\| \widetilde{\mathbf{x}}_{(i)}^{k} - \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^{k}\right) \right\|}_{\triangleq e_1^k} \\
&\overset{(3.245)}{\leq} \sum_{i=1}^{I} \left(1+\hat{L}_i\right) \frac{\phi_{ub}}{I} \sum_{j=k}^{t_k-1} \gamma^j \sum_{\ell=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(\ell)}^{j} \right\| + e_1^k \\
&\leq \sum_{i=1}^{I} \left(1+\hat{L}_i\right) \frac{\phi_{ub}}{I} \sum_{j=k+1}^{t_k-1} \gamma^\tau \sum_{\ell=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(\ell)}^{j} \right\| + e_1^k + \underbrace{\sum_{i=1}^{I} \left(1+\hat{L}_i\right) \frac{\phi_{ub}}{I} \gamma^k \sum_{\ell=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(\ell)}^{k} \right\|}_{\triangleq e_2^k} \\
&\overset{(b)}{\leq} \sum_{i=1}^{I} \left(1+\hat{L}_i\right) \frac{\phi_{ub}}{\delta} \sum_{j=k+1}^{t_k-1} \gamma^j \sum_{\ell=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(\ell)}^{j} \right\|^2 + e_1^k + e_2^k,
\end{aligned} \tag{3.263}$$

where in (a) we used Lemma 4.20(a); and in (b) we used (3.262) and Lemma 3.15.

Note that

1. $\lim_{k\to\infty} e_1^k = 0$, due to Lemma 4.21 [cf. (3.242)], Proposition 4.23 [cf. (3.249)], and Proposition 4.24 [cf. (3.252)];
2. $\lim_{k\to\infty} e_2^k = 0$, due to (3.260), which implies $\lim_{k\to\infty} \gamma^k \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{k} \right\|^2 = 0$, and thus $\lim_{k\to\infty} \gamma^k \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{k} \right\| = 0$ (recall that $\gamma^k \in (0, 1]$); and
3. $$\lim_{k\to\infty} \sum_{j=k+1}^{t_k-1} \gamma^j \sum_{i=1}^{I} \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{j} \right\|^2 = 0,$$

   due to (3.260).

This however contradicts (3.263). Therefore, it must be

$$\lim_{k\to\infty} \left\| \varDelta\widetilde{\mathbf{x}}_{(i)}^{k} \right\| = 0, \qquad i = 1, \ldots, I. \tag{3.264}$$

Step 3: $\lim_{k\to\infty} J(\bar{\mathbf{x}}^k) = 0$
This part of the proof can be found in [208], and is reported below for completeness.

Recalling the definition of $J(\bar{\mathbf{x}}^k) = \|\bar{\mathbf{x}}^k - \bar{\mathbf{x}}(\bar{\mathbf{x}}^k)\|$ [cf.(3.222)], where for notational simplicity we introduced

$$\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) \triangleq \underset{\mathbf{y}\in X}{\operatorname{argmin}} \left\{ \nabla F(\bar{\mathbf{x}}^k)^T(\mathbf{y}-\bar{\mathbf{x}}^k) + \frac{1}{2}\|\mathbf{y}-\bar{\mathbf{x}}^k\|_2^2 + G(\mathbf{y}) \right\}, \tag{3.265}$$

we start bounding J as follows

$$J(\bar{\mathbf{x}}^k) = \left\| \bar{\mathbf{x}}^k - \bar{\mathbf{x}}(\bar{\mathbf{x}}^k) \pm \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) \right\| \le \underbrace{\left\| \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) - \bar{\mathbf{x}}^k \right\|}_{\text{term I}} + \underbrace{\left\| \bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) \right\|}_{\text{term II}}, \tag{3.266}$$

where $\widehat{\mathbf{x}}_i(\bullet)$ is defined in (3.241). To prove $\lim_{k\to\infty} J(\bar{\mathbf{x}}^k) = 0$, it is then sufficient to show that both Term I and Term II in (3.266) are asymptotically vanishing. We study the two terms separately.

1. `Term I`: We prove $\lim_{k\to\infty} \|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) - \bar{\mathbf{x}}^k\| = 0$.

   We begin bounding $\|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) - \bar{\mathbf{x}}^k\|$ as

$$\begin{aligned} \left\| \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) - \bar{\mathbf{x}}^k \right\| &= \left\| \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) \pm \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) \pm \bar{\mathbf{x}}_\phi^k - \bar{\mathbf{x}}^k \right\| \\ &\le \left\| \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) - \bar{\mathbf{x}}_\phi^k \right\| + (1+\hat{L}_i)\left\| \bar{\mathbf{x}}_\phi^k - \bar{\mathbf{x}}^k \right\|. \end{aligned} \tag{3.267}$$

By (3.240) and Proposition 4.23 [cf. (3.249)] it follows that

$$\lim_{k\to\infty} \left\| \bar{\mathbf{x}}_\phi^k - \bar{\mathbf{x}}^k \right\| = 0. \tag{3.268}$$

Therefore, to prove $\lim_{k\to\infty} \|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) - \bar{\mathbf{x}}^k\| = 0$, it is sufficient to show that $\|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) - \bar{\mathbf{x}}_\phi^k\|$ is asymptotically vanishing, as proved next.

We bound $\left\|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) - \bar{\mathbf{x}}_\phi^k\right\|$ as:

$$\begin{aligned} &\left\| \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) - \bar{\mathbf{x}}_\phi^k \pm \widetilde{\mathbf{x}}_{(i)}^k \pm \widehat{\mathbf{x}}_i\left(\mathbf{x}_{(i)}^k\right) \right\| \\ &\le \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| + \left\| \widehat{\mathbf{x}}_i\left(\mathbf{x}_{(i)}^k\right) - \widetilde{\mathbf{x}}_{(i)}^k \right\| + \left\| \widehat{\mathbf{x}}_i\left(\mathbf{x}_{(i)}^k\right) - \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) \right\| \\ &\le \left\| \Delta\widetilde{\mathbf{x}}_{(i)}^k \right\| + B_{15} \sum_{j=1}^{I} \left\| \mathbf{x}_{(j)}^k - \bar{\mathbf{x}}_\phi^k \right\| + B_{16} \left\| \mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k \right\|, \end{aligned} \tag{3.269}$$

where B_{15} and B_{16} are some positive, finite, constants; and in the last inequality we used Lemmas 4.20 and 4.21 [cf. (3.242)]. Invoking

1. $\lim_{k\to\infty} \left\|\Delta\widetilde{\mathbf{x}}_{(i)}^k\right\| = 0$ [cf. (3.264)];
2. $\lim_{k\to\infty} \left\|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}_\phi^k\right\| = 0$, for all $i = 1, \ldots, I$, [cf. (3.249)]; and
3. $\lim_{k\to\infty} \left\|\mathbf{y}_{(i)}^k - \bar{\mathbf{y}}_\phi^k\right\| = 0$, for all $i = 1, \ldots, I$, [cf. (3.252)],

the desired result, $\lim_{k\to\infty} \left\|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}_\phi^k\right) - \bar{\mathbf{x}}_\phi^k\right\| = 0$, follows readily. This together with (3.268) and (3.267) prove

$$\lim_{k\to\infty} \left\|\widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right) - \bar{\mathbf{x}}^k\right\| = 0. \tag{3.270}$$

- `Term II`: We prove $\lim_{k\to\infty} \|\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i\left(\bar{\mathbf{x}}^k\right)\| = 0$.

We begin deriving a proper upper bound of $\|\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\|$. By the first order optimality condition of $\bar{\mathbf{x}}(\bar{\mathbf{x}}^k)$ and $\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)$ we have

$$\begin{aligned}
&\left(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) - \bar{\mathbf{x}}(\bar{\mathbf{x}}^k)\right)^T \left(\nabla F(\bar{\mathbf{x}}^k) + \bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \bar{\mathbf{x}}^k\right) + G\left(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\right) - G\left(\bar{\mathbf{x}}(\bar{\mathbf{x}}^k)\right) \geq 0, \\
&\left(\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\right)^T \left(\nabla \widetilde{f}_i(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) \,|\, \bar{\mathbf{x}}^k) + \nabla F(\bar{\mathbf{x}}^k) - \nabla f_i(\bar{\mathbf{x}}^k)\right) \\
&\quad + G\left(\bar{\mathbf{x}}(\bar{\mathbf{x}}^k)\right) - G\left(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\right) \geq 0,
\end{aligned}$$

which yields

$$\begin{aligned}
\left\|\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\right\| &\leq \left\|\nabla \widetilde{f}_i(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) \,|\, \bar{\mathbf{x}}^k) - \nabla f_i(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) + \bar{\mathbf{x}}^k\right\| \\
&\leq \left\|\nabla \widetilde{f}_i(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) \,|\, \bar{\mathbf{x}}^k) \pm \nabla f_i(\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)) - \nabla f_i(\bar{\mathbf{x}}^k)\right\| + \left\|\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) - \bar{\mathbf{x}}^k\right\| \\
&\leq B_{17} \left\|\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) - \bar{\mathbf{x}}^k\right\|,
\end{aligned} \tag{3.271}$$

where B_{17} is some positive, finite, constant; and in the last inequality we used the Lipschitz continuity of $\nabla f_i(\bullet)$ (cf. Assumption 4.1) and $\nabla \widetilde{f}_i(\mathbf{x} \,|\, \bullet)$ (cf. Assumption 4.14).

Using (3.270) and (3.271), we finally have

$$\lim_{k\to\infty} \left\|\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\right\| = 0. \tag{3.272}$$

Therefore, we conclude

$$\lim_{k\to\infty} J(\bar{\mathbf{x}}^k) \overset{(3.266)}{\leq} \underbrace{\lim_{k\to\infty} \left\|\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k) - \bar{\mathbf{x}}^k\right\|}_{\overset{(3.270)}{=} 0} + \underbrace{\lim_{k\to\infty} \left\|\bar{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k)\right\|}_{\overset{(3.272)}{=} 0} = 0. \tag{3.273}$$

This completes the proof of Step 3, and also the proof of Theorem 4.16. □

3.4.4 Applications

In this section, we test the performance of SONATA (Algorithm 10) on both convex and nonconvex problems. More specifically, as convex instance of Problem (3.167), we consider a distributed linear regression problem (cf. Sect. 3.4.4.1) whereas as nonconvex instances of (3.167), we study a target localization problem (cf. Sect. 3.4.4.2). Other applications and numerical results can be found in [208, 228, 232].

3.4.4.1 Distributed Robust Regression

The distributed robust linear regression problem is an instance of the empirical risk minimization considered in Example 1 in Sect. 3.4.1.1: Agents in the network want to cooperatively estimate a common parameter $\mathbf{x}$ of a linear model from a set of distributed measures corrupted by noise and outliers; let $D_i \triangleq \{d_{i\,1}, \ldots, d_{i\,n_i}\}$ be the set of n_i measurements taken by agent i. To be robust to the heavy-tailed errors or outliers in the response, a widely explored approach is to use the Huber's criterion as loss function, which leads to the following formulation

$$\underset{\mathbf{x}}{\text{minimize}} \quad F(\mathbf{x}) \triangleq \sum_{i=1}^{I}\sum_{j=1}^{n_i} H\left(\mathbf{b}_{ij}^T\,\mathbf{x} - d_{i\,j}\right), \tag{3.274}$$

where $\mathbf{b}_{ij} \in \mathbb{R}^m$ is the vector of features (or predictors) associated with the response d_{ij}, owned by agent i, with $j = 1, \ldots, n_i$ and $i = 1, \ldots, I$; and $H : \mathbb{R} \to \mathbb{R}$ is the Huber loss function, defined as:

$$H\,(r) = \begin{cases} r^2, & \text{if } |r| \leq \alpha, \\ \alpha \cdot (2\,|r| - \alpha)\,, & \text{otherwise;} \end{cases}$$

for some given $\alpha > 0$. This function is quadratic for small values of the residual r (like the least square loss) but grows linearly (like the absolute distance loss) for large values of r. The cut-off parameter α describes where the transition from quadratic to linear takes place. Note that H is convex (but not strongly convex) and differentiable, with derivative:

$$H'(r) = \begin{cases} -2\,\alpha, & \text{if } r < -\alpha, \\ 2\,r, & \text{if } r \in [-\alpha, \alpha], \\ 2\,\alpha, & \text{if } r > \alpha. \end{cases} \tag{3.275}$$

Introducing

$$f_i\left(\mathbf{x}\right) \triangleq \sum_{j=1}^{n_i} H\left(\mathbf{b}_{ij}^T\,\mathbf{x} - d_{i\,j}\right),$$

(3.274) is clearly an instance of Problem (3.167), with $F = \sum_{i=1}^{I} f_i$ and $G = 0$. It is not difficult to check that Assumption 4.1 is satisfied.

We apply SONATA to Problem (3.274) considering two alternative choices of the surrogate functions $\widetilde{f}_i$. The first choice is the linear approximation of f_i (plus a proximal regularization): given the local copy $\mathbf{x}_{(i)}^k$,

$$\widetilde{f}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) = f_i\left(\mathbf{x}_{(i)}^k\right) + \nabla f_i\left(\mathbf{x}_{(i)}^k\right)^T\left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right) + \frac{\tau_i}{2}\left\|\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right\|^2; \tag{3.276}$$

with

$$\nabla f_i\left(\mathbf{x}_{(i)}^k\right) = \sum_{j=1}^{n_i} \mathbf{b}_{ij}\cdot H'\left(\mathbf{b}_{ij}^T\,\mathbf{x}_{(i)}^k - d_{i\,j}\right),$$

where $H'(\bullet)$ is defined in (3.275). An alternative choice for $\widetilde{f}_i$ is a quadratic approximation of f_i at $\mathbf{x}_{(i)}^k$:

$$\widetilde{f}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) = \sum_{j=1}^{n_i} \widetilde{H}_{ij}\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) + \frac{\tau_i}{2}\|\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\|^2, \tag{3.277}$$

where $\widetilde{H}_{ij}$ is given by

$$\widetilde{H}_{ij}\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) = \begin{cases} \dfrac{\alpha}{r_{ij}^k}\cdot\left(\mathbf{b}_{ij}^T\,\mathbf{x}_{(i)} - d_{ij}\right)^2, & \text{if } \left|r_{ij}^k\right| \geq \alpha, \\ \left(\mathbf{b}_{ij}^T\,\mathbf{x}_{(i)} - d_{ij}\right)^2, & \text{if } \left|r_{ij}^k\right| < \alpha; \end{cases}$$

with $r_{ij}^k \triangleq \mathbf{b}_{ij}^T\,\mathbf{x}_{(i)}^k - d_{ij}$.

Note that the subproblems in Step 2 of Algorithm 10, with $\widetilde{f}_i$ given by (3.276) or (3.277), have a closed form solution. In particular, $\widetilde{\mathbf{x}}_{(i)}^k$ associated with the surrogate in (3.277) is given by

$$\widetilde{\mathbf{x}}_i^k = \left(\tau_i\,\mathbf{I} + 2\,\mathbf{B}_i^T\mathbf{D}_i^k\,\mathbf{B}_i\right)^{-1}\left(\tau_i\,\mathbf{x}^k - (I\cdot\mathbf{y}_{(i)}^k - \nabla f_i(\mathbf{x}_{(i)}^k)) + 2\,\mathbf{B}_i^T\,\mathbf{D}_i\,\mathbf{d}_i\right),$$

where $\mathbf{B}_i$ is the matrix whose j-th row is $\mathbf{b}_{ij}^T$; $\mathbf{d}_i$ is the vector whose j-th component is d_{ij}; and $\mathbf{D}_i^k$ is a diagonal matrix whose j-th diagonal entry is $\min\{1, \alpha/r_{ij}^k\}$.

Numerical Example We simulate a network of $I = 30$ agents, modeled as a directed time-varying graph. At each iteration, the graph is generated according to the following procedure. All the agents are connected through a time-varying ring; at each iteration the order of the agents in the ring is randomly permuted. In addition to the ring topology, each agent has one out-neighbor, selected at each iteration uniformly at random. In (3.274), the parameter $\mathbf{x}$ to estimate has dimension 200, with *i.i.d.* uniformly distributed entries in $[-1, 1]$; $n_i = 20$ (number of measures per agent); and the elements of the vectors $\mathbf{b}_{ij}$ are *i.i.d.* Gaussian distributed and then normalized so that $\|\mathbf{b}_{ij}\| = 1$; the noise affecting the measurements is generated according to a Gaussian distribution, with standard deviation $\sigma = 0.1$, and each agent has one measurement corrupted by one outlier following a Gaussian distribution, with standard deviation $5 \cdot \sigma$; and finally, the cut-off parameter α is set to be $\alpha = 3 \cdot \sigma$.

The free parameters in SONATA (Algorithm 10) are tuned as follows. The proximal parameters τ_i in (3.276) and (3.277) are set to $\tau_i = 2$ and $\tau_i = 1.5$, respectively, for all $i = 1, \ldots, I$. In both instances of the algorithm, the step-size γ^k is chosen according to the rule (3.108), with parameters $\gamma^0 = 0.1$ and $\mu = 0.01$. In the consensus and tracking updates (Step 3), the matrices $\mathbf{A}^k$ are chosen according to the push-sum rule: $a_{ij}^k = 1/d_j^k$. We will refer to the two instances of SONATA as *SONATA-L* ("L" stands for linearized) for the one associated with the surrogate in (3.276), and *SONATA-SCA* for the one associated with the surrogate (3.277). We compare SONATA with the subgradient-push algorithm proposed in [169]. The step-size in the algorithm is chosen according to (3.108), with parameters $\gamma^0 = 0.5$, $\mu = 0.01$ (this choice resulted in the best practical performance among all the tested tunings).

To compare the two algorithms, as merit functions, we used $J^k \triangleq \|\nabla F(\bar{\mathbf{x}}^k)\|_\infty$ and $D^k \triangleq (1/I) \cdot \sum_{i=1}^I \|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}^k\|^2$; the former measures the distance of the average of the iterates $\bar{\mathbf{x}}^k$ from stationarity whereas the latter is a measure of the consensus disagreement. In Fig. 3.15 we plot J^k and D^k for all the algorithms versus the number of messages exchanged at each iteration k. For the (sub)gradient-push algorithm, this number coincides with the iteration index k whereas for the two instances of SONATA it is $2 \cdot k$ (recall that SONATA employs two communications at each iteration). The curves are averaged over 100 Monte-Carlo simulations: in each trial, the parameter $\mathbf{x}$ does not change while the noise and graph connectivity are randomly generated (as described above). The analysis of the figure clearly shows that both instances of SONATA lock consensus and converge much faster than the (sub)gradient-push. Furthermore, SONATA-SCA outperforms SONATA-L, since it better exploits the convexity of the loss function.

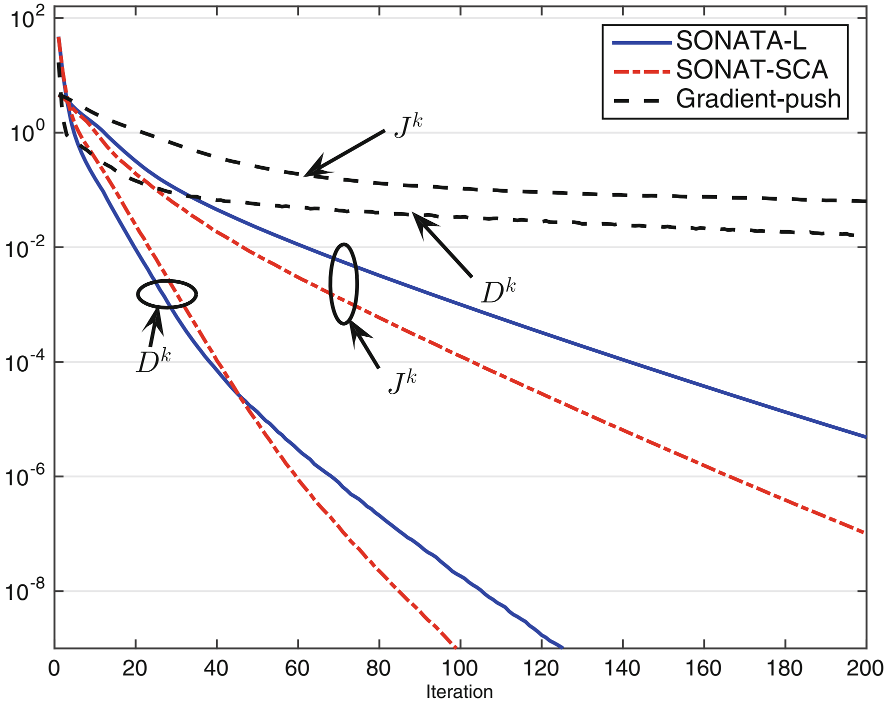


Fig. 3.15 Robust linear regression (3.274): distance from optimality J^k and consensus disagreement D^k versus the number of message exchanges. Both instances of SONATA significantly outperform the (sub)gradient-push [169]

3.4.4.2 Target Localization

Consider the target localization problem described in Example 3 of Sect. 3.4.1.1:

$$\begin{aligned} &\underset{\mathbf{x}\triangleq(\mathbf{x}_t)_{t=1}^n}{\text{minimize}} \quad F(\mathbf{x}) \triangleq \sum_{i=1}^{I}\sum_{t=1}^{n} p_{it}\left(d_{it} - \|\mathbf{x}_t - \mathbf{s}_i\|^2\right)^2 \\ &\text{subject to} \quad \mathbf{x}_{(i)} \in X \subseteq \mathbb{R}^m, \quad \forall i = 1, \ldots, I, \end{aligned} \tag{3.278}$$

where we recall that n is the number of targets; $\mathbf{s}_i$ is the vector of the coordinates of sensor i's location; d_{it} is the squared Euclidean distance between the position of sensor i and that of target t; $\mathbf{x}_t$ is an estimate (to be found) of the location of target t; X is a closed convex set; and $p_{it} \in \{0, 1\}$ is a scalar taking value zero if the ith agent does not have any measurement related to target t. Clearly (3.278) is an instance of

Problem (3.167), with $F = \sum_{i=1}^{I} f_i$, $G = 0$, and

$$f_i\,(\mathbf{x}) = \sum_{t=1}^{n} p_{it}\left(d_{it} - \|\mathbf{x}_t - \mathbf{s}_i\|^2\right)^2. \tag{3.279}$$

We consider next two instances of SONATA, associated with two alternative surrogate functions $\widetilde{f}_i$. A first choice is using the first order approximation of f_i in (3.279), that is,

$$\widetilde{f}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) = f_i\left(\mathbf{x}_{(i)}^k\right) + \nabla f_i\left(\mathbf{x}_{(i)}^k\right)^T\left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right) + \frac{\tau_i}{2}\left\|\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^k\right\|^2, \tag{3.280}$$

where $\mathbf{x}_{(i)}^k \triangleq (\mathbf{x}_{(i),t}^k)_{t=1}^n$, $\nabla f_i(\mathbf{x}_{(i)}^k) = [\nabla_1 f_i(\mathbf{x}_{(i),1}^k)^T, \ldots, \nabla_n f_i(\mathbf{x}_{(i),n}^k)^T]^T$, and

$$\nabla_t f_i(\mathbf{x}_{(i),t}^k) = -4 \cdot p_{it}\left(d_{it} - \|\mathbf{x}_{(i),t}^k - \mathbf{s}_i\|^2\right) \cdot \left(\mathbf{x}_{(i),t}^k - \mathbf{s}_i\right).$$

A second choice for $\widetilde{f}_i$ is motivated by the observation that (3.279) is a fourth-order polynomial in each $\mathbf{x}_t$; one may want to preserve the "partial" convexity in f_i by keeping the first and second order (convex) terms in each summand of (3.279) unaltered and linearizing the higher order terms. This leads to the following:

$$\widetilde{f}_i\left(\mathbf{x}_{(i)} \mid \mathbf{x}_{(i)}^k\right) = \sum_{t=1}^{n} p_{it}\left(\widetilde{f}_{i\,t}\left(\mathbf{x}_{(i),t} \mid \mathbf{x}_{(i)}^k\right) + \frac{\tau_i}{2}\left\|\mathbf{x}_{(i),t} - \mathbf{x}_{(i),t}^k\right\|^2\right), \tag{3.281}$$

where $\mathbf{x}_{(i)} \triangleq (\mathbf{x}_{(i),t})_{t=1}^n$,

$$\widetilde{f}_{i\,t}\left(\mathbf{x}_{(i),t} \mid \mathbf{x}^k\right) = \mathbf{x}_{(i),t}^T\,\mathbf{S}_i\,\mathbf{x}_{(i),t} - \mathbf{b}_{i\,t}^{kT}\left(\mathbf{x}_{(i),t} - \mathbf{x}_{(i),t}^k\right),$$

with $\mathbf{S}_i = 4 \cdot \mathbf{s}_i\,\mathbf{s}_i^T + 2 \cdot \|\mathbf{s}_i\|^2\mathbf{I}$, and

$$\mathbf{b}_{i\,t}^k = 4\|\mathbf{s}_i\|^2\mathbf{s}_i - 4\left(\|\mathbf{x}_{(i),t}^k\|^2 - d_{it}\right)\left(\mathbf{x}_{(i),t}^k - \mathbf{s}_i\right) + 8\left(\mathbf{s}_i^T\mathbf{x}_{(i),t}^k\right)\mathbf{x}_{(i),t}^k.$$

Numerical Example We simulate a network of $I = 30$ sensors, modeled as a directed time-varying graph; and we deploy $n = 5$ targets. The graph is generated as described in the previous example (cf. Sect. 3.4.4.1). The locations of the sensors and targets are generated uniformly at random in $[0, 1]^2$. The parameters p_{it} take values $\{0, 1\}$, with equal probability. The distances $d_{i\,t}$ are corrupted by *i.i.d.* zero-mean Gaussian noise, with standard deviation set to be the minimum pairwise distance between sensors and targets. We let the constraint set $X = \mathbb{R}^2$.

The free parameters in SONATA (Algorithm 10) are tuned as follows. The proximal parameters τ_i in (3.280) and (3.281) are set to $\tau_i = 7$ and $\tau_i = 5$, respectively, for all $i = 1, \ldots, I$. In both instances of the algorithm, the step-size γ^k is chosen according to the rule (3.108), with parameters $\gamma^0 = 0.1$ and $\mu = 10^{-4}$. In the consensus and tracking updates (Step 3), the matrices $\mathbf{A}^k$ are chosen according to the push-sum rule: $a_{ij}^k = 1/d_j^k$. We will refer to the two instances of SONATA as *SONATA-L* ("L" stands for linearized) for the one associated with the surrogate in(3.280), and *SONATA-SCA* for the one associated with the surrogate (3.281). We compare SONATA with the distributed gradient [233] for unconstrained optimization (adapted here to account for the constraints; note that for such a version there is no formal proof of convergence in the literature). The step-size in the algorithm [233] is chosen according to (3.108), with parameters $\gamma^0 = 0.5$, $\mu = 0.01$ (this choice resulted in the best practical performance among all the tested tunings).

A comparison of the algorithms is given in Fig. 3.16, where we plotted the merit functions $J^k \triangleq \|\nabla F(\bar{\mathbf{x}}^k)\|_\infty$ and $D^k \triangleq (1/I) \cdot \sum_{i=1}^I \|\mathbf{x}_{(i)}^k - \bar{\mathbf{x}}^k\|^2$ versus the number of messages exchanged at each iteration k (cf. Sect. 3.4.4.1). The curves

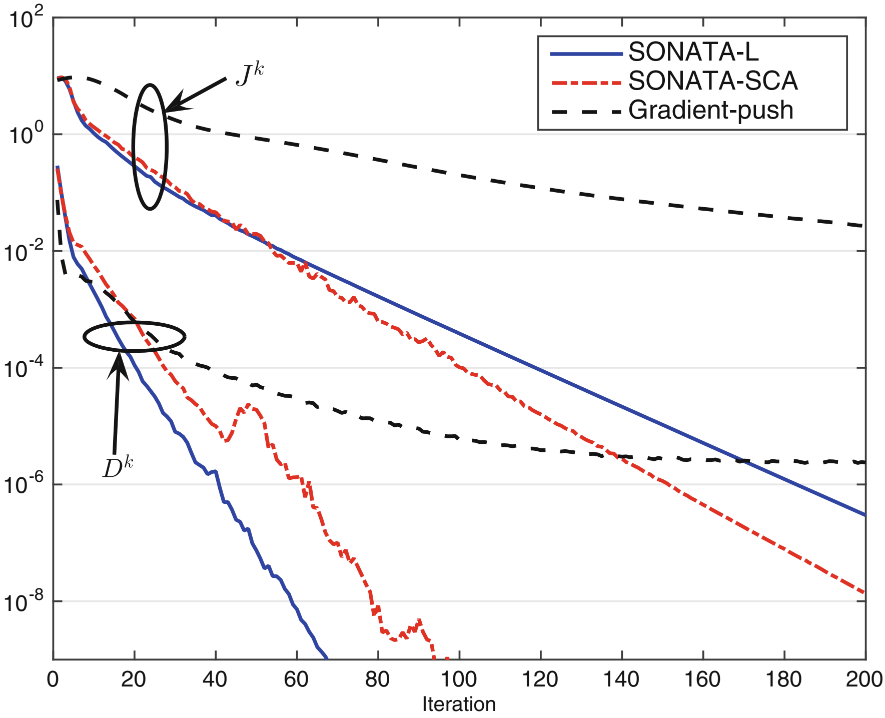


Fig. 3.16 Target localization problem (3.278): distance from optimality J^k and consensus disagreement D^k versus the number of message exchanges. Both instances of SONATA significantly outperform the distributed gradient algorithm [233]

are averaged over 100 Monte-Carlo simulations; in each trial, the position of the sensors and targets were kept fixed, while the noise and graph connectivity were randomly generated every trial. Figure 3.16 shows that both instances of SONATA converge much faster than the benchmark distributed gradient algorithm. This can be explained by the fact that SONATA better exploits the problem's structure and employs the tracking of the average of agents' gradients.

3.4.5 Sources and Notes

Distributed optimization has a rich literature, dating back to the 1980s [240]. Since then, due to the emergence of large-scale networks, advances in computing and storage, data acquisition, and communications, the development of distributed algorithms for solving optimization problems over networks has received significant attention. Besides the distributed approaches we are going to summarize below, many efforts have been made to solve optimization problems over networks with *specific topology*, including multilayer HNets (e.g., master-slave architectures and shared-memory systems)—see Fig. 3.13. However, all these schemes require some degree of centralization to be implemented, due to the use of master nodes. Since the focus of this lecture is on distributed algorithms implementable on general network architectures, in the following we omit to discuss the literature on algorithms for HNets. Although the lecture considered mainly nonconvex optimization, we begin overviewing the much abundant literature of distributed algorithms for convex problems. In fact, some of the methods and tools introduced in this lecture for nonconvex problems can be applied and are useful also for convex instances.

3.4.5.1 Distributed Convex Optimization

The literature on distributed solution methods for convex problems is vast and a comprehensive overview of current methods goes beyond the scope of this commentary. Here we briefly focus on the methods somehow related to the proposed SCA approach and distributed gradient tracking. Generally speaking, we roughly divide the literature in two groups, namely: primal and dual methods.

Primal Methods While substantially different, primal methods can be generically abstracted as a combination of a local (sub)gradient step and a subsequent consensus-like update (or multiple consensus updates); examples include [116, 169, 170, 216, 217]. Algorithms for adaptation and learning tasks based on in-network diffusion techniques were proposed in [39, 46, 202]. Schemes in [39, 46, 170, 202, 217] are applicable only to *undirected* graphs; [39, 170, 217] require the consensus matrices to be *doubly-stochastic* whereas [46, 202] use *row-stochastic* matrices but are applicable only to strongly convex agents' cost functions having a *common* minimizer. When the graph is *directed*, doubly-stochastic weight matrices compliant to the graph may not exist; and, when they exist, their construction is computationally prohibitive for networks with arbitrary topology [94]. Recently, some new algorithms have been developed that do not require doubly-stochastic weights. Examples, include [169], where the authors combined the sub-gradient algorithm [170] with the push-sum consensus protocol [123]; and [256, 257]. However, the schemes in [170, 256, 257] considered only *unconstrained* optimization problems; in fact, up until the very recent works [208, 228, 232], it was not clear how to leverage push-sum protocols to deal with constraints (e.g., to preserve the feasibility of the iterates).

In all the aforementioned algorithms but [208, 232], agents perform their local optimization using only the gradient of their *own* function, while neglecting the rest of the sum-cost function F [cf. (3.167)]. This makes the algorithms simple but usually slow: Even if the objective functions are differentiable and strongly convex, these methods still need to use a diminishing step-size to converge to a consensual solution, and thus converge at a sub-linear rate. On the other hand, with a constant (sufficiently small) step-size, these distributed methods can be faster, but they only converge to a neighborhood of the solution set. This phenomenon creates an exactness-speed dilemma. In addition, convergence of these algorithms was proved only under the assumption that the objective function has *bounded (sub)gradient*, which limits the range of applicability of these schemes.

A different class of distributed (primal) approaches that bypasses the above issues is the one based on the idea of *gradient tracking*: each agent updates its own local variables along a surrogate direction that tracks the gradient average $\sum_i \nabla f_i$. This idea was proposed independently in [70–72] for the general class of constrained nonsmooth nonconvex problems (3.167), and in [259] for the special case of strongly convex, unconstrained, smooth, optimization. The works [208, 232] extended the algorithms to (possibly) time-varying digraphs (still in the nonconvex setting of [70–72]). A convergence rate analysis of the scheme [259] was later developed in [172, 191], with [172] considering time-varying (directed) graphs. The benefit of these schemes is threefold: (1) a constant step-size can be employed, instead of a diminishing one; (2) the assumption that the gradient is bounded is no longer needed; and (3) the convergence speed of the algorithms significantly improves—linear convergence was in fact proved in [172, 191] for unconstrained, smooth, strongly convex $\sum_i f_i$, and more recently in [231] for the more general class of constrained, nonsmooth, problems (3.167) (with F strongly convex).

Finally, it is worth mentioning that a difference algorithmic structure was proposed in [216] to cancel the steady state error in the aforementioned plain decentralized gradient descent methods and converge to a consensual solution using a fixed (sufficiently small) step-size. Convergence at a sub-linear rate was proved when the objective function is smooth and convex (and the problem unconstrained), and a Q-linear rate when the objective function is (smooth) strongly convex (still unconstrained optimization).

Dual-Based Methods This class of algorithms is based on a different approach: slack variables are first introduced to decouple the sum-utility function while forcing consistency among these local copies by adding consensus equality constraints (matching the graph topology). Lagrangian dual variables are then introduced to deal with such coupling constraints. The resulting algorithms build on primal-dual updates, aiming at converging to a saddle point of the (augmented) Lagrangian function. Examples of such algorithms include ADMM-like methods [42, 115, 217, 247] as well as inexact primal-dual instances wherein a first order [43] or second-order [165, 166] approximation of the Lagrangian function is minimized at each iteration. All these algorithms can handle only *static and undirected* graphs. Dual methods cannot be trivially extended to time-varying graphs or digraphs, as it is not clear how to enforce consensus via equality constraints over time-varying or directed networks. Furthermore, all the above schemes but [42, 217] require the objective function to be *smooth* and the optimization problem to be *unconstrained*.

3.4.5.2 Distributed Nonconvex Optimization

For the general nonconvex problem (3.167), results are scarse. We remark that the parallel algorithm described in Lecture II as well as those discussed in Sect. 3.3.7 are not applicable to the distributed setting (3.167): either they require the graph to be complete (i.e., at each iteration, agents must be able to exchange information with *all* the others) or they assume full knowledge of F from the agents. There are only a few distributed algorithms designed for (3.167), including primal methods [20, 72, 233, 245] and dual methods [108, 275]. The key features of these algorithms are summarized in Table 3.8 and briefly discussed next.

Primal Methods The distributed stochastic projection scheme [20] implements random gossip between agents and uses diminishing step-size, and can handle *smooth* objective functions over *undirected static* graphs. In [233], the authors showed that the distributed push-sum gradient algorithm with diminishing step-size, earlier proposed for convex objectives in [169], converges also when applied to nonconvex *smooth unconstrained* problems. The first provably convergent distributed scheme for (3.167), with $G \neq 0$ and (convex) constraints X, over time-varying digraphs is NEXT [70–72]. A special instance of NEXT, applied to smooth V over *undirected static* graphs, was later proved in [245] to have sub-linear convergence rate. However, the consensus protocol employed in [72, 245] uses *doubly-stochastic* weight matrices, and thus is limited to special digraphs (or undirected graphs). Moreover, all the algorithms discussed above require that the (sub)gradient of V is *bounded* on X (or $\mathbb{R}^m$).

This issue was resolved in SONATA (Algorithm 10, Sect. 3.4.3), proposed in [208, 232], which combined a judiciously designed perturbed push-sum consensus scheme with SCA techniques. Based on a new line of analysis whereby optimization and consensus dynamics are *jointly* analyzed by introducing a novel Lyapunov function, SONATA was shown to converge sub-linearly to a stationary solution of (3.167), without requiring the (sub)gradient of V being bounded. SONATA is also the first provably convergent algorithm that (i) can deal with constrained,

Table 3.8 Overview of current distributed methods for nonconvex optimization over networks

		Proj-DGM [20]	NEXT [72]	Push-sum DGM [233]	Prox-PDA [108]	DeFW [245]	SONATA [208] (this chapter)
Nonsmooth G^+			✓				✓
Constraints		✓	✓			X compact	✓
Unbounded gradient					✓		✓
Network topology	Time-varying		✓	✓			✓
	Digraph		Restricted	✓			✓
Step-size	Constant				✓		✓
	Diminishing	✓	✓	✓	✓	✓	✓
Complexity				✓	✓	✓	✓

nonsmooth, (convex and nonconvex) problems in the form (3.167), and time-varying digraphs; and (ii) employs either a constant or diminishing step-size.

Dual-Based Methods As their convex counterparts, dual-based schemes [108, 275] cannot deal with time-varying or directed graphs. The algorithm proposed in [275] is an early attempt of distributed method for nonconvex problems. However, it calls for the solution of nonconvex subproblems and converges to (stationary) solutions of an auxiliary problem, which are not necessarily stationary for the original problem. The prox-PDA algorithm in [108], proposed for linearly constrained problems, can be applied to Problem (3.167), but it is limited to the special classes where the functions f_i are *smooth* and $X = \mathbb{R}^m$. The dual-based algorithms discussed above but [108] assume that the (sub)gradient of V is *bounded*.

Practical Implementation Issues and Open Questions The majority of the methods discussed above assume an "ideal" distributed computing environment wherein all the communication links operate reliably and noise-free, and the agents communicate and update their own variables in a synchronous fashion. There are however many practical issues that arise for these models, including difficulties associated with maintaining time-synchronization in face of computational and communication delays, reaching deadlock situations requiring re-initializations of the update process due to node/link failures.

Asynchronous methods based on randomization techniques were proposed in [18–20, 22, 112, 130, 140, 168, 176, 225, 247, 270]. These algorithms assume that agents are activated (wakeup) randomly (with no coordination), run some local computation, and then wake up (some or all) their neighbors to pass them their most recent update. Both random (e.g., based on gossip consensus schemes) [18–20, 112, 140, 168, 225, 247, 270] and deterministic (e.g., based on the push-sum protocol) activations [22, 130, 176] of the neighbors were proposed. While asynchronous in the activation of the agents performing the update, these schemes still impose a coordination among the agents, cannot tolerate (arbitrary) delays in the communications (packet losses are considered in [22, 130, 270]) or optimization (out-of-sync information), and require the agents to stay all the time in listening mode. A first attempt to deal with this issue was proposed in [255]: a dual-based scheme is employed wherein agents are randomly activated, and update their local copies using possibly outdated information. The probabilistic model for asynchrony requires that the random variables triggering the activation of the agents and the delay vector used by the agent to performs its update are *independent*. While this greatly simplifies the convergence analysis, it makes the model not realistic—see Sect. 3.3.7 in Lec. II for a discussion on this issue. Furthermore, [255] is applicable only to convex instances of (3.167), and over fixed undirected graphs. At the time of this writing, designing distributed fully asynchronous algorithms for the nonconvex problem (3.167)—in the sense that (1) agents update their own variables with no coordination; and (2) can handle arbitrary (deterministic or random) delay profiles—remains an open problem. A first attempt towards this direction can be found in [234].

Despite of asynchrony, another question of practical interest is how to design and implement distributed algorithms under low communication requirements, and what trade-offs are involved in such designs/implementations. For instance, in some sensor networks, broadcast communications can be too expensive in terms of consumed power. Also, communications are subject to rate constraints (systems are bandwidth limited). Furthermore, in the emerging distributed big-data applications, optimization problems are huge-scale: computing at each iteration the gradient with respect to all the optimization variables, communicating at each iteration, and sending to the neighbors the entire set of variables would incur in an unaffordable computational cost and communication overhead or is just infeasible. Some initial investigations along this lines were presented in the literature, following alternative, scattered, paths. For instance, in [177, 178], a distributed algorithm for convex and nonconvex instances of (3.167) was proposed, aiming at reducing both communication and computation costs by optimizing and transmitting at each iteration only a subset of the entire optimization variables. While asymptotic convergence of the method was proved, the tradeoff between communication and computation has not been studied yet. Some insight on the impact of multiple communications steps (without optimizing) or multiple optimization steps (without communicating) can be found in [13, 239]. Finally, some works proposed distributed methods for convex optimization problems implementing quantized communications. Of course, much more research needs to be done on these important topics.

Acknowledgements This work has been partially supported by the USA National Science Foundation under Grants CIF 1564044, CIF 1719205, and CAREER Award 1555850; and in part by the Office of Naval Research under Grant N00014-16-1-2244 and the Army Research Office under Grant W911NF1810238.

References

1. 3GPP TR 36.814
2. M. Ahn, J.-S. Pang, J. Xin, Difference-of-convex learning: directional stationarity, optimality, and sparsity. SIAM J. Optim. **27**(3), 1637–1665 (2017)
3. M. Allain, J. Idier, Y. Goussard, On global and local convergence of half-quadratic algorithms. IEEE Trans. Image Process. **15**(5), 1130–1142 (2006)
4. F. Bach, R. Jenatton, J. Mairal, G. Obozinski, Optimization with sparsity-inducing penalties. Found. Trends Mach. Learn. **4**(1), 1–106 (2012)
5. G.M. Baudet, Asynchronous iterative methods for multiprocessors. J. ACM (JACM) **25**(2), 226–244 (1978)
6. E. Bauer, D. Koller, Y. Singer, Update rules for parameter estimation in bayesian networks, in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (1997), pp. 3–13
7. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
8. A. Beck, M. Teboulle, Gradient-based algorithms with applications to signal recovery problems, in *Convex Optimization in Signal Processing and Communications*, chap. 2 (Cambridge University Press, Cambridge, 2010)

9. A. Beck, M. Teboulle, Z. Chikishev, Iterative minimization schemes for solving the single source localization problem. SIAM J. Optim. **19**(3), 1397–1416 (2008)
10. M.P. Becker, I. Yang, K. Lange, EM algorithms without missing data. Stat. Methods Med. Res. **6**(1), 38–54 (1997)
11. F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, M. Vetterli, Weighted gossip: distributed averaging using non-doubly stochastic matrices, in *Proceedings of the 2010 IEEE International Symposium on Information Theory Proceedings (ISIT)* (2010), pp. 1753–1757
12. K. Benidis, Y. Sun, P. Babu, D.P. Palomar, Orthogonal sparse PCA and covariance estimation via procrustes reformulation. IEEE Trans. Signal Process. **64**(23), 6211–6226 (2016)
13. A.S. Berahas, R. Bollapragada, N.S. Keskar, E. Wei, Balancing communication and computation in distributed optimization. arXiv:1709.02999 (2017, preprint)
14. D.P. Bertsekas, *Nonlinear Programming* (Athena Scientific, Belmont, 1995)
15. D.P. Bertsekas, P. Tseng, Partial proximal minimization algorithms for convex programming. SIAM J. Optim. **4**(3), 551–572 (1994)
16. D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd edn. (Athena Scientific Press, Belmont, 1989)
17. D.P. Bertsekas, J.N. Tsitsiklis, Gradient convergence in gradient methods with errors. SIAM J. Optim. **10**(3), 627–642 (2000)
18. P. Bianchi, W. Hachem, F. Iutzeler, A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization. IEEE Trans. Autom. Control **61**(10), 2947–2957 (2016)
19. P. Bianchi, G. Fort, W. Hachem, Performance of a distributed stochastic approximation algorithm. IEEE Trans. Inf. Theory **59**(11), 7405–7418 (2013)
20. P. Bianchi, J. Jakubowicz, Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. IEEE Trans. Autom. Control **58**(2), 391–405 (2013)
21. T. Blumensath, M. Yaghoobi, M.E. Davies, Iterative hard thresholding and l_0 regularisation, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, vol. 3 (2007), pp. 877–880
22. N. Bof, R. Carli, G. Notarstefano, L. Schenato, D. Varagnolo, Newton-Raphson consensus under asynchronous and lossy communications for peer-to-peer networks. arXiv:1707.09178 (2017, preprint)
23. D. Böhning, Multinomial logistic regression algorithm. Ann. Inst. Stat. Math. **44**(1), 197–200 (1992)
24. J.M. Borwein, A.S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS Books in Mathematics (Springer, Berlin, 2000)
25. L. Bottou, F.E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning. arXiv:1606.04838 (2016, preprint)
26. C. Bouman, K. Sauer, Fast numerical methods for emission and transmission tomographic reconstruction, in *Proceedings of Conference on Information Sciences and Systems* (1993), pp. 611–616
27. R.A. Boyles, On the convergence of the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **45**(1), 47–50 (1983)
28. P.S. Bradley, O.L. Mangasarian, Feature selection via concave minimization and support vector machines, in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 98')*, vol. 98 (1998), pp. 82–90
29. J.K. Bradley, A. Kyrola, D. Bickson, C. Guestrin, Parallel coordinate descent for l1-regularized loss minimization, in *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)* (2011), pp. 321–328
30. P. Bühlmann, S. van De Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications* (Springer, Berlin, 2011)
31. R.H. Byrd, J. Nocedal, F. Oztoprak, An inexact successive quadratic approximation method for l-1 regularized optimization. Math. Program. **157**(2), 375–396 (2016)
32. E.J. Candés, T. Tao, Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)

33. E.J. Candés, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory **52**(12), 5406–5425 (2006)
34. E. J. Candés, M.B. Wakin, An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 21–30 (2008)
35. E.J. Candés, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM **58**(3), 11 (2011)
36. E.J. Candés, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**(5–6), 877–905 (2008)
37. L. Cannelli, F. Facchinei, V. Kungurtsev, G. Scutari, Asynchronous parallel algorithms for nonconvex big-data optimization: model and convergence. arXiv:1607.04818 (2016, preprint)
38. L. Cannelli, F. Facchinei, V. Kungurtsev, G. Scutari, Asynchronous parallel algorithms for nonconvex big-data optimization Part II: complexity and numerical results. arXiv:1701.04900 (2017, preprint)
39. F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation. IEEE Trans. Signal Process. **58**(3), 1035–1048 (2010)
40. A. Chambolle, An algorithm for total variation minimization and applications. J. Math. Imaging Vision **20**(1–2), 89–97 (2004)
41. V. Chandrasekaran, S. Sanghavi, P.A. Parrilo, A.S. Willsky, Rank-sparsity incoherence for matrix decomposition. SIAM J. Optim. **21**(2), 572–596 (2011)
42. T.-H. Chang, A proximal dual consensus ADMM method for multi-agent constrained optimization. IEEE Trans. Signal Process. **64**(14), 3719–3734 (2014)
43. T.-H. Chang, M. Hong, X. Wang, Multi-agent distributed optimization via inexact consensus ADMM. IEEE Trans. Signal Process. **63**(2), 482–497 (2015)
44. T. Charalambous, M.G. Rabbat, M. Johansson, C.N. Hadjicostis, Distributed finite-time computation of digraph parameters: left-eigenvector, out-degree and spectrum. IEEE Trans. Control Netw. Syst. **3**(2), 137–148 (2016)
45. D. Chazan, W. Miranker, Chaotic relaxation. Linear Algebra Appl. **2**(2), 199–222 (1969)
46. J. Chen, A.H. Sayed, Diffusion adaptation strategies for distributed optimization and learning over networks. IEEE Trans. Signal Process. **60**(8), 4289–4305 (2012)
47. E. Chouzenoux, J.-C. Pesquet, Convergence rate analysis of the majorize-minimize subspace algorithm. IEEE Signal Process. Lett. **23**(9), 1284–1288 (2016)
48. E. Chouzenoux, J. Idier, S. Moussaoui, A majorize-minimize strategy for subspace optimization applied to image restoration. IEEE Trans. Image Process. **20**(6), 1517–1528 (2011)
49. E. Chouzenoux, S. Moussaoui, J. Idier, Majorize-minimize linesearch for inversion methods involving barrier function optimization. Inverse Prob. **28**(6), 065011 (2012)
50. E. Chouzenoux, J.-C. Pesquet, A. Repetti, Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. J. Optim. Theory Appl. **162**(1), 107–132 (2014)
51. G. Cohen, Optimization by decomposition and coordination: a unified approach. IEEE Trans. Autom. Control **23**(2), 222–232 (1978)
52. G. Cohen, Auxiliary problem principle and decomposition of optimization problems. J. Optim. Theory Appl. **32**(3), 277–305 (1980)
53. P.L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (Springer, Berlin, 2011), pp. 185–212
54. J.A. Costa, N. Patwari, A.O. Hero III, Distributed weighted-multidimensional scaling for node localization in sensor networks. ACM Trans. Sensor Netw. **2**(1), 39–64 (2006)
55. A. Daneshmand, Numerical comparison of hybrid random/deterministic parallel algorithms for nonconvex big data optimization. Technical report, Department of Electrical Engineering, SUNY Buffalo, August 2014
56. A. Daneshmand, F. Facchinei, V. Kungurtsev, G. Scutari, Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization. IEEE Trans. Signal Process. **63**(15), 3914–3929 (2015)

57. J.M. Danskin, The theory of max-min, with applications. SIAM J. Appl. Math. **14**(4), 641–664 (1966)
58. M.E. Daube-Witherspoon, G. Muehllehner, An iterative image space reconstruction algorthm suitable for volume ECT. IEEE Trans. Med. Imaging **5**(2), 61–66 (1986)
59. I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**(11), 1413–1457 (2004)
60. D. Davis, The asynchronous PALM algorithm for nonsmooth nonconvex problems. arXiv:1604.00526 (2016, preprint)
61. D. Davis, B. Edmunds, M. Udell, The sound of APALM clapping: faster nonsmooth nonconvex optimization with stochastic asynchronous PALM, in *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (Curran Associates, New York, 2016), pp. 226–234
62. J. de Leeuw, Applications of convex analysis to multidimensional scaling, in *Recent Developments in Statistics* (1977)
63. J. de Leeuw, Convergence of the majorization method for multidimensional scaling. J. Classif. **5**(2), 163–180 (1988)
64. J. de Leeuw, Block-relaxation algorithms in statistics, in *Information Systems and Data Analysis* (Springer, Berlin, 1994), pp. 308–324
65. J. de Leeuw, W.J. Heiser, Convergence of correction matrix algorithms for multidimensional scaling, in *Geometric Representations of Relational Data* (Mathesis Press, Ann Arbor, 1977), pp. 735–752 (1977)
66. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **39**, 1–38 (1977)
67. W. Deng, M.-J. Lai, Z. Peng, W. Yin, Parallel multi-block ADMM with $o(1/k)$ convergence. J. Sci. Comput. **71**(2), 712–736 (2017)
68. A.R. De Pierro, A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. IEEE Trans. Med. Imaging **14**(1), 132–137 (1995)
69. I.S. Dhillon, P.K. Ravikumar, A. Tewari, Nearest neighbor based greedy coordinate descent, in *Advances in Neural Information Processing Systems 24 (NIPS 2011)* (Curran Associates, New York, 2011), pp. 2160–2168
70. P. Di Lorenzo, G. Scutari, Distributed nonconvex optimization over networks, in *Proceedings of the IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2015)* (2015)
71. P. Di Lorenzo, G. Scutari, Distributed nonconvex optimization over time-varying networks, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2016)* (2016)
72. P. Di Lorenzo, G. Scutari, NEXT: in-network nonconvex optimization. IEEE Trans. Signal Inf. Process. Over Netw. **2**(2):120–136 (2016)
73. S. Doclo, M. Moonen, T.V. den Bogaert, J. Wouters, Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids. IEEE Trans. Audio Speech Lang. Process. **17**(1), 38–51 (2009)
74. S. Doclo, S. Gannot, M. Moonen, A. Spriet, Acoustic beamforming for hearing aid applications, in *Handbook on Array Processing and Sensor Networks* (Wiley, Hoboken, 2010), pp. 269–302
75. P.B. Eggermont, Multiplicative iterative algorithms for convex programming. Linear Algebra Appl. **130**, 25–42 (1990)
76. H.A. Eiselt, V. Marianov, Pioneering developments in location analysis, in *Foundations of Location Analysis, International Series in Operations Research & Management Science*, chap. 11 (Springer, Berlin, 2011), pp. 3–22
77. M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
78. F. Facchinei, J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problem* (Springer, New York, 2003)
79. F. Facchinei, G. Scutari, S. Sagratella, Parallel selective algorithms for nonconvex big data optimization. IEEE Trans. Signal Process. **63**(7), 1874–1889 (2015)

80. F. Facchinei, V. Kungurtsev, L. Lampariello, G. Scutari, Ghost penalties in nonconvex constrained optimization: diminishing stepsizes and iteration complexity. arXiv:1709.03384 (2017, preprint)
81. F. Facchinei, L. Lampariello, G. Scutari, Feasible methods for nonconvex nonsmooth problems with applications in green communications. Math. Program. **164**(1–2), 55–90 (2017)
82. J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001)
83. J.A. Fessler, A.O. Hero III, Space-alternating generalized expectation-maximization algorithm. IEEE Trans. Signal Process. **42**(10), 2664–2677 (1994)
84. J.A. Fessler, A.O. Hero III, Penalized maximum-likelihood image reconstruction using space-alternating generalized em algorithms. IEEE Trans. Image Process. **4**(10), 1417–1429 (1995)
85. C. Févotte, Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)* (2011), pp. 1980–1983
86. C. Févotte, J. Idier, Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput. **23**(9), 2421–2456 (2011)
87. M.A.T. Figueiredo, R.D. Nowak, A bound optimization approach to wavelet-based image deconvolution, in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2005)*, vol. 2 (2005)
88. M.A.T. Figueiredo, J.M. Bioucas-Dias, R.D. Nowak, Majorization-minimization algorithms for wavelet-based image restoration. IEEE Trans. Image Process. **16**(12), 2980–2991 (2007)
89. A. Frommer, D.B. Szyld, On asynchronous iterations. J. Comput. Appl. Math. **123**(1–2), 201–216 (2000)
90. W.J. Fu, Penalized regressions: the bridge versus the lasso. J. Comput. Graph. Stat. **7**(3), 397–416 (1998)
91. M. Fukushima, H. Mine, A generalized proximal point algorithm for certain non-convex minimization problems. Int. J. Syst. Sci. **12**(8), 989–1000 (1981)
92. D. Geman, G. Reynolds, Constrained restoration and the recovery of discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. **14**(3), 367–383 (1992)
93. D. Geman, C. Yang, Nonlinear image recovery with half-quadratic regularization. IEEE Trans. Image Process. **4**(7), 932–946 (1995)
94. B. Gharesifard, J. Cortés, When does a digraph admit a doubly stochastic adjacency matrix? in *Proceedings of the 2010 American Control Conference* (2010), pp. 2440–2445
95. G.B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, B.F. Wollenberg, Monitoring and optimization for power grids: a signal processing perspective. IEEE Signal Process. Mag. **30**(5), 107–128 (2013)
96. I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. IEEE Trans. Signal Process. **45**(3), 600–616 (1997)
97. J.-M. Gracia, Directional derivatives of the singular values of matrices depending on several real parameters. arXiv:1504.01679 (2015, preprint)
98. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics (Springer, New York, 2009)
99. T. Hastie, R. Tibshirani, M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations* (CRC Press, West Palm Beach, 2015)
100. Z. He, S. Xie, R. Zdunek, G. Zhou, A. Cichocki. Symmetric nonnegative matrix factorization: algorithms and applications to probabilistic clustering. IEEE Trans. Neural Netw. **22**(12), 2117–2131 (2011)
101. W.J. Heiser, Convergent computation by iterative majorization: theory and applications in multidimensional data analysis, in *Recent Advances in Descriptive Multivariate Analysis*, (Clarendon Press, Oxford, 1995), pp. 157–189
102. D.P. Helmbold, R.E. Schapire, Y. Singer, M.K. Warmuth, A comparison of new and old algorithms for a mixture estimation problem. Mach. Learn. **27**(1), 97–119 (1997)

103. J.M. Hendrickx, J.N. Tsitsiklis, Fundamental limitations for anonymous distributed systems with broadcast communications, in *Proceedings of the 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (2015), pp. 9–16
104. J.-B. Hiriart-Urruty, C. Lemaréchal, *Convex Analysis and Minimization Algorithms I: Fundamentals*, vol. 305 (Springer, Berlin, 1993)
105. A. Hjorungnes, *Complex-Valued Matrix Derivatives With Applications in Signal Processing and Communications* (Cambridge University Press, London, 2011)
106. W. Hogan, Point-to-set maps in mathematical programming. SIAM Rev. **15**(3), 591–603 (1973)
107. M. Hong, A distributed, asynchronous and incremental algorithm for nonconvex optimization: an ADMM approach. IEEE Trans. Control Netw. Syst. (2017)
108. M. Hong, D. Hajinezhad, M.-M. Zhao, Prox-PDA: the proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks, in *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, vol. 70 (2017), pp. 1529–1538
109. P.O. Hoyer, Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5**, 1457–1469 (2004)
110. D.R. Hunter, K. Lange, A tutorial on MM algorithms. Am. Stat. **58**(1), 30–37 (2004)
111. Z. Huo, H. Huang, Asynchronous stochastic gradient descent with variance reduction for non-convex optimization. arXiv:1604.03584 (2016, preprint)
112. F. Iutzeler, P. Bianchi, P. Ciblat, W. Hachem, Asynchronous distributed optimization using a randomized alternating direction method of multipliers, in *Proceedings of the 2013 IEEE 52nd Annual Conference on Decision and Control (CDC 2013)* (2013), pp. 3671–3676
113. M.W. Jacobson, J.A. Fessler, An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. IEEE Trans. Image Process. **16**(10), 2411–2422 (2007)
114. P. Jain, P. Netrapalli, S. Sanghavi, Low-rank matrix completion using alternating minimization, in *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing* (2013), pp. 665–674
115. D. Jakovetic, J. Xavier, J.M.F. Moura, Cooperative convex optimization in networked systems: augmented Lagrangian algorithms with directed gossip communication. IEEE Trans. Signal Process. **59**(8), 3889–3902 (2011)
116. D. Jakovetić, J. Xavier, J.M.F. Moura, Fast distributed gradient methods. IEEE Trans. Autom. Control **59**(5), 1131–1146 (2014)
117. M. Jamshidian, R.I. Jennrich, Conjugate gradient acceleration of the EM algorithm. *J. Am. Stat. Assoc.* **88**(421), 221–228 (1993)
118. M. Jamshidian, R.I. Jennrich, Acceleration of the EM algorithm by using quasi-Newton methods. J. R. Stat. Soc. Ser. B (Stat Methodol.) **59**(3), 569–587 (1997)
119. R. Jenatton, G. Obozinski, F. Bach, Structured sparse principal component analysis, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)* (2010), pp. 366–373
120. M. Journée, Y. Nesterov, P. Richtárik, R. Sepulchre, Generalized power method for sparse principal component analysis. J. Mach. Learn. Res. **11**, 517–553 (2010)
121. N. Karowski, A.C. Viana, A. Wolisz, Optimized asynchronous multi-channel neighbor discovery, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM 2011)* (2011), pp. 536–540
122. V. Kekatos, G.B. Giannakis, Distributed robust power system state estimation. IEEE Trans. Power Syst. **28**(2), 1617–1626 (2013)
123. D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science* (2003), pp. 482–491
124. S.-J. Kim, G.B. Giannakis, Optimal resource allocation for MIMO ad hoc cognitive radio networks. IEEE Trans. Inf. Theory **57**(5), 3117–3131 (2011)

125. H. Kim, H. Park, Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. SIAM J. Matrix Anal. Appl. **30**(2), 713–730 (2008)
126. K. Kreutz-Delgado, The complex gradient operator and the CR-calculus. Technical report, Electrical and Computer Engineering, Jacobs School of Engineering, University of California, San Diego, July 2009
127. B. Krishnapuram, L. Carin, M.A.T. Figueiredo, A.J. Hartemink, Sparse multinomial logistic regression: fast algorithms and generalization bounds. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 957–968 (2005)
128. D. Kuang, C. Ding, H. Park, Symmetric nonnegative matrix factorization for graph clustering, in *Proceedings of the 2012 SIAM International Conference on Data Mining* (2012), pp. 106–117
129. D. Kuang, S. Yun, H. Park, SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. J. Glob. Optim. **62**(3), 545–574 (2015)
130. S. Kumar, R. Jain, K. Rajawat, Asynchronous optimization over heterogeneous networks via consensus ADMM. IEEE Trans. Signal Inf. Process. Netw. **3**(1), 114–129 (2017)
131. C. Labat, J. Idier, Convergence of conjugate gradient methods with a closed-form stepsize formula. J. Optim. Theory Appl. **136**(1), 43–60 (2008)
132. A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E.D. Kolaczyk, N. Taft, Structural analysis of network traffic flows, in *ACM SIGMETRICS Performance Evaluation Review*, vol. 32 (2004), pp. 61–72
133. K. Lange, A gradient algorithm locally equivalent to the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **57**, 425–437 (1995)
134. K. Lange, A quasi-Newton acceleration of the EM algorithm. Stat. Sin. **5**, 1–18 (1995)
135. K. Lange, J.A. Fessler, Globally convergent algorithms for maximum a posteriori transmission tomography. IEEE Trans. Image Process. **4**(10), 1430–1438 (1995)
136. K. Lange, H. Zhou, MM algorithms for geometric and signomial programming. Math. Program. **143**(1–2), 339–356 (2014)
137. K. Lange, D.R. Hunter, I. Yang, Optimization transfer using surrogate objective functions. J. Comput. Graph. Stat. **9**(1), 1–20 (2000)
138. K. Lange, E.C. Chi, H. Zhou, A brief survey of modern optimization for statisticians. Int. Stat. Rev. **82**(1), 46–70 (2014)
139. R. Leblond, F. Pedregosa, S. Lacoste-Julien, ASAGA: asynchronous parallel SAGA, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2017), pp. 46–54
140. S. Lee, A. Nedić, Asynchronous gossip-based random projection algorithms over networks. IEEE Trans. Autom. Control **61**(4), 953–968 (2016)
141. D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788 (1999)
142. D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, ed. by T.K. Leen, T.G. Dietterich, V. Tresp (MIT Press, 2001, Cambridge), pp. 556–562
143. H.A. Le Thi, T.P. Dinh, H.M. Le, X.T. Vo, DC approximation approaches for sparse optimization. Eur. J. Oper. Res. **244**(1), 26–46 (2015)
144. R.M. Lewitt, G. Muehllehner, Accelerated iterative reconstruction for positron emission tomography based on the EM algorithm for maximum likelihood estimation. IEEE Trans. Med. Imaging **5**(1), 16–22 (1986)
145. Y. Li, S. Osher, Coordinate descent optimization for ℓ^1 minimization with application to compressed sensing; a greedy algorithm. Inverse Prob. Imaging **3**(3), 487–503 (2009)
146. M. Li, L. Zhou, Z. Yang, A. Li, F. Xia, D.G. Andersen, A. Smola, Parameter server for distributed machine learning, in *Big Learning NIPS Workshop*, vol. 6 (2013), p. 2
147. X. Lian, Y. Huang, Y. Li, J. Liu, Asynchronous parallel stochastic gradient for nonconvex optimization, in *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (Curran Associates, New York, 2015), pp. 2737–2745

148. T. Lipp, S. Boyd, Variations and extension of the convex-concave procedure. Optim. Eng. **17**(2), 263–287 (2016)
149. J. Liu, S.J. Wright, Asynchronous stochastic coordinate descent: parallelism and convergence properties. SIAM J. Optim. **25**(1), 351–376 (2015)
150. J. Liu, S.J. Wright, C. Ré, V. Bittorf, S. Sridhar, An asynchronous parallel stochastic coordinate descent algorithm. J. Mach. Learn. Res. **16**(1), 285–322 (2015)
151. Y. Lou, P. Yin, J. Xin, Point source super-resolution via non-convex l_1 based methods. J. Sci. Comput. **68**(3), 1082–1100 (2016)
152. T.A. Louis, Finding the observed information matrix when using the EM algorithm. J. R. Stat. Soc. Ser. B Methodol. **44**, 226–233 (1982)
153. J. Mairal, Incremental majorization-minimization optimization with application to large-scale machine learning. SIAM J. Optim. **25**(2), 829–855 (2015)
154. O.L. Mangasarian, *Machine Learning via Polyhedral Concave Minimization* (Physica-Verlag HD, Heidelberg, 1996), pp. 175–188
155. H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, M.I. Jordan, Perturbed iterate analysis for asynchronous stochastic optimization. arXiv:1507.06970 (2015, preprint)
156. M. Mardani, G. Mateos, G.B. Giannakis, Decentralized sparsity-regularized rank minimization: algorithms and applications. IEEE Trans. Signal Process. **61**(21), 5374–5388 (2013)
157. M. Mardani, G. Mateos, G.B. Giannakis, Dynamic anomalography: tracking network anomalies via sparsity and low rank. IEEE J. Sel. Top. Sign. Proces. **7**(1), 50–66 (2013)
158. G. Marjanovic, A.O. Hero III, l_0 sparse inverse covariance estimation. IEEE Trans. Signal Process. **63**(12), 3218–3231 (2015)
159. G. Marjanovic, M.O. Ulfarsson, A.O. Hero III, Mist: l_0 sparse linear regression with momentum, in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)* (2015), pp. 3551–3555
160. B.R. Marks, G.P. Wright, Technical note-A general inner approximation algorithm for nonconvex mathematical programs. Oper. Res. **26**(4), 681–683 (1978)
161. G. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, vol. 382 (Wiley, Hoboken, 2007)
162. L. Meier, S. van de Geer, P. Bühlmann, The group lasso for logistic regression. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **70**(1), 53–71 (2008)
163. I. Meilijson, A fast improvement to the EM algorithm on its own terms. J. R. Stat. Soc. Ser. B Methodol. **51**, 127–138 (1989)
164. H. Mine, M. Fukushima, A minimization method for the sum of a convex function and a continuously differentiable function. J. Optim. Theory Appl. **33**(1), 9–23 (1981)
165. A. Mokhtari, W. Shi, Q. Ling, A. Ribeiro, DQM: decentralized quadratically approximated alternating direction method of multipliers. IEEE Trans. Signal Process. **64**(19), 5158–5173 (2015)
166. A. Mokhtari, W. Shi, Q. Ling, A. Ribeiro, A decentralized second-order method with exact linear convergence rate for consensus optimization. IEEE Trans. Signal Inf. Process. Over Netw. **2**(4), 507–522 (2016)
167. I. Necoara, D. Clipici, Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC. J. Process Control **23**(3), 243–253 (2013)
168. A. Nedic, Asynchronous broadcast-based convex optimization over a network. IEEE Trans. Autom. Control **56**(6), 1337–1351 (2011)
169. A. Nedic, A. Olshevsky, Distributed optimization over time-varying directed graphs. IEEE Trans. Autom. Control **60**(3), 601–615 (2015)
170. A. Nedic, A. Ozdaglar, Distributed subgradient methods for multi-agent optimization. IEEE Trans. Autom. Control **54**(1), 48–61 (2009)
171. A. Nedić, D.P. Bertsekas, V.S. Borkar, Distributed asynchronous incremental subgradient methods. Stud. Comput. Math. **8**(C), 381–407 (2001)
172. A. Nedich, A. Olshevsky, W. Shi, Achieving geometric convergence for distributed optimization over time-varying graphs. arXiv:1607.03218 (2016, preprint)

173. R. Neelamani, C.E. Krohn, J.R. Krebs, J.K. Romberg, M. Deffenbaugh, J.E. Anderson, Efficient seismic forward modeling using simultaneous random sources and sparsity. Geophysics **75**(6), WB15–WB27 (2010)
174. Y. Nesterov, Gradient methods for minimizing composite functions. Math. Program. **140**(1), 125–161 (2013)
175. P. Netrapalli, P. Jain, S. Sanghavi, Phase retrieval using alternating minimization. IEEE Trans. Signal Process. **63**(18), 4814–4826 (2015)
176. I. Notarnicola, G. Notarstefano, A randomized primal distributed algorithm for partitioned and big-data non-convex optimization, in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC 2016)* (2016), pp. 153–158
177. I. Notarnicola, Y. Sun, G. Scutari, G. Notarstefano, Distributed big-data optimization via block communications in *IEEE International Conference on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2017)* (2017)
178. I. Notarnicola, Y. Sun, G. Scutari, G. Notarstefano, Distributed big-data optimization via block-iterative convexification and averaging, in *Proceedings of the IEEE 56th Conference on Decision and Control (CDC 2017)* (2017)
179. P. Oguz-Ekim, J.P. Gomes, J. Xavier, P. Oliveira, Robust localization of nodes and time-recursive tracking in sensor networks using noisy range measurements. IEEE Trans. Signal Process. **59**(8), 3930–3942 (2011)
180. B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1? Vis. Res. **37**(23), 3311–3325 (1997)
181. J. Ortega, W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Society for Industrial and Applied Mathematics, Philadelphia, 2000)
182. D.P. Palomar, J.R. Fonollosa, Practical algorithms for a family of waterfilling solutions. IEEE Trans. Signal Process. **53**(2), 686–695 (2005)
183. J. Papandriopoulos, J.S. Evans, SCALE: a low-complexity distributed protocol for spectrum balancing in multiuser DSL networks. IEEE Trans. Inf. Theory **55**(8), 3711–3724 (2009)
184. N. Parikh, S. Boyd, Proximal algorithms. Found. Trends Optim. **1**(3), 127–239 (2014)
185. M. Patriksson, Cost approximation: a unified framework of descent algorithms for nonlinear programs. SIAM J. Optim. **8**(2), 561–582 (1998)
186. F. Pedregosa, R. Leblond, S. Lacoste-Julien, Breaking the nonsmooth barrier: a scalable parallel method for composite optimization, in *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (Curran Associates, Nwe Yotrk, 2017), pp. 56–65
187. Z. Peng, M. Yan, W. Yin, Parallel and distributed sparse optimization, in *Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers* (2013), pp. 659–646
188. Z. Peng, Y. Xu, M. Yan, W. Yin, ARock: an algorithmic framework for asynchronous parallel coordinate updates. SIAM J. Sci. Comput. **38**(5), A2851–A2879 (2016)
189. Z. Peng, Y. Xu, M. Yan, W. Yin, On the convergence of asynchronous parallel iteration with arbitrary delays. arXiv:1612.04425 (2016, preprint)
190. T. Qiu, P. Babu, D.P. Palomar, PRIME: phase retrieval via majorization-minimization. IEEE Trans. Signal Process. **64**(19), 5174–5186 (2016)
191. G. Qu, N. Li, Harnessing smoothness to accelerate distributed optimization. IEEE Trans. Control Netw Syst. (2017)
192. T.D. Quoc, M. Diehl, Sequential convex programming methods for solving nonlinear optimization problems with DC constraints. arXiv:1107.5841 (2011, preprint)
193. B.D. Rao, K. Kreutz-Delgado, An affine scaling methodology for best basis selection. IEEE Trans. Signal Process. **47**(1), 187–200 (1999)
194. M. Razaviyayn, M. Hong, Z.-Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J. Optim. **23**(2), 1126–1153 (2013)
195. M. Razaviyayn, H.-W. Tseng, Z.-Q. Luo, Dictionary learning for sparse representation: complexity and algorithms, in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)* (2014), pp. 5247–5251

196. B. Recht, C. Re, S. Wright, F. Niu, Hogwild: a lock-free approach to parallelizing stochastic gradient descent, in *Advances in Neural Information Processing Systems 24 (NIPS 2011)* (Curran Associates, New York, 2011), pp. 693–701.
197. P. Richtárik, M. Takáč, On optimal probabilities in stochastic coordinate descent methods. Optim. Lett. **10**(6), 1233–1243 (2016)
198. P. Richtárik, M. Takáč, Parallel coordinate descent methods for big data optimization. Math. Program. **156**(1–2), 433–484 (2016)
199. P. Richtárik, M. Takáč, S.D. Ahipaşaoğlu, Alternating maximization: unifying framework for 8 sparse PCA formulations and efficient parallel codes. arXiv:1212.4137 (2012, preprint)
200. S. Robinson, R. Day, A sufficient condition for continuity of optimal sets in mathematical programming. J. Math. Anal. Appl. **45**(2), 506–511 (1974)
201. R. Salakhutdinov, S.T. Roweis, Adaptive overrelaxed bound optimization methods, in *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)* (2003), pp. 664–671
202. A.H. Sayed, Adaptation, learning, and optimization over networks. Found. Trends Mach. Learn. **7**(4–5), 311–801 (2014)
203. A.H. Sayed, Diffusion adaptation over networks, in *Academic Press Library in Signal Processing*, vol. 3 (Elsevier, Netherlands, 2014), pp. 323–453
204. D.S. Scherber, H.C. Papadopoulos, Locally constructed algorithms for distributed computations in ad-hoc networks, in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks* (2004), pp. 11–19
205. C. Scherrer, A. Tewari, H. Mahantesh, H. David, Feature clustering for accelerating parallel coordinate descent, in *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (Curran Associates, New York, 2012), pp. 28–36
206. I.D. Schizas, Distributed informative-sensor identification via sparsity-aware matrix decomposition. IEEE Trans. Signal Process. **61**(18), 4610–4624 (2013)
207. D.A. Schmidt, C. Shi, R.A. Berry, M.L. Honig, W. Utschick, Distributed resource allocation schemes. IEEE Signal Process. Mag. **26**(5), 53–63 (2009)
208. G. Scutari, Y. Sun, Distributed nonconvex constrained optimization over time-varying digraphs (2018, Preprint). arXiV:1809.01106
209. G. Scutari, F. Facchinei, J.-S. Pang, D.P. Palomar, Real and complex monotone communication games. IEEE Trans. Inf. Theory **60**(7), 4197–4231 (2014)
210. G. Scutari, F. Facchinei, P. Song, D.P. Palomar, J.-S. Pang, Decomposition by partial linearization: parallel optimization of multi-agent systems. IEEE Trans. Signal Process. **62**(3), 641–656 (2014)
211. G. Scutari, F. Facchinei, L. Lampariello, Parallel and distributed methods for constrained nonconvex optimization-Part I: theory. IEEE Trans. Signal Process. **65**(8), 1929–1944 (2017)
212. G. Scutari, F. Facchinei, L. Lampariello, S. Sardellitti, P. Song, Parallel and distributed methods for constrained nonconvex optimization-Part II: applications in communications and machine learning. IEEE Trans. Signal Process. **65**(8), 1945–1960 (2017)
213. F. Sha, L.K. Saul, D.D. Lee, Multiplicative updates for nonnegative quadratic programming in support vector machines, in *Advances in Neural Information Processing Systems 15 (NIPS 2002)* (MIT Press, Cambridge, 2003), pp. 1065–1072
214. S.K. Shevade, S.S. Keerthi, A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics **19**(17), 2246–2253 (2003)
215. Q. Shi, M. Razaviyayn, Z.-Q. Luo, C. He, An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel. IEEE Trans. Signal Process. **59**(9), 4331–4340 (2011)
216. W. Shi, Q. Ling, G. Wu, W. Yin, EXTRA: an exact first-order algorithm for decentralized consensus optimization. SIAM J. Optim. **25**(2), 944–966 (2015)
217. W. Shi, Q. Ling, G. Wu, W. Yin, A proximal gradient algorithm for decentralized composite optimization. IEEE Trans. Signal Process. **63**(22), 6013–6023 (2015)
218. H.-J.M. Shi, S. Tu, Y. Xu, W. Yin, A primer on coordinate descent algorithms. arXiv:1610.00040 (2016, preprint)

219. N.D. Sidiropoulos, T.N. Davidson, Z.-Q. Luo, Transmit beamforming for physical-layer multicasting. IEEE Trans. Signal Process. **54**(6), 2239–2251 (2006)
220. A.P. Singh, G.J. Gordon, A unified view of matrix factorization models, in *Machine Learning and Knowledge Discovery in Databases* (Springer, Berlin, 2008)
221. P. Song, Numerical comparison of decomposition algorithms for nonconvex sum-utility problems. Technical report, Department of Electrical Engineering, University at Buffalo, the State University of New York, Jan. 2013
222. J. Song, P. Babu, D.P. Palomar, Optimization methods for designing sequences with low autocorrelation sidelobes. IEEE Trans. Signal Process. **63**(15), 3998–4009 (2015)
223. J. Song, P. Babu, D.P. Palomar, Sparse generalized eigenvalue problem via smooth optimization. IEEE Trans. Signal Process. **63**(7), 1627–1642 (2015)
224. B.K. Sriperumbudur, D.A. Torres, G.R.G. Lanckriet, A majorization-minimization approach to the sparse generalized eigenvalue problem. Mach. Learn. **85**(1–2), 3–39 (2011)
225. K. Srivastava, A. Nedić, Distributed asynchronous constrained stochastic optimization. IEEE J. Sel. Top. Sign. Proces. **5**(4), 772–790 (2011)
226. P. Stoica, Y. Selén, Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: a refresher. IEEE Signal Process. Mag. **21**(1), 112–114 (2004)
227. P. Stoica, P. Babu, J. Li, SPICE: a sparse covariance-based estimation method for array processing. IEEE Trans. Signal Process. **59**(2), 629–638 (2011)
228. Y. Sun, G. Scutari, Distributed nonconvex optimization for sparse representation, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017)* (2017), pp. 4044–4048
229. Y. Sun, P. Babu, D.P. Palomar, Robust estimation of structured covariance matrix for heavy-tailed elliptical distributions. IEEE Trans. Signal Process. **64**(14), 3576–3590 (2016)
230. Y. Sun, P. Babu, D.P. Palomar, Majorization-minimization algorithms in signal processing, communications, and machine learning. IEEE Trans. Signal Process. **65**(3), 794–816 (2017)
231. Y. Sun, A. Daneshmand, G. Scutari, Convergence rate of distributed convex and nonconvex optimization methods with gradient tracking. Technical report, Purdue University, March 2018
232. Y. Sun, G. Scutari, D.P. Palomar, Distributed nonconvex multiagent optimization over time-varying networks, in *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers* (2016), pp. 788–794
233. T. Tatarenko, B. Touri, Non-convex distributed optimization. IEEE Trans. Autom. Control **62**(8), 3744–3757 (2017)
234. Y. Tian, Y. Sun, B. Du, G. Scutari, ASY-SONATA: achieving geometric convergence for distributed asynchronous optimization. arXiv:1803.10359 (2018, preprint)
235. R. Tibshirani, Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B Methodol. **58**, 267–288 (1996)
236. P.A. Traganitis, K. Slavakis, G.B. Giannakis, Sketch and validate for big data clustering. IEEE J. Sel. Top. Sign. Proces. **9**(4), 678–690 (2015)
237. P. Tseng, On the rate of convergence of a partially asynchronous gradient projection algorithm. SIAM J. Optim. **1**(4), 603–619 (1991)
238. P. Tseng, S. Yun, A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**(1–2), 387–423 (2009)
239. K. Tsianos, S. Lawlor, M.G. Rabbat, Communication/computation tradeoffs in consensus-based distributed optimization, in *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (Curran Associates, New York, 2012), pp. 1943–1951
240. J. Tsitsiklis, Problems in decentalized decision making and computation. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, 1984
241. F. Vaida, Parameter convergence for EM and MM algorithms. Stat. Sin. **15**, 831–840 (2005)
242. R. Varadhan, C. Roland, Simple and globally convergent methods for accelerating the convergence of any EM algorithm. Scand. J. Stat. **35**(2), 335–353 (2008)
243. S. Vasudevan, M. Adler, D. Goeckel, D. Towsley, Efficient algorithms for neighbor discovery in wireless networks. IEEE/ACM Trans. Netw. **21**(1), 69–83 (2013)

244. J. von Neumann, Some matrix-inequalities and metrization of matric-space. Tomsk Univ. Rev. **1**, 286–300 (1937)
245. H.T. Wai, J. Lafond, A. Scaglione, E. Moulines, Decentralized Frank-Wolfe algorithm for convex and non-convex problems. IEEE Trans. Autom. Control **62**(11), 5522–5537 (2017)
246. Z. Wang, G.B. Giannakis, Wireless multicarrier communications. IEEE Signal Process. Mag. **17**(3), 29–48 (2000)
247. E. Wei, A. Ozdaglar, On the $o(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers, in *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013)* (2013), pp. 551–554
248. J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, Use of the zero-norm with linear models and kernel methods. J. Mach. Learn. Res. **3**, 1439–1461 (2003)
249. D.P. Wipf, B.D. Rao, Sparse Bayesian learning for basis selection. IEEE Trans. Signal Process. **52**(8), 2153–2164 (2004)
250. S.J. Wright, Coordinate descent algorithms. Math. Program. **151**(1), 3–34 (2015)
251. S.J. Wright, R.D. Nowak, M.A.T. Figueiredo, Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. **57**(7), 2479–2493 (2009)
252. C.F.J. Wu, On the convergence properties of the EM algorithm. Ann. Stat. **11**(1), 95–103 (1983)
253. T. Wu, K. Lange, The MM alternative to EM. Stat. Sci. **25**(4), 492–505 (2010)
254. L. Wu, P. Babu, D.P. Palomar, Cognitive radar-based sequence design via SINR maximization. IEEE Trans. Signal Process. **65**(3), 779–793 (2017)
255. T. Wu, K. Yuan, Q. Ling, W. Yin, A.H. Sayed, Decentralized consensus optimization with asynchrony and delays, in *Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers* (2016), pp. 992–996
256. C. Xi, U.A. Khan, DEXTRA: a fast algorithm for optimization over directed graphs. IEEE Trans. Autom. Control **62**(10), 4980–4993 (2017)
257. C. Xi, R. Xin, U.A. Khan, ADD-OPT: accelerated distributed directed optimization. IEEE Trans. Autom. Control **63**(5), 1329–1339 (2018)
258. L. Xiao, S. Boyd, S.-J. Kim, Distributed average consensus with least-mean-square deviation. J. Parallel Distrib. Comput. **67**(1), 33–46 (2007)
259. J. Xu, S. Zhu, Y.C. Soh, L. Xie, Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes, in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015)* (2015), pp. 2055–2060
260. Y. Yang, G. Scutari, D.P. Palomar, M. Pesavento, A parallel stochastic optimization method for nonconvex multi-agent optimization problems. IEEE Trans. Signal Process. **64**(11), 2949–2964 (2016)
261. P. Yin, Y. Lou, Q. He, J. Xin, Minimization of ℓ_{1-2} for compressed sensing. SIAM J. Sci. Comput. **37**(1), A536–A563 (2015)
262. M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **68**(1), 49–67 (2006)
263. X.-T. Yuan, T. Zhang, Truncated power method for sparse eigenvalue problems. J. Mach. Learn. Res. **14**, 899–925 (2013)
264. A.L. Yuille, R. Anand, The concave-convex procedure (CCCP), in *Advances in Neural Information Processing Systems 14 (NIPS 2001)* (MIT Press, Cambridge, 2002), pp. 1033–1040
265. H. Yun, H.-F. Yu, C.-J. Hsieh, S.V.N. Vishwanathan, I. Dhillon, NOMAD: non-locking, stochastic multi-machine algorithm for asynchronous and decentralized matrix completion, in *Proceedings of the VLDB Endowment*, vol. 7 (2014), pp. 975–986
266. Y. Zhang, L.E. Ghaoui, Large-scale sparse principal component analysis with application to text data, in *Advances in Neural Information Processing Systems 24 (NIPS 2011)* (Curran Associates, New York, 2011), pp. 532–539
267. Y. Zhang, X. Lin, DiSCO: distributed optimization for self-concordant empirical loss, in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)* (2015), pp. 362–370

268. S. Zhang, J. Xin, Minimization of transformed l_1 penalty: theory, difference of convex function algorithm, and robust application in compressed sensing. arXiv:1411.5735 (2014, preprint)
269. Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, A survey of sparse representation: algorithms and applications. IEEE Access **3**, 490–530 (2015)
270. X. Zhao, A.H. Sayed, Asynchronous adaptation and learning over networks-Part I: modeling and stability analysis. IEEE Trans. Signal Process. **63**(4), 811–826 (2015)
271. L. Zhao, W.-Z. Song, X. Ye, Fast decentralized gradient descent method and applications to in-situ seismic tomography, in *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data)* (2015), pp. 908–917
272. L. Zhao, J. Song, P. Babu, D.P. Palomar, A unified framework for low autocorrelation sequence design via majorization-minimization. IEEE Trans. Signal Process. **65**(2), 438–453 (2017)
273. H. Zhou, D. Alexander, K. Lange, A quasi-Newton acceleration for high-dimensional optimization algorithms. Stat. Comput. **21**(2), 261–273 (2011)
274. H. Zhou, L. Hu, J. Zhou, K. Lange, MM algorithms for variance components models. arXiv:1509.07426 (2015, preprint)
275. M. Zhu, S. Martínez, An approximate dual subgradient algorithm for multi-agent non-convex optimization. IEEE Trans. Autom. Control **58**(6), 1534–1539 (2013)

LECTURE NOTES IN MATHEMATICS Springer

Editorial Policy

1. Lecture Notes aim to report new developments in all areas of mathematics and their applications – quickly, informally and at a high level. Mathematical texts analysing new developments in modelling and numerical simulation are welcome.

   Manuscripts should be reasonably self-contained and rounded off. Thus they may, and often will, present not only results of the author but also related work by other people. They may be based on specialised lecture courses. Furthermore, the manuscripts should provide sufficient motivation, examples and applications. This clearly distinguishes Lecture Notes from journal articles or technical reports which normally are very concise. Articles intended for a journal but too long to be accepted by most journals, usually do not have this "lecture notes" character. For similar reasons it is unusual for doctoral theses to be accepted for the Lecture Notes series, though habilitation theses may be appropriate.

2. Besides monographs, multi-author manuscripts resulting from SUMMER SCHOOLS or similar INTENSIVE COURSES are welcome, provided their objective was held to present an active mathematical topic to an audience at the beginning or intermediate graduate level (a list of participants should be provided).

   The resulting manuscript should not be just a collection of course notes, but should require advance planning and coordination among the main lecturers. The subject matter should dictate the structure of the book. This structure should be motivated and explained in a scientific introduction, and the notation, references, index and formulation of results should be, if possible, unified by the editors. Each contribution should have an abstract and an introduction referring to the other contributions. In other words, more preparatory work must go into a multi-authored volume than simply assembling a disparate collection of papers, communicated at the event.

3. Manuscripts should be submitted either online at www.editorialmanager.com/lnm to Springer's mathematics editorial in Heidelberg, or electronically to one of the series editors. Authors should be aware that incomplete or insufficiently close-to-final manuscripts almost always result in longer refereeing times and nevertheless unclear referees' recommendations, making further refereeing of a final draft necessary. The strict minimum amount of material that will be considered should include a detailed outline describing the planned contents of each chapter, a bibliography and several sample chapters. Parallel submission of a manuscript to another publisher while under consideration for LNM is not acceptable and can lead to rejection.

4. In general, **monographs** will be sent out to at least 2 external referees for evaluation.

   A final decision to publish can be made only on the basis of the complete manuscript, however a refereeing process leading to a preliminary decision can be based on a pre-final or incomplete manuscript.

   Volume Editors of **multi-author works** are expected to arrange for the refereeing, to the usual scientific standards, of the individual contributions. If the resulting reports can be forwarded to the LNM Editorial Board, this is very helpful. If no reports are forwarded or if other questions remain unclear in respect of homogeneity etc, the series editors may wish to consult external referees for an overall evaluation of the volume.

5. Manuscripts should in general be submitted in English. Final manuscripts should contain at least 100 pages of mathematical text and should always include

   – a table of contents;
   – an informative introduction, with adequate motivation and perhaps some historical remarks: it should be accessible to a reader not intimately familiar with the topic treated;
   – a subject index: as a rule this is genuinely helpful for the reader.
   – For evaluation purposes, manuscripts should be submitted as pdf files.

6. Careful preparation of the manuscripts will help keep production time short besides ensuring satisfactory appearance of the finished book in print and online. After acceptance of the manuscript authors will be asked to prepare the final LaTeX source files (see LaTeX templates online: https://www.springer.com/gb/authors-editors/book-authors-editors/manuscriptpreparation/5636) plus the corresponding pdf- or zipped ps-file. The LaTeX source files are essential for producing the full-text online version of the book, see http://link.springer.com/bookseries/304 for the existing online volumes of LNM). The technical production of a Lecture Notes volume takes approximately 12 weeks. Additional instructions, if necessary, are available on request from lnm@springer.com.

7. Authors receive a total of 30 free copies of their volume and free access to their book on SpringerLink, but no royalties. They are entitled to a discount of 33.3 % on the price of Springer books purchased for their personal use, if ordering directly from Springer.

8. Commitment to publish is made by a *Publishing Agreement*; contributing authors of multiauthor books are requested to sign a *Consent to Publish form*. Springer-Verlag registers the copyright for each volume. Authors are free to reuse material contained in their LNM volumes in later publications: a brief written (or e-mail) request for formal permission is sufficient.

Addresses: