Distributed Optimization for Control

Article *in* Annual Review of Control Robotics and Autonomous Systems · May 2018

DOI: 10.1146/annurev-control-060117-105131

Angelia Nedic
Arizona State University
261 PUBLICATIONS **16,053** CITATIONS
SEE PROFILE

Ji Liu
Stony Brook University
161 PUBLICATIONS **1,692** CITATIONS
SEE PROFILE

ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Distributed Optimization for Control

Angelia Nedić[1] and Ji Liu[2]

[1] School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA; email: angelia.nedich@asu.edu

[2] Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, New York 11794, USA; email: ji.liu@stonybrook.edu

Annu. Rev. Control Robot. Auton. Syst. 2018. 1:77–103

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-060117-105131

Keywords

distributed optimization, multi-agent systems, agent networks

Abstract

Advances in wired and wireless technology have necessitated the development of theory, models, and tools to cope with the new challenges posed by large-scale control and optimization problems over networks. The classical optimization methodology works under the premise that all problem data are available to a central entity (a computing agent or node). However, this premise does not apply to large networked systems, where each agent (node) in the network typically has access only to its private local information and has only a local view of the network structure. This review surveys the development of such distributed computational models for time-varying networks. To emphasize the role of the network structure in these approaches, we focus on a simple direct primal (sub)gradient method, but we also provide an overview of other distributed methods for optimization in networks. Applications of the distributed optimization framework to the control of power systems, least squares solutions to linear equations, and model predictive control are also presented.

1. INTRODUCTION

Recent advances in wired and wireless technology have led to the emergence of large-scale networks, including the Internet, mobile ad hoc networks, and wireless sensor networks. These networks have, in turn, given rise to new network application domains, such as data-based networks, robotic networks, unmanned aerial vehicle systems, social and economic networks, smart power networks, and epidemic networks. Such applications often require decentralized in-network control and optimization techniques to support various operations, including resource allocation, coordination, learning, and estimation.

As a result, there is a need to develop new models and tools for the design and performance analysis of large, complex networked systems. The problems arising in such networks stem mainly from two aspects: a lack of central authority or coordinator (master node) and an inherent dynamic of the network connectivity structure. The lack of central authority in a network system naturally requires a decentralized architecture for operations over the network (as in the case of the Internet). In some applications, the decentralized architecture is often preferred over a centralized architecture for several reasons: (*a*) The size of the network (the number of agents) and the resources needed to coordinate (i.e., communicate with) a large number of agents make a centralized architecture impractical, (*b*) a centralized network architecture is not robust to the failure of the central entity, and (*c*) the privacy of agent information often cannot be preserved in a centralized systems. Additional challenges in decentralized operations over such networks arise from the network connectivity structure, which can vary over time owing to unreliable communication links or the mobility of the network agents. The challenge is to control, coordinate, and analyze the performance of such networks.

Over the past decade, there has been considerable interest in distributed computation and decision-making problems; most notable among these are consensus and flocking problems, multiagent coverage problems, rendezvous problems, localization of sensors in a multi-sensor network, and distributed management of multi-agent formations. More recently, substantial efforts have been made to develop distributed optimization algorithms that can be deployed in such networks without a central coordinator, but with the ability to exploit the network connectivity to achieve a global network performance objective. Such algorithms have the following properties: (*a*) They rely only on local information and observations (i.e., the agents can exchange some limited information with their one-hop neighbors only), (*b*) they are robust to the changes in the network topology (the topology is not necessarily static, as the communication links may not function perfectly), and (*c*) they are easily implementable in the sense that the local computations performed by the agents are not expensive.

We next provide some examples of networks and control applications in such networks.

> **Example 1 (sensor networks).** Sensor networks are a new computing concept based on a system of small sensors (also referred to as motes or smart dust sensors) that have some computational, sensing, and communication capabilities. They can be used in many different ways, e.g., to monitor the health structure of buildings and bridges (smart structures) or the load on a power grid (smart grids). A more concrete problem of interest that supports several applications in sensor networks, such as building a piecewise approximation of a coverage area, enabling multi-sensor target localization, and solving tracking problems, is the problem of determining 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 that sensor among all other sensors (1). After determining 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 networks). In a peer-to-peer network consisting of m nodes, each node i has its local data or files stored with average size θ_i, which is known to node i only. The nodes are connected over a static undirected network, 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 (2–8). An optimization formulation of the problem is $\min_{x\in\mathbb{R}} \sum_{i=1}^{m}(x-\theta_i)^2$, which 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 must compute it in a network in a distributed manner using only local communication. 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 planar region with the same speed but different headings (they are refracted from the boundary to prevent them from leaving the area) (5, 9). The objective is to design a local protocol that will ensure the alignment of the agent headings while the agent communications are constrained by a given maximal distance.

Additional examples of distributed problems can be found in recent books and monographs on robotic networks (e.g., 10–12). In the following sections, we often refer to networks as graphs, and we use the terms agent and node interchangeably. The remainder of the article is organized as follows. In Section 2, we formally describe a multi-agent optimization problem in a network and discuss some related aspects of the consensus protocol. Section 3 presents a distributed subgradient method for solving the multi-agent optimization problem. Section 4 provides an overview of related literature, including the most recent research directions. Section 5 contains the applications of distributed optimization for control of power networks and model predictive control.

2. THE DISTRIBUTED OPTIMIZATION MODEL

This section provides a formal multi-agent optimization problem description and gives the underlying assumptions of the multi-agent problem. The agents are embedded in a communication graph that accommodates distributed computations through the use of consensus protocols.

2.1. The Problem and Assumptions

We focus on solving distributed optimization problems of the generic form

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

in a network of m agents, where each function f_i is known only to agent i, whereas the constraint set $X \subseteq \mathbb{R}^n$ is known to all agents. We assume that the problem in Equation 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 the problem in Equation 1 has a solution. In such cases, we let f^* denote the optimal value of the problem and X^* denote the set of its solutions, i.e.,

$f^* = \min_{x \in X} f(x)$ and $X^* = \{x^* \in X \mid f(x^*) = f^*\}$. We work with the Euclidean norm, denoted by $\|\cdot\|$, unless explicitly stated otherwise. We use $\langle \cdot, \cdot \rangle$ to denote the inner product. We view all vectors as column vectors unless stated otherwise. We use a prime to denote the transpose of a matrix and a vector. We assume that the agents are embedded in a communication network, which allows them to exchange some limited information with their immediate (one-hop) neighbors. Multi-hop communications are not allowed in this setting. The goal of the multi-agent system is to collaboratively solve the problem in Equation 1. 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., $[m] = \{1, \ldots, m\}$, and 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: $N_i(k) = \{j \in [m] \mid i \leftrightarrow j \in E_k\} \cup \{i\}$. The neighbor set $N_i(k)$ includes agent i itself because agent i always has access to its own information.

The agents' desire to solve Equation 1 jointly through local communication translates to the following problem that agents face at time k:

$$\begin{aligned} &\text{minimize} && \mathbf{f}(x_1, \ldots, x_m) \quad \text{with} \quad \mathbf{f}(x_1, \ldots, x_m) = \sum_{i=1}^m f_i(x_i) \\ &\text{subject to} && x_i = x_j \text{ for all } j \in N_i(k) \text{ and all } i \in [m], \\ &&& x_i \in X \text{ for all } i \in [m]. \end{aligned} \qquad 2.$$

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 this end, we impose the following assumption on the graphs G_k.

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 adopt Assumption 2.

Let C_k be the constraint set of the problem in Equation 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, one can note that the constraint sets C_k are all the same, i.e., for all k,

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

Thus, the optimization problem in Equation 2 is over a static constraint set C. However, the algebraic description of set C is given to the agents through a different set of equations at different time instances owing to the variable communication graph structure.

In view of the preceding discussion, one can formally associate a limit problem with the sequence of problems in Equation 2, where the limit problem is defined by the following:

$$\begin{aligned} &\text{minimize} && \mathbf{f}(x_1, \ldots, x_m) \quad \text{with} \quad \mathbf{f}(x_1, \ldots, x_m) = \sum_{i=1}^m f_i(x_i) \\ &\text{subject to} && (x_1, \ldots, x_m) \in \cap_{k=1}^{\infty} C_k. \end{aligned} \qquad 3.$$

As noted above, all sets C_k are the same under Assumption 2. However, we keep the notation C_k to capture the fact that the agents have a different set of equations describing the constraint set C at different times. The preceding formulation of the limit problem is suitable for modeling more general problems with time-varying constraints. In particular, it captures the situation where the graphs G_k are not necessarily connected but their unions over B consecutive time instances are connected [there exists a positive integer B such that the graph $([m], E_{\ell B} \cup \cdots \cup E_{(\ell+1)B-1})$ is connected for all $\ell = 0, 1, \ldots$].

2.2. The Consensus Problem and Algorithm

Consensus and distributed averaging are fundamental problems in distributed control and computation that play an important role in many other problems, such as Google's PageRank, clock synchronization, rendezvous, and distributed estimation. It turns out that the consensus problem is a special case of the limit problem in Equation 3, where each $f_i \equiv 0$ and $X = \mathbb{R}^n$. Concretely, the consensus problem is as follows:

$$\begin{aligned} &\text{minimize} && 0 \\ &\text{subject to} && (x_1, \ldots, x_m) \in \cap_{k=1}^{\infty} C_k, \end{aligned} \qquad 4.$$

with $C_k = \{(x_1, \ldots, x_m) \mid x_i = x \text{ for some } x \in \mathbb{R}^n \text{ and all } i \in [m]\}$ for all $k \geq 0$. The consensus problem is a feasibility problem where the agents collectively determine a vector $\mathbf{x} = (x_1, \ldots, x_m)$ satisfying the constraint in Equation 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:

$$\min_{x \in \mathbb{R}^n} \sum_{j \in N_i(k)} p_{ij}(k) \|x - x_j\|^2, \qquad 5.$$

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 $j \in N_i(k)$ (in fact, x_j can depend on time k—i.e., we can have $x_j = x_j(k)$—but we suppress this dependence on k since it is nonessential for our discussion). This problem can be viewed as a penalty problem associated with the constraints in the set C_k that involve only the agent i decision variable. The objective function in Equation 5 is strongly convex (in x), so the minimization problem in Equation 5 has a unique solution, denoted by $\hat{x}_i(k)$, i.e.,

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

By setting the gradient of the function $\sum_{j \in N_i(k)} p_{ij}(k) \|x - x_j\|^2$ (with respect to x) to zero, we can find the solution $\hat{x}_i(k)$ in a closed form. Specifically, the solution $\hat{x}_i(k)$ is given by

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

which shows that $\hat{x}_i(k)$ is a convex combination (or weighted average) of the points x_j, $j \in N_i(k)$.

Alternatively, a penalty problem associated with agent i feasible set C_{ik} at time k can also be considered in the following form:

$$\min_{x \in \mathbb{R}^n} \sum_{j \in N_i(k)} w_{ij}(k) \|x - x_j\|^2, \qquad 6.$$

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

$$w_{ij}(k) > 0 \quad \text{for all } j \in N_i(k), \qquad \sum_{j \in N_i(k)} w_{ij}(k) = 1. \tag{7.}$$

For the problems in Equations 5 and 6 to be equivalent, we can set $w_{ij}(k) = p_{ij}(k)/\sum_{j \in N_i(k)} p_{ij}(k)$. In this case, the corresponding solution $\hat{x}_i(k)$ that solves both problems is given by $\hat{x}_i(k) = \sum_{j \in N_i(k)} w_{ij}(k) x_j$.

The preceding discussion motivates the following simple algorithm, known as a consensus algorithm (with projections), for solving the constrained consensus problem in Equation 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. Every agent i then 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

$$x_i(k+1) = \sum_{j=1}^{m} w_{ij}(k) x_j(k) \qquad \text{for all } i \in [m] \text{ and all } k \geq 0. \tag{8.}$$

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

We note that if a (convex) constraint set $X \subseteq \mathbb{R}^n$ is known to all agents, then the constrained consensus problem in Equation 4 can also be solved by the consensus algorithm in Equation 8 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 Equation 7) and that 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 Equation 8 has attracted much attention since the work by Jadbabaie et al. (5); Section 4 provides an overview of the consensus-related literature.

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 Equation 7. To state one such assumption, we introduce some additional terminology and notation. We let $W(k)$ be the matrix with ijth entry equal to $w_{ij}(k)$. We say that a matrix W is (row) stochastic if its entries are nonnegative and the sum of its entries in each row is equal to 1. We 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:

- $W(k)$ is doubly stochastic.
- $W(k)$ is compatible with the structure of the graph G_k, i.e., $w_{ij}(k) = 0$ if $i \leftrightarrow j \notin E_k$.
- $W(k)$ has positive diagonal entries, i.e., $w_{ii}(k) > 0$ for all $i \in [m]$.
- There is an $\eta > 0$ such that $w_{ii}(k) \geq \eta$ for all $i \in [m]$ and $w_{ij}(k) \geq \eta$ if $i \leftrightarrow j \in E_k$.

Assumptions 2 and 3 guarantee that the consensus is reached when the matrices are time varying. (Assumption 3 is stronger than what is typically assumed to ensure the consensus.) In fact, under these assumptions, we have that $\lim_{k\to\infty}[W(k)W(k-1)\cdots W(s)] = \frac{1}{m}\mathbf{1}\mathbf{1}'$ 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 1 (from Reference 13, lemma 5). Let the graph sequence $\{G_k\}$ satisfy Assumption 2, and let the matrix sequence $\{W(k)\}$ satisfy Assumption 3. Then, for all $s \geq 0$ and $k \geq s$, we have

$$\left([W(k)W(k-1)\cdots W(s+1)W(s)]_{ij} - \frac{1}{m}\right)^2 \leq \left(1 - \frac{\eta}{2m^2}\right)^{k-s} \qquad \text{for all } i, j \in [m].$$

Lemma 1 provides a key insight into the behavior of the products of the matrices $W(k)$, which implies that the consensus method in Equation 8 converges geometrically to $\frac{1}{m}\sum_{i=1}^{m} x_i(0)$.

3. THE DISTRIBUTED SUBGRADIENT METHOD

3.1. A Synchronous Algorithm

We now consider a distributed algorithm for solving the problem in Equation 3. We assume that the set X is closed and convex and has a simple structure for the projection operation (i.e., determining the projection of a point x on the set X is not computationally expensive). The idea is to construct an algorithm that can be executed locally by each agent i and that at every instant k involves two steps: one step aimed at aligning its iterate with its neighbor's iterates and one aimed at minimizing its objective cost f_i over the set X. Thus, the first step is similar to the consensus update in Equation 8, while the second is a 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., $\delta_X(x) = 0$ if $x \in X$, and $\delta_X(x) = +\infty$ otherwise—and the weights $w_{ij}(k)$, $j \in N_i(k)$, are convex combinations (see Equation 7).

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 $\min_{x\in\mathbb{R}^n}\{\langle\nabla f_i(\hat{x}_i(k)), x\rangle + \delta_X(x) + \frac{1}{2\alpha_k}\|x - \hat{x}_i(k)\|^2\}$, 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 step size. 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., $\Pi_X[z] = \text{argmin}_{x\in X}\|x - z\|^2$ 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 $h(x) + \langle g(x), y - x\rangle \leq h(y)$ for all $y \in \mathbb{R}^n$. In what follows, we 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 $z = \hat{x}_i(k)$.

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 $j \in N_i(k)$. It then updates as follows:

$$\begin{aligned} y_i(k+1) &= \sum_{j\in N_i(k)} w_{ij}(k)x_j(k), \\ x_i(k+1) &= \Pi_X[y_i(k+1) - \alpha_{k+1}g_i(k+1)], \end{aligned} \qquad 9.$$

where $\alpha_{k+1} > 0$ is a step size 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 $i \in [m]$. By introducing 0-weights for nonexisting links in the graph G_k—i.e., by defining $w_{ij}(k) = 0$ when $j \notin N_i(k)$—we can rewrite the method in Equation 9 as follows: for all $k \geq 0$ and all $i \in [m]$,

$$y_i(k+1) = \sum_{j=1}^{m} w_{ij}(k) x_j(k),$$
$$x_i(k+1) = \Pi_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)]. \qquad 10.$$

3.2. Convergence Analysis

In this section, we provide a main convergence result (Theorem 1) showing that the iterates $x_i(k)$ for all agents $i \in [m]$ converge to a solution of the problem in Equation 1 as $k \to \infty$. The proof of this result relies on a basic relation satisfied by the iterates of the algorithm, as given below in Proposition 1. The proof of this proposition is constructed through several auxiliary results that are provided below in Lemmas 2–4. Specifically, Lemma 2 provides an elementary relation for the iterates $x_i(k)$ for a single agent without the use of the network aspect. By viewing Equation 10 as a perturbation of the consensus algorithm, Lemma 3 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 3 is refined in Lemma 4 by taking into account that the perturbation to the consensus algorithm comes from a subgradient influence controlled by a step-size choice.

3.2.1. Relation for single-agent iterates.

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

$$\|\Pi_X[z] - x\|^2 \leq \|z - x\|^2 - \|\Pi_X[z] - z\|^2 \qquad \text{for all } x \in X \text{ and all } z \in \mathbb{R}^n. \qquad 11.$$

The preceding projection relation follows from a more general relation given by Facchinei & Pang (14, vol. II, p. 1120, lemma 12.1.13).

Lemma 2. Let the problem be convex (Assumption 1 holds) and let $\alpha_{k+1} > 0$. Then, for the iterate $x_i(k+1)$ of Equation 10, we have for all $x \in X$ and all $i \in [m]$

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

Proof. From the projection relation in Equation 11 and the definition of $x_i(k+1)$, we obtain for any $x \in X$

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

By expanding the squared-norm terms, we further have

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

Since $g_i(k+1)$ is a subgradient of f_i at $y_i(k+1)$, by the 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 can be estimated by using the Cauchy–Schwarz inequality, in order to obtain

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

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

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

3.2.2. 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 then be used in Lemma 2 to obtain some insights into the behavior of $\|x_i(k) - x^*\|$ for an optimal solution x^* of the problem. To do so, we rewrite the iterations of the method in Equation 10 as follows:

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

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

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k)x_j(k) + \epsilon_i(k+1), \\ \epsilon_i(k+1) &= \Pi_X[y_i(k+1) - \alpha_{k+1}g_i(k+1)] - y_i(k+1), \\ y_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k)x_j(k). \end{aligned} \qquad 12.$$

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 1, 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 3. Let the graphs G_k satisfy Assumption 2 and the matrices $W(k)$ satisfy Assumption 3. Then, for the iterate process in Equation 12, we have for all $k \geq 0$

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

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 the fourth property in Assumption 3).

Proof. We write the evolution of the iterates $x_i(k+1)$ of Equation 12 in a matrix representation. Letting $\ell \in [n]$ be any coordinate index, we can write for the ℓth coordinate (denoted by a superscript) $x_i^{\ell}(k+1) = \sum_{j=1}^{m} w_{ij}(k) x_j^{\ell}(k) + \epsilon_i^{\ell}(k+1)$ for all $\ell \in [n]$. Stacking all of the ℓth coordinates in a column vector, denoted by $x^{\ell}(k+1)$, we have $x^{\ell}(k+1) = W(k)x_j^{\ell}(k) + \epsilon^{\ell}(k+1)$ 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 $x_i(k+1)$:

$$\mathbf{X}(k+1) = W(k)\mathbf{X}(k) + \mathbf{E}(k+1) \qquad \text{for all } k \geq 0. \tag{13.}$$

Using the recursion, from Equation 13 we see that, for all $k \geq 0$,

$$\begin{aligned} \mathbf{X}(k+1) &= W(k)\mathbf{X}(k) + \mathbf{E}(k+1) \\ &= W(k)W(k-1)\mathbf{X}(k-1) + W(k)\mathbf{E}(k) + \mathbf{E}(k+1) \\ &= \cdots \\ &= W(k:0)\mathbf{X}(0) + \left(\sum_{t=1}^{k} W(k:t)\mathbf{E}(t)\right) + \mathbf{E}(k+1), \end{aligned} \tag{14.}$$

where $W(k:t) = W(k)W(k-1)\cdots W(t+1)W(t)$ for all $k \geq t \geq 0$.

By multiplying both sides of Equation 14 with the matrix $\frac{1}{m}\mathbf{1}\mathbf{1}'$, we have

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

where the last equality follows from the fact that the matrices $W(k:t)$ are column stochastic [because $W(k)$'s are column stochastic]. By subtracting the preceding relation from Equation 14, we obtain

$$\begin{aligned} \mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1) = {} & \left(W(k:0) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{X}(0) + \sum_{t=1}^{k}\left(W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{E}(t) \\ & + \left(I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right)\mathbf{E}(k+1), \end{aligned} \tag{15.}$$

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 Equation 15, we further obtain

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

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

$$\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F \le \left\|W(k:0) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \|\mathbf{X}(0)\|_F + \left(\sum_{t=1}^{k}\left\|W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \|\mathbf{E}(t)\|_F\right) + \left\|I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \|\mathbf{E}(k+1)\|_F. \qquad 16.$$

By Lemma 1, we have $\left([W(k:t)]_{ij} - \frac{1}{m}\right)^2 \le q^{k-t}$ for all $k \ge t \ge 0$, 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 that $\sqrt{1-\mu} \le 1 - \mu/2$ for any $\mu \in (0,1)$, we see that for all $k \ge t \ge 0$,

$$\left\|W(k:t) - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F \le mp^{k-t} \qquad \text{with } p = 1 - \frac{\eta}{4m^2}. \qquad 17.$$

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

$$\left\|I - \frac{1}{m}\mathbf{1}\mathbf{1}'\right\|_F = \sqrt{m\left(1 - \frac{1}{m}\right)^2 + (m-1)m\frac{1}{m^2}} = \sqrt{m-1}. \qquad 18.$$

Using Equations 17 and 18 in Equation 16, we obtain

$$\left\|\mathbf{X}(k+1) - \frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k+1)\right\|_F \le mp^k\|\mathbf{X}(0)\|_F + m\left(\sum_{t=1}^{k} p^{k-t}\|\mathbf{E}(t)\|_F\right) + \sqrt{m-1}\|\mathbf{E}(k+1)\|_F. \qquad 19.$$

We next interpret Equation 19 in terms of the iterates $x_i(k+1)$ and the vectors $\epsilon_i(k+1)$, as given in Equation 12. Recalling that the ℓth column of $\mathbf{X}(k)$ consists of the vector $x^\ell(k)$ with the entries $x_i^\ell(k)$, $i \in [m]$, $\ell \in [n]$, 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, $\frac{1}{m}\mathbf{1}'\mathbf{X}(k) = x_{av}'(k)$, where $x_{av}(k) = \frac{1}{m}\sum_{j=1}^m x_j(k)$, and $\frac{1}{m}\mathbf{1}\mathbf{1}'\mathbf{X}(k) = \mathbf{1}x_{av}'(k)$ 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, Equation 19 is equivalent to

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

3.2.3. 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 Equation 12).

Thus, there is a structure in $\epsilon_i(k+1)$ that can be further exploited. In particular, we can refine Lemma 3 under the assumption of bounded subgradients $g_i(k+1)$, as given in the following lemma.

Lemma 4. Let the problem be convex (i.e., Assumption 1 holds). In addition, assume that the subgradients of f_i are bounded over the set X for all i, i.e., there exists a constant C_f such that $\|s\| \le C_f$ for every subgradient s of $f_i(z)$ 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 in Equation 10 and their averages $x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k)$, we have for all $i \in [m]$ and $k \ge 0$

$$\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} \le mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m\sqrt{m}C_f \sum_{t=1}^{k} p^{k-t}\alpha_t + mC_f\alpha_{k+1},$$

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

Proof. By Lemma 3, for all $k \ge 0$ we have

$$\sqrt{\sum_{i=1}^{m} \|x_i(k+1) - x_{av}(k+1)\|^2} \le mp^k \sqrt{\sum_{i=1}^{m} \|x_i(0)\|^2} + m\left(\sum_{t=1}^{k} p^{k-t} \sqrt{\sum_{i=1}^{m} \|\epsilon_i(t)\|^2}\right) + \sqrt{m-1}\sqrt{\sum_{i=1}^{m} \|\epsilon_i(k+1)\|^2}. \qquad 20.$$

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 $k \ge 0$, $\left\|\Pi_X[y_i(k+1) - \alpha_{k+1}g_i(k+1)] - y_i(k+1)\right\| \le \alpha_{k+1}\|g_i(k+1) \le \alpha_{k+1}C_f$. Therefore, for all i and $k \ge 0$, $\|\epsilon_i(k+1)\|^2 \le \alpha_{k+1}^2 C_f^2$, implying that

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

By substituting the preceding estimate in Equation 19, we obtain

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

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

We now put together Lemmas 2 and 4 to provide a key result for establishing the convergence of the method. We assume some conditions on the step size, 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_f such that $\|s\| \leq C_f$ for every subgradient s of $f_i(z)$ at any $z \in X$. Also, let the step size satisfy the following conditions: $0 < \alpha_{k+1} \leq \alpha_k$ for all $k \geq 1$, and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$. Then, for the iterates $x_i(k)$ of Equation 10, 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 $x_{av}(k) = \frac{1}{m}\sum_{j=1}^{m} x_j(k)$ and $s_k = 2\alpha_{k+1} C_f \sqrt{m}\sqrt{\sum_{j=1}^{m} \|x_j(k) - x_{av}(k)\|^2} + m\alpha_{k+1}^2 C_f^2$, with s_k being summable, i.e., $\sum_{k=0}^{\infty} s_k < \infty$.

Proof. By Lemma 2, for all i, all $k \geq 0$, and all $x \in X$, we have

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

By the convexity of the squared norm and the definition of $y_i(k+1)$, it follows that

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

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

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

By exchanging the order of summation in the double-sum term and using $\mathbf{1}'W(k) = \mathbf{1}'$, we can see that $\sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij}(k)\|x_j(k) - x\|^2 = \sum_{j=1}^{m} \|x_j(k) - x\|^2$. Therefore,

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

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

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

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

$$\begin{aligned}\sum_{i=1}^{m} \|x_i(k+1) - x\|^2 \leq \sum_{j=1}^{m} \|x_j(k) - x\|^2 - 2\alpha_{k+1}\left(f(x_{av}(k)) - f(x)\right) \\ + 2\alpha_{k+1}C_f\sum_{i=1}^{m}\|y_i(k+1) - x_{av}(k)\| + m\alpha_{k+1}^2 C_f^2. \qquad 21.\end{aligned}$$

Consider now the vectors $y_i(k+1)$, $i \in [m]$. By the definition of $y_i(k+1)$, the convexity of the norm, it follows that $\sum_{i=1}^m \|y_i(k+1) - y\| \leq \sum_{i=1}^m \sum_{j=1}^m w_{ij}(k)\|x_j(k) - y\| = \sum_{j=1}^m \|x_j(k) - y\|$, 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

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

where the last inequality follows by Hölder's inequality. By substituting the preceding estimate in Equation 21, we have for all $k \geq 0$ and all $x \in X$

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

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

$$s_k = 2\alpha_{k+1}C_f\sqrt{m}\sqrt{\sum_{j=1}^m \|x_j(k) - x_{av}(k)\|^2} + m\alpha_{k+1}^2 C_f^2, \qquad 22.$$

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

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

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 4, for all $k \geq 1$, we have

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

Letting

$$r_k = \alpha_{k+1}\sqrt{\sum_{j=1}^m \|x_j(k) - x_{av}(k)\|^2} \qquad 23.$$

and using the assumption that the step size α_k is nonincreasing, we see that

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

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

$$\sum_{k=2}^K r_k \leq m\left(\sum_{k=1}^K p^{k-1}\right)\alpha_1\sqrt{\sum_{i=1}^m \|x_i(0)\|^2} + m\sqrt{m}C_f\sum_{k=2}^K\sum_{t=1}^{k-1} p^{k-t-1}\alpha_t^2 + mC_f\sum_{k=2}^K \alpha_k^2$$
$$< \frac{m}{1-p}\alpha_1\sqrt{\sum_{i=1}^m \|x_i(0)\|^2} + m\sqrt{m}C_f\sum_{s=1}^{K-1}\sum_{t=1}^{s} p^{s-t}\alpha_t^2 + mC_f\sum_{k=2}^K \alpha_k^2,$$

where we use the fact that $p \in (0, 1)$ and 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,

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

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 $s_k = 2C_f\sqrt{m} r_k + m\alpha_{k+1}^2 C_f^2$ (see Equations 22 and 23), it follows that $\sum_{k=0}^{\infty} s_k < \infty$. □

3.2.4. 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_f such that $\|s\| \leq C_f$ for every subgradient s of $f_i(z)$ at any $z \in X$. Let the step size satisfy the following conditions: $0 < \alpha_{k+1} \leq \alpha_k$ for all $k \geq 1$, $\sum_{k=1}^{\infty} \alpha_k = \infty$, and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$. In addition, assume that the problem in Equation 1 has a solution. Then, the iterate sequences $\{x_i(k)\}$, $i \in [m]$, generated by the method in Equation 10, converge to an optimal solution of the problem in Equation 1, i.e., $\lim_{k\to\infty} \|x_i(k) - x^*\| = 0$ for all $i \in [m]$ 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 $k \geq 0$

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

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 rearranging the terms, we further obtain for all $x^* \in X^*$ and all $T \geq K \geq 0$

$$\sum_{i=1}^{m} \|x_i(T+1) - x^*\|^2 + 2\sum_{k=K}^{T} \alpha_{k+1} \left(f(x_{av}(k)) - f(x^*)\right) \leq \sum_{j=1}^{m} \|x_j(K) - x^*\|^2 + \sum_{k=K}^{T} s_k. \quad 24.$$

Note that $f(x_{av}(k)) - f(x^*) > 0$ because $x_{av}(k) \in X$. Thus, the preceding relation implies that the sequences $\{x_i(k)\}$, $i \in [m]$, are bounded. It also implies that $\sum_{k=0}^{\infty} \alpha_{k+1} \left(f(x_{av}(k)) - f^*\right) < \infty$ because $\sum_{k=0}^{\infty} s_k < \infty$, 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 limit inferior, i.e.,

$$\lim_{\ell\to\infty} f(x_{av}(k_\ell)) = f^*. \quad 25.$$

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 convergent 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. Also note that f is continuous on $\mathbb{R}^n$ since it is convex on $\mathbb{R}^n$. Hence, $\lim_{\ell\to\infty} f(x_{av}(k_\ell)) = f(\hat{x})$ with $\hat{x} \in X$, which together with Equation 25 yields $f(\hat{x}) = f^*$. Therefore, $\hat{x}$ is an optimal point, i.e., $\hat{x} \in X^*$.

Now we show that $\{x_i(k_\ell)\}$ converges to $\hat{x}$ for all i. By Lemma 4, we have for all $k \geq 0$ and $i \in [m]$

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

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

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

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

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

For the last limit superior, we have

$$\begin{aligned}
\limsup_{\ell\to\infty} \sum_{t=1}^{k_\ell - 1} p^{k_\ell - 1 - t}\alpha_t &= \lim_{k\to\infty} \sum_{t=1}^{k-1} p^{k-1-t}\alpha_t \\
&= \lim_{k\to\infty} \left(\left(\sum_{\tau=1}^{k-1} p^{k-1-\tau} \right) \frac{1}{\sum_{\tau=1}^{k-1} p^{k-1-\tau}} \sum_{t=1}^{k-1} p^{k-1-t}\alpha_t \right) \\
&= \lim_{k\to\infty} \left(\sum_{\tau=1}^{k-1} p^{k-1-\tau} \right) \left(\lim_{k\to\infty} \frac{1}{\sum_{\tau=1}^{k-1} p^{k-1-\tau}} \sum_{t=1}^{k-1} p^{k-1-t}\alpha_t \right) \\
&= \frac{1}{1-p} \lim_{t\to\infty} \alpha_t.
\end{aligned} \tag{26}$$

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

$$\lim_{\ell\to\infty} x_i(k_\ell) = \hat{x} \quad \text{for all } i \in [m], \qquad \text{with} \quad \hat{x} \in X^*. \tag{27}$$

Since $\hat{x} \in X^*$, we can set $x^* = \hat{x}$ in Equation 24. We then let $K = k_\ell$ in Equation 24, and by omitting the term involving the function values, from Equation 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 Equation 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 $\hat{x} \in X^*$, we have $\lim_{k\to\infty} \|x_i(k) - \hat{x}\| = 0$ for all $i \in [m]$. □

4. LITERATURE ON DISTRIBUTED METHODS FOR OPTIMIZATION IN NETWORKS

This section provides a literature review for distributed optimization methods and new research directions. Many of the distributed algorithms rely on the consensus approach, which has been extensively studied in the past decade (for a literature overview of consensus algorithms, see 15).

4.1. Weighted-Averaging-Based Approaches

The approaches that use consensus models with stochastic matrices, such as the algorithm in Equation 10, are often referred to as weighted-averaging methods. Tsitsiklis (4) described the early work on consensus-based optimization, in which the agents share a common objective function, and Lopes & Sayed (16) and Nedić & Ozdaglar (17, 18) described the first work on distributed optimization in a network with agent-based local objective functions. Nedić & Ozdaglar (17, 18) also considered a slightly different algorithm:

$$x_i(k+1) = \sum_{j=1}^{m} w_{ij}(k)x_j(k) - \alpha_k d_i(k), \qquad 28.$$

where $d_i(k)$ is a subgradient of $f_i(x)$ at $x = x_i(k)$. Nedić & Ozdaglar (19) investigated the convergence rate of this algorithm, Nedić et al. (20) investigated an extension of this algorithm to the case of quantized messages, and Lobel & Ozdaglar (21) studied its implementation over random networks. Lopes & Sayed (22) and Tu & Sayed (23) studied both of these alternative approaches (i.e., the algorithms in Equations 10 and 28) for distributed estimation. A variant of the algorithm in Equation 10 with distributed constraints X_i for agents to solve the problem of minimizing $\sum_{i=1}^{m} f_i(x)$ over $X = \cap_{i=1}^{m} X_i$ was proposed by Nedić et al. (24). The algorithm is based on Equation 10, where for each agent i, the set X is replaced with agent's constraint set X_i:

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

with a subgradient $g_i(k)$ of $f_i(x)$ at $x = \sum_{j=1}^{m} w_{ij}(k)x_j(k)$. Lee (25) and Lee & Nedić (26, 27) studied this algorithm, including random set selections, for synchronous updates over time-varying graphs and for gossip-based asynchronous updates over a static graph. Srivastava and colleagues (28–30) studied a different variant of this algorithm (using the Laplacian formulation of the consensus problem) for distributed optimization with distributed constraints in noisy networks. Distributed algorithms for special quadratic convex problems arising in parameter estimation in sensor networks were developed and studied by Sayed and colleagues (16, 22, 23, 31–33). The consensus-based algorithms for other types of network objective functions were considered by Ram et al. (34). The application of distributed methods to hypothesis-testing problems in graphs has recently attracted attention, resulting in a stream of papers (35–38). While these works deal mainly with finitely many hypothesis, a recent paper (39) extended the framework to the case of infinitely many hypotheses. Nedić et al. (38) considered several algorithms that use different types of consensus models, namely weighted-averaging and push-sum models, which we discuss in more detail in Section 4.2.

Srivastava et al. (40) developed a Bregman-distance-based distributed algorithm as well as consensus-based algorithms for solving certain min-max problems. Koshal et al. (41) studied distributed algorithms, both synchronous and asynchronous, for solving a special type of game (aggregative games). Duchi et al. (42) proposed a distributed dual Nesterov algorithm, Shi et al. (43) proposed a distributed algorithm using the gradient differences, Lu & Tang (44) proposed a distributed algorithm based on the idea of preserving an optimality condition at every stage of the algorithm, and Gharesifard & Cortés (45) investigated distributed convex optimization algorithms for weight-balanced directed graphs in continuous time. Li & Marden (46) proposed a different type of distributed algorithm for convex optimization, in which each agent keeps an estimate for all agents' decisions. This algorithm solves a problem where the agents must minimize a global cost function $f(x_1, \ldots, x_m)$ while each agent i can control only its variable x_i. The algorithm was extended to the online optimization setting by Nedić et al. (47) and Lee et al. (48). Jakovetić et al. studied distributed algorithms using an augmented Lagrangian approach with gossip-type communications (49) and proposed and studied accelerated versions of distributed gradient methods (50).

Srivastava et al. (40) and Zhu & Martínez (51, 52) studied a consensus-based algorithm for solving problems with a separable constraint structure and the use of primal-dual distributed methods, and Chang et al. (53) explored a distributed primal-dual approach with perturbations. Wang & Elia (54) provided algorithms for centralized and distributed convex optimization from a continuous-time dynamical system perspective, and Wan & Lemmon (55) considered an event-triggered distributed optimization for sensor networks. Bürger et al. (56) developed a distributed simplex algorithm for linear programming problems, and Zanella et al. (57) proposed a Newton–Raphson consensus-based method for distributed convex problems. In all of the prior work cited in this subsection, the weights are state independent (i.e., do not depend on the agents' iterates). Lobel et al. (58) proposed and analyzed a consensus-based algorithm employing state-dependent weights.

4.2. Push-Sum-Based Approaches

Another class of distributed algorithms has recently been developed that employs a different type of consensus algorithm, known as a push-sum algorithm. This algorithm has also been referred to as a doubly linear iteration or ratio consensus algorithm owing to its form, which involves the 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 by Kempe et al. (6) for consensus problems over a static network (in a random gossip-based form) and was investigated by Dominguez-Garcia & Hadjicostis (59) in a deterministic setting. It was also extended to time-varying networks by Benezit et al. (60).

Tsianos et al. (61) were the first to employ the push-sum consensus model to develop distributed optimization methods, which was then further investigated in other papers (62–64). This work 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 problems. Tsianos (64) also offered a push-sum algorithm that can deal with constraints by using a Nesterov dual-averaging approach. Nedić & Olshevsky (65, 66) extended the push-sum consensus-based algorithm to the subgradient push algorithm, which can deal with convex optimization problems over time-varying directed graphs. More recently, Sun et al. (67) extended the push-sum algorithm to a larger class of distributed algorithms that are applicable to nonconvex objectives, convex constraint sets, and time-varying graphs (for more detail, see Section 4.4).

4.3. Alternate Direction Multiplier Method–Based Approaches

Another approach to solving Equation 1 in a distributed fashion over a static network can be constructed by using the alternate direction multiplier method (ADMM), which is based on an equivalent formulation of consensus constraints. Unlike the consensus-based (sub)gradient method represented by Equation 10, 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 the consensus requirement). As in any dual method, the ADMM is applicable to problems where the structure of the objective functions f_i is simple enough that the ADMM updates can be executed efficiently. The algorithm has the potential to solve the problem with a geometric convergence rate, which requires global knowledge of some parameters, including eigenvalues of a weight matrix associated with the graph. Boyd et al. (68) provided a survey of the ADMM and its applications. The first work to address the development of distributed ADMM over a network was published by Wei & Ozdaglar (69, 70); it has since been further investigated by Ling & Ribeiro (71), and its linear rate was demonstrated by Shi et al. (72). Aybat et al. (73) proposed the ADMM with linearization for special composite optimization over graphs.

4.4. New Directions

Within the area of distributed (multi-agent) optimization over networks, loosely speaking, there are two main directions of research: toward efficiency improvements (to develop fast distributed algorithms, whose performance can meet the best-known performance guarantees in a centralized setting) and toward addressing nonconvex 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 a geometric convergence rate even when the overall objective function is strongly convex. Shi et al. (43, 74) developed EXTRA (exact first-order algorithm) 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 step size (17, 18). EXTRA converges at an $O(1/k)$ rate when the objective network function is convex and has a geometric rate when the objective function is strongly convex. These developments have considered a static and undirected graph.

Xi & Khan (75) and Zeng & Yin (76) combined EXTRA with the push-sum protocol of Kempe et al. (6) to produce the DEXTRA (directed extra push) algorithm for optimization over a directed graph. DEXTRA converges at a geometric (R-linear) rate for a strong convex objective function, but it requires a careful step-size selection. Xi & Khan (75) noted that the feasible region of step sizes that guarantees this convergence rate can be empty in some cases.

Xu and colleagues (77, 78) utilized an adapt-then-combine strategy (79, 80) of the dynamic weighted-average consensus approach (81) to develop a distributed algorithm termed Aug-DGM (augmented distributed gradient method). This algorithm can be used over static directed or undirected graphs but requires a doubly stochastic matrix. The most interesting aspect of the Aug-DGM algorithm is that it can produce convergent iterates even when different agents use different (constant) step sizes. Simultaneously and independently, Xu et al. (77) and Di Lorenzo & Scutari (82) proposed the idea of tracking the gradient averages through the use of consensus for convex unconstrained problems and nonconvex problems with convex constraints. Di Lorenzo & Scutari (82–84) developed a large class of distributed algorithms, referred to as NEXT (in-network successive convex approximation), which utilizes various function-surrogate modules, thus providing great flexibility in its use and creating a new class of algorithms that subsumes many of the existing distributed algorithms. The work by Di Lorenzo & Scutari (83, 84) and Xu (78) was also proposed independently, with the former preceding the latter. The algorithm framework of Di Lorenzo & Scutari (82–84) is applicable to nonconvex problems with convex constraint sets over time-varying graphs but requires the use of doubly stochastic matrices. Sun et al. (67) removed this assumption by using column-stochastic matrices, which are more general than the degree-based column-stochastic matrices of the push-sum method. Simultaneously and independently, Di Lorenzo & Scutari (84) and Tatarenko & Touri (85) treated nonconvex problems over graphs. Tatarenko & Touri (85) proposed and analyzed a distributed gradient method based on the push-sum consensus in a deterministic and stochastic setting for unconstrained problems.

Nedić et al. (86) used the idea of utilizing a consensus process to track gradients in order to develop a distributed algorithm, referred to as DIGing (distributed inexact gradient method and gradient tracking technique), with a geometric convergence rate over time-varying graphs. This was the first paper to establish such a rate for consensus-based algorithms for convex optimization over time-varying graphs. The algorithm uses a fixed step size, and the rate result is applicable to the problems with a strongly convex smooth objective function. Nedić et al. (87) showed that a variant of the DIGing algorithm converges geometrically fast even if the step sizes differ among the agents.

5. APPLICATIONS IN CONTROL

In this section, we provide some distributed optimization problems arising in control applications.

5.1. Control of Power Systems

Distributed optimization can be applied to various control problems in power networks. Notable among them is the problem of optimally dispatching a set of distributed energy resources (DERs) without a centralized decision maker. In power system operation, the economic dispatch problem aims to minimize the total generation cost while meeting the demand and satisfying generator capacity limits. Specifically, consider a network of m DERs whose communication relationships are described by an undirected or directed graph. Each DER can provide a certain resource (e.g., active or reactive power) at some cost, with the additional constraint that the amount of resource that each DER provides is upper and lower bounded by its capacity limits. For each DER i, let $x_i \in [0, \infty)$ be its power generation; $C_i(\cdot) : [0, \infty) \to [0, \infty)$ be its cost function; $x_i^{\min}$ and $x_i^{\max}$ be the lower and upper bounds of its power generation, respectively; and D be the total demand satisfying $\sum_{i=1}^{m} x_i^{\min} \leq D \leq \sum_{i=1}^{n} x_i^{\max}$. The economic dispatch problem is then formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} C_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{m} x_i = D, \\ & x_i \in [x_i^{\min}, x_i^{\max}], \qquad i \in [m]. \end{aligned} \tag{29}$$

Each cost function $C_i(\cdot)$ is assumed to be strictly convex and continuously differentiable. Define the Lagrangian function as

$$L(\lambda) = \sum_{i=1}^{m} C_i(x_i) - \lambda \left(\sum_{i=1}^{m} x_i - D \right).$$

The corresponding Lagrange dual problem can then be written as

$$\max_{\lambda} \sum_{i=1}^{m} f_i(\lambda),$$

where

$$f_i(\lambda) = \min_{x_i \in [x_i^{\min}, x_i^{\max}]} C_i(x_i) - \lambda(x_i - D_i).$$

Here, D_i is a virtual local demand such that $\sum_{i=1}^{m} D_i = D$ (for details, see 88).

References 89–92 used the idea of consensus to study this problem for the case when the cost functions are quadratic. Yang et al. (88) have recently studied the case for general convex cost functions by making use of the idea of the subgradient push algorithm (65, 66).

Similar problem formulation has been applied to other power system problems, including the optimal load control problem, which balances load and generation while minimizing the end-use disutility of participating in load control (93); the optimal load-sharing control problem in power systems (94); the distributed optimal generation control problem (95); and the problem of distributed energy management for both the generation and demand sides in smart grids (96). The distributed optimization problem (29) also finds application in the optimal resource allocation problem for parallel computing (97).

5.2. Least Squares Solutions to Linear Equations

Consider a network of m autonomous agents that are able to receive information from their neighbors, where neighbor relationships are characterized by a time-dependent directed graph G_t with m vertices and a set of arcs defined so that there is an arc in the graph from vertex j to vertex i whenever agent j is a neighbor of agent i. Thus, the directions of arcs represent the directions of information flow. Each agent i has a real-time dependent state vector $x_i(t)$ taking values in $\mathbb{R}^n$, and we assume that the information that agent i receives from neighbor j is only the current state vector of neighbor j. We also assume that agent i knows only a pair of real-valued matrices $(A_i^{n_i \times n}, b_i^{n_i \times 1})$. The problem of interest is to devise local algorithms, one for each agent, which will enable all m agents to iteratively compute the same least squares solution to the linear equation $Ax = b$, where

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}_{\bar{n} \times n}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}_{\bar{n} \times 1},$$

and $\bar{n} = \sum_{i=1}^{m} n_i$. Note that

$$\min_x \|Ax - b\|^2 = \min_x \sum_{i=1}^{m} \|A_i x - b_i\|^2 = \min_x \sum_{i=1}^{m} f_i(x),$$

where

$$f_i(x) = \|A_i x - b_i\|^2,$$

which is a distributed optimization problem. This problem finds application in estimating the electromechanical oscillation modes of large power system networks using synchrophasors (98).

5.3. Model Predictive Control

Consider a network of m independent agents that are allowed to communicate information according to an undirected graph. Each agent i has a discrete-time linear dynamics given by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t),$$

where $x_i(t) \in \mathbb{R}^{n_i}$ is the state; $u_i(t) \in \mathbb{R}^{m_i}$ is the control input; and A_i and B_i are update and input matrices, respectively. The goal of model predictive control is to predict the future behavior of the system and determine inputs to optimize a performance index over a finite horizon, as follows:

$$\begin{aligned} \text{minimize} \qquad & J = \sum_{t=0}^{T} \sum_{i=1}^{m} V_i\left(\{x_j(t), u_j(t), j \in \mathcal{N}_i\}\right) \\ & \quad + \sum_{t=0}^{T} \sum_{i=1}^{m} U_i\left(\{x_j(T), u_j(T), j \in \mathcal{N}_i\}\right) \\ \text{subject to} \qquad & x_i(t+1) = A_i x_i(t) + B_i u_i(t), \\ & x_i(t) \in \mathcal{X}_i(t), \quad u_i(t) \in \mathcal{U}_i(t), \quad i \in [m], \end{aligned}$$

where $\mathcal{N}_i$ is the set of neighbors of agent i, including itself; $V_i(\cdot)$ is the convex stage cost function; $U_i(\cdot)$ is the terminal cost function; and $\mathcal{X}_i(t)$ and $\mathcal{U}_i(t)$ are the convex constraint sets for local state and control input, respectively. The above problem can be rewritten in a distributed optimization form as follows:

$$\begin{aligned} \text{minimize} \qquad & \sum_{i=1}^{m} f_i(x) \\ \text{subject to} \qquad & x_i(t+1) = A_i x_i(t) + B_i u_i(t), \\ & x_i(t) \in \mathcal{X}_i(t), \quad u_i(t) \in \mathcal{U}_i(t), \quad i \in [m], \end{aligned}$$

where x is the vector obtained by stacking the x_i, and

$$f_i(x) = \sum_{t=0}^{T} \left(V_i\left(\{x_j(t), u_j(t), j \in \mathcal{N}_i\}\right) + U_i\left(\{x_j(T), u_j(T), j \in \mathcal{N}_i\}\right)\right).$$

This problem was studied by Summers & Lygeros (99) using the distributed ADMM, and compared with the dual-decomposition-based distributed gradient algorithm by Conte et al. (100). Mota et al. (101) considered the case when each agent has a more general nonlinear dynamics. Christofides et al. (102) provided a tutorial review of recent results in the design of distributed model predictive control systems.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

J.L. wishes to thank Jie Lu (ShanghaiTech University) and Tao Yang (University of North Texas) for useful discussions of the economic dispatch problem.

LITERATURE CITED

1. Bash BA, Desnoyers PJ. 2007. Exact distributed Voronoi cell computation in sensor networks. In *2007 6th International Symposium on Information Processing in Sensor Networks*, pp. 236–43. New York: IEEE
2. DeGroot M. 1974. Reaching a consensus. *J. Am. Stat. Assoc.* 69:118–21
3. Borkar V, Varaiya P. 1982. Asymptotic agreement in distributed estimation. *IEEE Trans. Autom. Control* 27:650–55

4. Tsitsiklis J. 1984. *Problems in decentralized decision making and computation*. PhD Thesis, Dep. Electr. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, MA
5. Jadbabaie A, Lin J, Morse A. 2003. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* 48:988–1001
6. Kempe D, Dobra A, Gehrke J. 2003. Gossip-based computation of aggregate information. In *2003 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 482–91. New York: IEEE
7. Olfati-Saber R, Murray RM. 2004. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* 49:1520–33
8. Moreau L. 2005. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Autom. Control* 50:169–82
9. Vicsek T, Czirok A, Ben-Jacob E, Cohen I, Schochet O. 1995. Novel type of phase transitions in a system of self-driven particles. *Phys. Rev. Lett.* 75:1226–29
10. Bullo F, Cortés J, Martínez S. 2009. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ: Princeton Univ. Press
11. Mesbahi M, Egerstedt M. 2010. *Graph Theoretic Methods for Multiagent Networks*. Princeton, NJ: Princeton Univ. Press
12. Martinoli A, Mondada F, Mermoud G, Correll N, Egerstedt M, et al., eds. 2013. *Distributed Autonomous Robotic Systems: The 10th International Symposium*. Berlin: Springer
13. Nedić A, Olshevsky A, Ozdaglar A, Tsitsiklis J. 2009. On distributed averaging algorithms and quantization effects. *IEEE Trans. Autom. Control* 54:2506–17
14. Facchinei F, Pang JS. 2003. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. New York: Springer
15. Nedić A. 2015. *Convergence Rate of Distributed Averaging Dynamics and Optimization in Networks*. Found. Trends Syst. Control Vol. 2, No. 1. Hanover, MA: Now
16. Lopes CG, Sayed AH. 2007. Distributed processing over adaptive networks. In *2007 9th International Symposium on Signal Processing and Its Applications*. New York: IEEE. **https://doi.org/10.1109/ISSPA.2007.4555636**
17. Nedić A, Ozdaglar A. 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control* 54:48–61
18. Nedić A, Ozdaglar A. 2010. Cooperative distributed multi-agent optimization. In *Convex Optimization in Signal Processing and Communications*, ed. Y Eldar, D Palomar, pp. 340–86. Cambridge, UK: Cambridge Univ. Press
19. Nedić A, Ozdaglar A. 2007. On the rate of convergence of distributed subgradient methods for multi-agent optimization. In *2007 46th IEEE Conference on Decision and Control*, pp. 4711–16. New York: IEEE
20. Nedić A, Olshevsky A, Ozdaglar A, Tsitsiklis J. 2008. Distributed subgradient methods and quantization effects. In *2008 47th IEEE Conference on Decision and Control*, pp. 4177–84. New York: IEEE
21. Lobel I, Ozdaglar A. 2011. Distributed subgradient methods for convex optimization over random networks. *IEEE Trans. Autom. Control* 56:1291–306
22. Lopes C, Sayed A. 2008. Diffusion least-mean squares over adaptive networks: formulation and performance analysis. *IEEE Trans. Signal Process.* 56:3122–36
23. Tu SY, Sayed A. 2012. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.* 60:6217–34
24. Nedić A, Ozdaglar A, Parrilo P. 2010. Constrained consensus and optimization in multi-agent networks. *IEEE Trans. Autom. Control* 55:922–38
25. Lee S. 2013. *Optimization over networks: efficient algorithms and analysis*. PhD Thesis, Dep. Electr. Comput. Eng., Univ. Ill., Urbana-Champaign
26. Lee S, Nedić A. 2012. Distributed random projection algorithm for convex optimization. *IEEE J. Sel. Top. Signal Process.* 48:988–1001
27. Lee S, Nedić A. 2016. Asynchronous gossip-based random projection algorithms over networks. *IEEE Trans. Autom. Control* 61:953–68
28. Srivastava K, Nedić A, Stipanović D. 2010. Distributed constrained optimization over noisy networks. In *2010 49th IEEE Conference on Decision and Control (CDC)*, pp. 1945–50. New York: IEEE

29. Srivastava K, Nedić A. 2011. Distributed asynchronous constrained stochastic optimization. *IEEE J. Sel. Top. Signal Process.* 5:772–90

30. Srivastava K. 2011. *Distributed optimization with applications to sensor networks and machine learning*. PhD Thesis, Dep. Ind. Enterp. Syst. Eng., Univ. Ill., Urbana-Champaign

31. Cattivelli F, Sayed A. 2010. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* 58:1035–48

32. Tu SY, Sayed A. 2012. On the influence of informed agents on learning and adaptation over networks. *IEEE Trans. Signal Process.* 61:1339–56

33. Chen J, Sayed A. 2012. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* 60:4289–305

34. Ram S, Nedić A, Veeravalli V. 2012. A new class of distributed optimization algorithms: application to regression of distributed data. *Optim. Methods Softw.* 27:71–88

35. Shahrampour S, Jadbabaie A. 2013. Exponentially fast parameter estimation in networks using distributed dual averaging. In *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, pp. 6196–201. New York: IEEE

36. Shahrampour S, Rakhlin A, Jadbabaie A. 2015. Distributed detection: finite-time analysis and impact of network topology. *IEEE Trans. Autom. Control* 61:3256–68

37. Lalitha A, Javidi T, Sarwate A. 2014. Social learning and distributed hypothesis testing. In *2014 IEEE International Symposium on Information Theory (ISIT)*, pp. 551–55. New York: IEEE

38. Nedić A, Olshevsky A, Uribe C. 2015. Nonasymptotic convergence rates for cooperative learning over time-varying directed graphs. In *2015 American Control Conference (ACC)*, pp. 5884–89. New York: IEEE

39. Nedić A, Olshevsky A, Uribe C. 2016. Network independent rates in distributed learning. In *2016 American Control Conference (ACC)*, pp. 1072–77. New York: IEEE

40. Srivastava K, Nedić A, Stipanovic D. 2013. Distributed Bregman-distance algorithms for min-max optimization. In *Agent-Based Optimization*, ed. I Czarnowski, P Jedrzejowicz, J Kacprzyk, pp. 143–74. Berlin: Springer

41. Koshal J, Nedić A, Shanbhag U. 2016. Distributed algorithms for aggregative games on graphs. *Oper. Res.* 64:680–704

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

43. Shi W, Ling Q, Wu G, Yin W. 2015. EXTRA: an exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.* 25:944–66

44. Lu J, Tang C. 2012. Zero-gradient-sum algorithms for distributed convex optimization: the continuous-time case. *IEEE Trans. Autom. Control* 57:2348–54

45. Gharesifard B, Cortés J. 2012. Distributed continuous-time convex optimization on weight-balanced digraphs. *Eur. J. Control* 18:539–57

46. Li N, Marden JR. 2013. Designing games for distributed optimization. *IEEE J. Sel. Top. Signal Process.* 7:230–42

47. Nedić A, Lee S, Raginsky M. 2015. Decentralized online optimization with global objectives and local communication. In *2015 American Control Conference (ACC)*, pp. 4497–503. New York: IEEE

48. Lee S, Nedić A, Raginsky M. 2018. Coordinate dual averaging for decentralized online optimization with non-separable global objectives. *IEEE Trans. Control Netw. Syst.* 5:34–44

49. Jakovetić D, Xavier J, Moura J. 2011. Cooperative convex optimization in networked systems: augmented Lagrangian algorithms with directed gossip communication. *IEEE Trans. Signal Process.* 59:3889–902

50. Jakovetić D, Xavier J, Moura J. 2014. Fast distributed gradient methods. *IEEE Trans. Autom. Control* 59:1131–46

51. Zhu M, Martínez S. 2012. On distributed convex optimization under inequality and equality constraints. *IEEE Trans. Autom. Control* 57:151–64

52. Zhu M, Martínez S. 2013. An approximate dual subgradient algorithm for distributed non-convex constrained optimization. *IEEE Trans. Autom. Control* 58:1534–39

53. Chang TH, Nedić A, Scaglione A. 2014. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Trans. Autom. Control* 59:1524–38

54. Wang J, Elia N. 2011. A control perspective for centralized and distributed convex optimization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 3800–5. New York: IEEE
55. Wan P, Lemmon M. 2009. Event-triggered distributed optimization in sensor networks. In *2009 International Conference on Information Processing of Sensor Networks*, pp. 49–60. New York: IEEE
56. Bürger M, Notarstefano G, Bullo F, Allgöwer F. 2012. A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. *Automatica* 48:2298–304
57. Zanella F, Varagnolo D, Cenedese A, Pillonetto G, Schenato L. 2011. Newton-Raphson consensus for distributed convex optimization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 5917–22. New York: IEEE
58. Lobel I, Ozdaglar A, Feijer D. 2011. Distributed multi-agent optimization with state-dependent communication. *Math. Prog.* 129:255–84
59. Dominguez-Garcia A, Hadjicostis C. 2011. Distributed strategies for average consensus in directed graphs. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 2124–29. New York: IEEE
60. Benezit F, Blondel V, Thiran P, Tsitsiklis J, Vetterli M. 2010. Weighted gossip: distributed averaging using non-doubly stochastic matrices. In *2010 IEEE International Symposium on Information Theory (ISIT)*, pp. 1753–57. New York: IEEE
61. Tsianos K, Lawlor S, Rabbat M. 2012. Consensus-based distributed optimization: practical issues and applications in large-scale machine learning. In *2012 50th Allerton Conference on Communication, Control, and Computing*, pp. 1543–50. New York: IEEE
62. Tsianos K, Lawlor S, Rabbat M. 2012. Push-sum distributed dual averaging for convex optimization. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 5453–58. New York: IEEE
63. Tsianos K, Rabbat M. 2011. Distributed consensus and optimization under communication delays. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing*, pp. 974–82. New York: IEEE
64. Tsianos K. 2013. *The role of the network in distributed optimization algorithms: convergence rates, scalability, communication/computation tradeoffs and communication delays*. PhD Thesis, Dep. Electr. Comput. Eng., McGill Univ., Montreal, Can.
65. Nedić A, Olshevsky A. 2015. Distributed optimization over time-varying directed graphs. *IEEE Trans. Autom. Control* 60:601–5
66. Nedić A, Olshevsky A. 2016. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Trans. Autom. Control* 61:3936–47
67. Sun Y, Scutari G, Palomar D. 2016. Distributed nonconvex multiagent optimization over time-varying networks. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 788–94. New York: IEEE
68. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. 2010. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Found. Trends Mach. Learn. Vol. 3, No. 1. Hanover, MA: Now
69. Wei E, Ozdaglar A. 2012. Distributed alternating direction method of multipliers. In *2012 IEEE 51st Conference on Decision and Control (CDC)*, pp. 5445–50. New York: IEEE
70. Wei E, Ozdaglar A. 2013. On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers. In *2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 551–54. New York: IEEE
71. Ling Q, Ribeiro A. 2014. Decentralized dynamic optimization through the alternating direction method of multiplier. *IEEE Trans. Signal Process.* 62:1185–97
72. Shi W, Ling Q, Yuan K, Wu G, Yin W. 2014. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Trans. Signal Process.* 62:1750–61
73. Aybat NS, Wang Z, Lin T, Ma S. 2017. Distributed linearized alternating direction method of multipliers for composite convex consensus optimization. *IEEE Trans. Autom. Control* 63:5–20
74. Shi W, Ling Q, Wu G, Yin W. 2015. A proximal gradient algorithm for decentralized composite optimization. *IEEE Trans. Signal Process.* 63:6013–23

75. Xi C, Khan U. 2015. On the linear convergence of distributed optimization over directed graphs. arXiv:1510.02149

76. Zeng J, Yin W. 2015. ExtraPush for convex smooth decentralized optimization over directed networks. arXiv:1511.02942

77. Xu J, Zhu S, Soh Y, Xie L. 2015. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *2015 IEEE 54th Conference on Decision and Control (CDC)*, pp. 2055–60. New York: IEEE

78. Xu J. 2016. *Augmented distributed optimization for networked systems*. PhD Thesis, Sch. Electr. Electron. Eng., Nanyang Technol. Univ., Singapore

79. Sayed A. 2013. Diffusion adaptation over networks. In *Array and Statistical Signal Processing*, ed. AM Zoubir, M Viberg, R Chellappa, S Theodoridis, pp. 323–453. Acad. Press Libr. Signal Process. Vol. 3. Oxford, UK: Academic

80. Sayed AH. 2014. Adaptation, learning, and optimization over networks. *Found. Trends Mach. Learn.* 7:311–801

81. Zhu M, Martínez S. 2010. Discrete-time dynamic average consensus. *Automatica* 46:322–29

82. Di Lorenzo P, Scutari G. 2015. Distributed nonconvex optimization over networks. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2015)*, pp. 229–32. New York: IEEE

83. Di Lorenzo P, Scutari G. 2016. Distributed nonconvex optimization over time-varying networks. In *2016 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 16)*, pp. 4124–28. New York: IEEE

84. Di Lorenzo P, Scutari G. 2016. NEXT: in-network nonconvex optimization. *IEEE Trans. Signal Inf. Process. Netw.* 2:120–36

85. Tatarenko T, Touri B. 2016. Non-convex distributed optimization. arXiv:1512.00895

86. Nedić A, Olshevsky A, Shi W. 2016. Achieving geometric convergence for distributed optimization over time-varying graphs. *IEEE Trans. Autom. Control* 62:3744–57

87. Nedić A, Olshevsky A, Shi W, Uribe C. 2017. Geometrically convergent distributed optimization with uncoordinated step-sizes. In *2017 American Control Conference (ACC)*, pp. 3950–55. New York: IEEE

88. Yang T, Lu J, Wu D, Wu J, Shi G, et al. 2017. A distributed algorithm for economic dispatch over time-varying directed networks with delays. *IEEE Trans. Ind. Electron.* 64:5095–106

89. Kar S, Hug G. 2012. Distributed robust economic dispatch in power systems: a consensus + innovations approach. In *2012 IEEE Power and Energy Society General Meeting*. New York: IEEE. **https://doi.org/10.1109/PESGM.2012.6345156**

90. Dominguez-Garcia A, Cady S, Hadjicostis CN. 2012. Decentralized optimal dispatch of distributed energy resources. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 3688–93. New York: IEEE

91. Yang S, Tan S, Xu JX. 2013. Consensus based approach for economic dispatch problem in a smart grid. *IEEE Trans. Power Syst.* 28:4416–26

92. Xing H, Mou Y, Fu M, Lin Z. 2015. Distributed bisection method for economic power dispatch in smart grid. *IEEE Trans. Power Syst.* 30:3024–35

93. Zhao C, Topcu U, Low SH. 2013. Optimal load control via frequency measurement and neighborhood area communication. *IEEE Trans. Power Syst.* 28:3576–87

94. Yi P, Hong Y, Liu F. 2015. Distributed gradient algorithm for constrained optimization with application to load sharing in power systems. *Syst. Control Lett.* 83:45–52

95. Zhang W, Liu W, Wang X, Liu L, Ferrese F. 2013. Online optimal generation control based on constrained distributed gradient algorithm. *IEEE Trans. Power Syst.* 30:35–45

96. Zhao C, He J, Cheng P, Chen J. 2016. Consensus-based energy management in smart grid with transmission losses and directed communication. *IEEE Trans. Smart Grid* 8:2049–61

97. Xiao L, Boyd S. 2006. Optimal scaling of a gradient method for distributed resource allocation. *J. Optim. Theory Appl.* 129:469–88

98. Nabavi S, Zhang J, Chakrabortty A. 2015. Distributed optimization algorithms for wide-area oscillation monitoring in power systems using interregional PMU-PDC architectures. *IEEE Trans. Smart Grid* 6:2529–38

99. Summers TH, Lygeros J. 2012. Distributed model predictive consensus via the alternating direction method of multipliers. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 79–84. New York: IEEE

100. Conte C, Summers TH, Zeilinger MN, Morari M, Jones CN. 2012. Computational aspects of distributed optimization in model predictive control. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 6819–24. New York: IEEE

101. Mota JFC, Xavier JMF, Aguiar PMQ, Püschel M. 2015. Distributed optimization with local domains: applications in MPC and network flows. *IEEE Trans. Autom. Control* 60:2004–09

102. Christofides PD, Scattolini R, de la Peña DM, Liu J. 2013. Distributed model predictive control: a tutorial review and future research directions. *Comput. Chem. Eng.* 51:21–41

Annual Review of
Control, Robotics,
and Autonomous
Systems

Volume 1, 2018

Contents

Toward Robotic Manipulation
Matthew T. Mason 1

Autonomous Flight
Sarah Tang and Vijay Kumar 29

Soft Micro- and Nanorobotics
Chengzhi Hu, Salvador Pané, and Bradley J. Nelson 53

Distributed Optimization for Control
Angelia Nedić and Ji Liu 77

Game Theory and Control
Jason R. Marden and Jeff S. Shamma 105

The Bountiful Intersection of Differential Geometry, Mechanics, and Control Theory
Andrew D. Lewis 135

Sampling-Based Methods for Motion Planning with Constraints
Zachary Kingston, Mark Moll, and Lydia E. Kavraki 159

Planning and Decision-Making for Autonomous Vehicles
Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus 187

Synthesis for Robots: Guarantees and Feedback for Robot Behavior
Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman 211

Invariant Kalman Filtering
Axel Barrau and Silvère Bonnabel 237

Data-Driven Predictive Control for Autonomous Systems
Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli 259

A Separation Principle for Control in the Age of Deep Learning
Alessandro Achille and Stefano Soatto 287

Privacy in Control and Dynamical Systems
Shuo Han and George J. Pappas 309

Hamilton–Jacobi Reachability: Some Recent Theoretical Advances and Applications in Unmanned Airspace Management
Mo Chen and Claire J. Tomlin 333

Design of Materials and Mechanisms for Responsive Robots
Elliot W. Hawkes and Mark R. Cutkosky 359

Haptics: The Present and Future of Artificial Touch Sensation
Heather Culbertson, Samuel B. Schorr, and Allison M. Okamura 385

Programming Cells to Work for Us
Yili Qian, Cameron McBride, and Domitilla Del Vecchio 411

Autonomy in Rehabilitation Robotics: An Intersection
Brenna D. Argall 441

Medical Technologies and Challenges of Robot-Assisted Minimally Invasive Intervention and Diagnostics
Nabil Simaan, Rashid M. Yasin, and Long Wang 465

Errata

An online log of corrections to *Annual Review of Control, Robotics, and Autonomous Systems* articles may be found at http://www.annualreviews.org/errata/control