Convex Optimization in Signal Processing and Communications

March 20, 2009

CAMBRIDGE
UNIVERSITY PRESS

Contents

List of contributors

Angelia Nedić

Angelia Nedic received her B.S. degree from the University of Montenegro in 1987 and M.S. degree from the University of Belgrade in 1990, both in Mathematics. She received her Ph.D. degrees from Moscow State University in 1994 in Mathematics and Mathematical Physics, and from Massachusetts Institute of Technology in Electrical Engineering and Computer Science in 2002. She has been at the BAE Systems Advanced Information Technology from 2002-2006. Since 2006 she has been an Assistant Professor at the Department of Industrial and Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign. Her general interest is in optimization including fundamental theory, models, algorithms, and applications.

Asuman Ozdaglar

Asuman Ozdaglar is the Class of 1943 Associate Professor in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology. She is also a member of the Laboratory for Information and Decision Systems and the Operations Research Center. She received the B.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, in 1996, and the S.M. and the Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1998 and 2003, respectively. Her research interests include optimization theory, with emphasis on nonlinear programming and convex analysis, game theory, with applications in communication, social, and economic networks, and distributed optimization methods.

Part I

1 Cooperative Distributed Multi-Agent Optimization

Angelia Nedić and Asuman Ozdaglar

Angelia Nedić is with the University of Illinois at Urbana-Champaign.
Asuman Ozdaglar is with the Massachusetts Institute of Technology.

This chapter presents distributed algorithms for cooperative optimization among multiple agents connected through a network. The goal is to optimize a global objective function which is a combination of local objective functions known by the agents only. We focus on two related approaches for the design of distributed algorithms for this problem. The first approach relies on using Lagrangian decomposition and dual subgradient methods. We show that this methodology leads to distributed algorithms for optimization problems with special structure. The second approach involves combining consensus algorithms with subgradient methods. In both approaches, our focus is on providing convergence rate analysis for the generated solutions that highlight the dependence on problem parameters.

1.1 Introduction and Motivation

There has been much recent interest in distributed control and coordination of networks consisting of multiple agents, where the goal is to collectively optimize a global objective. This is motivated mainly by the emergence of large scale networks and new networking applications such as mobile ad hoc networks and wireless sensor networks, characterized by the lack of centralized access to information and time-varying connectivity. Control and optimization algorithms deployed in such networks should be completely distributed, relying only on local observations and information, robust against unexpected changes in topology, such as link or node failures, and scalable in the size of the network.

This chapter studies the problem of distributed optimization and control of multi-agent networked systems. More formally, we consider a multi-agent network model, where m agents exchange information over a connected network. Each agent i has a *local convex objective function* $f_i(x)$, with $f_i : \mathbb{R}^n \to \mathbb{R}$, and a nonempty *local convex constraint set* X_i, with $X_i \subset \mathbb{R}^n$, known by this agent only. The vector $x \in \mathbb{R}^n$ represents a global decision vector that the agents are collectively trying to decide on.

The goal of the agents is to *cooperatively optimize* a global objective function, denoted by $f(x)$, which is a combination of the local objective functions, i.e.,

$$f(x) = T\Big(f_1(x), \ldots, f_m(x)\Big),$$

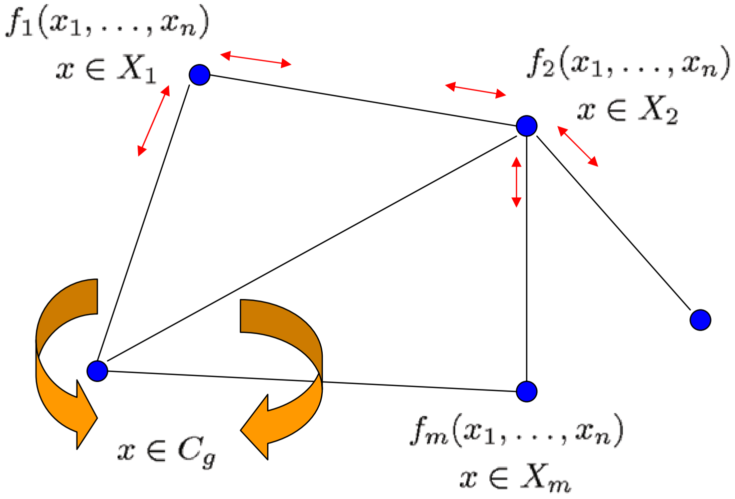


Figure 1.1 Multiagent cooperative optimization problem.

where $T : \mathbb{R}^m \to \mathbb{R}$ is an increasing convex function.[1] The decision vector x is constrained to lie in a set, $x \in C$, which is a combination of local constraints and additional global constraints that may be imposed by the network structure, i.e.,

$$C = \left(\cap_{i=1}^m X_i\right) \cap C_g,$$

where C_g represents the global constraints. This model leads to the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in C, \end{aligned} \tag{1.1}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$ is a *convex objective function* and the set C is a *convex constraint set* (see Figure 1.1). The decision vector x in problem (1.1) can be viewed as either a resource vector whose components correspond to resources allocated to each agent, or a global parameter vector to be estimated by the agents using local information.

Our goal in this chapter is to develop optimization methods that the agents can use to solve problem (1.1) within the informational structure available to them. Our development relies on using *first-order methods*, i.e., gradient-based methods (or subgradient methods for the case when the local objective functions

[1] By an increasing function, we mean for any $w,\ y \in \mathbb{R}^m$ with $w \geq y$ with respect to the usual vector order (i.e., the inequality holds componentwise), we have $T(w) \geq T(y)$.

f_i are nonsmooth). Due to simplicity of computations per iteration, first-order methods have gained popularity in the last few years as low overhead alternatives to interior-point methods, that may lend themselves to distributed implementations. Despite the fact that first-order methods have slower convergence rate (compared to interior point methods) in finding high-accuracy solutions, they are particularly effective in large scale multi-agent optimization problems where the goal is to generate near-optimal *approximate solutions* in relatively small number of iterations.

This chapter will present both classical results and recent advances on the design and analysis of distributed optimization algorithms. The theoretical development will be complemented with recent application areas for these methods. Our development will focus on two major methodologies.

The first approach relies on using Lagrangian dual decomposition and dual methods for solving problem (1.1). We will show that this approach leads to distributed optimization algorithms when problem (1.1) is *separable* (i.e., problems where local objective functions and constraints decompose over the components of the decision vector). This methodology has been used extensively in the networking literature to design cross-layer resource allocation mechanisms (see [23], [27], [48], [50], and [14]). Our focus in this chapter will be on generating approximate (primal) solutions from the dual algorithm and providing convergence rate estimates. Despite the fact that duality yields distributed methods primarily for separable problems, our methods and rate analysis are applicable for general convex problems and will be covered here in their general version.

When problem (1.1) is not separable, dual decomposition approach will not lead to distributed methods. For such problems, we present optimization methods that use *consensus algorithms* as a building block. Consensus algorithms involve each agent maintaining estimates of the decision vector x and updating it based on local information that becomes available through the communication network. These algorithms have attracted much attention in the cooperative control literature for distributed coordination of a system of dynamic agents (see [6] [7], [8], [9], [10] [11] [12], [13], [20], [21], [22], [29], [44] [45], [46] [54], [55]). These works mainly focus on the canonical consensus problem, where the goal is to design distributed algorithms that can be used by a group of agents to agree on a common value. Here, we show that consensus algorithms can be combined with first-order methods to design distributed methods that can optimize general convex local objective functions over a time-varying network topology.

The chapter is organized into four sections. In Section 1.2, we present distributed algorithms designed using Lagrangian duality and subgradient methods. We show that for (separable) network resource allocation problems, this methodology yields distributed optimization methods. We present recent results on generating approximate primal solutions from dual subgradient methods and provide convergence rate analysis. In Section 1.3, we develop distributed methods for optimizing the sum of general (non-separable) convex objective functions corresponding to multiple agents connected over a time-varying topology. These methods will involve a combination of first-order methods and consensus algorithms. Section 1.4 focuses on extensions of the distributed methods to handle local constraints and imperfections associated with implementing optimization algorithms over networked systems, such as delays, asynchronism, and quantization effects, and studies the implications of these considerations on the network algorithm performance. Section 1.5 suggests a number of areas for future research.

1.2 Distributed Optimization Methods using Dual Decomposition

This section focuses on subgradient methods for solving the dual problem of a convex constrained optimization problem obtained by Lagrangian relaxation of some of the constraints. For separable problems, this method leads to decomposition of the computations at each iteration into subproblems that each agent can solve using his local information and the prices (or dual variables).

In the first part of the section, we formally define the dual problem of a (primal) convex constrained optimization problem. We establish relations between the primal and the dual optimal values, and investigate properties of the dual optimal solution set. In Section 1.2.3, we introduce the utility-based network resource allocation problem and show that Lagrangian decomposition and dual subgradient methods yield distributed optimization methods for solving this problem. Since the main interest in most practical applications is to obtain near-optimal solutions to problem (1.1), the remainder of the section focuses on obtaining approximate primal solutions using information directly available from dual subgradient methods and presents the corresponding rate analysis.

We start by defining the basic notation and terminology used throughout the chapter.

1.2.1 Basic Notation and Terminology

We consider the n-dimensional vector space $\mathbb{R}^n$ and the m-dimensional vector space $\mathbb{R}^m$. We view a vector as a column vector, and we denote by $x'y$ the inner product of two vectors x and y. We use $\|y\|$ to denote the standard Euclidean norm, $\|y\| = \sqrt{y'y}$. We write $dist(\bar{y}, Y)$ to denote the standard Euclidean distance of a vector $\bar{y}$ from a set Y, i.e.,

$$dist(\bar{y}, Y) = \inf_{y \in Y} \|\bar{y} - y\|.$$

For a vector $u \in \mathbb{R}^m$, we write u^+ to denote the projection of u on the nonnegative orthant in $\mathbb{R}^m$, i.e., u^+ is the component-wise maximum of the vector u and the zero vector:

$$u^+ = (\max\{0, u_1\}, \cdots, \max\{0, u_m\})' \quad \text{for } u = (u_1, \cdots, u_m)'.$$

For a convex function $F : \mathbb{R}^n \to [-\infty, \infty]$, we denote the domain of F by $\mathrm{dom}(F)$, where

$$\mathrm{dom}(F) = \{x \in \mathbb{R}^n \mid F(x) < \infty\}.$$

We use the notion of a subgradient of a convex function $F(x)$ at a given vector $\bar{x} \in \mathrm{dom}(F)$. A subgradient $s_F(\bar{x})$ of a convex function $F(x)$ at any $\bar{x} \in \mathrm{dom}(F)$ provides a linear underestimate of the function F. In particular, $s_F(\bar{x}) \in \mathbb{R}^n$ is a *subgradient of a convex function* $F : \mathbb{R}^n \to \mathbb{R}$ *at a given vector* $\bar{x} \in \mathrm{dom}(F)$ when the following relation holds:

$$F(\bar{x}) + s_F(\bar{x})'(x - \bar{x}) \leq F(x) \qquad \text{for all } x \in \mathrm{dom}(F). \tag{1.2}$$

The set of all subgradients of F at $\bar{x}$ is denoted by $\partial F(\bar{x})$.

Similarly, for a concave function $q : \mathbb{R}^m \to [-\infty, \infty]$, we denote the domain of q by $\mathrm{dom}(q)$, where

$$\mathrm{dom}(q) = \{\mu \in \mathbb{R}^m \mid q(\mu) > -\infty\}.$$

A subgradient of a concave function is defined through a subgradient of a convex function $-q(\mu)$. In particular, $s_q(\bar{\mu}) \in \mathbb{R}^m$ is a *subgradient of a concave function* $q(\mu)$ *at a given vector* $\bar{\mu} \in \mathrm{dom}(q)$ when the following relation holds:

$$q(\bar{\mu}) + s_q(\bar{\mu})'(\mu - \bar{\mu}) \geq q(\mu) \qquad \text{for all } \mu \in \mathrm{dom}(q). \tag{1.3}$$

The set of all subgradients of q at $\bar{\mu}$ is denoted by $\partial q(\bar{\mu})$.

1.2.2 Primal and Dual Problem

We consider the following constrained optimization problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0 \\ & x \in X, \end{aligned} \tag{1.4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $g = (g_1, \ldots, g_m)'$ and each $g_j : \mathbb{R}^n \to \mathbb{R}$ is a convex function, and $X \subset \mathbb{R}^n$ is a nonempty closed convex set. We refer to problem (1.4) as the *primal problem* . We denote the primal optimal value by f^* and the primal optimal set by X^*. Throughout this section, we assume that the value f^* is finite.

We next define the *dual problem* for problem (1.4). The dual problem is obtained by first relaxing the inequality constraints $g(x) \leq 0$ in problem (1.4), which yields the *dual function* $q : \mathbb{R}^m \to \mathbb{R}$ given by

$$q(\mu) = \inf_{x \in X} \{f(x) + \mu' g(x)\}. \tag{1.5}$$

The dual problem is then given by

$$\begin{aligned} &\text{maximize} && q(\mu) \\ &\text{subject to} && \mu \geq 0 \\ & && \mu \in \mathbb{R}^m. \end{aligned} \tag{1.6}$$

We denote the dual optimal value by q^* and the dual optimal set by M^*.

The primal and the dual problem can be visualized geometrically by considering the set V of constraint-cost function values as x ranges over the set X, i.e.,

$$V = \{(g(x), f(x)) \mid x \in X\},$$

(see Figure 1.2). In this figure, the primal optimal value f^* corresponds to the minimum vertical axis value of all points on the left-half plane, i.e., all points of the form $\{g(x) \leq 0 \mid x \in X\}$. Similarly, for a given dual feasible solution $\mu \geq 0$, the dual function value $q(\mu)$ corresponds to the vertical intercept value of all hyperplanes with normal $(\mu, 1)$ and support the set V from below.[2] The dual optimal value q^* then corresponds to the maximum intercept value of such hyperplanes over all $\mu \geq 0$ [see Figure 1.2(b)]. This figure provides much insight about the relation between the primal and the dual problems and the structure of dual optimal solutions, and has been used recently to develop a duality theory based on geometric principles (see [2] for convex constrained optimization problems, and [36, 37, 42] for nonconvex constrained optimization problems).

Duality Gap and Dual Solutions

It is clear from the geometric picture that the primal and dual optimal values satisfy $q^* \leq f^*$, which is the well-known *weak duality* relation (see Bertsekas *et al.* [4]). When $f^* = q^*$, we say that *there is no duality gap* or *strong duality holds.* The next condition guarantees that there is no duality gap.

Assumption 1. *(Slater Condition)* There exists a vector $\bar{x} \in X$ such that

$$g_j(\bar{x}) < 0 \qquad \text{for all } j = 1, \ldots, m.$$

We refer to a vector $\bar{x}$ satisfying the Slater condition as *a Slater vector.*

[2] A *hyperplane* $H \subset \mathbb{R}^n$ is an $(n-1)$-dimensional affine set, which is defined through its nonzero normal vector $a \in \mathbb{R}^n$ and a scalar b as

$$H = \{x \in \mathbb{R}^n \mid a'x = b\}.$$

Any vector $\bar{x} \in H$ can be used to determine the constant b as $a'\bar{x} = b$, thus yielding an equivalent representation of the hyperplane H as

$$H = \{x \in \mathbb{R}^n \mid a'x = a'\bar{x}\}.$$

Here, we consider hyperplanes in $\mathbb{R}^{r+1}$ with normal vectors given by $(\mu, 1) \in \mathbb{R}^{r+1}$.

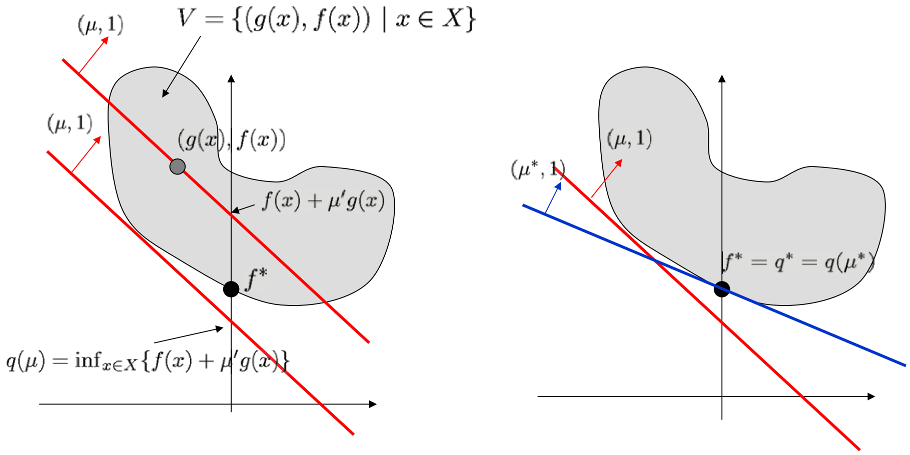


Figure 1.2 Illustration of the primal and the dual problem.

Under the convexity assumptions on the primal problem (1.4) and the assumption that f^* is finite, it is well-known that the Slater condition is sufficient for no duality gap as well as for the existence of a dual optimal solution (see for example Bertsekas [3] or Bertsekas *et. al* [4]). Furthermore, under Slater condition, the dual optimal set is bounded (see Uzawa [53] and Hiriart-Urruty and Lemaréchal [19]). Figure 1.3 provides some intuition for the role of convexity and the Slater condition in establishing no duality gap and the boundedness of the dual optimal solution set.

The following lemma extends the result on the optimal dual set boundedness under the Slater condition. In particular, it shows that the Slater condition also guarantees the boundedness of the (level) sets $\{\mu \geq 0 \mid q(\mu) \geq q(\bar{\mu})\}$.

Lemma 1.1. Let the Slater condition hold [cf. Assumption 1]. Then, the set $Q_{\bar{\mu}} = \{\mu \geq 0 \mid q(\mu) \geq q(\bar{\mu})\}$ is bounded and, in particular, we have

$$\max_{\mu \in Q_{\bar{\mu}}} \|\mu\| \leq \frac{1}{\gamma}\left(f(\bar{x}) - q(\bar{\mu})\right),$$

where $\gamma = \min_{1\leq j\leq m}\{-g_j(\bar{x})\}$ and $\bar{x}$ is a Slater vector.

Proof. We have for any $\mu \in Q_{\bar{\mu}}$,

$$q(\bar{\mu}) \leq q(\mu) = \inf_{x\in X}\{f(x) + \mu' g(x)\} \leq f(\bar{x}) + \mu' g(\bar{x}) = f(\bar{x}) + \sum_{j=1}^{m} \mu_j g_j(\bar{x}),$$

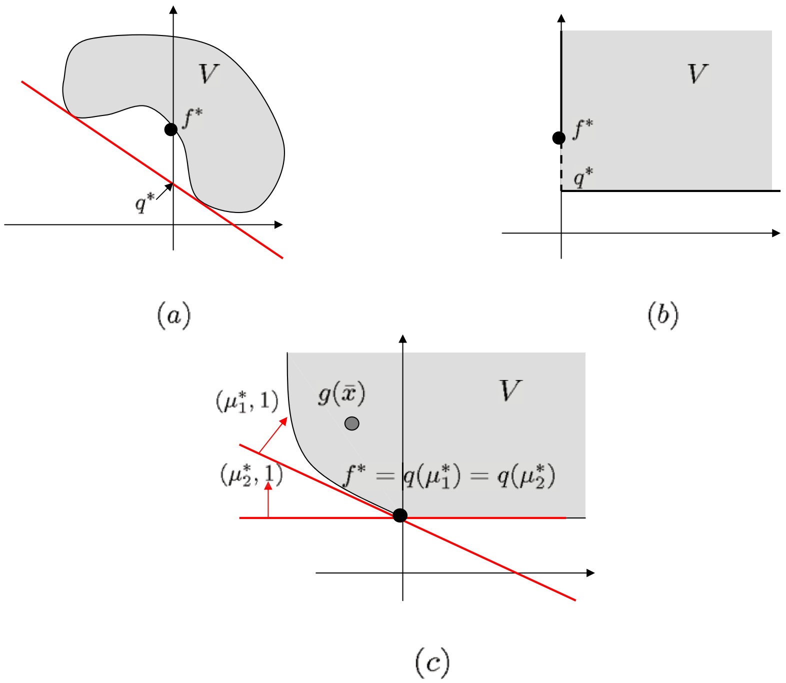


Figure 1.3 Parts (a) and (b) provide two examples where there is a duality gap [due to lack of convexity in (a) and lack of "continuity around origin" in (b)]. Part (c) illustrates the role of the Slater condition in establishing no duality gap and boundedness of the dual optimal solutions. Note that dual optimal solutions correspond to the normal vectors of the (nonvertical) hyperplanes supporting set V from below at the point $(0, q^*)$.

implying that

$$-\sum_{j=1}^{m} \mu_j g_j(\bar{x}) \leq f(\bar{x}) - q(\bar{\mu}).$$

Because $g_j(\bar{x}) < 0$ and $\mu_j \geq 0$ for all j, it follows that

$$\min_{1 \leq j \leq m} \{-g_j(\bar{x})\} \sum_{j=1}^{m} \mu_j \leq -\sum_{j=1}^{m} \mu_j g_j(\bar{x}) \leq f(\bar{x}) - q(\bar{\mu}).$$

Therefore,

$$\sum_{j=1}^{m} \mu_j \leq \frac{f(\bar{x}) - q(\bar{\mu})}{\min_{1 \leq j \leq m} \{-g_j(\bar{x})\}}.$$

Since $\mu \geq 0$, we have $\|\mu\| \leq \sum_{j=1}^{m} \mu_j$ and the estimate follows. **Q.E.D.**

It can be seen from the preceding lemma that under the Slater condition, the dual optimal set M^* is nonempty. In particular, by noting that $M^* = \{\mu \geq 0 \mid q(\mu) \geq q^*\}$ and by using Lemma 1.1, we see that

$$\max_{\mu^* \in M^*} \|\mu^*\| \leq \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right), \tag{1.7}$$

with $\gamma = \min_{1 \leq j \leq m}\{-g_j(\bar{x})\}$.

Dual Subgradient Method

Since the dual function $q(\mu)$ given by Eq. (1.5) is the infimum of a collection of affine functions, it is a concave function (see [4]). Hence, we can use a subgradient method to solve the dual problem (1.6). In view of its implementation simplicity, we consider the classical subgradient algorithm with a constant stepsize:

$$\mu_{k+1} = [\mu_k + \alpha g_k]^+ \qquad \text{for } k = 0, 1, \ldots, \tag{1.8}$$

where the vector $\mu_0 \geq 0$ is an initial iterate, the scalar $\alpha > 0$ is a stepsize, and the vector g_k is a subgradient of q at μ_k. Due to the form of the dual function q, the subgradients of q at a vector μ are related to the primal vectors x_μ attaining the minimum in Eq. (1.5). Specifically, the set $\partial q(\mu)$ of subgradients of q at a given $\mu \geq 0$ is given by

$$\partial q(\mu) = \text{conv}\left(\{g(x_\mu) \mid x_\mu \in X_\mu\}\right), \quad X_\mu = \{x_\mu \in X \mid q(\mu) = f(x_\mu) + \mu' g(x_\mu)\}, \tag{1.9}$$

where conv(Y) denotes the convex hull of a set Y (see [4]).

1.2.3 Distributed Methods for Utility-based Network Resource Allocation

In this section, we consider a utility-based network resource allocation problem and briefly discuss how dual decomposition and subgradient methods lead to decentralized optimization methods that can be used over a network. This approach was proposed in the seminal work of Kelly [23] and further developed by Low and Lapsley [27], Shakkottai and Srikant [48], and Srikant [50].

Consider a network that consists of a set $\mathcal{S} = \{1, \ldots, S\}$ of sources and a set $\mathcal{L} = \{1, \ldots, L\}$ of undirected links, where a link l has capacity c_l. Let $\mathcal{L}(i) \subset \mathcal{L}$ denote the set of links used by source i. The application requirements of source i is represented by a concave increasing utility function $u_i : [0, \infty) \to [0, \infty)$, i.e., each source i gains a utility $u_i(x_i)$ when it sends data at a rate x_i. We further assume that rate x_i is constrained to lie in the interval $I_i = [0, M_i]$ for all $i \in \mathcal{S}$, where the scalar M_i denotes the maximum allowed rate for source i. Let $\mathcal{S}(l) = \{i \in \mathcal{S} \mid l \in \mathcal{L}(i)\}$ denote the set of sources that use link l. The goal of the *network utility maximization problem* is to allocate the source rates as the optimal solution of the problem

$$\text{maximize} \quad \sum_{i\in\mathcal{S}} u_i(x_i) \tag{1.10}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{S}(l)} x_i \le c_l \quad \text{for all } l \in \mathcal{L} \tag{1.11}$$

$$x_i \in I_i \quad \text{for all } i \in \mathcal{S}.$$

This problem is a special case of the multi-agent optimization problem (1.1), where the local objective function of each agent (or source) $f_i(x)$ is given by $f_i(x) = -u_i(x_i)$, i.e., the local objective function of each agent depends only on one component of the decision vector x and the overall objective function $f(x)$ is the sum of the local objective functions, $f(x) = -\sum_i u_i(x_i)$, i.e., the global objective function $f(x)$ is separable in the components of the decision vector. Moreover, the global constraint set C_g is given by the link capacity constraints (1.11), and the local constraint set of each agent X_i is given by the interval I_i. Note that only agent i knows his utility function $u_i(x_i)$ and his maximum allowed rate M_i, which specifies the local constraint $x_i \in I_i$.

Solving problem (1.10) directly by applying existing subgradient methods requires coordination among sources and therefore may be impractical for real networks. This is in view of the fact that in large-scale networks, such as the Internet, there is no central entity that has access to both the source utility functions and constraints, and the capacity of all the links in the network. Despite this information structure, in view of the separable structure of the objective and constraint functions, the dual problem can be evaluated exactly using decentralized information. In particular, the dual problem of (1.10) is given by (1.6), where the dual function takes the form

$$\begin{aligned} q(\mu) &= \max_{x_i\in I_i,\ i\in\mathcal{S}} \sum_{i\in\mathcal{S}} u_i(x_i) - \sum_{l\in\mathcal{L}} \mu_l \Big(\sum_{i\in\mathcal{S}(l)} x_i - c_l \Big) \\ &= \max_{x_i\in I_i,\ i\in\mathcal{S}} \sum_{i\in\mathcal{S}} \Big(u_i(x_i) - x_i \sum_{l\in\mathcal{L}(i)} \mu_l \Big) + \sum_{l\in\mathcal{L}} \mu_l c_l. \end{aligned}$$

Since the optimization problem on the right-hand side of the preceding relation is separable in the variables x_i, the problem decomposes into subproblems for each source i. Letting $\mu_i = \sum_{l\in\mathcal{L}(i)} \mu_l$ for each i (i.e., μ_i is the sum of the multipliers corresponding to the links used by source i), we can write the dual function as

$$q(\mu) = \sum_{i\in\mathcal{S}} \max_{x_i\in I_i} \{u_i(x_i) - x_i\mu_i\} + \sum_{l\in\mathcal{L}} \mu_l c_l.$$

Hence, to evaluate the dual function, each source i needs to solve the one-dimensional optimization problem $\max_{x_i\in I_i}\{u_i(x_i) - x_i\mu_i\}$. This involves only its own utility function u_i and the value μ_i, which is available to source i in practical networks (through a direct feedback mechanism from its destination).

Using a subgradient method to solve the dual problem (1.6) yields the following distributed optimization method, where at each iteration $k \ge 0$, links and

sources update their prices (or dual solution values) and rates respectively in a decentralized manner:
Link Price Update: Each link l updates its price μ_l according to

$$\mu_l(k+1) = [\mu_l(k) + \alpha g_l(k)]^+,$$

where $g_l(k) = \sum_{i \in \mathcal{S}(l)} x_i(k) - c_l$, i.e., $g_l(k)$ is the value of the link l capacity constraint (1.11) at the primal vector $x(k)$ [see the relation for the subgradient of the dual function $q(\mu)$ in Eq. (1.9)].
Source Rate Update: Each source i updates its rate x_i according to

$$x_i(k+1) = \arg\max_{x_i \in I_i} \{u_i(x_i) - x_i \mu_i\}.$$

The preceding methodology has motivated much interest in using dual decomposition and subgradient methods to solve network resource allocation problems in an iterative decentralized manner (see Chiang *et. al* [14]). Other problems where the dual problem has a structure that allows exact evaluation of the dual function using local information include the problem of processor speed control considered by Mutapcic *et. al* [30], and the traffic equilibrium and road pricing problems considered by Larsson *et. al* [24], [25], [26].

1.2.4 Approximate Primal Solutions and Rate Analysis

We first establish some basic relations that hold for a sequence $\{\mu_k\}$ obtained by the subgradient algorithm of Eq. (1.8).

Lemma 1.2. Let the sequence $\{\mu_k\}$ be generated by the subgradient algorithm (1.8). For any $\mu \geq 0$, we have

$$\|\mu_{k+1} - \mu\|^2 \leq \|\mu_k - \mu\|^2 - 2\alpha\left(q(\mu) - q(\mu_k)\right) + \alpha^2 \|g_k\|^2 \qquad \text{for all } k \geq 0.$$

Proof. By using the nonexpansive property of the projection operation, from relation (1.8) we obtain for any $\mu \geq 0$ and all k,

$$\|\mu_{k+1} - \mu\|^2 = \left\|[\mu_k + \alpha g_k]^+ - \mu\right\|^2 \leq \|\mu_k + \alpha g_k - \mu\|^2 .$$

Therefore,

$$\|\mu_{k+1} - \mu\|^2 \leq \|\mu_k - \mu\|^2 + 2\alpha g_k'(\mu_k - \mu) + \alpha^2 \|g_k\|^2 \qquad \text{for all } k.$$

Since g_k is a subgradient of q at μ_k [cf. Eq. (1.3)], we have

$$g_k'(\mu - \mu_k) \geq q(\mu) - q(\mu_k),$$

implying that

$$g_k'(\mu_k - \mu) \leq -\left(q(\mu) - q(\mu_k)\right).$$

Hence, for any $\mu \geq 0$,

$$\|\mu_{k+1} - \mu\|^2 \leq \|\mu_k - \mu\|^2 - 2\alpha\left(q(\mu) - q(\mu_k)\right) + \alpha^2\|g_k\|^2 \qquad \text{for all } k.$$

Q.E.D.

Boundedness of Dual Iterates

Here, we show that the dual sequence $\{\mu_k\}$ generated by the subgradient algorithm is bounded under the Slater condition and a boundedness assumption on the subgradient sequence $\{g_k\}$. We formally state the latter requirement in the following.

Assumption 2. *(Bounded Subgradients)* The subgradient sequence $\{g_k\}$ is bounded, i.e., there exists a scalar $L > 0$ such that

$$\|g_k\| \leq L \qquad \text{for all } k \geq 0.$$

This assumption is satisfied, for example, when the primal constraint set X is compact. Due to the convexity of the constraint functions g_j over $\mathbb{R}^n$, each g_j is continuous over $\mathbb{R}^n$. Thus, $\max_{x\in X}\|g(x)\|$ is finite and provides an upper bound on the norms of the subgradients g_k, i.e.,

$$\|g_k\| \leq L \qquad \text{for all } k \geq 0, \qquad \text{with } L = \max_{x\in X}\|g(x)\|.$$

In the following lemma, we establish the boundedness of the dual sequence generated by the subgradient method.

Lemma 1.3. Let the dual sequence $\{\mu_k\}$ be generated by the subgradient algorithm of Eq. (1.8). Also, let the Slater condition and the Bounded Subgradients assumption hold [cf. Assumptions 1 and 2]. Then, the sequence $\{\mu_k\}$ is bounded and, in particular, we have

$$\|\mu_k\| \leq \frac{2}{\gamma}\left(f(\bar{x}) - q^*\right) + \max\left\{\|\mu_0\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \alpha L\right\},$$

where $\gamma = \min_{1\leq j\leq m}\{-g_j(\bar{x})\}$, $\bar{x}$ is a Slater vector, L is the subgradient norm bound, and $\alpha > 0$ is the stepsize.

Proof. Under the Slater condition the optimal dual set M^* is nonempty. Consider the set Q_α defined by

$$Q_\alpha = \left\{\mu \geq 0 \mid q(\mu) \geq q^* - \frac{\alpha L^2}{2}\right\},$$

which is nonempty in view of $M^* \subset Q_\alpha$. We fix an arbitrary $\mu^* \in M^*$ and we first prove that for all $k \geq 0$,

$$\|\mu_k - \mu^*\| \leq \max\left\{\|\mu_0 - \mu^*\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \|\mu^*\| + \alpha L\right\}, \qquad (1.12)$$

where $\gamma = \min_{1 \leq j \leq m}\{-g_j(\bar{x})\}$ and L is the bound on the subgradient norms $\|g_k\|$. Then, we use Lemma 1.1 to prove the desired estimate.

We show that relation (1.12) holds by induction on k. Note that the relation holds for $k = 0$. Assume now that it holds for some $k > 0$, i.e.,

$$\|\mu_k - \mu^*\| \leq \max\left\{\|\mu_0 - \mu^*\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \|\mu^*\| + \alpha L\right\} \text{ for some } k > 0. \tag{1.13}$$

We now consider two cases: $q(\mu_k) \geq q^* - \alpha L^2/2$ and $q(\mu_k) < q^* - \alpha L^2/2$.

Case 1: $q(\mu_k) \geq q^* - \alpha L^2/2$. By using the definition of the iterate μ_{k+1} in Eq. (1.8) and the subgradient boundedness, we obtain

$$\|\mu_{k+1} - \mu^*\| \leq \|\mu_k + \alpha g_k - \mu^*\| \leq \|\mu_k\| + \|\mu^*\| + \alpha L.$$

Since $q(\mu_k) \geq q^* - \alpha L^2/2$, it follows that $\mu_k \in Q_\alpha$. According to Lemma 1.1, the set Q_α is bounded and, in particular, $\|\mu\| \leq \frac{1}{\gamma}\left(f(\bar{x}) - q^* + \alpha L^2/2\right)$ for all $\mu \in Q_\alpha$. Therefore

$$\|\mu_k\| \leq \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma}.$$

By combining the preceding two relations, we obtain

$$\|\mu_{k+1} - \mu^*\| \leq \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \|\mu^*\| + \alpha L,$$

thus showing that the estimate in Eq. (1.12) holds for $k + 1$.

Case 2: $q(\mu_k) < q^* - \alpha L^2/2$. By using Lemma 1.2 with $\mu = \mu^*$, we obtain

$$\|\mu_{k+1} - \mu^*\|^2 \leq \|\mu_k - \mu^*\|^2 - 2\alpha\left(q^* - q(\mu_k)\right) + \alpha^2\|g_k\|^2.$$

By using the subgradient boundedness, we further obtain

$$\|\mu_{k+1} - \mu^*\|^2 \leq \|\mu_k - \mu^*\|^2 - 2\alpha\left(q^* - q(\mu_k) - \frac{\alpha L^2}{2}\right).$$

Since $q(\mu_k) < q^* - \alpha L^2/2$, it follows that $q^* - q(\mu_k) - \alpha L^2/2 > 0$, which when combined with the preceding relation yields

$$\|\mu_{k+1} - \mu^*\| < \|\mu_k - \mu^*\|.$$

By the induction hypothesis [cf. Eq. (1.13)], it follows that the estimate in Eq. (1.12) holds for $k + 1$ as well. Hence, the estimate in Eq. (1.12) holds for all $k \geq 0$.

From Eq. (1.12) we obtain for all $k \geq 0$,

$$\begin{aligned}\|\mu_k\| &\leq \|\mu_k - \mu^*\| + \|\mu^*\| \\ &\leq \max\left\{\|\mu_0 - \mu^*\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \|\mu^*\| + \alpha L\right\} + \|\mu^*\|.\end{aligned}$$

By using $\|\mu_0 - \mu^*\| \leq \|\mu_0\| + \|\mu^*\|$, we further have for all $k \geq 0$,

$$\begin{aligned}\|\mu_k\| &\leq \max\left\{\|\mu_0\| + \|\mu^*\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \|\mu^*\| + \alpha L\right\} + \|\mu^*\| \\ &= 2\|\mu^*\| + \max\left\{\|\mu_0\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \alpha L\right\}.\end{aligned}$$

Since $M^* = \{\mu \geq 0 \mid q(\mu) \geq q^*\}$, according to Lemma 1.1, we have the following bound on the dual optimal solutions

$$\max_{\mu^* \in M^*} \|\mu^*\| \leq \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right),$$

implying that for all $k \geq 0$,

$$\|\mu_k\| \leq \frac{2}{\gamma}\left(f(\bar{x}) - q^*\right) + \max\left\{\|\mu_0\|, \frac{1}{\gamma}\left(f(\bar{x}) - q^*\right) + \frac{\alpha L^2}{2\gamma} + \alpha L\right\}.$$

Q.E.D.

Convergence Rate Estimates

In this section, we generate approximate primal solutions by considering the running averages of the primal sequence $\{x_k\}$ obtained in the implementation of the subgradient method. We show that under the Slater condition, we can provide bounds for the number of subgradient iterations needed to generate a primal solution within a given level of constraint violation. We also derive upper and lower bounds on the gap from the optimal primal value.

To define the approximate primal solutions, we consider the dual sequence $\{\mu_k\}$ generated by the subgradient algorithm in (1.8), and the corresponding sequence of primal vectors $\{x_k\} \subset X$ that provide the subgradients g_k in the algorithm , i.e.,

$$g_k = g(x_k), \qquad x_k \in \arg\min_{x \in X}\{f(x) + \mu_k' g(x)\} \qquad \text{for all } k \geq 0, \tag{1.14}$$

[see the subdifferential relation in (1.9)]. We define $\hat{x}_k$ as the average of the vectors $x_0, \ldots, x_{k-1}$, i.e.,

$$\hat{x}_k = \frac{1}{k}\sum_{i=0}^{k-1} x_i \qquad \text{for all } k \geq 1. \tag{1.15}$$

The average vectors $\hat{x}_k$ lie in the set X because X is convex and $x_i \in X$ for all i. However, these vectors need not satisfy the primal inequality constraints $g_j(x) \leq 0$, $j = 1, \ldots, m$, and therefore, they can be primal infeasible.

The next proposition provides a bound on the amount of feasibility violation of the running averages $\hat{x}_k$. It also provides upper and lower bounds on the primal cost of these vectors. These bounds are given per iteration, as seen in the following.

Theorem 1.1. Let the sequence $\{\mu_k\}$ be generated by the subgradient algorithm (1.8). Let the Slater condition and Bounded Subgradients assumption hold [cf. Assumptions 1 and 2]. Also, let

$$B^* = \frac{2}{\gamma}\,(f(\bar{x}) - q^*) + \max\left\{\|\mu_0\|, \frac{1}{\gamma}\,(f(\bar{x}) - q^*) + \frac{\alpha L^2}{2\gamma} + \alpha L\right\}. \tag{1.16}$$

Let the vectors $\hat{x}_k$ for $k \geq 1$ be the averages given by Eq. (1.15). Then, the following holds for all $k \geq 1$:

(a) An upper bound on the amount of constraint violation of the vector $\hat{x}_k$ is given by

$$\|g(\hat{x}_k)^+\| \leq \frac{B^*}{k\alpha}.$$

(b) An upper bound on the primal cost of the vector $\hat{x}_k$ is given by

$$f(\hat{x}_k) \leq f^* + \frac{\|\mu_0\|^2}{2k\alpha} + \frac{\alpha L^2}{2}.$$

(c) A lower bound on the primal cost of the vector $\hat{x}_k$ is given by

$$f(\hat{x}_k) \geq f^* - \frac{1}{\gamma}\,[\, f(\bar{x}) - q^* \,]\ \|g(\hat{x}_k)^+\|.$$

Proof. (a) By using the definition of the iterate μ_{k+1} in Eq. (1.8), we obtain

$$\mu_k + \alpha g_k \leq [\mu_k + \alpha g_k]^+ = \mu_{k+1} \qquad \text{for all } k \geq 0.$$

Since $g_k = g(x_k)$ with $x_k \in X$, it follows that

$$\alpha g(x_k) \leq \mu_{k+1} - \mu_k \qquad \text{for all } k \geq 0.$$

Therefore,

$$\sum_{i=0}^{k-1} \alpha g(x_i) \leq \mu_k - \mu_0 \leq \mu_k \qquad \text{for all } k \geq 1,$$

where the last inequality in the preceding relation follows from $\mu_0 \geq 0$. Since $x_k \in X$ for all k, by the convexity of X, we have $\hat{x}_k \in X$ for all k. Hence, by the convexity of each of the functions g_j, it follows that

$$g(\hat{x}_k) \leq \frac{1}{k}\sum_{i=0}^{k-1} g(x_i) = \frac{1}{k\alpha}\sum_{i=0}^{k-1} \alpha g(x_i) \leq \frac{\mu_k}{k\alpha} \qquad \text{for all } k \geq 1.$$

Because $\mu_k \geq 0$ for all k, we have $g(\hat{x}_k)^+ \leq \mu_k/(k\alpha)$ for all $k \geq 1$ and, therefore,

$$\left\|g(\hat{x}_k)^+\right\| \leq \frac{\|\mu_k\|}{k\alpha} \qquad \text{for all } k \geq 1. \tag{1.17}$$

By Lemma 1.3 we have

$$\|\mu_k\| \leq \frac{2}{\gamma}\,(f(\bar{x}) - q^*) + \max\left\{\|\mu_0\|, \frac{1}{\gamma}\,(f(\bar{x}) - q^*) + \frac{\alpha L^2}{2\gamma} + \alpha L\right\} \qquad \text{for all } k \geq 0.$$

By the definition of B^* in Eq. (1.16), the preceding relation is equivalent to

$$\|\mu_k\| \leq B^* \qquad \text{for all } k \geq 0.$$

Combining this relation with Eq. (1.17), we obtain

$$\|g(\hat{x}_k)^+\| \leq \frac{\|\mu_k\|}{k\alpha} \leq \frac{B^*}{k\alpha} \qquad \text{for all } k \geq 1.$$

(b) By the convexity of the primal cost $f(x)$ and the definition of x_k as a minimizer of the Lagrangian function $f(x) + \mu_k' g(x)$ over $x \in X$ [cf. Eq. (1.14)], we have

$$f(\hat{x}_k) \leq \frac{1}{k}\sum_{i=0}^{k-1} f(x_i) = \frac{1}{k}\sum_{i=0}^{k-1}\{f(x_i) + \mu_i' g(x_i)\} - \frac{1}{k}\sum_{i=0}^{k-1}\mu_i' g(x_i).$$

Since $q(\mu_i) = f(x_i) + \mu_i' g(x_i)$ and $q(\mu_i) \leq q^*$ for all i, it follows that for all $k \geq 1$,

$$f(\hat{x}_k) \leq \frac{1}{k}\sum_{i=0}^{k-1} q(\mu_i) - \frac{1}{k}\sum_{i=0}^{k-1}\mu_i' g(x_i) \leq q^* - \frac{1}{k}\sum_{i=0}^{k-1}\mu_i' g(x_i). \tag{1.18}$$

From the definition of the algorithm in Eq. (1.8), by using the nonexpansive property of the projection, and the facts $0 \in \{\mu \in \mathbb{R}^m \mid \mu \geq 0\}$ and $g_i = g(x_i)$, we obtain

$$\|\mu_{i+1}\|^2 \leq \|\mu_i\|^2 + 2\alpha\mu_i' g(x_i) + \alpha^2\|g(x_i)\|^2 \qquad \text{for all } i \geq 0,$$

implying that

$$-\mu_i' g(x_i) \leq \frac{\|\mu_i\|^2 - \|\mu_{i+1}\|^2 + \alpha^2\|g(x_i)\|^2}{2\alpha} \qquad \text{for all } i \geq 0.$$

By summing over $i = 0, \ldots, k-1$ for $k \geq 1$, we have

$$-\frac{1}{k}\sum_{i=0}^{k-1}\mu_i' g(x_i) \leq \frac{\|\mu_0\|^2 - \|\mu_k\|^2}{2k\alpha} + \frac{\alpha}{2k}\sum_{i=0}^{k-1}\|g(x_i)\|^2 \qquad \text{for all } k \geq 1.$$

Combining the preceding relation and Eq. (1.18), we further have

$$f(\hat{x}_k) \leq q^* + \frac{\|\mu_0\|^2 - \|\mu_k\|^2}{2k\alpha} + \frac{\alpha}{2k}\sum_{i=0}^{k-1}\|g(x_i)\|^2 \qquad \text{for all } k \geq 1.$$

Under the Slater condition, there is zero duality gap, i.e., $q^* = f^*$. Furthermore, the subgradients are bounded by a scalar L [cf. Assumption 2], so that

$$f(\hat{x}_k) \leq f^* + \frac{\|\mu_0\|^2}{2k\alpha} + \frac{\alpha L^2}{2} \qquad \text{for all } k \geq 1,$$

yielding the desired estimate.

(c) Given a dual optimal solution μ^*, we have

$$f(\hat{x}_k) = f(\hat{x}_k) + (\mu^*)' g(\hat{x}_k) - (\mu^*)' g(\hat{x}_k) \geq q(\mu^*) - (\mu^*)' g(\hat{x}_k).$$

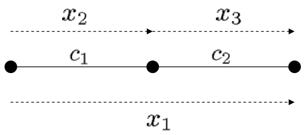


Figure 1.4 A simple network with two links of capacities $c_1 = 1$ and $c_2 = 2$, and three users, each sending data at a rate x_i.

Because $\mu^* \geq 0$ and $g(\hat{x}_k)^+ \geq g(\hat{x}_k)$, we further have

$$-(\mu^*)' g(\hat{x}_k) \geq -(\mu^*)' g(\hat{x}_k)^+ \geq -\|\mu^*\| \|g(\hat{x}_k)^+\|.$$

From the preceding two relations and the fact $q(\mu^*) = q^* = f^*$ it follows that

$$f(\hat{x}_k) \geq f^* - \|\mu^*\| \|g(\hat{x}_k)^+\|.$$

By using Lemma 1.1 with $\bar{\mu} = \mu^*$, we see that the dual set is bounded and, in particular, $\|\mu^*\| \leq \frac{1}{\gamma}(f(\bar{x}) - q^*)$ for all dual optimal vectors μ^*. Hence,

$$f(\hat{x}_k) \geq f^* - \frac{1}{\gamma} \left[f(\bar{x}) - q^* \right] \|g(\hat{x}_k)^+\| \qquad \text{for all } k \geq 1.$$

Q.E.D.

1.2.5 Numerical Example

In this section, we study a numerical example to illustrate the performance of the dual subgradient method with primal averaging for the utility-based network resource allocation problem described in Section 1.2.3. Consider the network illustrated in Figure 1.4 with 2 serial links and 3 users each sending data at a rate x_i for $i = 1, 2, 3$. Link 1 has a capacity $c_1 = 1$ and link 2 has a capacity $c_2 = 2$. Assume that each user has an identical concave utility function $u_i(x_i) = \sqrt{x_i}$, which represents the utility gained from sending rate x_i. We consider allocating rates among the users as the optimal solution of the problem

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{3} \sqrt{x_i} \\ \text{subject to} & x_1 + x_2 \leq 1, \quad x_1 + x_3 \leq 2, \\ & x_i \geq 0, \quad i = 1, 2, 3. \end{array}$$

The optimal solution of this problem is $x^* = [0.2686, 0.7314, 1.7314]$ and the optimal value is $f^* \approx 2.7$. We consider solving this problem using the dual subgradient method of Eq. (1.8) (with a constant stepsize $\alpha = 1$) combined with primal averaging. In particular, when evaluating the subgradients of the dual function in Eq. (1.14), we obtain the primal sequence $\{x_k\}$. We generate the sequence $\{\hat{x}_k\}$ as the running average of the primal sequence [cf. Eq. (1.15)].

Figure 1.5 illustrates the behavior of the sequences $\{x_{ik}\}$ and $\{\hat{x}_{ik}\}$ for each user $i = 1, 2, 3$. As seen in this figure, for each user i, the sequences $\{x_{ik}\}$ exhibit

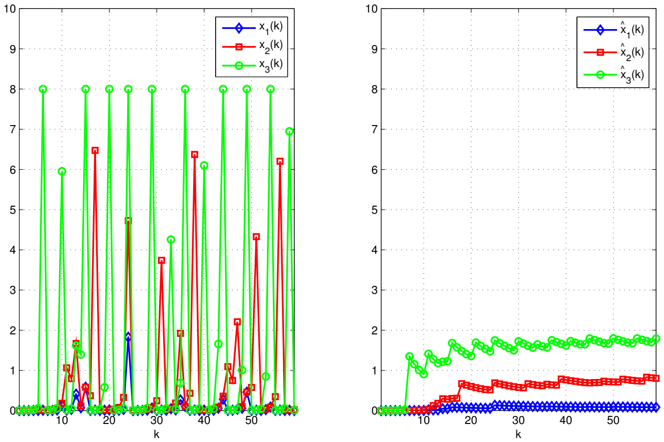


Figure 1.5 The convergence behavior of the primal sequence $\{x_k\}$ (on the left) and $\{\hat{x}_k\}$ (on the right).

oscillations whereas the average sequences $\{\hat{x}_{ik}\}$ converge smoothly to near-optimal solutions within 60 subgradient iterations.

Figure 1.6 illustrates the results for the constraint violation and the primal objective value for the sequences $\{x_k\}$ and $\{\hat{x}_k\}$. The plot to the left in Figure 1.6 shows the convergence behavior of the constraint violation $\|g(x)^+\|$ for the two sequences, i.e., $\|g(x_k)^+\|$ and $\|g(\hat{x}_k)^+\|$. Note that the constraint violation for the sequence $\{x_k\}$ oscillates within a large range while the constraint violation for the average sequence $\{\hat{x}_k\}$ rapidly converges to 0. The plot to the right in Figure 3 shows a similar convergence behavior for the primal objective function values $f(x)$ along the sequences $\{x_k\}$ and $\{\hat{x}_k\}$.

1.3 Distributed Optimization Methods using Consensus Algorithms

In this section, we develop distributed methods for minimizing the sum of non-separable convex functions corresponding to multiple agents connected over a network with time-varying topology. These methods combine first-order methods and a consensus algorithm . The consensus part serves as a basic mechanism for distributing the computations among the agents and allowing us to solve the problem in a decentralized fashion.

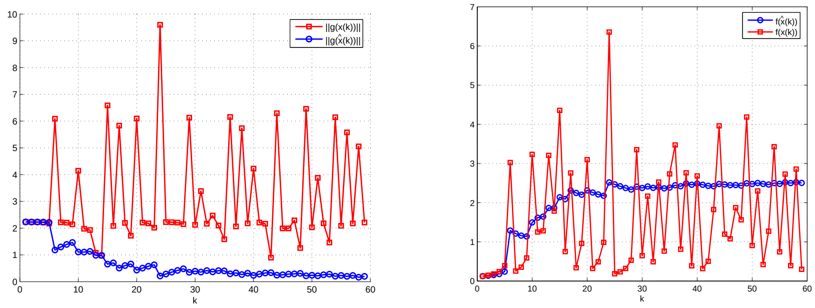


Figure 1.6 The figure on the left shows the convergence behavior of the constraint violation for the two primal sequences, $\{x_k\}$ and $\{\hat{x}_k\}$. Similarly, the figure on the right shows the convergence of the corresponding primal objective function values.

In contrast with the setting considered in Section 1.2.3 where each agent has an objective function that depends only on the resource allocated to that agent, the model discussed in this section allows for the individual cost functions to depend on the entire resource allocation vector. In particular, the focus here is on a distributed optimization problem in a network consisting of m agents that communicate locally. The global objective is to cooperatively minimize the cost function $\sum_{i=1}^{m} f_i(x)$, where the function $f_i : \mathbb{R}^n \to \mathbb{R}$ represents the cost function of agent i, known by this agent only, and $x \in \mathbb{R}^n$ is a decision vector. The decision vector can be viewed as either a resource vector where sub-components correspond to resources allocated to each agent, or a global decision vector which the agents are trying to compute using local information.

The approach presented here builds on the seminal work of Tsitsiklis [51] (see also Tsitsiklis *et al.* [52], Bertsekas and Tsitsiklis [5]), who developed a framework for the analysis of distributed computation models.[3] As mentioned earlier, the approach here is to use the consensus as a mechanism for distributing the computations among the agents. The problem of reaching a consensus on a particular scalar value, or computing exact averages of the initial values of the agents, has attracted much recent attention as natural models of cooperative behavior in networked-systems (see Vicsek *et al.* [54], Jadbabaie *et al.* [20], Boyd *et al.* [8], Olfati-Saber and Murray [44], Cao *et al.* [11], and Olshevsky and Tsitsiklis [45]). Exploiting the consensus idea, recent work [40] (see also the short paper [33]) has proposed a distributed model for optimization over a network.

[3] This framework focuses on the minimization of a (smooth) function $f(x)$ by distributing the processing of the components of vector $x \in \mathbb{R}^n$ among n agents.

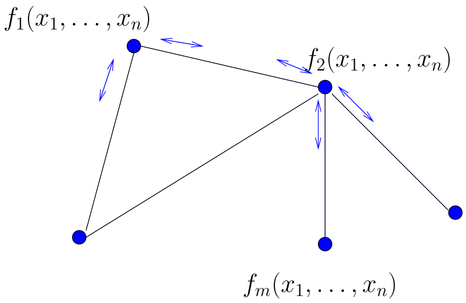


Figure 1.7 Illustration of the network with each agent having its local objective and communicating locally with its neighbors.

1.3.1 Problem and Algorithm

In this section, we formulate the problem of interest and present a distributed algorithm for solving the problem.

Problem

We consider the problem of optimizing the sum of convex objective functions corresponding to m agents connected over a time-varying topology. The goal of the agents is to cooperatively solve the unconstrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} f_i(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array} \tag{1.19}$$

where each $f_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function, representing the local objective function of agent i, which is known only to this agent. This problem is an unconstrained version of the multi-agent optimization problem (1.1), where the global objective function $f(x)$ is given by the sum of the individual local objective functions $f_i(x)$, i.e.,

$$f(x) = \sum_{j=1}^{m} f_i(x),$$

(see Figure 1.7). We denote the optimal value of problem (1.19) by f^* and the set of optimal solutions by X^*.

To keep our discussion general, we do not assume differentiability of any of the functions f_i. Since each f_i is convex over the entire $\mathbb{R}^n$, the function is differentiable almost everywhere (see [4] or [47]). At the points where the function fails to be differentiable, a subgradient exists [as defined in Eq. (1.2)] and it can be used in "the role of a gradient".

Algorithm

We next introduce a distributed subgradient algorithm for solving problem (1.19). The main idea of the algorithm is the use of consensus as a mechanism for distributing the computations among the agents. In particular, each agent starts with an initial estimate $x_i(0) \in \mathbb{R}^n$ and updates its estimate at discrete times $t_k, k = 1, 2, \ldots$. We denote by $x_i(k)$ the vector estimate maintained by agent i at time t_k. When updating, an agent i combines its current estimate x_i with the estimates x_j received from its neighboring agents j. Specifically, agent i updates its estimates by setting

$$x_i(k+1) = \sum_{j=1}^{m} a_{ij}(k) x_j(k) - \alpha d_i(k), \tag{1.20}$$

where the scalar $\alpha > 0$ is a stepsize and the vector $d_i(k)$ is a subgradient of the agent i cost function $f_i(x)$ at $x = x_i(k)$. The scalars $a_{i1}(k), \ldots, a_{im}(k)$ are nonnegative weights that agent i gives to the estimates $x_1(k), \ldots, x_m(k)$. These weights capture two aspects:

1. The active links (j, i) at time k. In particular, the neighbors j that communicate with agent i at time k, will be captured by assigning $a_{ij}(k) > 0$ (including i itself). The neighbors j that do not communicate with i at time k, as well as those that are not neighbors of i, are captured by assigning $a_{ij}(k) = 0$;
2. The weight that agent i gives to the estimates received from its neighbors.

When all objective functions are zero, i.e., $f_i(x) = 0$ for all x and i, the method in (1.20) reduces to

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

which is the consensus algorithm . In view of this, algorithm (1.20) can be seen as a combination of the "consensus step" $\sum_{j=1}^{m} a_{ij}(k) x_j(k)$ and the subgradient step $-\alpha d_i(k)$. The subgradient step is taken by the agent to minimize its own objective $f_i(x)$, while the consensus step serves to align its decision x_i with the decisions of its neighbors. When the network is sufficiently often connected in time to ensure the proper mixing of the agents' estimates, one would expect that all agents have the same estimate after some time, at which point the algorithm would start behaving as a "centralized" method. This intuition is behind the construction of the algorithm and also behind the analysis of its performance.

Representation using Transition Matrices

In the subsequent development, we find it useful to introduce $A(k)$ to denote the *weight matrix* $[a_{ij}(k)]_{i,j=1,\ldots,m}$. Using these matrices, we can capture the evolution of the estimates $x_i(k)$ generated by Eq. (1.20) over a window of time. In particular, we define a *transition matrix* $\Phi(k, s)$ for any s and k with $k \geq s$, as follows:

$$\Phi(k, s) = A(k)A(k-1) \cdots A(s+1)A(s).$$

Through the use of transition matrices, we can relate the estimate $x_i(k+1)$ to the estimates $x_1(s), \ldots, x_m(s)$ for any $s \leq k$. Specifically, for the iterates generated by Eq. (1.20), we have for any i, and any s and k with $k \geq s$,

$$x_i(k+1) = \sum_{j=1}^{m} [\Phi(k,s)]_{ij} x_j(s) - \alpha \sum_{r=s}^{k-1} \sum_{j=1}^{m} [\Phi(k,r+1)]_{ij} d_j(r) - \alpha\, d_i(k). \quad (1.21)$$

As seen from the preceding relation, to study the asymptotic behavior of the estimates $x_i(k)$, we need to understand the behavior of the transition matrices $\Phi(k,s)$. We do this under some assumptions on the agent interactions that translate into some properties of transition matrices, as seen in the next section.

1.3.2 Information Exchange Model

The agent interactions and information aggregation at time k are modeled through the use of the matrix $A(k)$ of agent weights $a_{ij}(k)$. At each time k, this weight matrix captures the information flow (or the communication pattern) among the agents, as well as how the information is aggregated by each agent, i.e., how much actual weight each agent i assigns to its own estimate $x_i(k)$ and the estimates $x_j(k)$ received from its neighbors.

For the proper mixing of the agent information, we need some assumptions on the weights $a_{ij}(k)$ and the agent connectivity in time. When discussing these assumptions, we use the notion of a stochastic vector and a stochastic matrix, defined as follows. A vector a is said to be a *stochastic vector* when its components a_i are nonnegative and $\sum_i a_i = 1$. A square matrix A is said to be *stochastic* when each row of A is a stochastic vector, and it is said to be *doubly stochastic* when both A and its transpose A' are stochastic matrices.

The following assumption puts conditions on the weights $a_{ij}(k)$ in Eq. (1.20).

Assumption 3. *For all $k \geq 0$, the weight matrix $A(k)$ is doubly stochastic with positive diagonal. Additionally, there is a scalar $\eta > 0$ such that if $a_{ij}(k) > 0$, then $a_{ij}(k) \geq \eta$.*

The doubly stochasticity assumption on the weight matrix will guarantee that the function f_i of every agent i receives the same weight in the long run. This ensures that the agents optimize the sum of the functions f_i as opposed to some weighted sum of these functions. The second part of the assumption states that each agent gives significant weight to its own value and to the values of its neighbors. This is needed to ensure that new information is aggregated into the agent system persistently in time.

We note that the lower bound η on weights in Assumption 3 need not be available to any of the agents. The existence of such a bound is merely used in the analysis of the system behavior and the algorithm's performance. Note also that such a bound η exists when each agent has a lower bound η_i on its own weights $a_{ij}(k)$, $j = 1, \ldots, m$, in which case we can define $\eta = \min_{1 \leq i \leq m} \eta_i$.

The following are some examples of how to ensure in a distributed manner that the weight matrix $A(k)$ satisfies Assumption 3 when the agent communications are bidirectional.

Example 1.1: Metropolis-based weights [55] are given by for all i and j with $j \neq i$,

$$a_{ij}(k) = \begin{cases} \frac{1}{1+\max\{n_i(k), n_j(k)\}} & \text{if } j \text{ communicates with } i \text{ at time } k, \\ 0 & \text{otherwise,} \end{cases}$$

with $n_i(k)$ being the number of neighbors of communicating with agent i at time k. Using these, the weights $a_{ii}(k)$ for all $i = 1, \ldots, m$ as follows

$$a_{ii}(k) = 1 - \sum_{j \neq i} a_{ij}(k).$$

The next example can be viewed as a generalization of the Metropolis weights.

Example 1.2: Each agent i has planned weights $\tilde{a}_{ij}(k), j = 1 \ldots, m$ that the agent communicates to its neighbors together with the estimate $x_i(k)$, where the matrix $\tilde{A}(k)$ of planned weights is a (row) stochastic matrix satisfying Assumption 3, except for doubly stochasticity. In particular, at time k, if agent j communicates with agent i, then agent i receives $x_j(k)$ and the planned weight $\tilde{a}_{ji}(k)$ from agent j. At the same time, agent j receives $x_i(k)$ and the planned weight $\tilde{a}_{ij}(k)$ from agent i. Then, the actual weights that an agent i uses are given by

$$a_{ij}(k) = \min\{\tilde{a}_{ij}(k),\, \tilde{a}_{ji}(k)\},$$

if i and j talk at time k, and $a_{ij}(k) = 0$ otherwise; while

$$a_{ii}(k) = 1 - \sum_{\{j \mid j \leftrightarrow i \,\text{at time}\, k\}} a_{ij}(k),$$

where the summation is over all j communicating with i at time k. It can be seen that the weights $a_{ij}(k)$ satisfy Assumption 3.

We need the agent network to be connected to ensure that the information state of each and every agent influences the information state of other agents. However, the network need not be connected at every time instance but rather frequently enough to persistently influence each other. To formalize this assumption, we introduce the index set $\mathcal{N} = \{1, \ldots, m\}$ and we view the agent network as a directed graph with node set $\mathcal{N}$ and time-varying link set. We define $\mathcal{E}(A(k))$

to be the set of directed links at time k induced by the weight matrix $A(k)$. In particular, the link set $\mathcal{E}(A(k))$ is given by

$$\mathcal{E}(A(k)) = \{(j,i) \mid a_{ij}(k) > 0,\ i,j = 1\ldots,m\} \qquad \text{for all } k.$$

Note that each set $\mathcal{E}(A(k))$ includes self-edges (i,i) for all i. Now, the agents' connectivity can be represented by a directed graph $G(k) = (\mathcal{N}, \mathcal{E}(A(k)))$.

The next assumption states that the agent network is frequently connected.

Assumption 4. *There exists an integer $B \geq 1$ such that the directed graph*

$$\Big(\mathcal{N}, \mathcal{E}(A(kB)) \cup \cdots \cup \mathcal{E}(A((k+1)B-1))\Big)$$

is strongly connected for all $k \geq 0$.

Note that the bound B need not be known by any of the agents. This is another parameter that is used in the analysis of the network properties and the algorithm.

1.3.3 Convergence of Transition Matrices

Here, we study the behavior of the transition matrices $\Phi(k,s) = A(k)\cdots A(s)$ that govern the evolution of the estimates over a window of time, as seen from Eq. (1.21). Under Assumptions 3 and 4, we provide some results that we use later in the convergence analysis of method (1.20). These results are of interest on their own for consensus problems and distributed averaging.[4]

To understand the convergence of the transition matrices $\Phi(k,s)$, we start by considering a related "consensus-type" update rule of the form

$$z(k+1) = A(k)z(k), \tag{1.22}$$

where $z(0) \in \mathbb{R}^m$ is an initial vector. This update rule captures the averaging part of Eq. (1.20), as it operates on a particular component of the agent estimates, with the vector $z(k) \in \mathbb{R}^m$ representing the estimates of the different agents for that component.

We define

$$V(k) = \sum_{j=1}^{m} (z_j(k) - \bar{z}(k))^2 \qquad \text{for all } k \geq 0,$$

where $\bar{z}(k)$ is the average of the entries of the vector $z(k)$. Under the doubly stochasticity of $A(k)$, the initial average $\bar{z}(0)$ is preserved by the update rule (1.22), i.e., $\bar{z}(k) = \bar{z}(0)$ for all k. Hence, the function $V(k)$ measures the "disagreement" in agent values.

[4] More detailed development of these results for distributed averaging can be found in [31].

In the next lemma, we give a bound on the decrease of the agent disagreement $V(kB)$, which is linear in η and quadratic in m^{-1}. This bound plays a crucial role in establishing the convergence and rate of convergence of transition matrices, which subsequently are used to analyze the method.

Lemma 1.4. *Let Assumptions 3 and 4 hold. Then, $V(k)$ is nonincreasing in k. Furthermore,*

$$V((k+1)B) \leq \left(1 - \frac{\eta}{2m^2}\right) V(kB) \quad \text{for all } k \geq 0.$$

Proof. This is an immediate consequence of Lemma 5 in [31], stating that[5] under Assumptions 3 and 4, for all k with $V(kB) > 0$,

$$\frac{V(kB) - V((k+1)B)}{V(kB)} \geq \frac{\eta}{2m^2}.$$

Q.E.D.

Using Lemma 1.4, we establish the convergence of the transition matrices $\Phi(k,s)$ of Eq. (1.21) to the matrix with all entries equal to $\frac{1}{m}$, and we provide a bound on the convergence rate. In particular, we show that the difference between the entries of $\Phi(k,s)$ and $\frac{1}{m}$ converges to zero with a geometric rate.

Theorem 1.2. *Let Assumptions 3 and 4 hold. Then, for all i,j and all k,s with $k \geq s$, we have*

$$\left| [\Phi(k,s)]_{ij} - \frac{1}{m} \right| \leq \left(1 - \frac{\eta}{4m^2}\right)^{\lceil \frac{k-s+1}{B} \rceil - 2}.$$

Proof. By Lemma 1.4, we have for all $k \geq s$,

$$V(kB) \leq \left(1 - \frac{\eta}{2m^2}\right)^{k-s} V(sB).$$

Let k and s be arbitrary with $k \geq s$, and let

$$\tau B \leq s < (\tau+1)B, \quad tB \leq k < (t+1)B,$$

with $\tau \leq t$. Hence, by the nonincreasing property of $V(k)$, we have

$$\begin{aligned} V(k) &\leq V(tB) \\ &\leq \left(1 - \frac{\eta}{2m^2}\right)^{t-\tau-1} V((\tau+1)B) \\ &\leq \left(1 - \frac{\eta}{2m^2}\right)^{t-\tau-1} V(s). \end{aligned}$$

Note that $k - s < (t-\tau)B + B$ implying that $\frac{k-s+1}{B} \leq t - \tau + 1$, where we used the fact that both sides of the inequality are integers. Therefore $\lceil \frac{k-s+1}{B} \rceil - 2 \leq$

[5] The assumptions in [31] are actually weaker.

$t - \tau - 1$, and we have for all k and s with $k \geq s$,

$$V(k) \leq \left(1 - \frac{\eta}{2m^2}\right)^{\lceil \frac{k-s+1}{B} \rceil - 2} V(s). \tag{1.23}$$

By Eq. (1.22), we have $z(k+1) = A(k)z(k)$, and therefore $z(k+1) = \Phi(k,s)z(s)$ for all $k \geq s$. Let $e_i \in \mathbb{R}^m$ denote the vector with entries all equal to 0, except for the i-th entry which is equal to 1. Letting $z(s) = e_i$ we obtain $z(k+1) = [\Phi(k,s)]_i'$, where $[\Phi(k,s)]_i'$ denotes the transpose of the i-th row of the matrix. Using the inequalities (1.23) and $V(e_i) \leq 1$, we obtain

$$V([\Phi(k,s)]_i') \leq \left(1 - \frac{\eta}{2m^2}\right)^{\lceil \frac{k-s+1}{B} \rceil - 2}.$$

The matrix $\Phi(k,s)$ is doubly stochastic, because it is the product of doubly stochastic matrices. Thus, the average entry of $[\Phi(k,s)]_i$ is $1/m$ implying that for all i and j,

$$\begin{aligned} \left([\Phi(k,s)]_{ij} - \frac{1}{m}\right)^2 &\leq V([\Phi(k,s)]_i') \\ &\leq \left(1 - \frac{\eta}{2m^2}\right)^{\lceil \frac{k-s+1}{B} \rceil - 2}. \end{aligned}$$

From the preceding relation and $\sqrt{1 - \eta/(2m^2)} \leq 1 - \eta/(4m^2)$, we obtain

$$\left|[\Phi(k,s)]_{ij} - \frac{1}{m}\right| \leq \left(1 - \frac{\eta}{4m^2}\right)^{\lceil \frac{k-s+1}{B} \rceil - 2}.$$

Q.E.D.

1.3.4 Convergence Analysis of the Subgradient Method

Here, we study the convergence properties of the subgradient method (1.20) and, in particular, we obtain a bound on the performance of the algorithm. In what follows, we assume the uniform boundedness of the set of subgradients of the cost functions f_i at all points: for some scalar $L > 0$, we have for all $x \in \mathbb{R}^n$ and all i,

$$\|g\| \leq L \qquad \text{for all } g \in \partial f_i(x), \tag{1.24}$$

where $\partial f_i(x)$ is the set of all subgradients of f_i at x.

The analysis combines the proof techniques used for consensus algorithms and approximate subgradient methods. The consensus analysis rests on the convergence rate result of Theorem 1.2 for transition matrices, which provides a tool for measuring the "agent disagreements" $\|x_i(k) - x_j(k)\|$ in time. Equivalently, we can measure $\|x_i(k) - x_j(k)\|$ in terms of the disagreements $\|x_i(k) - y(k)\|$ with respect to an auxiliary sequence $\{y(k)\}$, defined appropriately. The sequence $\{y_k\}$ will also serve as a basis for understanding the effects of subgradient steps in the algorithm. In fact, we will establish suboptimality property of the sequence

$\{y_k\}$, and then using the estimates for the disagreements $\|x_i(k) - y(k)\|$, we will provide a performance bound for the algorithm.

Disagreement Estimate

To estimate the agent "disagreements", we use an auxiliary sequence $\{y(k)\}$ of reference points, defined as follows[6]:

$$y(k+1) = y(k) - \frac{\alpha}{m}\sum_{i=1}^{m} d_i(k), \tag{1.25}$$

where $d_i(k)$ is the same subgradient of $f_i(x)$ at $x = x_i(k)$ that is used in the method (1.20), and

$$y(0) = \frac{1}{m}\sum_{i=1}^{m} x_i(0).$$

In the following lemma, we estimate the norms of the differences $x_i(k) - y(k)$ at each time k. The result relies on Theorem 1.2.

Lemma 1.5. *Let Assumptions 3 and 4 hold. Assume also that the subgradients of each f_i are uniformly bounded by some scalar L [cf. Eq. (1.24)]. Then for all i and $k \geq 1$,*

$$\|x_i(k) - y(k)\| \leq \beta^{\lceil \frac{k}{B} \rceil - 2}\sum_{j=1}^{m}\|x_j(0)\| + \alpha L\left(2 + \frac{mB}{\beta(1-\beta)}\right),$$

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

Proof. From the definition of the sequence $\{y(k)\}$ in (1.25) it follows for all k,

$$y(k) = \frac{1}{m}\sum_{i=1}^{m} x_i(0) - \frac{\alpha}{m}\sum_{r=0}^{k-1}\sum_{i=1}^{m} d_i(r). \tag{1.26}$$

As given in equation (1.21), for the agent estimates $x_i(k)$ we have for all k,

$$x_i(k+1) = \sum_{j=1}^{m}[\Phi(k,s)]_{ij}x_j(s) - \alpha\sum_{r=s}^{k-1}\sum_{j=1}^{m}[\Phi(k,r+1)]_{ij}d_j(r) - \alpha\, d_i(k).$$

From this relation (with $s = 0$), we see that for all $k \geq 1$,

$$x_i(k) = \sum_{j=1}^{m}[\Phi(k-1,0)]_{ij}x_j(0) - \alpha\sum_{r=0}^{k-2}\sum_{j=1}^{m}[\Phi(k-1,r+1)]_{ij}d_j(r) - \alpha\, d_i(k-1). \tag{1.27}$$

[6] The iterates $y(k)$ can be associated with a stopped process related to algorithm (1.20), as discussed in [40].

By using the relations in Eqs. (1.26) and (1.27), we obtain for all $k \geq 1$,

$$\begin{aligned} x_i(k) - y(k) = & \sum_{j=1}^{m} \left([\Phi(k-1,0)]_{ij} - \frac{1}{m} \right) x_j(0) \\ & - \alpha \sum_{r=0}^{k-2} \sum_{j=1}^{m} \left([\Phi(k-1,r+1)]_{ij} - \frac{1}{m} \right) d_j(r) \\ & - \alpha\, d_i(k-1) + \frac{\alpha}{m} \sum_{i=1}^{m} d_i(k-1). \end{aligned}$$

Using the subgradient boundedness, we obtain for all $k \geq 1$,

$$\begin{aligned} \|x_i(k) - y(k)\| \leq & \sum_{j=1}^{m} \left| [\Phi(k-1,0)]_{ij} - \frac{1}{m} \right| \|x_j(0)\| \\ & + \alpha L \sum_{s=1}^{k-1} \sum_{j=1}^{m} \left| [\Phi(k-1,s)]_{ij} - \frac{1}{m} \right| + 2\alpha L. \end{aligned}$$

By using Theorem 1.2, we can bound the terms $\left|[\Phi(k-1,s)]_{ij} - \frac{1}{m}\right|$ and obtain for all i and any $k \geq 1$,

$$\begin{aligned} \|x_i(k) - y(k)\| & \leq \sum_{j=1}^{m} \beta^{\lceil \frac{k}{B} \rceil - 2} \|x_j(0)\| + \alpha L \sum_{s=1}^{k-1} \sum_{j=1}^{m} \beta^{\lceil \frac{k-s}{B} \rceil - 2} + 2\alpha L \\ & = \beta^{\lceil \frac{k}{B} \rceil - 2} \sum_{j=1}^{m} \|x_j(0)\| + \alpha L m \sum_{s=1}^{k-1} \beta^{\lceil \frac{k-s}{B} \rceil - 2} + 2\alpha L. \end{aligned}$$

By using $\sum_{s=1}^{k-1} \beta^{\lceil \frac{k-s}{B} \rceil - 2} \leq \sum_{r=1}^{\infty} \beta^{\lceil \frac{r}{B} \rceil - 2} = \frac{1}{\beta} \sum_{r=1}^{\infty} \beta^{\lceil \frac{r}{B} \rceil - 1}$, and

$$\sum_{r=1}^{\infty} \beta^{\lceil \frac{r}{B} \rceil - 1} = \sum_{r=1}^{\infty} \beta^{\lceil \frac{r}{B} \rceil - 1} \leq B \sum_{t=0}^{\infty} \beta^{t} = \frac{B}{1-\beta},$$

we obtain

$$\sum_{s=1}^{k-1} \beta^{\lceil \frac{k-s}{B} \rceil - 2} \leq \frac{B}{\beta(1-\beta)}.$$

Therefore, it follows that for all $k \geq 1$,

$$\|x_i(k) - y(k)\| \leq \beta^{\lceil \frac{k}{B} \rceil - 2} \sum_{j=1}^{m} \|x_j(0)\| + \alpha L \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

Q.E.D.

Estimate for the Auxiliary Sequence
We next establish a result that estimates the objective function $f = \sum_{i=1}^m f_i$ at the running averages of the vectors $y(k)$ of Eq. (1.25). Specifically, we define

$$\hat{y}(k) = \frac{1}{k} \sum_{h=1}^{k} y(h) \qquad \text{for all } k \geq 1,$$

and we estimate the function values $f(\hat{y}(k))$. We have the following result.

Lemma 1.6. *Let Assumptions 3 and 4 hold, and assume that the subgradients are uniformly bounded as in Eq. (1.24). Also, assume that the set X^* of optimal solutions of problem (1.19) is nonempty. Then, the average vectors $\hat{y}(k)$ satisfy for all $k \geq 1$,*

$$f(\hat{y}(k)) \leq f^* + \frac{\alpha L^2 C}{2} + \frac{2mLB}{k\beta(1-\beta)} \sum_{j=1}^{m} \|x_j(0)\| + \frac{m}{2\alpha k} \left(\text{dist}(y(0), X^*) + \alpha L\right)^2,$$

where $y(0) = \frac{1}{m} \sum_{j=1}^m x_j(0)$, $\beta = 1 - \frac{\eta}{4m^2}$ and

$$C = 1 + 4m \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

Proof. From the definition of the sequence $y(k)$ it follows for any $x^* \in X^*$ and all k,

$$\|y(k+1) - x^*\|^2 = \|y(k) - x^*\|^2 + \frac{\alpha^2}{m^2} \left\| \sum_{i=1}^{m} d_j(k) \right\|^2 - 2\frac{\alpha}{m} \sum_{i=1}^{m} d_i(k)'(y(k) - x^*). \tag{1.28}$$

We next estimate the terms $d_i(k)'(y(k) - x^*)$ where $d_i(k)$ is a subgradient of f_i at $x_i(k)$. For any i and k, we have

$$d_i(k)'(y(k) - x^*) = d_i(k)'(y(k) - x_i(k)) + d_i(k)'(x_i(k) - x^*).$$

By the subgradient property in (1.2), we have $d_i(k)(x_i(k) - x^*) \geq f_i(x_i(k)) - f_i(x^*)$ implying

$$\begin{aligned} d_i(k)'(y(k) - x^*) &\geq d_i(k)'(y(k) - x_i(k)) + f_i(x_i(k)) - f_i(x^*) \\ &\geq -L\|y(k) - x_i(k)\| + [f_i(x_i(k)) - f_i(y(k))] + [f_i(y(k)) - f_i(x^*)], \end{aligned}$$

where the last inequality follows from the subgradient boundedness. We next consider $f_i(x_i(k)) - f_i(y(k))$, for which by subgradient property (1.2) we have

$$f_i(x_i(k)) - f_i(y(k)) \geq \tilde{d}_i(k)'(x_i(k) - y(k)) \geq -L\|x_i(k) - y(k)\|,$$

where $\tilde{d}_i(k)$ is a subgradient of f_i at $y(k)$, and the last inequality follows from the subgradient boundedness. Thus, by combining the preceding two relations, we have for all i and k,

$$d_i(k)'(y(k) - x^*) \geq -2L\|y(k) - x_i(k)\| + f_i(y(k)) - f_i(x^*).$$

By substituting the preceding estimate for $d_i(k)'(y(k)-x^*)$ in relation (1.28), we obtain

$$\begin{aligned}\|y(k+1)-x^*\|^2 \leq \|y(k)-x^*\|^2 + \frac{\alpha^2}{m^2}\left\|\sum_{i=1}^{m} d_j(k)\right\|^2 + \frac{4L\alpha}{m}\sum_{i=1}^{m}\|y(k)-x_i(k)\| \\ -\frac{2\alpha}{m}\sum_{i=1}^{m} f_i(y(k)) - f_i(x^*).\end{aligned}$$

By using the subgradient boundedness and noting that $f=\sum_{i=1}^{m} f_i$ and $f(x^*) = f^*$, we can write

$$\begin{aligned}\|y(k+1)-x^*\|^2 \leq \|y(k)-x^*\|^2 + \frac{\alpha^2 L^2}{m} + \frac{4\alpha L}{m}\sum_{i=1}^{m}\|y(k)-x_i(k)\| \\ -\frac{2\alpha}{m}\left(f(y(k))-f^*\right).\end{aligned}$$

Taking the minimum over $x^*\in X^*$ in both sides of the preceding relation, we obtain

$$\begin{aligned}\mathrm{dist}^2(y(k+1),X^*) \leq \mathrm{dist}^2(y(k),X^*) + \frac{\alpha^2 L^2}{m} + \frac{4\alpha L}{m}\sum_{i=1}^{m}\|y(k)-x_i(k)\| \\ -\frac{2\alpha}{m}\left(f(y(k))-f^*\right).\end{aligned}$$

Using Lemma 1.5 to bound each of the terms $\|y(k)-x_i(k)\|$, we further obtain

$$\begin{aligned}\mathrm{dist}^2(y(k+1),X^*) \leq \mathrm{dist}^2(y(k),X^*) + \frac{\alpha^2 L^2}{m} + 4\alpha L\beta^{\lceil \frac{k}{B}\rceil -2}\sum_{j=1}^{m}\|x_j(0)\| \\ +4\alpha^2 L^2\left(2+\frac{mB}{\beta(1-\beta)}\right) - \frac{2\alpha}{m}\left[f(y(k))-f^*\right].\end{aligned}$$

Therefore, by regrouping the terms and introducing

$$C = 1+4m\left(2+\frac{mB}{\beta(1-\beta)}\right),$$

we have for all $k\geq 1$,

$$\begin{aligned}f(y(k)) \leq f^* + \frac{\alpha L^2 C}{2} + 2mL\beta^{\lceil \frac{k}{B}\rceil -2}\sum_{j=1}^{m}\|x_j(0)\| \\ +\frac{m}{2\alpha}\left(\mathrm{dist}^2(y(k),X^*) - \mathrm{dist}^2(y(k+1),X^*)\right).\end{aligned}$$

By adding these inequalities for different values of k, we obtain

$$\begin{aligned}\frac{1}{k}\sum_{h=1}^{k} f(y(h)) \leq f^* + \frac{\alpha L^2 C}{2} + \frac{2mLB}{k\beta(1-\beta)}\sum_{j=1}^{m}\|x_j(0)\| \\ +\frac{m}{2\alpha k}\left(\mathrm{dist}^2(y(1),X^*) - \mathrm{dist}^2(y(k),X^*)\right), \qquad (1.29)\end{aligned}$$

where we use the following inequality for $t \geq 1$,

$$\sum_{k=1}^{t} \beta^{\lceil \frac{k}{B} \rceil - 2} \leq \frac{1}{\beta} \sum_{k=1}^{\infty} \beta^{\lceil \frac{k}{B} \rceil - 1} \leq \frac{B}{\beta} \sum_{s=1}^{\infty} \beta^{s} = \frac{B}{\beta(1-\beta)}.$$

By discarding the nonpositive term on the right hand side in relation (1.29) and by using the convexity of f,

$$\frac{1}{k} \sum_{h=1}^{k} f(y(h)) \leq f^* + \frac{\alpha L^2 C}{2} + \frac{2mLB}{k\beta(1-\beta)} \sum_{j=1}^{m} \|x_j(0)\| + \frac{m}{2\alpha k} \operatorname{dist}^2(y(1), X^*).$$

Finally, by using the definition of $y(k)$ in (1.25) and the subgradient boundedness, we see that

$$\operatorname{dist}^2(y(1), X^*) \leq \left(\operatorname{dist}(y(0), X^*) + \alpha L\right)^2,$$

which when combined with the preceding relation yields the desired inequality. **Q.E.D.**

Performance Bound for the Algorithm

We establish a bound on the performance of the algorithm at the time-average of the vectors $x_i(k)$ generated by method (1.20). In particular, we define the vectors $\hat{x}_i(k)$ as follows:

$$\hat{x}_i(k) = \frac{1}{k} \sum_{h=1}^{k} x_i(h).$$

The use of these vectors allows us to bound the objective function improvement at every iteration, by combining the estimates for $\|x_i(k) - y(k)\|$ of Lemma 1.5 and the estimates for $f(\hat{y}(k))$ of Lemma 1.6. We have the following.

Theorem 1.3. *Let Assumptions 3 and 4 hold, and assume that the set X^* of optimal solutions of problem (1.19) is nonempty. Let the subgradients be bounded as in Eq. (1.24). Then, the averages $\hat{x}_i(k)$ of the iterates obtained by the method (1.20) satisfy for all i and $k \geq 1$,*

$$f(\hat{x}_i(k)) \leq f^* + \frac{\alpha L^2 C_1}{2} + \frac{4mLB}{k\beta(1-\beta)} \sum_{j=1}^{m} \|x_j(0)\| + \frac{m}{2\alpha k} \left(\operatorname{dist}(y(0), X^*) + \alpha L\right)^2,$$

where $y(0) = \frac{1}{m}\sum_{j=1}^{m} x_j(0)$, $\beta = 1 - \frac{\eta}{4m^2}$ and

$$C_1 = 1 + 8m\left(2 + \frac{mB}{\beta(1-\beta)}\right).$$

Proof. By the convexity of the functions f_j, we have, for any i and $k \geq 1$,

$$f(\hat{x}_i(k)) \leq f(\hat{y}(k)) + \sum_{j=1}^{m} g_{ij}(k)'(\hat{x}_i(k) - \hat{y}(k)),$$

where $g_{ij}(k)$ is a subgradient of f_j at $\hat{x}_i(k)$. Then, by using the subgradient boundedness, we obtain for all i and $k \geq 1$,

$$f(\hat{x}_i(k)) \leq f(\hat{y}(k)) + \frac{2L}{k} \sum_{i=1}^{m} \left(\sum_{t=1}^{k} \|x_i(t) - y(t)\| \right). \tag{1.30}$$

By using the bound for $\|x_i(k) - y(k)\|$ of Lemma 1.5, we have for all i and $k \geq 1$,

$$\sum_{t=1}^{k} \|x_i(t) - \hat{y}(t)\| \leq \left(\sum_{t=1}^{k} \beta^{\lceil \frac{t}{B} \rceil - 2} \right) \sum_{j=1}^{m} \|x_j(0)\| + \alpha k L \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

Noting that

$$\sum_{t=1}^{k} \beta^{\lceil \frac{t}{B} \rceil - 2} \leq \frac{1}{\beta} \sum_{t=1}^{\infty} \beta^{\lceil \frac{t}{B} \rceil - 1} \leq \frac{B}{\beta} \sum_{s=1}^{\infty} \beta^{s} = \frac{B}{\beta(1-\beta)},$$

we obtain for all i and $k \geq 1$,

$$\sum_{t=1}^{k} \|x_i(t) - \hat{y}(t)\| \leq \frac{B}{\beta(1-\beta)} \sum_{j=1}^{m} \|x_j(0)\| + \alpha k L \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

Hence, by summing these inequalities over all i and by substituting the resulting estimate in relation (1.30), we obtain

$$f(\hat{x}_i(k)) \leq f(\hat{y}(k)) + \frac{2mLB}{k\beta(1-\beta)} \sum_{j=1}^{m} \|x_j(0)\| + 2m\alpha L^2 \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

The result follows by using the estimate for $f(\hat{y}(k))$ of Lemma 1.6. **Q.E.D.**

The result of Theorem 1.3 provides an estimate on the values $f(\hat{x}_i(k))$ per iteration k. As the number of iterations increases to infinity the last two terms of the estimate diminish, resulting with

$$\limsup_{k \to \infty} f(\hat{x}_i(k)) \leq f^* + \frac{\alpha L^2 C_1}{2} \qquad \text{for all } i.$$

As seen from Theorem 1.3, the constant C_1 increases only polynomially with m. When α is fixed and the parameter η is independent of m, the largest error is of the order of m^4, indicating that for high accuracy, the stepsize needs to be very small. However, our bound is for general convex functions and network topologies, and further improvements of the bound are possible for special classes of convex functions and special topologies.

1.4 Extensions

Here, we consider extensions of the distributed model of Section 1.3 to account for various network effects. We focus on two such extensions. The *first* is an extension of the optimization model (1.20) to a scenario where the agents communicate over a network with finite bandwidth communication links, and the *second* is an extension of the consensus problem to the scenario where each agent is facing some constraints on its decisions. We discuss these extensions in the following sections.

1.4.1 Quantization Effects on Optimization

Here, we discuss the agent system where the agents communicate over a network consisting of finite bandwidth links. Thus, the agents cannot exchange continuous-valued information (real numbers), but instead can only send quantized information. This problem has recently gained interest in the networking literature [22, 12, 13, 31]. In what follows, we present recent results dealing with the effects of quantization on the multi-agent distributed optimization over a network. More specifically, we discuss a "quantized" extension of the subgradient method (1.20) and provide a performance bound.[7]

We consider the case where the agents exchange quantized data, but they can store continuous data. In particular, we assume that each agent receives and sends only quantized estimates, i.e., vectors whose entries are integer multiples of $1/Q$, where Q is some positive integer. At time k, an agent receives quantized estimates $x_j^Q(k)$ from some of its neighbors and updates according to the following rule:

$$x_i^Q(k+1) = \left\lfloor \sum_{j=1}^{m} a_{ij}(k) x_j^Q(k) - \alpha \tilde{d}_i(k) \right\rfloor, \tag{1.31}$$

where $\tilde{d}_i(k)$ is a subgradient of f_i at $x_i^Q(k)$, and $\lfloor y \rfloor$ denotes the operation of (componentwise) rounding the entries of a vector y to the nearest multiple of $1/Q$. We also assume that the agents' initial estimates $x_j^Q(0)$ are quantized.

We can view the agent estimates in Eq. (1.31) as consisting of a consensus part $\sum_{j=1}^m a_{ij}(k) x_j^Q(k)$, and the term due to the subgradient step and an error (due to extracting the consensus part). Specifically, we rewrite Eq. (1.31) as follows:

$$x_i^Q(k+1) = \sum_{j=1}^{m} a_{ij}(k) x_j^Q(k) - \alpha \tilde{d}_i(k) - \epsilon_i(k+1), \tag{1.32}$$

[7] The result presented here can be found with more details in [32].

where the error vector $\epsilon_i(k+1)$ is given by

$$\epsilon_i(k+1) = \sum_{j=1}^{m} a_{ij}(k) x_j^Q(k) - \alpha \tilde{d}_i(k) - x_i^Q(k+1).$$

Thus, the method can be viewed as a subgradient method using consensus and with external (possibly persistent) noise, represented by $\epsilon_i(k+1)$. Due to the rounding down to the nearest multiple of $1/Q$, the error vector $\epsilon_i(k+1)$ satisfies

$$0 \leq \epsilon_i(k+1) \leq \frac{1}{Q}\mathbf{1} \qquad \text{for all } i \text{ and } k,$$

where the inequalities above hold componentwise and $\mathbf{1}$ denotes the vector in $\mathbb{R}^n$ with all entries equal to 1. Therefore, the error norms $\|\epsilon_i(k)\|$ are uniformly bounded in time and across agents. In fact, it turns out that these errors converge to 0 as k increases. These observations are guiding the analysis of the algorithm.

Performance Bound for the Quantized Method

We next give a performance bound for the method (1.31) assuming that the agents can store perfect information (infinitely many bits). We consider the time-average of the iterates $x_i^Q(k)$, defined by

$$\hat{x}_i^Q(k) = \frac{1}{k} \sum_{h=1}^{k} x_i^Q(h) \qquad \text{for } k \geq 1.$$

We have the following result (see [32] for the proof).

Theorem 1.4. *Let Assumptions 3 and 4 hold, and assume that the optimal set X^* of problem (1.19) is nonempty. Let subgradients be bounded as in Eq. (1.24). Then, for the averages $\hat{x}_i^Q(k)$ of the iterates obtained by the method (1.31) satisfy for all i and all $k \geq 1$,*

$$f(\hat{x}_i^Q(k)) \leq f^* + \frac{\alpha L^2 \tilde{C}_1}{2} + \frac{4mLB}{k\beta(1-\beta)} \sum_{j=1}^{m} \|x_j^Q(0)\| + \frac{m}{2\alpha k}\left(\mathrm{dist}(\tilde{y}(0), X^*) + \alpha L + \frac{\sqrt{n}}{Q}\right)^2,$$

where $\tilde{y}(0) = \frac{1}{m}\sum_{j=1}^{m} x_j^Q(0)$, $\beta = 1 - \frac{\eta}{4m^2}$ and

$$\tilde{C}_1 = 1 + 8m\left(1 + \frac{\sqrt{n}}{\alpha LQ}\right)\left(2 + \frac{mB}{\beta(1-\beta)}\right).$$

Theorem 1.4 provides an estimate on the values $f(\hat{x}_i^Q(k))$ per iteration k. As the number of iterations increases to infinity the last two terms of the estimate vanish, yielding

$$\limsup_{k\to\infty} f(\hat{x}_i^Q(k)) \leq f^* + \frac{\alpha L^2 \tilde{C}_1}{2} \qquad \text{for all } i,$$

with

$$\tilde{C}_1 = 1 + 8m\left(1 + \frac{\sqrt{n}}{\alpha L Q}\right)\left(2 + \frac{mB}{\beta(1-\beta)}\right).$$

The constant $\tilde{C}_1$ increases only polynomially with m. In fact, the growth with m is the same as that of the bound given in Theorem 1.3, since the result in Theorem 1.3 follows from Theorem 1.4. In particular, by letting the quantization level Q be increasingly finer (i.e., $Q \to \infty$), we see that the constant $\tilde{C}_1$ satisfies

$$\lim_{Q\to\infty} \tilde{C}_1 = 1 + 8m\left(2 + \frac{mB}{\beta(1-\beta)}\right),$$

which is the same as the constant C_1 in Theorem 1.3. Hence, in the limit as $Q \to \infty$, the estimate in Theorem 1.4 yields the estimate in Theorem 1.3.

1.4.2 Consensus with Local Constraints

Here, we focus only on the problem of reaching a consensus when the estimates of different agents are constrained to lie in different constraint sets and each agent only knows its own constraint set. Such constraints are significant in a number of applications including signal processing within a network of sensors, network motion planning and alignment, rendezvous problems and distributed constrained multi-agent optimization problems[8], where each agent's position is limited to a certain region or range.

As in the preceding, we denote by $x_i(k)$ the estimate generated and stored by agent i at time slot k. The agent estimate $x_i(k) \in \mathbb{R}^n$ is constrained to lie in a nonempty closed convex set $X_i \subseteq \mathbb{R}^n$ known only to agent i. The agents' objective is to cooperatively *reach a consensus on a common vector through a sequence of local estimate updates (subject to the local constraint set) and local information exchanges (with neighboring agents only).*

To generate the estimate at time $k+1$, agent i forms a convex combination of its estimate $x_i(k)$ with the estimates received from other agents at time k, and takes the projection of this vector on its constraint set X_i. More specifically, agent i at time $k+1$ generates its new estimate according to the following relation:

$$x_i(k+1) = P_{X_i}\left[\sum_{j=1}^{m} a_{ij}(k)x_j(k)\right]. \tag{1.33}$$

Through the rest of the discussion, the constraint sets $X_1, \ldots, X_m$ are assumed to be *closed convex* subsets of $\mathbb{R}^n$.

The relation in (1.33) defines the *projected consensus algorithm* . The method can be viewed as a distributed algorithm for finding a point in common to the closed convex sets $X_1, \ldots, X_m$. This problem can be formulated as an uncon-

[8] See also [28] for constrained consensus arising in connection with potential games.

strained convex minimization, as follows

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sum_{i=1}^{m}\|x-P_{X_i}[x]\|^2 \\ \text{subject to} & x\in\mathbb{R}^n. \end{array} \tag{1.34}$$

In view of this optimization problem, the method in (1.33) can be interpreted as a distributed algorithm where an agent i is assigned an objective function $f_i(x) = \frac{1}{2}\|x-P_{X_i}[x]\|^2$. Each agent updates its estimate by taking a step (with steplength equal to 1) along the negative gradient of its own objective function $f_i = \frac{1}{2}\|x-P_{X_i}\|^2$ at $x=\sum_{j=1}^{m}a_{ij}(k)x_j(k)$. This interpretation of the update rule motivates our line of analysis of the projected consensus method. In particular, we use $\sum_{i=1}^{m}\|x_i(k)-x\|^2$ with $x\in\cap_{i=1}^{m}X_i$ as a function measuring the progress of the algorithm.

Let us note that the method of Eq. (1.33), with the right choice of the weights $a_{ij}(k)$, corresponds to the classical *alternating or cyclic projection method.* These methods generate a sequence of vectors by projecting iteratively on the sets (either cyclically or with some given order), see Figure 1.8(a) . The convergence behavior of these methods has been established by Von Neumann [43] and Aronszajn [1], Gubin *et al.* [17], Deutsch [15], and Deutsch and Hundal [16]. The projected consensus algorithm can be viewed as a version of the alternating projection algorithm, where the iterates are combined with the weights varying over time and across agents, and then projected on the individual constraint sets.

To study the convergence behavior of the agent estimates $\{x_i(k)\}$ defined in Eq. (1.33), we find it useful to decompose the representation of the estimates in a linear part (corresponding to nonprojected consensus) and a nonlinear part (corresponding to the difference between the projected and nonprojected consensus). Specifically, we re-write the update rule in (1.33) as

$$x_i(k+1)=\sum_{j=1}^{m}a_{ij}(k)x_j(k)+e^i(k), \tag{1.35}$$

where $e^i(k)$ represents the error due to the projection operation, given by

$$e^i(k)=P_{X_i}\left[\sum_{j=1}^{m}a_{ij}(k)x_j(k)\right]-\sum_{j=1}^{m}a_{ij}(k)x_j(k). \tag{1.36}$$

As indicated by the preceding two relations, the evolution dynamics of the estimates $x_i(k)$ for each agent is decomposed into a sum of a linear (time-varying) term $\sum_{j=1}^{m}a_{ij}(k)x_j(k)$ and a nonlinear term $e^i(k)$. The linear term captures the effects of mixing the agent estimates, while the nonlinear term captures the nonlinear effects of the projection operation. This decomposition can be exploited to analyze the behavior and estimate the performance of the algorithm. It can be seen [41] that, under the doubly stochasticity assumption on the weights, the nonlinear terms $e^i(k)$ are diminishing in time for each i, and therefore, the evolution of agent estimates is "almost linear". Thus, the nonlinear term can be viewed as a non-persistent disturbance in the linear evolution of the estimates.

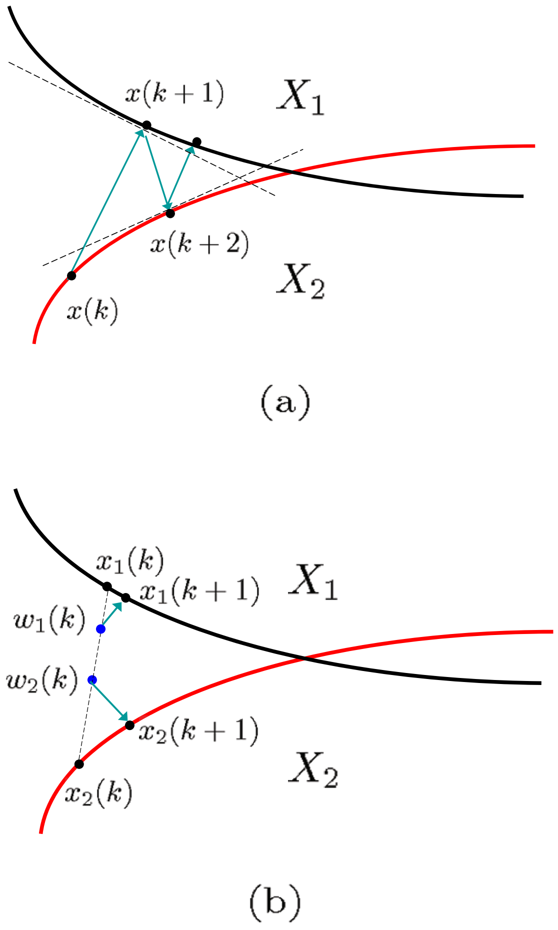


Figure 1.8 The connection between the alternating/cyclic projection method and the projected consensus algorithm for two closed convex sets X_1 and X_2. In plot (a), the alternating projection algorithm generates a sequence $\{x(k)\}$ by iteratively projecting onto sets X_1 and X_2, i.e., $x(k+1) = P_{X_1}[x(k)]$, $x(k+2) = P_{X_2}[x(k+1)]$. In plot (b), the projected consensus algorithm generates sequences $\{x_i(k)\}$ for agents $i = 1, 2$ by first combining the iterates with different weights and then projecting on respective sets X_i, i.e., $w_i(k) = \sum_{j=1}^{m} a_{ij}(k)x_j(k)$ and $x_i(k+1) = P_{X_i}[w_i(k)]$ for $i = 1, 2$.

Convergence and Rate of Convergence Results

We show that the projected consensus algorithm converges to some vector that is common to all constraint sets X_i, under Assumptions 3 and 4. Under an additional assumption that the sets X_i have an interior point in common, we provide a convergence rate estimate.

The following result shows that the agents reach a consensus asymptotically, i.e., the agent estimates $x_i(k)$ converge to the same point as k goes to infinity (see [41]).

Theorem 1.5. *Let the constraint sets $X_1, \ldots, X_m$ be closed convex subsets of $\mathbb{R}^n$, and let the set $X = \cap_{i=1}^m X_i$ be nonempty. Also, let Assumptions 3 and 4 hold. Let the sequences $\{x_i(k)\}$, $i = 1 \ldots, m$, be generated by the projected consensus algorithm (1.33). We then have for some $\tilde{x} \in X$,*

$$\lim_{k\to\infty} \|x_i(k) - \tilde{x}\| = 0 \qquad \textit{for all } i.$$

We next provide a rate estimate for the projected consensus algorithm (1.33). It is difficult to access the convergence rate in the absence of any specific structure on the constraint sets X_i. To deal with this, we consider a special case when the weights are time-invariant and equal, i.e., $a_{ij}(k) = 1/m$ for all i, j and k, and the intersection of the sets X_i has a nonempty interior. In particular, we have the following rate result (see [41]).

Theorem 1.6. *Let the constraint sets $X_1, \ldots, X_m$ be closed convex subsets of $\mathbb{R}^n$. Let $X = \cap_{i=1}^m X_i$, and assume that there is a vector $\bar{x} \in X$ and a scalar $\delta > 0$ such that*

$$\{z \mid \|z - \bar{x}\| \le \delta\} \subset X.$$

Also, let Assumption 4 hold. Let the sequences $\{x_i(k)\}$, $i = 1 \ldots, m$ be generated by the algorithm (1.33), where the weights are uniform, i.e., $a_{ij}(k) = 1/m$ for all i, j and k. We then have

$$\sum_{i=1}^m \|x_i(k) - \tilde{x}\|^2 \le \left(1 - \frac{1}{4R^2}\right)^k \sum_{i=1}^m \|x_i(0) - \tilde{x}\|^2 \qquad \textit{for all } k \ge 0,$$

where $\tilde{x} \in X$ is the common limit of the sequences $\{x_i(k)\}$, $i = 1 \ldots, m$, and $R = \frac{1}{\delta} \sum_{i=1}^m \|x_i(0) - \bar{x}\|$.

The result shows that the projected consensus algorithm converges with a geometric rate under the interior point and uniform weights assumptions.

1.5 Future Work

The models presented so far highlight a number of fruitful areas for future research. These include but are not limited to the following topics.

1.5.1 Optimization with Delays

The distributed subgradient algorithm we presented in Section 1.3 [cf. Eq. (1.20)] assumes that at any time $k \geq 0$, agent i has access to estimates $x_j(k)$ of its neighbors. This may not be possible in communication networks where there are delays associated with transmission of agent estimates over a communication channel. A natural extension therefore is to study an asynchronous operation of the algorithm (1.20) using delayed agent values, i.e., agent i at time k has access to outdated values of agent j.

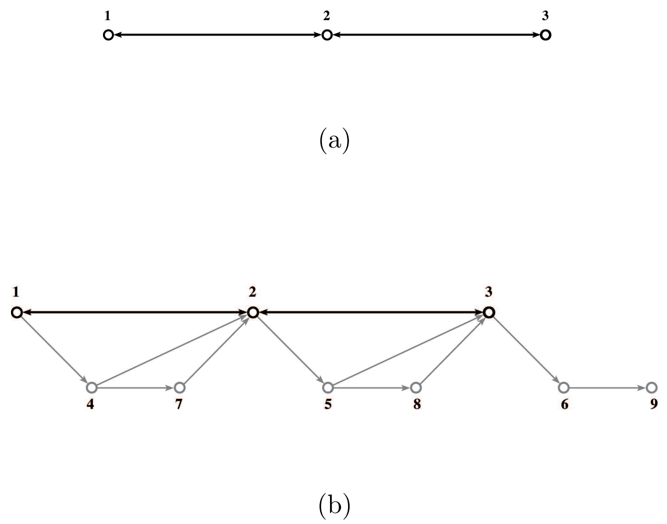
More formally, we consider the following update rule for agent i: Suppose agent j sends its estimate $x_j(s)$ to agent i. If agent i receives the estimate $x_j(s)$ at time k, then the delay is $t_{ij}(k) = k - s$ and agent i assigns a weight $a_{ij}(k) > 0$ to the estimate $x_j(s)$. Otherwise, agent i uses $a_{ij}(k) = 0$. Hence, each agent i updates its estimate according to the following relation:

$$x_i(k+1) = \sum_{j=1}^{m} a_{ij}(k) x_j(k - t_{ij}(k)) - \alpha d_i(k) \qquad \text{for } k = 0, 1, 2, \ldots, \tag{1.37}$$

where the vector $x_i(0) \in \mathbb{R}^n$ is the initial estimate of agent i, the scalar $t_{ij}(k)$ is nonnegative and it represents the delay of a message from agent j to agent i, while the scalar $a_{ij}(k)$ is a nonnegative weight that agent i assigns to a delayed estimate $x_j(s)$ arriving from agent j at time k.

Establishing the convergence and rate properties of the update rule (1.37) is essential in understanding the robustness of the optimization algorithm to delays and dynamics associated with information exchange over finite bandwidth communication channels. Under the assumption that all delay values are bounded [i.e., there exists a scalar $B > 0$ such that $t_{ij}(k) \leq B$ for all $i, j \in \mathcal{N}$ and all $k \geq 0$], the update rule (1.37) can be analyzed by considering an *augmented model*, where we introduce "artificial" agents for handling the delayed information only. In particular, with each agent i of the original model, we associate a new agent for each of the possible values of the delay that a message originating from agent i may experience. In view of the bounded delay values assumption, it suffices to add finitely many new agents handling the delays. This augmentation reduces the delayed multi-agent model into a model without delays (see Figure 1.5.1).

The augmented agent model is used in [35] to study the consensus problem in the presence of delays. In particular, this paper shows that agents reach consensus on a common decision even with delayed information and establishes convergence rate results. Future work includes analyzing the optimization algorithm of (1.20) in the presence of delays. The analysis of the optimization algorithm is more challenging in view of the fact that due to delays, an agent may receive different amount of information from different agents. This difference in the update frequencies results in the overall consensus value to be influenced more by some of the agents' information (about their local objective function). Thus, the value that agents reach a consensus on need not be the optimal solution of the problem of minimizing the sum of the local objective functions of the agents. To address



(a)

(b)

Figure 1.9 Plot (a) illustrates an agent network with 3 agents, where agents 1 and 2, and agents 2 and 3 communicate directly. Plot (b) illustrates the augmented network associated with the original network of part (a), when the delay value between agents is bounded by 3. The artificial agents introduced in the system are $4, \ldots, 9$. Agents 4, 5, and 6 model the delay of 1 while agents 7, 8, and 9 model the delay of 2 for the original nodes 1, 2 and 3, respectively.

this issue, the optimization algorithm should be modified to include the update frequency information in the agent exchange model.

1.5.2 Optimization with Constraints

Section 1.3 presents a distributed subgradient method for solving the unconstrained optimization problem (1.19). An important extension is to develop optimization methods for solving multi-agent optimization problems in which each agent i has a local convex closed constraint set $X_i \subseteq \mathbb{R}^n$ know by agent i only. Note that the case when there is a global constraint $C_g \subseteq \mathbb{R}^n$ is a special case of this problem with $X_i = C_g$ for all $i \in \mathcal{N}$ (see Introduction).

An immediate solution for this problem is to combine the subgradient algorithm (1.20) with the projected consensus algorithm studied in Section 1.4.2. More specifically, we denote by $x_i(k)$ the estimate maintained by agent i at time slot k. Agent i updates this estimate by forming a convex combination of this estimate with the estimates received from his neighbors at time k, taking a step (with stepsize α) in the direction of the subgradient of his local convex objective function f_i at $x_i(k)$, and taking the projection of this vector on its constraint set X_i, i.e., agent i at time k generates its new estimate according to

$$x_i(k+1) = P_{X_i}\left[\sum_{j=1}^{m} a_{ij}(k)x_j(k) - \alpha d_i(k)\right] \qquad \text{for } k = 0, 1, 2, \ldots. \tag{1.38}$$

The convergence analysis of this algorithm involves combining the ideas and methods of Sections 1.3 and 1.4.2, i.e., understanding the behavior of the transition matrices, the approximate subgradient method, and the projection errors. It is more challenging due to the dependencies of the error terms involved in the analysis.

When the global constraint set C_g has more structure, e.g., when it can be expressed as finitely many equality and inequality constraints, it may be possible to develop *primal-dual algorithms*, which combine the primal step (1.20) with a dual step for updating the dual solutions (or multipliers), as in Section 1.2. Primal-dual subgradient methods have been analyzed in recent work [38] for a model in which each agent has the same information set, i.e., at each time slot, each agent has access to the same estimate. It would be of great interest to combine this model with the multi-agent model of Section 1.3 that incorporates different local information structures.

1.5.3 Nonconvex Local Objective Functions

The distributed optimization framework presented in Section 1.3 is very general in that it encompasses local information structures, operates with time-varying connectivity, and optimizes general *convex local objective functions subject to convex constraints*. However, there are many applications such as inelastic rate control for voice communication (see [49, 18]) and rendezvous problems with constraints (see [28]) in which the local objective functions and constraints are not convex. Under smoothness assumptions on the objective functions, the methods presented in this chapter can still be used and guarantee convergence to stationary points. An important future direction is the design of algorithms that can guarantee convergence to global optimal in the presence of nonconvexities and in decentralized environments.

1.6 Conclusions

This chapter presents a general framework for distributed optimization of a multi-agent networked system. In particular, we consider multiple agents, each with its own private local objective function and local constraint, which exchange information over a network with time-varying connectivity. The goal is to design algorithms that the agents can use to cooperatively optimize a global objective function, which is a function of the local objective functions, subject to local and global constraints. A key characteristic of these algorithms is their operation within the informational constraints of the model, specified by the local information structures and the underlying connectivity of the agents.

Our development focuses on two key approaches. The first approach uses Lagrangian duality and dual subgradient methods to design algorithms. These algorithms yield distributed methods for problems with separable structure (i.e., problems where local objective functions and constraints decompose over the components of the decision vector), as illustrated in Section 1.2.3. Since these methods operate in the dual space, a particular interest is in producing primal near-feasible and near-optimal solutions using the information generated by the dual subgradient algorithm. The analysis presented in Section 1.2 is from our recent work [34, 39], which discusses approximate primal solution recovery and provides convergence rate estimates on the generated solutions.

The second approach combines subgradient methods with consensus algorithms to optimize general convex local objective functions in decentralized settings. Even though the nonseparable structure of the local objective functions does not immediately lead to decomposition schemes, the consensus part included in the algorithm serves as a mechanism to distribute the computations among the agents. The material presented in Sections 1.3 and 1.4 combines results from a series of recent papers (see [40, 33, 35, 41, 31, 32]).

This chapter illustrates the challenges associated with optimization algorithm design for multi-agent networked systems. Optimization methodologies have played a key role in providing a systematic framework for the design of architectures and new protocols in many different networks. For such applications, it is clear that many of the assumptions we take for granted in the development and analysis of algorithms for solving constrained optimization problems are not valid. These include assumptions such as global access to input data and ability to exchange real-valued variables instantly between different processors. Moreover, practical considerations divert our attention from complicated stepsize rules that can guarantee convergence to an optimal solution to simple stepsize rules (such as a constant stepsize rule), which may not guarantee convergence, but nevertheless can provide an "approximate solution" within reasonable time constraints. The importance of optimization methods for such practical applications motivate development of new frameworks and methods that can operate within the constraints imposed by the underlying networked system.

1.7 Problems

Exercise 1. *Let $\{\mu_k\}$ be a dual sequence generated by the subgradient method with a constant stepsize α [cf. Eq. (1.8)]. Assume that the subgradients $\{g_k\}$ are uniformly bounded, i.e., $\|g_k\| \leq L$ for all k. Assume also that the dual optimal solution set M^* is nonempty.*

At each iteration k, consider an approximate dual solution generated by averaging the vectors $\mu_0, \dots, \mu_{k-1}$, i.e.,

$$\hat{\mu}_k = \frac{1}{k} \sum_{i=0}^{k-1} \mu_i \qquad \text{for all } k \geq 1.$$

Show that

$$q(\hat{\mu}_k) \geq q^* - \frac{\mathrm{dist}^2(\mu_0, M^*)}{2\alpha k} - \frac{\alpha L^2}{2} \qquad \text{for all } k \geq 1.$$

Exercise 2. *(Minimum Cost Network Flow Problem)*

Consider a directed connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the node set and $\mathcal{E}$ is the edge set. At each node i, there is a given external source flow b_i that enters (if $b_i > 0$) or leaves (if $b_i < 0$) node i. Let b be the vector $b = [b_i]_{i \in \mathcal{N}}$. We define the node-edge incidence matrix A as the $|\mathcal{N}| \times |\mathcal{E}|$ matrix given as follows: the $(i,j)^{th}$ entry $[A]_{ij}$ is given by $+1$ if edge j leaves node i; by -1 if edge j enters node i; and 0 otherwise.

Each edge has a convex cost function $f_i(x_i)$, where x_i denotes the flow on edge i. We are interested in finding a flow vector $x = [x_i]_{i \in \mathcal{E}}$ that minimizes the sum of the edge cost functions. This problem can be formulated as an optimization problem as follows:

$$\begin{aligned} \textit{minimize} \quad & \sum_{i \in \mathcal{E}} f_i(x_i) \\ \textit{subject to} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{1.39}$$

where the constraint $Ax = b$ captures the conservation of flow constraints.

Use Lagrangian decomposition and the dual subgradient algorithm to solve this problem. Show that this approach leads to a distributed optimization method for solving problem (1.39).

Exercises 3–5 are the steps involved in proving the result of Theorem 1.4.

Exercise 3. *Consider the quantized method given in (1.31) of Section 1.4.1. Define the transition matrices $\Phi(k,s)$ from time s to time k, as follows*

$$\Phi(k,s) = A(k)A(k-1)\cdots A(s) \qquad \text{for all } s \text{ and } k \text{ with } k \geq s,$$

(see Section 1.3.1).

(a) Using the transition matrices and the decomposition of the estimate evolution of Eqs. (1.35)–(1.36), show that the relation between $x_i(k+1)$ and the estimates $x_1(0), \ldots, x_m(0)$ is given by

$$\begin{aligned} x_i^Q(k+1) = {} & \sum_{j=1}^{m} [\Phi(k,0)]_{ij} x_j^Q(0) - \alpha \sum_{s=1}^{k} \sum_{j=1}^{m} [\Phi(k,s)]_{ij} \tilde{d}_j(s-1) \\ & - \sum_{s=1}^{k} \sum_{j=1}^{m} [\Phi(k,s)]_{ij} \epsilon_j(s) - \alpha \tilde{d}_i(k) - \epsilon_i(k+1). \end{aligned}$$

(b) Consider an auxiliary sequence $\{y(k)\}$ defined by

$$y(k) = \frac{1}{m} \sum_{j=1}^{m} x_i^Q(k).$$

Show that for all k,

$$y(k) = \frac{1}{m} \sum_{j=1}^{m} x_j^Q(0) - \frac{\alpha}{m} \sum_{s=1}^{k} \sum_{j=1}^{m} \tilde{d}_j(s-1) - \frac{1}{m} \sum_{s=1}^{k} \sum_{j=1}^{m} \epsilon_j(s).$$

(c) Suppose that Assumptions 3 and 4 hold. Using the relations in parts (a) and (b), and Theorem 1.2, show that

$$\|x_i^Q(k) - y(k)\| \leq \beta^{\lceil \frac{k}{B} \rceil - 2} \sum_{j=1}^{m} \|x_j^Q(0)\| + \left(\alpha L + \frac{\sqrt{n}}{Q} \right) \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

Exercise 4. *Let $y(k)$ be a sequence defined in Exercise 3. Consider the running averages $\hat{y}(k)$ of the vectors $y(k)$, given by*

$$\hat{y}(k) = \frac{1}{k} \sum_{h=1}^{k} y(h) \qquad \text{for all } k \geq 1.$$

Let Assumptions 3 and 4 hold, and assume that the set X^* of optimal solutions of problem (1.19) is nonempty. Also, assume that the subgradients are uniformly bounded as in Eq. (1.24). Then, the average vectors $\hat{y}(k)$ satisfy for all $k \geq 1$,

$$f(\hat{y}(k)) \leq f^* + \frac{\alpha L^2 \tilde{C}}{2} + \frac{2mLB}{k\beta(1-\beta)} \sum_{j=1}^{m} \|x_j^Q(0)\|$$
$$+ \frac{m}{2\alpha k} \left(\mathrm{dist}(y(0), X^*) + \alpha L + \frac{\sqrt{n}}{Q} \right)^2,$$

where $y(0) = \frac{1}{m} \sum_{j=1}^{m} x_j^Q(0)$, $\beta = 1 - \frac{\eta}{4m^2}$, and

$$\tilde{C} = 1 + 4m \left(1 + \frac{\sqrt{n}}{\alpha L Q} \right) \left(2 + \frac{mB}{\beta(1-\beta)} \right).$$

Exercise 5. *Prove Theorem 1.4 using the results of Exercises 3 and 4.*

References

[1] N. Aronszajn, *Theory of reproducing kernels*, Transactions of the American Mathematical Society **68** (1950), no. 3, 337–404.

[2] D. P. Bertsekas, A. Nedić, and A. Ozdaglar, *Min common/max crossing duality: A simple geometric framework for convex optimization and minimax theory*, Tech. Report LIDS 2536, Massachusetts Institute of Technology, 2002.

[3] D.P. Bertsekas, *Nonlinear programming*, Athena Scientific, Belmont, Massachusetts, 1999.

[4] D.P. Bertsekas, A. Nedić, and A.E. Ozdaglar, *Convex analysis and optimization*, Athena Scientific, Belmont, Massachusetts, 2003.

[5] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*, Athena Scientific, Belmont, MA, 1997.

[6] P.-A. Bliman and G. Ferrari-Trecate, *Average consensus problems in networks of agents with delayed communications*, Automatica **44** (2008), no. 8, 1985–1995.

[7] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis, *Convergence in multi-agent coordination, consensus, and flocking*, Proceedings of IEEE CDC, 2005, pp. 2996–3000.

[8] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, *Gossip algorithms: Design, analysis, and applications*, Proceedings of IEEE INFOCOM, vol. 3, 2005, pp. 1653–1664.

[9] M. Cao, A.S. Morse, and B.D.O. Anderson, *Reaching a consensus in a dynamically changing environment: A graphical approach*, SIAM J. on Control and Opt. **47** (2008), no. 2, 575–600.

[10] ———, *Reaching a consensus in a dynamically changing environment: Convergence rates, measurement delays, and asynchronous events*, SIAM J. on Control and Opt. **47** (2008), no. 2, 601–623.

[11] M. Cao, D.A. Spielman, and A.S. Morse, *A lower bound on convergence of a distributed network consensus algorithm*, Proceedings of IEEE CDC, 2005, pp. 2356–2361.

[12] R. Carli, F. Fagnani, P. Frasca, T. Taylor, and S. Zampieri, *Average consensus on networks with transmission noise or quantization*, Proceedings of European Control Conference, 2007.

[13] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, *Communication constraints in coordinated consensus problem*, Proceedings of IEEE American Control Conference, 2006, pp. 4189–4194.

[14] M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle, *Layering as optimization decomposition: A mathematical theory of network architectures*, Proceedings of the IEEE **95** (2007), no. 1, 255–312.

[15] F. Deutsch, *Rate of convergence of the method of alternating projections*, Parametric Optimization and Approximation (B. Brosowski and F. Deutsch, eds.), vol. 76, Birkhuser, Basel, 1983, pp. 96–107.

[16] F. Deutsch and H. Hundal, *The rate of convergence for the cyclic projections algorithm i: Angles between convex sets*, Journal of Approximation Theory **142** (2006), 36–55.

[17] L.G. Gubin, B.T. Polyak, and E.V. Raik, *The method of projections for finding the common point of convex sets*, U.S.S.R Computational Mathematics and Mathematical Physics **7** (1967), no. 6, 1211–1228.

[18] P. Hande, S. Zhang, and M. Chiang, *Distributed rate allocation for inelastic flows*, IEEE/ACM Transactions on Networking **15** (2007), no. 6, 1240–1253.

[19] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms*, Springer-Verlag, Berlin, 1996.

[20] A. Jadbabaie, J. Lin, and S. Morse, *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Transactions on Automatic Control **48** (2003), no. 6, 988–1001.

[21] S. Kar and J. Moura, *Distributed consensus algorithms in sensor networks: Link and channel noise*, available at http://arxiv.org/abs/0711.3915, 2007.

[22] A. Kashyap, T. Basar, and R. Srikant, *Quantized consensus*, Automatica **43** (2007), no. 7, 1192–1203.

[23] F.P. Kelly, A.K. Maulloo, and D.K. Tan, *Rate control for communication networks: shadow prices, proportional fairness, and stability*, Journal of the Operational Research Society **49** (1998), 237–252.

[24] T. Larsson, M. Patriksson, and A. Strömberg, *Ergodic results and bounds on the optimal value in subgradient optimization*, Operations Research Proceedings (P. Kelinschmidt et al., ed.), Springer, 1995, pp. 30–35.

[25] T. Larsson, M. Patriksson, and A. Strömberg, *Ergodic convergence in subgradient optimization*, Optimization Methods and Software **9** (1998), 93–120.

[26] ———, *Ergodic primal convergence in dual subgradient schemes for convex programming*, Mathematical Programming **86** (1999), 283–312.

[27] S. Low and D.E. Lapsley, *Optimization flow control, I: Basic algorithm and convergence*, IEEE/ACM Transactions on Networking **7** (1999), no. 6, 861–874.

[28] J.R. Marden, G. Arslan, and J.S. Shamma, *Connections between cooperative control and potential games illustrated on the consensus problem*, Preprint, 2008.

[29] L. Moreau, *Stability of multiagent systems with time-dependent communication links*, IEEE Transactions on Automatic Control **50** (2005), no. 2, 169–182.

[30] A. Mutapcic, S. Boyd, S. Murali, D. Atienza, G. De Micheli, and R. Gupta, *Processor speed control with thermal constraints*, submitted for publication, 2007.

[31] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, *On distributed averaging algorithms and quantization effects*, IEEE Transactions on Automatic Control, forthcoming, available at http://arxiv.org/abs/0803.1202 (2009).

[32] A. Nedić, A. Olshevsky, A. Ozdaglar, and J.N. Tsitsiklis, *Distributed subgradient methods and quantization effects*, Proceedings of IEEE CDC, 2008.

[33] A. Nedić and A. Ozdaglar, *On the rate of convergence of distributed subgradient methods for multi-agent optimization*, Proceedings of IEEE CDC, 2007, pp. 4711–4716.

[34] ———, *Approximate primal solutions and rate analysis for dual subgradient methods*, SIAM Journal on Optimization, forthcoming (2008).

[35] ______, *Convergence rate for consensus with delays*, Journal of Global Optimization, forthcoming (2008).

[36] ______, *A geometric framework for nonconvex optimization duality using augmented Lagrangian functions*, Journal of Global Optimization **40** (2008), no. 4, 545–573.

[37] ______, *Separation of nonconvex sets with general augmenting functions*, Mathematics of Operations Research **33** (2008), no. 3, 587–605.

[38] ______, *Subgradient methods for saddle-point problems*, Journal of Optimization Theory and Applications, forthcoming (2008).

[39] ______, *Subgradient methods in network resource allocation: Rate analysis*, Proceedings of CISS, 2008.

[40] ______, *Distributed subgradient method for multi-agent optimization*, IEEE Transactions on Automatic Control, forthcoming (2009).

[41] A. Nedić, A. Ozdaglar, and P.A. Parrilo, *Constrained consensus and optimization in multi-agent networks*, LIDS Technical Report 2779, available at http://arxiv.org/abs/0802.3922, 2008.

[42] A. Nedić, A. Ozdaglar, and A. Rubinov, *Abstract convexity for non-convex optimization duality*, Optimization **56** (2007), 655–674.

[43] J. Von Neumann, *Functional operators*, Princeton University Press, Princeton, 1950.

[44] R. Olfati-Saber and R.M. Murray, *Consensus problems in networks of agents with switching topology and time-delays*, IEEE Transactions on Automatic Control **49** (2004), no. 9, 1520–1533.

[45] A. Olshevsky and J.N. Tsitsiklis, *Convergence rates in distributed consensus averaging*, Proceedings of IEEE CDC, 2006, pp. 3387–3392.

[46] ______, *Convergence speed in distributed consensus and averaging*, SIAM Journal on Control and Optimization, forthcoming (2008).

[47] R. T. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[48] S. Shakkottai and R. Srikant, *Network optimization and control*, Foundations and Trends in Networking **2** (2007), no. 3, 271–379.

[49] S. Shenker, *Fundamental design issues for the future internet*, IEEE Journal on Selected Areas in Communication **13** (1995), no. 7, 1176–1188.

[50] R. Srikant, *Mathematics of Internet congestion control*, Birkhauser, 2004.

[51] 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.

[52] J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans, *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE Transactions on Automatic Control **31** (1986), no. 9, 803–812.

[53] H. Uzawa, *Iterative methods in concave programming*, Studies in Linear and Nonlinear Programming (K. Arrow, L. Hurwicz, and H. Uzawa, eds.), Stanford University Press, 1958, pp. 154–165.

[54] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Schochet, *Novel type of phase transitions in a system of self-driven particles*, Physical Review Letters **75** (1995), no. 6, 1226–1229.

[55] L. Xiao, S. Boyd, and S.-J. Kim, *Distributed average consensus with least mean square deviation*, Journal of Parallel and Distributed Computing **67** (2007), no. 1, 33–46.

Index