Distributed Optimization and Control: Primal–Dual, Online, and Event-Triggered Algorithms

XINLEI YI

Doctoral Thesis
Stockholm, Sweden 2020

KTH Royal Institute of Technology
School of Electrical Engineering and Computer Science
Division of Decision and Control Systems
SE-100 44 Stockholm
SWEDEN

TRITA-EECS-AVL-2020:45
ISBN 978-91-7873-640-9

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i elektro- och systemteknik onsdag den 14 oktober 2020 klockan 14.00 i sal F3, Lindstedtsvägen 26, KTH Campus, Stockholm.

Tryck: Universitetsservice US AB

Abstract

In distributed optimization and control, each network node performs local computation based on its own information and information received from its neighbors through a communication network to achieve a global objective. Although many distributed optimization and control algorithms have been proposed, core theoretical problems with important practical relevance remain. For example, what convergence properties can be obtained for nonconvex problems? How to tackle time-varying cost and constraint functions? Can these algorithms work under limited communication resources? This thesis contributes to answering these questions by providing new algorithms with better convergence rates under less information exchange than existing algorithms. It consists of three parts.

In the first part, we consider distributed nonconvex optimization problems. It is hard to solve these problems and often only stationary points can be found. We propose distributed primal–dual optimization algorithms under different information feedback settings. Specifically, when full-information feedback or deterministic zeroth-order oracle feedback is available, we show that the proposed algorithms converge sublinearly to a stationary point if each local cost function is smooth. They converge linearly to a global optimum if the global cost function also satisfies the Polyak–Łojasiewicz condition. This condition is weaker than strong convexity, which is a standard condition in the literature for proving linear convergence of distributed optimization algorithms. When stochastic gradient feedback or stochastic zeroth-order oracle feedback is available, we show that the proposed algorithms achieve linear speedup convergence rates, meaning that the convergence rates decrease linearly with the number of computing nodes.

In the second part, distributed online convex optimization problems are considered. For such problems, the cost and constraint functions are revealed at the end of each time slot. We focus on time-varying coupled inequality constraints and time-varying directed communication networks. We propose one primal–dual dynamic mirror descent algorithm and two bandit primal–dual algorithms. It is shown that these distributed algorithms achieve the same sublinear regret and constraint violation bounds as existing centralized algorithms.

In the third and final part, in order to achieve a common control objective for a networked system, we propose distributed event-triggered algorithms to reduce the amount of information exchanged. Specifically, we propose dynamic event-triggered control algorithms to solve the average consensus problem for first-order systems, the global consensus problem for systems with input saturation, and the formation control problem with connectivity preservation for first- and second-order systems. We show that these algorithms do not exhibit Zeno behavior and that they achieve exponential convergence rates.

Sammanfattning

Vid distribuerad optimering och reglering utför varje nätverksnod lokala beräkningar baserat på sin egen information och information som mottas från sina grannar via ett kommunikationsnätverk för att uppnå ett globalt mål. Även om många distribuerade optimerings- och regleralgoritmer har föreslagits kvarstår fundamentala teoretiska problem av stor praktisk relevans. Till exempel, vilka konvergensegenskaper kan erhållas för icke-konvexa problem? Hur hanterar man tidsvarierande kostnadsfunktioner och bivillkor? Kan dessa algoritmer fungera under begränsade kommunikationsresurser? Denna avhandling bidrar till att svara på dessa frågor genom att ge nya algoritmer med bättre konvergensegenskaper med mindre informationsutbyte än befintliga algoritmer. Avhandlingen består av tre delar.

I den första delen studerar vi distribuerade icke-konvexa optimeringsproblem. Det är svårt att lösa dessa problem och ofta kan bara stationära punkter hittas. Vi föreslår distribuerade primal-duala optimeringsalgoritmer under olika förutsättningar för återkoppling av information. När återkoppling av fullständig information eller återkoppling med hjälp av ett deterministiskt nollte ordningens orakel är tillgänglig, visar vi att de föreslagna algoritmerna konvergerar sublinjärt till en stationär punkt om varje lokal kostnadsfunktion är slät. De konvergerar linjärt till ett globalt optimum om den globala kostnadsfunktionen också uppfyller Polyak- Lojasiewicz-villkoret. Detta villkor är svagare än stark konvexitet, vilket är ett standardvillkor i litteraturen som används för att bevisa linjär konvergens av distribuerade optimeringsalgoritmer. När återkoppling baserad på stokastisk gradientinformation eller ett stokastiskt nollte ordningens orakel är tillgänglig visar vi att de föreslagna algoritmerna uppnår linjära konvergenshastigheter, vilket innebär att konvergenshastigheterna minskar linjärt med antalet beräkningsnoder.

I den andra delen av avhandlingen studerar vi distribuerad konvex optimering som utförs i realtid (online). För sådana optimeringsproblem ges kostnadsfunktionen och bivillkoren i slutet av varje tidsperiod. Vi fokuserar på tidsvarierande kommunikationsnätverk med tidsvarierande kopplade bivillkor angivna som olikheter. Vi föreslår en primal-dual dynamisk gradientalgoritm och två primal-duala banditalgoritmer. Vår analys visar att dessa distribuerade algoritmer uppnår samma sublinjära gränser för ånger (eng: regret) och överträdelse av bivillkor som befintliga centraliserade algoritmer.

I den tredje och sista delen, för att uppnå ett gemensamt reglermål för ett nätverkssystem, föreslår vi distribuerade händelsestyrda algoritmer för att minska mängden information som utbyts. Mer specifikt så föreslår vi dynamiska händelseutlösta regleralgoritmer för att lösa det genomsnittliga konsensusproblemet för första ordningens system, det globala konsensusproblemet för system med styrsignalssaturation och formationsproblemet med anslutningsbevaring för första och andra ordningens system. Vi visar att dessa algoritmer inte uppvisar Zeno-beteende och att de uppnår en exponentiell konvergenshastighet.

Acknowledgments

I would like to use "things to be respected according to Confucian thought" to express my deepest gratitude to my supervisor Professor Karl Henrik Johansson. Heartfelt thanks to him for providing me the opportunity to study at KTH and the freedom to explore any questions that I am interested in. He gives me insightful guidance when I am lost; he provides me invaluable support when I meet problems; he gives me warm encouragement when I have achievements no matter how small they are. It is really a great pleasure to work with him and he is the perfect supervisor in my mind.

I wish to express my sincere gratitude to my co-supervisors Professor John S. Baras for offering the inspiration and guidance to me, and Professor Dimos V. Dimarogonas for giving me detailed support in research. I would like to thank Professor Lihua Xie from Nanyang Technological University for being the opponent for my licentiate defense and hosting me on a productive exchange trimester, and Professor Mikael Johansson for being the advance reviewer for both my licentiate and doctoral theses as well as for chairing the public defence. I also would like to thank Professor Antonis Papachristodoulou from University of Oxford for being the opponent, and Professor Na Li from Harvard University, Professor Maurice Heemels from Eindhoven University of Technology, and Professor Anders Hansson from Linköping University for being the committee members. I am grateful to Professors Henrik Sandberg and Alexandre Proutiere for willing be the substitute members.

Heartfelt thanks to my collaborators Wen Du, Jemin George, Matin Jafarian, Xiuxian Li, Kun Liu, Mohammad Pirani, Jieqiang Wei, Junfeng Wu, Tao Yang, and Shengjun Zhang for the fruitful discussions with them and their interesting ideas and careful attentions to our work. Appreciation to Hao Chen for pointing out the mistakes in one proof. Special thanks to Matin Jafarian, Yuchao Li, Jieqiang Wei, and Tao Yang for proof reading this thesis, and Robert Mattila for translating the abstract of both my licentiate and doctoral theses into Swedish.

I would also express my appreciation to all my colleagues (current and former) at the Division of Decision and Control Systems for creating a friendly environment and an active working atmosphere, and for their continuous support for everything that I needed. I also want to thank the administrative staffs in our division Karin Karlsson Eklund, Felicia Gustafsson, Hanna Holmqvist, Tord Christer Magnusson, Anneli Ström, and Silvia Cardenas Svensson for their assistance and support.

Last, but not least, I would like to thank my family and friends for their unconditional love and constant support. I am deeply grateful to my partner, Zhengrong, for her understanding, support, and love.

Xinlei Yi
Stockholm, September 2020

Contents

List of Acronyms

ADMM	Alternating Direction Method of Multipliers
CC	Connected Component
CNN	Convolutional Neural Network
DERs	Distributed Energy Resources
DNN	Deep Neural Network
DZO	Deterministic Zeroth-Order
FO	First-Order
K–Ł	Kurdyka–Łojasiewicz
L-ADMM	Linearized Alternating Direction Method of Multipliers
NN	Neural Networks
NoSPPI	Number of Sampled Points Per Iteration
OCO	Online Convex Optimization
P–Ł	Polyak–Łojasiewicz
RSI	Restricted Secant Inequality
SCC	Strongly Connected Component
SGD	Stochastic Gradient Descent
SOS	Second-Order Stationary
SZO	Stochastic Zeroth-Order
ZO	Zeroth-Order

Notations

Real analysis

$\mathbb{R}^p$	the real Euclidean space of dimension p
$\mathbb{R}^p_+$	the nonnegative subspace of $\mathbb{R}^p$
$\mathbb{B}^p$	the unit ball centered around the origin in $\mathbb{R}^p$
$\mathbb{S}^p$	the unit sphere centered around the origin in $\mathbb{R}^p$
$\boldsymbol{e}_m$	the m-th element of the standard basis of $\mathbb{R}^p$
$\mathbb{N}_0$	the set of nonnegative integers
$\mathbb{N}_+$	the set of positive integers
$n!$	the factorial of n
$[n]$	the set $\{1, \dots, n\}$
$\mathcal{O}(\alpha_t)$	$\beta_t = \mathcal{O}(\alpha_t)$ if $\limsup_{t\to\infty}(\beta_t/\alpha_t)$ is bounded
$\mathbf{o}(\alpha_t)$	$\beta_t = \mathcal{O}(\alpha_t)$ if $\lim_{t\to\infty}(\beta_t/\alpha_t) = 0$
$\mathcal{P}_{\mathbb{K}}(\cdot)$	the projection operator onto set $\mathbb{K}$
$[\cdot]_+$	the projection operator onto $\mathbb{R}^p_+$
$\langle x, y\rangle$	the standard inner product of two vectors x and y
$\lceil\cdot\rceil$	the ceiling function
$\lfloor\cdot\rfloor$	the floor function
$\mathbf{1}_{(\cdot)}$	the indicator function
$\lvert\cdot\rvert$	the absolute value of a real number or the magnitude of a complex number
f^*	the global minimum value of function f
x^*	the global minimum point of function f
$\mathbb{X}^*$	the optimal set of function f, i.e., the set of global minimum points of function f
∇f	the (sub)gradient of function f
$\partial f(x)$	the set of all subgradients of function f
$\hat{\nabla}_1 f$	the one-point sampling based random gradient estimator
$\hat{\nabla}_2 f$	the two-point sampling based random gradient estimator

$\hat{\nabla}_p f$	the p-point sampling based deterministic gradient estimator
$\nabla^2 f$	the Hessian matrix of function f
$\mathcal{D}_\psi(\cdot, \cdot)$	Bregman divergence associated with strongly convex function ψ

Linear algebra

$\mathbb{R}^{n\times m}$	the space of n-by-m real matrices
$\|\cdot\|$	Euclidean norm for vectors or the induced 2-norm for matrices
$\|\cdot\|_1$	absolute sum for vectors or the induced 1-norm for matrices
$\|x\|_A^2$	the value of $x^\top Ax$, where x is a vector and A is a matrix
$\mathbf{1}_p$	a p-by-1 vector of all ones
$\mathbf{0}_p$	a p-by-1 vector of all zeros
$\boldsymbol{I}_n$	a n-by-n identity matrix
$\rho(\cdot)$	the spectral radius for matrices
$\rho_2(\cdot)$	the minimum positive eigenvalue for matrices having positive eigenvalues
$M^\top$	the transpose of real matrix M
$x^\top$	the transpose of real vector x
rank(M)	the rank of matrix M
null(M)	the null space of matrix M
det(M)	the determinant of square matrix M
$M > N$	$M - N$ is positive definite
$M \geq N$	$M - N$ is positive semidefinite
$M \otimes N$	the Kronecker product of two matrices M and N
Diag(x)	a diagonal matrix with the vector x on its diagonal
col$(z_1, \ldots, z_k)$	the concatenated column vector of vectors $z_i \in \mathbb{R}^{p_i},\ i \in [k]$
$[x]_i$	the i-th element of vector x
$[M]_{i,:}$	the i-th rom of matrix M
$[M]_{i,j}$	the element of matrix M in the i-th row and j-th column; when necessary, also denoted by M_{ij} or m_{ij}
$c_l(x)$	l-th component of vector x
$x \perp y$	vector x is orthogonal to vector y, i.e., $x^\top y = 0$
$\emptyset$	an empty set
$\|S\|$	the cardinality of set S

Graph theory

$\mathcal{G}$	undirected graph or directed graph
$\mathcal{G}_t$	time-varying undirected graph or directed graph

$\mathcal{V}$	the vertex set
$\mathcal{E}$	the edge set; when necessary, also denoted by $\mathcal{E}(\mathcal{G})$
$\mathcal{E}_t$	the time-varying edge set; when necessary, also denoted by $\mathcal{E}_t(\mathcal{G}_t)$
n	the number of vertices
(i, j)	an edge in a graph, i.e., a directed link from vertex i to vertex j
$\mathcal{N}_i$	neighbors of vertex i in an undirected graph; when necessary, also denoted by $\mathcal{N}_i(\mathcal{G})$
$\mathcal{N}_i^{\text{in}}$	in-neighbors of vertex i in a directed graph; when necessary, also denoted by $\mathcal{N}_i^{\text{in}}(\mathcal{G})$
$\mathcal{N}_i^{\text{out}}$	out-neighbors of vertex i in a directed graph; when necessary, also denoted by $\mathcal{N}_i^{\text{out}}(\mathcal{G})$
A	the (weighted) adjacency matrix of $\mathcal{G}$
L	the (weighted) Laplacian matrix of $\mathcal{G}$
K_n	$K_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is the Laplacian matrix of a complete graph,
W	the mixing matrix of $\mathcal{G}$
W_t	the time-varying mixing matrix of $\mathcal{G}_t$
$B(\mathcal{G})$	the incidence matrix of $\mathcal{G}$
SCC_m	the m-th strongly connected component of a directed graph
CC_m	the m-th connected component of an undirected graph

Other

$a := b$	a is defined as b
$a \Leftrightarrow b$	a and b are equivalent
$a \Rightarrow b$	a implies b
$\forall$	for all
$\text{sat}_h(\cdot)$	the saturation function with saturation level $h > 0$
t_k^i	the k-th triggering time of agent i
$t_{k_i(t)}^i$	the latest triggering time of agent i before time instant t

Chapter 1

Introduction

In recent years, rapid developments in digital systems, communication, and sensing technologies have led to the emergence of networked systems. These systems consist of a large number of interconnected subsystems (agents), which are required to cooperate in order to achieve a desirable global objective through local interactions. Such networked systems have been extensively studied in various disciplines over the past decades, and they have broad applications in various areas, for instance, surveillance [1], monitoring [2], manufacturing [3], data mining [4], learning [5, 6], software engineering [7], power grid [8, 9], transportation [10], and logistics [11]. Due to their distributed nature, these applications often require distributed optimization and control techniques. Traditional centralized strategies are often not suitable since they are subject to single point of failure, high communication requirement, substantial computation burden, and limited flexibility and scalability. All of these have made imperative the need of developing new distributed approaches to solve optimization and control problems in networked systems.

This chapter is organized as follows. Section 1.1 provides some applications that have motivated the work presented in this thesis. Section 1.2 briefly introduces distributed optimization, online convex optimization, and distributed event-triggered control. Section 1.3 presents the problems studied in this thesis. Section 1.4 gives the thesis outline and describes the contributions of the author.

1.1 Motivating examples

In this section, nine examples are provided to motivate the problems considered in this thesis.

Motivating example 1: Distributed regularized logistic regression

Logistic regression is used to classify an observation into one of two classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to the two classes. The key question in logistic regression is how to fit the logistic

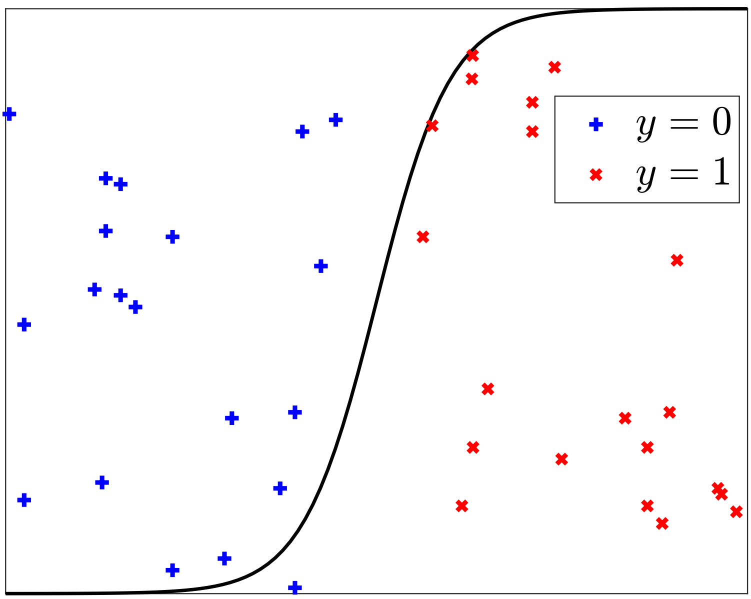


Figure 1.1: Illustration of logistic regression with labeled observations.

regression model using labeled observations. Figure 1.1 illustrates logistic regression with two classes of labeled observations separated by an S-shaped curve.

To compute the regression coefficients of the model, the negative of the log likelihood function, also called the objective function, is minimized:

$$f(x) = \frac{1}{m}\sum_{i=1}^{m}(y_i \log(1 + \exp(-x^\top z_i)) + (1 - y_i)\log(1 + \exp(x^\top z_i))),$$

where $x \in \mathbb{R}^p$ is the regression coefficient vector with p being the number of features, m is the number of independent observations, and $\{z_i \in \mathbb{R}^p\}_{i=1}^m$ are independent observations with known labels $\{y_i \in \{0, 1\}\}_{i=1}^m$.

Logistic regression is prone to overfitting if there are large number of features. Regularization can be used to train models that generalize better to unseen data, by preventing the algorithm from overfitting [12]. The objective function then normally has a regularization term:

$$f(x) = \frac{1}{m}\sum_{i=1}^{m}(y_i \log(1 + \exp(-x^\top z_i)) + (1 - y_i)\log(1 + \exp(x^\top z_i))) + \lambda r(x), \tag{1.1}$$

where $\lambda > 0$ is a regularization parameter and $r(x)$ is a regularization function. Various convex and nonconvex regularization functions for logistic regression have been proposed [13], for example,

$$r(x) = \sum_{s=1}^{p} \frac{\mu[x]_s^2}{1 + \mu[x]_s^2},$$

where $\mu > 0$ is another parameter.

Traditionally, the above optimization problem is solved by a single machine using the complete data set. However, it is sometimes necessary to solve it in a distributed manner. For example, when the data set of labeled observations is very large and cannot fit the memory of a single machine. Another motivating scenario is when data is collected from multiple distributed data sources by a group of machines and is stored distributively due to data ownership and privacy concerns. Thus, it is sometimes necessary to fit the logistic regression model distributively. Specifically, suppose there are n computing nodes and each node i has m_i labeled observations satisfying $\sum_{i=1}^{n} m_i = m$. All nodes collaborate to solve the optimization problem

$$\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x),$$

where each function f_i is held privately by node i and is given by

$$f_i(x) = \frac{n}{m} \sum_{l=1}^{m_i} (y_{il} \log(1 + \exp(-x^\top z_{il})) + (1 - y_{il}) \log(1 + \exp(x^\top z_{il}))) + \sum_{s=1}^{p} \frac{\lambda\mu[x]_s^2}{1 + \mu[x]_s^2}, \tag{1.2}$$

where $z_{il} \in \mathbb{R}^p$ is the l-th observation with label $y_{il} \in \{0, 1\}$ owned by node i.

In Chapter 3, we show that the above distributed regularized logistic regression problem can be solved by our new distributed primal–dual first-order (FO) algorithm with a faster convergence rate than state-of-the-art distributed first-order algorithms.

Motivating example 2: Distributed phase retrieval

The classic linear inverse problem is to recover an unknown signal $x \in \mathbb{R}^p$ from m linear measurements of the form $Bx = y$, where $B \in \mathbb{C}^{m \times p}$ is a known linear measurement operator matrix, and $y \in \mathbb{C}^m$ is a noisy but known vector. In contrast, phase retrieval is to recover the unknown signal x from the noisy squared magnitude of the linear measurements

$$y_i = |b_i^\top x|^2 + w_i, \ \forall i \in [m],$$

where $b_i \in \mathbb{C}^p$ is the i-th linear measurement operator and $y_i \in \mathbb{R}$ is the corresponding noisy squared magnitude, $|\cdot|$ is the magnitude of a complex number, and $w_i \in \mathbb{R}$ is noise. Phase retrieval has a long history and can be traced back at least to the 1970's [14–16].

Recently, it has gained increased interest from the optimization community, e.g., [17–21]. Phase retrieval can be reformulated as the nonconvex optimization problem

$$\min_{x\in\mathbb{R}^p} f(x),$$

where $f(x) = \frac{1}{m}\sum_{i=1}^{m}(y_i - |b_i^\top x|^2)^2$.

In practice, sometimes the linear measurement operators and the corresponding noisy squared magnitudes are recorded by different detectors [22]. It is then natural for large data sets to split the cost function across detectors and thus reformulate the centralized optimization problem as the distributed optimization problem

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x),$$

where

$$f_i(x) = \frac{n}{m}\sum_{l=1}^{m_i}(y_{il} - |b_{il}^\top x|^2)^2 = \frac{n}{m}\sum_{l=1}^{m_i}(y_{il} - (x^\top b_{il}^R)^2 - (x^\top b_{il}^I)^2)^2 \tag{1.3}$$

with m_i being the number of data points recorded by detector i, $b_{il} = b_{il}^R + \mathrm{i}b_{il}^I \in \mathbb{C}^p$ being the phase of the linear operator used in the l-th measurement by detector i, and $y_{il} \in \mathbb{R}$ being the corresponding noisy squared magnitude.

In Chapter 3, we show that the above distributed phase retrieval problem can be solved by our proposed distributed alternating direction method of multipliers (ADMM) algorithm with a faster convergence rate than state-of-the-art distributed ADMM algorithms.

Motivating example 3: Distributed training of neural networks

In the deep learning literature, it has been observed that performance can be dramatically improved when increasing the number of model parameters and/or the number of training examples, e.g., [23–25]. However, training neural networks is very tedious. Many neural networks have millions, even billions, model parameters and large amounts of data are needed to learn these parameters. This is a computationally intensive process which takes a lot of time. It can even take days to train a deep neural network [26]. Moreover, sometimes the training data set is too large to be stored on a single machine. Therefore it is important to come up with distributed algorithms to drastically reduce the training time. Two novel methodologies, data and model parallelisms have been proposed, e.g., [27–30]. Specifically, data parallelism means the partition of the training data across multiple machines and it allows each machine to read and update all model parameters. Model parallelism means the partition of the model parameters across multiple machines and it makes each machine responsible for updating only its assigned portion of parameters (either using the full data set or a subset).

In this example, we focus on data parallelism as illustrated in Figure 1.2. In this methodology we spawn n workers and assign a share of the data set to each worker. Using

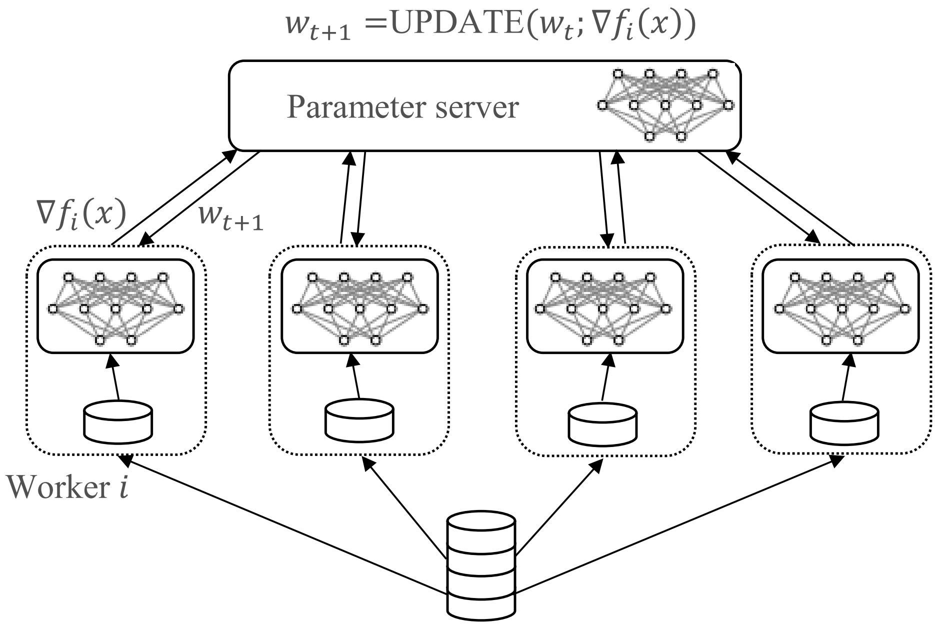


Figure 1.2: Illustration of data parallelism. A parameter server is responsible for the aggregation of model updates and parameter requests coming from workers. All workers get a copy of the central model with parameters w_t. The data is split into several partitions, where a specific worker is responsible for the computation of its own partition. Each worker samples mini-batches from its own data to produce the gradient $\nabla f_i(x)$ and then communicates it with the parameter server. The parameter server integrates this gradient by applying a specific update procedure to produce w_{t+1}. This process repeats itself until all workers have sampled all mini-batches from their shard. *Source:* https://joerihermans.com/ramblings/distributed-deep-learning-part-1-an-introduction/

this data, worker i iterates through mini-batches of data to produce a gradient, $\nabla f_i(x)$ for every mini-batch x. Next, $\nabla f_i(x)$ is sent to the parameter server, which incorporates the gradient using an update mechanism. Data parallelism is thus based on the master–worker architecture.

Although numerous distributed training algorithms based on data parallelism have been proposed, many of them are not truly distributed since they follow a master–worker architecture and do not involve any peer-to-peer communication. These algorithms are not always robust and they are useless if the server fails. In Chapter 4, we propose a distributed primal–dual stochastic gradient descent (SGD) algorithm, suitable for arbitrarily connected communication networks and any smooth (possibly nonconvex) cost functions. This algorithm achieves linear speedup in the number of partitions (agents), which enables us to scale up the computing capacity by adding more agents [31–33].

Motivating example 4: Black-box adversarial attacks

As machine learning is being more widely used, security concerns are attracting more attentions, especially for safety-critical applications [35, 36]. Many recent studies have shown that neural networks are vulnerable to adversarial attacks, e.g., [34, 37–41]. The outputs of neural networks can be altered arbitrarily with slightly perturbed inputs. For example, it has been shown in [34, 37–39] that a slightly modified image can be easily generated and misguide a well-trained image classifier into producing incorrect results. Figure 1.3 gives four examples to illustrate how carefully crafted small perturbations of the original inputs, often imperceptible to the human eye, misguide the network into producing incorrect outputs. The original images are in the left column, while the corresponding perturbed images produced by the algorithm proposed in [34] are shown in the right column. The perturbed images are misclassified by the network proposed in [42].

Attacks on machine learning models can be divided into white-box and black-box attacks. White-box attacks mean that the adversary has complete knowledge of the target model, whereas for black-box attacks the adversary only queries the target model, which may return complete or partial information [43]. Black-box attacks normally are more relevant in many practical scenarios since in most applications internal configurations of machine learning models, including the network structure and weights, are not released.

Designing adversarial attacks on a given network can be formulated as an optimization problem with the objective to find the smallest perturbation that leads to misclassification, e.g., [44, 45]. Note that under black-box attacks the adversary only accesses the input and output of a machine learning model. In other words, the adversary has to generate adversarial perturbations without access to the target model to compute gradients. Therefore it is intuitive to cast the problems of generating black-box attack examples as gradient-free optimization problems, e.g., [46–48]. Although various centralized and distributed gradient-free optimization algorithms have been proposed to generate adversarial black-box attacks, core theoretical questions remain. For instance, can distributed gradient-free optimization algorithms achieve comparable convergence rates as their first-order counterparts? Can they have similar convergence properties as their centralized counterparts? Can they even achieve linear speedup? In Chapter 5, we provide positive answers to these questions.

Motivating example 5: Multi-target tracking

Consider a multi-target tracking problem in which n agents follow n targets. Figure 1.4 shows how each agent i tracks each target i from time t to $t+1$. Let $z_i(s)$ and $\tilde{z}_i(s)$ denote the positions of agent i and target i at time s, respectively. To model agent and target paths, we introduce a parameterization:

$$\begin{aligned} z_i(s) &= \sum_{k=1}^{p_i} [x_{i,t}]_k c_{k,t}(s), \\ \tilde{z}_i(s) &= \sum_{k=1}^{p_i} [\xi_{i,t}]_k c_{k,t}(s), \;\; s \in [t, t+1), \end{aligned}$$

(a) A stingray misclassified as a sea lion.

(b) An ostrich misclassified as a goose.

(c) A jay misclassified as a junco.

(d) A water ouzel misclassified as a redshank.

Figure 1.3: Examples to illustrate how carefully crafted small perturbations of the original inputs can misguide the network into producing incorrect outputs. The left column shows the original images and the right column shows the perturbed images. *Source:* https://davidstutz.de/simple-black-box-adversarial-attacks-on-deep-neural-networks/ and [34].

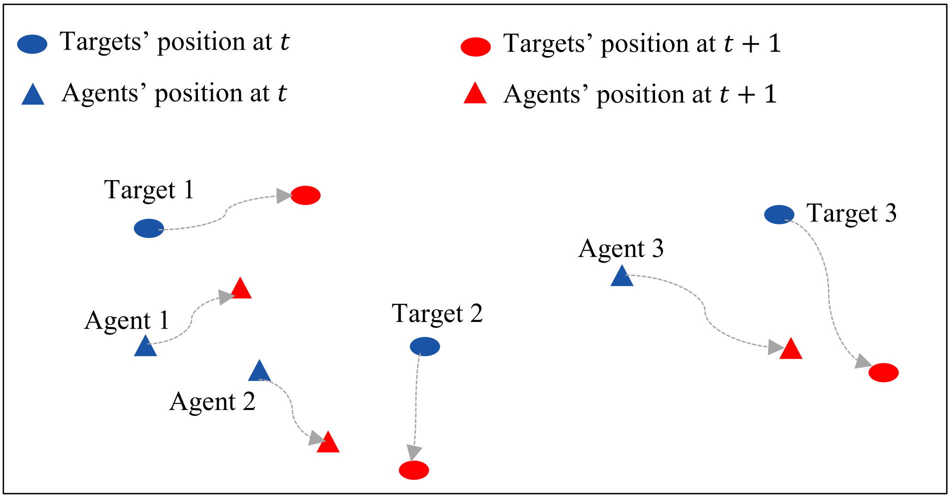


Figure 1.4: Illustration of multi-target tracking.

where $c_{k,t}(s)$ are vector functions that parameterize the space of possible trajectories over time $[t, t+1)$ and satisfy

$$\int_t^{t+1} \langle c_{k,t}(s), c_{l,t}(s) \rangle ds = \begin{cases} 1, & \text{if } k = l \\ 0, & \text{else.} \end{cases}$$

The action spaces of agent i and target i are given by $x_{i,t} = \mathrm{col}([x_{i,t}]_1, \dots, [x_{i,t}]_{p_i}) \in X_i \subseteq \mathbb{R}^{p_i}$ and $\xi_{i,t} = \mathrm{col}([\xi_{i,t}]_1, \dots, [\xi_{i,t}]_{p_i}) \in \mathbb{R}^{p_i}$, respectively. At time t, agent i repositions itself by selecting an action $x_{i,t}$ such that it could stay as close as possible to target i during time interval $[t, t+1)$. At the same time it wants the selection cost $\langle \pi_{i,t}, x_{i,t} \rangle$ to be as small as possible, where $\pi_{i,t} \in \mathbb{R}_+^{p_i}$ is the price vector. This goal can be captured by defining a local cost function

$$\begin{aligned} f_{i,t}(x_{i,t}) &= \zeta_{i,1} \langle \pi_{i,t}, x_{i,t} \rangle + \zeta_{i,2} \int_t^{t+1} \|z_i(s) - \tilde{z}_i(s)\|^2 ds \\ &= \zeta_{i,1} \langle \pi_{i,t}, x_{i,t} \rangle + \zeta_{i,2} \|x_{i,t} - \xi_{i,t}\|^2, \end{aligned}$$

where $\zeta_{i,1}$ and $\zeta_{i,2}$ are nonnegative constants to trade-off the two goals. Here, target i's action $\xi_{i,t}$ and the price vector $\pi_{i,t}$ are observed only after the selection. Agents need to cooperatively take into account energy and communication constraints. In some cases, they can be represented as linear local constraint functions $g_{i,t}(x_{i,t}) = D_{i,t} x_{i,t} - d_{i,t}$, where $D_{i,t} \in \mathbb{R}^{m \times p_i}$ and $d_{i,t} \in \mathbb{R}^m$ are time-varying and unknown at time t. These coupling constraints determine the limits on the available resources to be shared among the agents. Chapter 6 shows how such a multi-target tracking problem can be solved by a novel distributed online primal–dual dynamic mirror descent algorithm proposed in that chapter.

Motivating example 6: Coordination of distributed energy resources

In the past decades, the power system has been undergoing a transition from a system with conventional generation through few power plants and inflexible loads to a system with a large number of distributed generators, energy storages, and flexible loads, e.g., [49–51]. The new distributed energy resources (DERs) are small and highly flexible compared with conventional generators and can be aggregated to provide power necessary to meet varying demands. As the electricity grid continues to modernize, DERs can facilitate the transition to a smarter grid.

In order to achieve an effective and efficient deployment of DERs, one needs to properly design their coordination scheme. Specifically, consider a power grid with n power generation units. Each unit i has p_i conventional and renewable power generators. The units can communicate through a communication infrastructure. At stage t, let $x_{i,t} \in \mathbb{X}_i$ and $\mathbb{X}_i \subset \mathbb{R}^{p_i}$ be the output and the set of feasible outputs of the generators in unit i, respectively. To generate the output, each unit i suffers a cost $f_{i,t}(x_{i,t})$. This local cost is described by a quadratic function [52], but is unknown in advance, since fossil fuel price is fluctuating and renewable energy is uncertain and unpredictable. In addition to the local generator constraints $\mathbb{X}_i$, all units need to cooperatively take into account global constraints, such as power balance and emission constraints. The global constraints can be modelled as $\sum_{i=1}^{n} g_{i,t}(x_{i,t}) \le \mathbf{0}_m$, where $g_{i,t}$ is unit i's local constraint function. Again, the precise form of the constraint functions is typically unknown in advance. The goal of the units is to reduce the global cost while satisfying the constraints. Chapter 7 shows how this DERs coordination problem can be solved by the distributed bandit online primal–dual optimization algorithms proposed in that chapter.

Motivating example 7: Satellite formation flying

Multiple satellites may work together to accomplish the objective of one larger, usually more expensive, satellite. This reduces cost and adds flexibility to space programs [53]. An important component of such a strategy is satellite formation flying. Figure 1.5 shows the PRISMA formation flying mission. PRISMA was a Swedish-led technology mission to demonstrate formation flying and rendezvous technologies. The mission consisted of two spacecrafts, a bigger one with advanced and highly maneuverable capability, called MAIN, and a smaller one without a maneuvering capability, called TARGET. TARGET simply followed the trajectory into which it was injected by the launch system. MAIN had full translational capability, and performed a series of maneuvers around TARGET, on both close and long range, using sensors provided [54].

The satellite formation flying problem of PRISMA is a resource-constrained two-agent system. There are several constraints in this system, but here we only discuss two of them. The first one is energy. MAIN has six thrusters arranged to provide torque-free translational capability in all directions. Thus, the control input of MAIN should be optimized such that the energy consumed to perform the maneuvers is saved. The second constraint is communication. Although there are two deployable solar panels to power MAIN and there is one body-mounted solar panel to power TARGET, energy used for communication

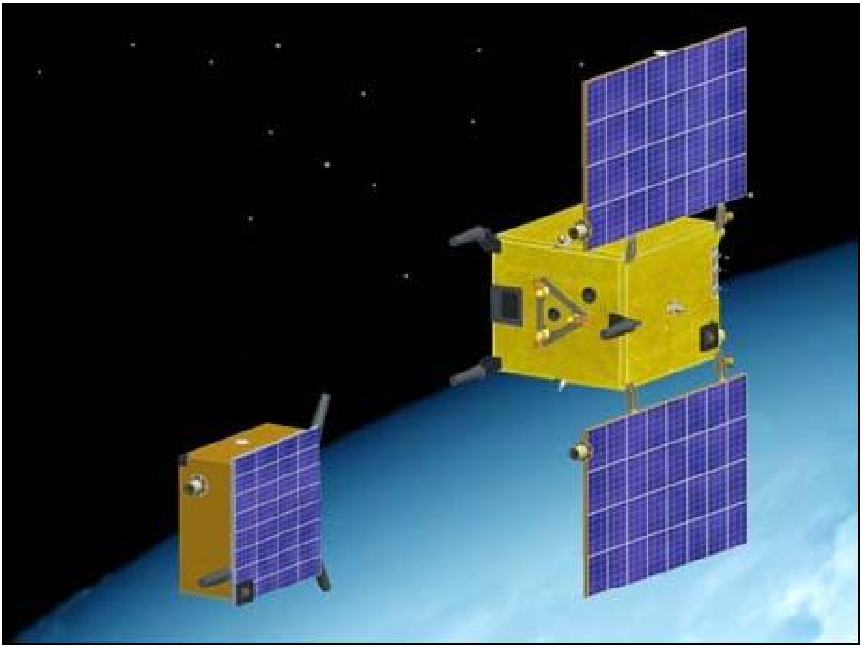


Figure 1.5: Illustration of the PRISMA formation flying mission [55].

should be limited. One way to partially satisfy these two constraints is by using the event-triggered control strategies investigated in Chapter 8.

Motivating example 8: Heavy-duty vehicle platooning

The formation of a group of heavy-duty vehicles at close intervehicular distances, similar to cyclists in a race, reduces fuel consumption thanks to reduced air resistance. A platooning with three vehicles is shown in Figure 1.6. In [56], the authors present an architecture for heavy-duty vehicle platooning to improve the efficiency of freight transportation. Experimental results show a significant decrease in fuel and energy consumption.

Vehicle platooning is a formation control problem with input saturation. The desired formation is a line graph. The input saturation follows from that the vehicles have limitations such as maximum acceleration and deceleration. Moreover, continuous communication among vehicles is impossible. One way to model such a system is using event-triggered multi-agent systems with input saturation as studied in Chapter 9.

Motivating example 9: Autonomous surface vehicle tracking

Autonomous surface vehicles can be used for target tracking, environmental sampling, hydrographic or oceanographic surveys, water surface cleaning, etc. One specific example

Figure 1.6: A platooning of heavy-duty vehicles. *Source:* https://www.scania.com

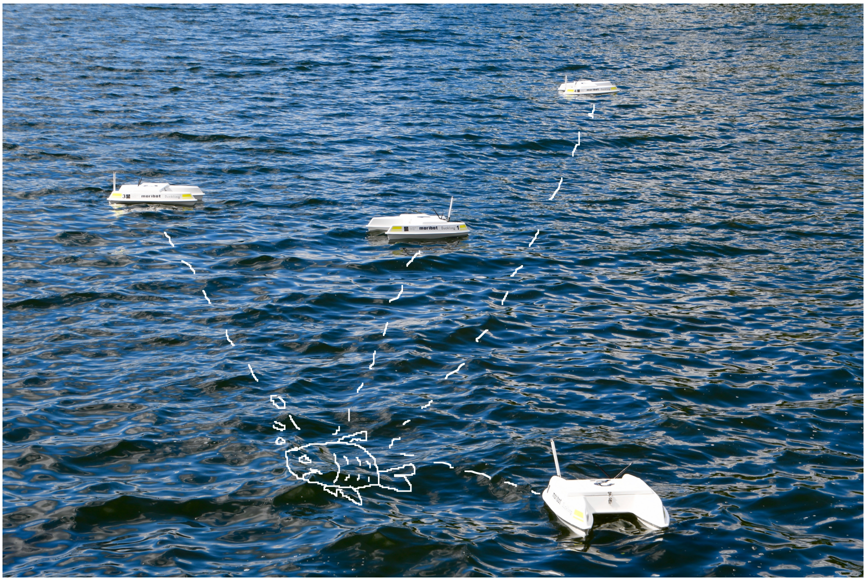


Figure 1.7: Illustration of autonomous surface vehicle tracking [57].

of autonomous surface vehicle tracking is collaborative tracking of fish [57], see Figure 1.7. The autonomous surface vehicles measure the location of the underwater target (the fish) by using sonar. The vehicles create a formation around the target to keep the fish within sensing range.

Fish tracking is a formation control problem of a resource-constrained multi-agent system. There are several constraints in this system. The first one is that each vehicle has limited energy since it is battery-powered. Motion and communication consume energy, so

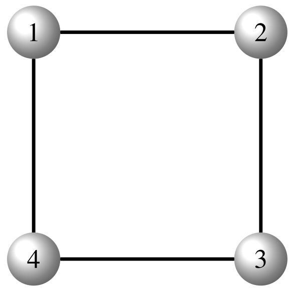


Figure 1.8: An example of a network of four computing agents.

it is important to design a proper control law. The second constraint is that the transceiver in each vehicle is simple and has limited communication range. The relative distance between any two vehicles may change during operation, so the connectivity of the underlying interaction graph cannot be guaranteed. One way to handle these constraints is to consider event-triggered formation control with connectivity preservation using relative positions as considered in Chapter 10.

1.2 Distributed optimization and control

The examples presented above motivate us to propose new distributed optimization and control algorithms. In this section, we briefly review related studies of distributed optimization and control in the literature, including distributed optimization, online convex optimization, and distributed event-triggered control.

1.2.1 Distributed optimization

Consider a networked system of n agents, each of which has a local private cost function $f_i(x)$, where $x \in \mathbb{R}^p$ is the decision variable and p is its dimension. The objective of distributed optimization is to minimize a global cost function, which is a sum of the local cost functions of all agents,

$$\min_{x\in\mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} f_i(x), \tag{1.4}$$

in a distributed manner by local computation and communication. The underlying communication network is described by a (directed or undirected) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set of vertices (or nodes) $\mathcal{V} = [n]$ and the set of edges (links) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Figure 1.8 shows an example with four agents connected through an undirected ring graph.

When each local cost function is convex, the optimization problem (1.4) is called a distributed convex optimization problem, which has a long history and can be traced back at least to the 1980's [58–60]. It has gained renewed interests in recent years due to its wide applications in power systems, machine learning, and sensor networks, just to name a few. Various distributed algorithms have been developed and their convergence rates have also been analyzed. Here convergence rates mean how quickly the output sequence of the algorithm approaches the global optimum. In these algorithms, each agent performs local computation based on its own information and information received from its neighbors. For example, the following distributed first-order (sub)gradient descent algorithm was proposed in [61]:

$$x_{i,k+1} = \sum_{j=1}^{n} [W_k]_{ij} x_{j,k} - \eta_k \nabla f_i(x_{i,k}),$$

where $x_{i,k} \in \mathbb{R}^p$ is agent i's estimate of the optimal solution at time instant k, W_k is the mixing matrix of the underlying time-varying communication network, $\eta_k > 0$ is the stepsize, and $\nabla f_i(x_{i,k})$ is the (sub)gradient of f_i. It was shown in [61] that this algorithm finds a global optimum with an $O(\ln(k)/\sqrt{k})$ convergence rate, i.e.,

$$f(\bar{x}_k) - f^* = O(\ln(k)/\sqrt{k}),$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$ and $f^* = \min_{x\in\mathbb{R}^p} f(x)$, which is a sublinear convergence rate. Sublinear convergence rate is described in terms of a power function of the iteration counter k [62]. Other sublinear convergence rates, such as $O(1/\sqrt{k})$, $O(1/k)$, and $O(1/k^2)$, have also been achieved by other distributed algorithms, e.g., [63–67]. Linear convergence rate, which is given in terms of an exponential function of the iteration counter, can be established under more stringent strong convexity conditions. For instance, in [68–87] and [88–91], the authors assumed that each local cost function and the global cost function are strongly convex, respectively, and showed that their proposed distributed algorithms achieve a linear convergence rate, i.e.,

$$f(\bar{x}_k) - f^* = O(c^k),$$

where c is a constant in the interval $(0, 1)$. For recent overviews we refer to the surveys [92–99] and the books [100–103].

In many applications, such as optimal power flow [104], resource allocation [105], and empirical risk minimization [106], the cost functions are usually nonconvex. Thus, it is important to develop distributed algorithms to solve also nonconvex optimization problems. These challenging yet important problems have drawn attention recently from control, signal processing, and machine learning. For example, unconstrained and constrained distributed nonconvex optimization problems were considered in [107–120] and [121–128], respectively. In these studies, convergence results typically ensure that the distributed algorithms find (first-order) stationary points

$$\{x \in \mathbb{R}^p : \nabla f(x) = \mathbf{0}_p\},$$

which could be local maxima or minima. Global optima are hard to find. In [111–116,124], it was shown that when each local cost function is smooth, first-order stationary points can be found with an $\mathcal{O}(1/k)$ convergence rate, i.e.,

$$\|\nabla f(\bar{x}_k)\|^2 = \mathcal{O}(1/k).$$

Second-order stationary points

$$\{x \in \mathbb{R}^p : \nabla f(x) = \mathbf{0}_p \text{ and } \nabla^2 f(x) \geq 0\}$$

can be found if additional assumptions are made, such as imposing the Kurdyka–Łojasiewicz condition, assuming a Lipschitz-continuous Hessian, or making a suitably initialization, e.g., [109, 116–118, 120].

There is a correspondence between the convergence rate and the iteration complexity. The upper bound for the iterations to attain an ϵ-accuracy, i.e., $f(\bar{x}_k) - f^* \leq \epsilon$ for convex problems or $\|\nabla f(\bar{x}_k)\|^2 \leq \epsilon$ for nonconvex problems, where $\epsilon > 0$ is a constant, is an inverse function of convergence rate. For example, if an algorithm has an $\mathcal{O}(1/\sqrt{k})$ convergence rate for an optimization problem, then it takes $\mathcal{O}(1/\epsilon^2)$ iterations to attain an ϵ-accuracy. Similarly, if another algorithm has an $\mathcal{O}(1/\sqrt{nk})$ convergence rate for the same optimization problem, then it takes $\mathcal{O}(1/(n\epsilon^2))$ iterations, which is n times smaller than $\mathcal{O}(1/\epsilon^2)$, to attain an ϵ-accuracy. In this sense, the second algorithm is n times faster than the first one, and thus achieves a linear speedup in the number of agents. Linear speedup enables us to scale up the computing capacity by adding more agents.

Note that aforementioned algorithms use at least gradient information of the cost functions, and sometimes even second- or higher-order information. However, in many applications explicit expressions of the gradients are often unavailable or at least difficult to obtain. For example, in empirical risk minimization, the actual gradient has to be calculated from the entire data set, which results in a heavy computational burden. A stochastic gradient can be calculated from a randomly selected subset of the data and is often an efficient way to replace the actual gradient. Various distributed SGD algorithms have been proposed, e.g., [31–33, 129–141]. Convergence properties of these algorithms have been analyzed in detail. In particular, in [31–33, 132, 133, 135–137], an $\mathcal{O}(1/\sqrt{nk})$ convergence rate has been established for SGD algorithms and smooth nonconvex cost functions. This rate is n times faster than the well-known $\mathcal{O}(1/\sqrt{k})$ convergence rate established by SGD over a single agent [142], and thus a linear speedup in the number of agents is achieved. Moreover, in [140, 141], an $\mathcal{O}(1/(nk))$ convergence rate has been established for smooth strongly convex cost functions. This rate is also n times faster than the optimal convergence rate $\mathcal{O}(1/k)$ established for centralized SGD algorithms [143], and thus linear speedup is also achieved. However, existing distributed SGD algorithms obtaining linear speedup require restrictive assumptions on the cost functions or the communication network.

In many applications, even stochastic gradients are unavailable [144–146]. For example, many cost functions of big data problems that deal with complex data-generating processes cannot be explicitly defined [46]. Motivated by this, some recent works have started to modify distributed gradient-based optimization algorithms to zeroth-order, e.g., [147–155]. However, it is unclear whether linear speedup can be achieved by these algorithms.

Although many distributed optimization algorithms have been proposed, the study is far from being complete. For example, it is interesting trying to achieve linear convergence without the strong convexity assumption, since many practical applications do not have strongly convex cost functions [156]. Another interesting direction is to develop distributed SGD algorithms that not only achieve linear speedup convergence rates $O(1/\sqrt{nk})$ and $O(1/(nk))$, but also do not require restrictive assumptions on the cost functions or the communication networks. It is also revelant to develop distributed zeroth-order algorithms to achieve linear speedup compared with centralized such algorithms.

1.2.2 Online convex optimization

Online convex optimization is a promising methodology for modeling sequential tasks and has important applications in machine learning [157], smart grids [158], sensor networks [159], and so on. It has been studied since the 1990's [160–168]. Online convex optimization can be understood as a repeated game between a learner and an adversary [157]. At round t of the game, the learner chooses a point x_t from a known feasible region $\mathbb{X} \subseteq \mathbb{R}^p$, which is a closed convex set. Then, the adversary observes x_t and chooses a convex loss function $f_t : \mathbb{X} \to \mathbb{R}$. After that, the loss function f_t is revealed to the learner who suffers a loss $f_t(x_t)$. Note that at each round the loss function can be arbitrarily chosen by the adversary, especially with no probabilistic model imposed on the choices. This is the key difference between online and stochastic convex optimization. An adversary with the power to arbitrarily choose the loss functions is said to be a completely adaptive adversary [164]. The goal of the learner is to choose a sequence $\boldsymbol{x}_{[T]} = (x_1, \dots, x_T)$ such that her regret

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) = \sum_{t=1}^{T} (f_t(x_t) - f_t(y_t)) \tag{1.5}$$

is minimized, where T is the total number of rounds and $\boldsymbol{y}_{[T]} = (y_1, \dots, y_T)$ is a comparator sequence. In the literature, there are two commonly used comparator sequences. One is the optimal dynamic decision sequence $\boldsymbol{y}_{[T]} = \boldsymbol{x}^*_{[T]} = (x^*_1, \dots, x^*_T)$ solving the following constrained convex optimization problem when the sequence of cost functions is known a priori:

$$\min_{\boldsymbol{x}_{[T]} \in \mathbb{X}^T} \sum_{t=1}^{T} f_t(x_t).$$

In this case $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})$ is called the dynamic regret for $\boldsymbol{x}_{[T]}$. Another comparator sequence is $\boldsymbol{y}_{[T]} = \check{\boldsymbol{x}}^*_{[T]} = (\check{x}^*_T, \dots, \check{x}^*_T)$, where $\check{x}^*_T$ is the optimal static decision solving

$$\min_{x \in \mathbb{X}} \sum_{t=1}^{T} f_t(x).$$

In this case $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})$ is called the static regret. It is straightforward to see that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) \le \mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})$, $\forall \boldsymbol{y}_{[T]} \in \mathbb{X}^T$, and that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) \le \mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})$. In online convex optimization, we are usually interested in an upper bound on the worst case regret of an algorithm. Intuitively, an algorithm performs well if its static regret is sublinear as a function of T, since this implies that on the average the algorithm performs as well as the best fixed strategy in hindsight as T goes to infinity [157, 165].

It is known that the simple and popular projection-based online gradient descent algorithm

$$x_{t+1} = \mathcal{P}_{\mathbb{X}}(x_t - \alpha\nabla f_t(x_t)), \tag{1.6}$$

where $\mathcal{P}_{\mathbb{X}}(\cdot)$ is the projection onto the closed convex set $\mathbb{X}$ and $\alpha > 0$ is the stepsize, achieves an $\mathcal{O}(\sqrt{T})$ static regret bound for loss functions with bounded subgradients [163], i.e.,

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(\sqrt{T}).$$

As a result, when the convex cost function is fixed, i.e., $f_t = f$, the above result implies that $f(\sum_{t=1}^{T} x_t/T) - f^* = \mathcal{O}(1/\sqrt{T})$, where $f^* = \min_{x\in\mathbb{X}} f(x)$. It was later shown that $\mathcal{O}(\sqrt{T})$ is a tight bound up to constant factors [166]. The static regret bound can be reduced under more stringent strong convexity conditions on the objective functions [157, 165–167] or by allowing to sample the gradient of the objective function multiple times per round [168].

Despite the simplicity of the algorithm (1.6), its computational cost is crucial for its applicability. The projection $\mathcal{P}_{\mathbb{X}}(\cdot)$ is easy to compute and even has a closed form solution when $\mathbb{X}$ is a simple set, e.g., a box or a ball. However, in practice, the constraint set $\mathbb{X}$ is often complex. For example, if $\mathbb{X}$ is characterized by inequalities as $\mathbb{X} = \{x : g(x) \leq \mathbf{0}_m,\ x \in \mathbb{R}^p\}$, where $g(x) = \mathrm{col}(g_1(x), \dots, g_m(x))$ with each $g_i : \mathbb{R}^p \to \mathbb{R}$ being a convex function, then the projection $\mathcal{P}_{\mathbb{X}}(\cdot)$ yields a heavy computational burden. To tackle this challenge, online convex optimization with long-term constraints was considered in [169]. In this case, instead of requiring $g(x_t) \leq \mathbf{0}_m$ at each round, the constraint should only be satisfied in the long run. More specifically, the constraint violation

$$\Big\|\Big[\sum_{t=1}^{T} g(x_t)\Big]_+\Big\| \tag{1.7}$$

should grow sublinearly. In this case, $\sum_{t=1}^{T} x_t/T \in \mathbb{X}$ as $T \to \infty$. In other words, the learner is allowed sometimes to make decisions that do not belong to $\mathbb{X}$, but the overall sequence of chosen decisions must obey the constraint at the end by a vanishing convergence rate. This problem is normally solved by online primal–dual algorithms [169–172]. The problem can be extended to the case when the constraint function is time-varying and revealed to the learner after her decision is chosen [173–176].

Not only centralized, but also distributed online convex optimization problems have been studied. For example, distributed unconstrained online optimization problems have been considered in [177] by proposing an online subgradient descent algorithm with proportional-integral disagreement and in [178] by designing a distributed online subgradient push-sum algorithm. Some other variations of distributed online convex optimization algorithms have also been proposed, e.g., the Nesterov based primal–dual algorithm [179], variant of the Arrow–Hurwicz saddle point algorithm [180], the mirror descent algorithm

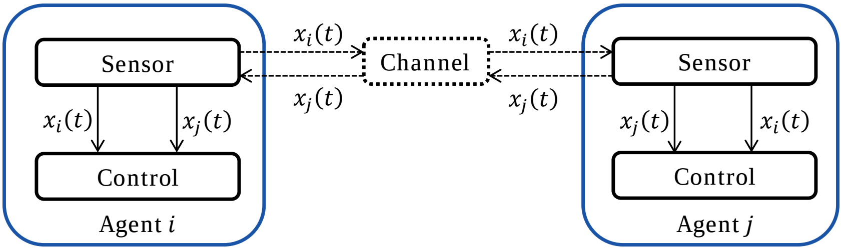


Figure 1.9: Illustration of how agents communicate when the control input is continuous.

[181], and the dual subgradient averaging algorithm [182]. For more studies on distributed online convex optimization, we refer to [183–191]. There are open problems on distributed online convex optimization. For instance, how to handle time-varying constraints. It is also interesting to develop gradient-free online algorithms, such as bandit online algorithms.

1.2.3 Distributed event-triggered control

Consider the continuous-time multi-agent systems described by integral dynamics

$$\dot{x}_i(t) = u_i(t), \ i \in [n], \ t \geq t_0, \tag{1.8}$$

$$u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t), \tag{1.9}$$

where $x_i \in \mathbb{R}^p$ is the state of agent i, which might represent physical variables such as attitude, position, temperature, or voltage, u_i is the control input, t_0 is a common initial time, and L_{ij} is the element of the Laplacian matrix of the underlying communication network $\mathcal{G}$. Such a system with two agents is illustrated by Figure 1.9. Each agent has a sensor component to measure and broadcast its state information, and to listen to and receive its neighbors' state information. Each agent also has a component to generate the control input based on the information it receives from the sensor.

To implement the control (1.9), continuous-time state information from neighbors is needed. In other words, each agent i has to continuously broadcast its own state $x_i(t)$, and continuously listen to and receive its neighbors' states $x_j(t), \ j \in \mathcal{N}_i$. Moreover, each agent i has to continuously update its control input $u_i(t) = \sum_{j=1}^{n} L_{ij} x_j(t)$. It is in most applications impractical to require continuous communications and updating of control inputs.

Reducing the frequency of information exchange among agents is essential. In order to realize this, we introduce a model where each agent $i \in [n]$ prefers to only broadcast its state at discrete time instants $\{t_1^i, t_2^i, \dots\}$. In this case, the state information received by agent i is $\{x_j(t_k^j), \ j \in \mathcal{N}_i\}_{k=1}^{\infty}$. In other words, at any t, agent i knows $x_j(t_{k_j(t)}^j), \ j \in \mathcal{N}_i$, where $t_{k_j(t)}^j = \max\{t_k^j : \ t_k^j \leq t\}$ is the latest broadcasting time of agent j. Then, the control input is

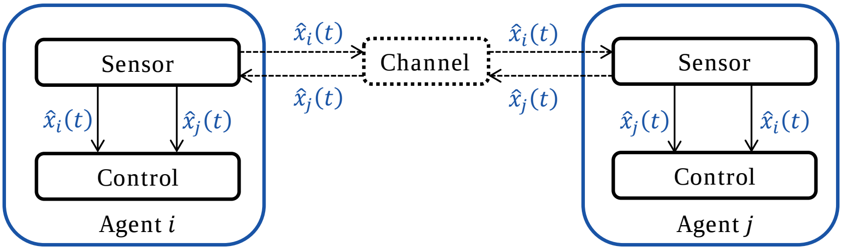


Figure 1.10: Illustration of how agents communicate when the control input is event-triggered.

computed as

$$u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t^j_{k_j(t)}). \tag{1.10}$$

For simplicity, let $\hat{x}_i(t) = x_i(t^i_{k_i(t)})$. Figure 1.10 shows that agent i broadcasts its state $x_i(t^i_k)$ at time instants $\{t^i_k\}_{k=1}^{\infty}$ and receives its neighbors' states $\{x_j(t^j_k),\ j \in \mathcal{N}_i\}$ at time instants $\{t^j_k,\ j \in \mathcal{N}_i\}_{k=1}^{\infty}$. An essential question is how to determine the communication instances $\{t^i_k,\ i \in [n]\}_{k=1}^{\infty}$ such that desired closed-loop system properties are maintained. In the literature, researchers often consider time-triggered, event-triggered, and self-triggered communications. We discuss each one of them next.

Time-triggered communication

The traditional way for agents to share information is to communicate equidistantly (periodically), i.e.,

$$t^i_1 = t_0,\ t^i_{k+1} = t^i_k + T_0,\ i \in [n],$$

where $T_0 > 0$ is the sampling period. This is called time-triggered or periodic sampling. Note that the triggering sequence is the same for each agent. A nice feature of this approach is that the analysis and design becomes rather straightforward and the vast literature on sample-data control can be used [192]. Drawbacks are that agents need to take actions in a synchronous manner (which is often hard to implement for large distributed systems) and it is not energy-efficient to communicate even if the state has not changed.

Event-triggered communication

To make the sampling period T_0 adaptive, we can let communication occur only when a predefined condition is satisfied. This is called event-triggered communication and the control input (1.10) with event-triggered communication is called distributed event-triggered control. Triggering times $\{t^i_1, t^i_2, \dots\}$ are in this case different for different agents. We call $\{t^i_{k+1} - t^i_k\}_{k=1}^{\infty}$ the inter-event times of agent i. Advantages of event-triggered approaches are that they can be implemented in a distributed manner and can sometimes give better performance than periodic sampling. However, the analysis and design methodologies are less developed.

One common choice of event-triggered communication is to use a triggering law defined by

$$t^i_1 = t_0,\ t^i_{k+1} = \min\{t:\ F_i(x_i(t), \hat{x}_i(t), \{x_j(t), \hat{x}_j(t)\}_{j\in\mathcal{N}_i}) \geq 0,\ t \geq t^i_k\},\ i \in [n], \tag{1.11}$$

where $F_i(\cdot)$ is a function to be designed. We call (1.11) a static triggering law since it does not involve any extra dynamic variables. There are two well-known ways to define the function $F_i(\cdot)$. The first one was introduced in [193]:

$$F_i(\cdot) = (\hat{x}_i(t) - x_i(t))^2 - \frac{\sigma_i a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\Big(\sum_{j=1}^{n}(x_j(t) - x_i(t))\Big)^2, \tag{1.12}$$

and the second one in [194]:

$$F_i(\cdot) = (\hat{x}_i(t) - x_i(t))^2 - \frac{\sigma_i a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|}\Big(\sum_{j=1}^{n}(\hat{x}_j(t) - \hat{x}_i(t))\Big)^2, \tag{1.13}$$

where $0 < \sigma_i < 1$ and $0 < a < \frac{1}{|\mathcal{N}_i|}$ are design parameters. The function $F_i(\cdot)$ in (1.12) and (1.13) do not involve any extra dynamic variables but the agent state variables $x_i(t), \hat{x}_i(t)$ and $x_j(t), \hat{x}_j(t)$, $j \in \mathcal{N}_i$.

Another common form of event-triggered communication is

$$t^i_1 = t_0,\ t^i_{k+1} = \min\{t:\ F_i(x_i(t), \hat{x}_i(t), \{x_j(t), \hat{x}_j(t)\}_{j\in\mathcal{N}_i}) \geq \eta_i(t),\ t \geq t^i_k\},\ i \in [n], \tag{1.14}$$

where $\eta_i(t)$ is an internal dynamic variable to be defined. We call (1.14) a dynamic triggering law since it involves an extra dynamic variable. One well-known dynamic triggering law introduced in [195] is

$$t^i_1 = t_0,\ t^i_{k+1} = \min\{t:\ |\hat{x}_i(t) - x_i(t)| \geq c_0 + c_1 e^{-\alpha t},\ t \geq t^i_k\},\ i \in [n] \tag{1.15}$$

with constants $c_0 \geq 0, c_1 \geq 0,\ c_0 + c_1 > 0$, and $0 < \alpha < \rho_2(L)$, where $\rho_2(L)$ is the minimum positive eigenvalue of the Laplacian matrix L of the underlying undirected graph $\mathcal{G}$.

Self-triggered communication

For event-triggered communication, each agent needs to continuously monitor the triggering laws. However, agent i could instead at its current triggering time t^i_k predict its next triggering time t^i_{k+1} and broadcast it to its neighbors. In this case, agent i only needs to listen and receive information at $\{t^j_k\}_{k=1}^{\infty}$, $j \in \mathcal{N}_i$ since it knows when these time instances will happen in advance. Each agent broadcasts at its own triggering times, and listen to incoming information from its neighbors at their triggering times. This is called self-triggered communication. Note that it is at the current triggering time instant that next triggering time is determined.

One common form of self-triggered communication is to use a triggering law defined by

$$t_1^i = t_0,\ t_{k+1}^i = \min\Big\{t:\ G_i\Big(t, x_i(t_k^i), t_k^i, \Big\{t^j_{k_j(t_k^i)}, t^j_{k_j(t_k^i)+1}, x_j(t^j_{k_j(t_k^i)})\Big\}_{j\in N_i}\Big) = 0,\ t \geq t_k^i\Big\},\ i \in [n], \tag{1.16}$$

where $G_i(\cdot)$ is a function to be designed, which is often chosen related to the function $F_i(\cdot)$ in the event-triggered communication.

Although there are numerous results on distributed event-triggered control in existing literature, there still remain some key challenges. For example, one key challenge is to exclude Zeno behavior when designing the triggering laws. Zeno is the behavior that there are infinite number of triggers in a finite time interval [196], i.e., that for some i

$$\lim_{k\to+\infty} t_k^i < \infty. \tag{1.17}$$

In other words, the non-existence of Zeno behavior is equivalent to that in every finite time interval there are only finite number of triggers. Thus, if Zeno behavior does not happen, it is guaranteed that during every finite time interval, the inter-event times are greater than a positive constant. Another challenge is to take into account resource constraints, such as energy, communication, sensing, and control constraints, which normally appear in applications. Resource constraints are essential for the control design of multi-agent systems as a constrained system can have completely different behavior compared to the unconstrained one. Therefore it is important to mathematically model resource-constrained multi-agent systems and to properly design their control laws such that a common objective is achieved while resource constraints are satisfied.

1.3 Problem formulation

In this section, we introduce the problems considered in this thesis, which can be categorized into three topics.

Distributed nonconvex optimization

The first considered problem is distributed nonconvex optimization. Specifically, consider a network of n agents, each of which has a local smooth (possibly nonconvex) cost function $f_i : \mathbb{R}^p \to \mathbb{R}$. All agents collaboratively solve the optimization problem

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x).$$

Each agent i only has information about its local cost function f_i. It can communicate with its neighbors through the underlying communication network which is modeled by an undirected graph. Different settings on the information feedback are investigated.

We first consider the case where full-information feedback is available. The problem to solve is to design (i) a distributed first-order algorithm such that linear convergence can be achieved without the strong convexity assumption; and (ii) a distributed ADMM algorithm that not only is suitable for arbitrarily connected communication networks, but also has linear convergence without the strong convexity assumption on the cost function.

We then consider the case where stochastic gradient feedback is available. The problem to solve is to design a distributed SGD algorithm that not only is suitable for arbitrarily connected communication networks, but also achieves linear speedup.

We finally consider the case where zeroth-order (ZO) oracle feedback is available. The problem to solve is to design (i) distributed algorithms based on deterministic zeroth-order (DZO) oracle feedback such that it has the same convergence properties as its first-order counterpart; and (ii) distributed algorithms based on stochastic zeroth-order (SZO) oracle feedback such that they not only are suitable for arbitrarily connected communication networks, but also achieve linear speedup.

Distributed online convex optimization

The second considered problem is distributed online convex optimization with time-varying coupled inequality constraints under different settings on the information feedback.

We first consider the full-information feedback setting. Specifically, consider a network of n agents indexed by $i \in [n]$. For each i, let the local decision set $\mathbb{X}_i \subseteq \mathbb{R}^{p_i}$ be a closed convex set with p_i being a positive integer. Let $\{f_{i,t} : \mathbb{X}_i \to \mathbb{R}\}$, $\{r_{i,t} : \mathbb{X}_i \to \mathbb{R}\}$, and $\{g_{i,t} : \mathbb{X}_i \to \mathbb{R}^m\}$ be arbitrary sequences of local convex cost, regularization, and constraint functions over time $t = 1, 2, \ldots$, respectively, where m is a positive integer. At time t, each agent i selects a decision $x_{i,t} \in \mathbb{X}_i$. After the selection, the agent receives its cost function $f_{i,t}$ and regularization $r_{i,t}$ together with its constraint function $g_{i,t}$, and obtains the loss $l_{i,t}(x_{i,t}) = f_{i,t}(x_{i,t}) + r_{i,t}(x_{i,t})$. At the same moment, the agents exchange data with their neighbors over a time-varying directed graph $\mathcal{G}_t$. The problem to solve is to develop distributed online optimization algorithms with guaranteed performance measured by the regret and constraint violation.

We also consider the bandit feedback setting, i.e., only the values of cost and constraint functions are revealed at the sampling instance. In this case, the problem can be defined as a repeated game between a group of n learners indexed by $i \in [n]$ and an adversary. At round t of the game, the adversary first arbitrarily chooses n local loss functions $\{f_{i,t} : \mathbb{X}_i \to \mathbb{R},\ i \in [n]\}$ and n local constraint functions $\{g_{i,t} : \mathbb{X}_i \to \mathbb{R}^m,\ i \in [n]\}$, where each $\mathbb{X}_i \subseteq \mathbb{R}^{p_i}$ is a known closed convex set with p_i and m being positive integers. Then, without knowing $\{f_{i,t},\ i \in [n]\}$ and $\{g_{i,t},\ i \in [n]\}$, all learners simultaneously choose their decisions $\{x_{i,t} \in \mathbb{X}_i,\ i \in [n]\}$. Each learner i samples the values of $f_{i,t}$ and $g_{i,t}$ at the point $x_{i,t}$ as well as at other potential points, i.e., the learners receive bandit feedback from the adversary. These values are held privately by each learner. At the same moment, the learners exchange data with their neighbors over a time-varying directed graph $\mathcal{G}_t$. The problem to solve is to develop distributed bandit online optimization algorithms with guaranteed performance measured by expected regret and constraint violation.

Distributed event-triggered control

The third and final problem is how to achieve consensus and formation control for multi-agent systems under limited communication resource constraint.

We first consider the average consensus problem for first-order continuous-time multi-agent systems with event-triggered control input over undirected graphs, i.e.,

$$\dot{x}_i(t) = u_i(t),\ i \in [n],\ t \geq t_0,$$
$$u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t^j_{k_j(t)}).$$

The problem to solve is to distributively determine the triggering times such that average consensus is reached, while continuous exchange of information, continuous update of actuators, and Zeno behavior are avoided.

We then consider the global consensus problem for multi-agent systems with input saturation over directed graphs, i.e.,

$$\dot{x}_i(t) = \mathrm{sat}_h(u_i(t)),\ i \in [n],\ t \geq t_0,$$
$$u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t) \text{ or } u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t^j_{k_j(t)}),$$

where $\mathrm{sat}_h(\cdot)$ is the saturation function with saturation level $h > 0$. For any $s = \mathrm{col}(s_1, \dots, s_p) \in \mathbb{R}^p$, the saturation function $\mathrm{sat}_h(s)$ is defined (with slight abuse of notation) as

$$\mathrm{sat}_h(s) = \mathrm{col}(\mathrm{sat}_h(s_1), \dots, \mathrm{sat}_h(s_p)),$$

where

$$\mathrm{sat}_h(s_i) = \begin{cases} h, & \text{if } s_i \geq h, \\ s_i, & \text{if } |s_i| < h, \\ -h, & \text{if } s_i \leq -h. \end{cases}$$

The problem to solve is to find sufficient and necessary connectivity conditions to guarantee that global consensus is reached, again, under the assumption that there are no continuous communication or system updates.

We finally consider formation control with connectivity preservation for both first-order multi-agent systems

$$\dot{x}_i(t) = u_i(t),\ i \in [n],\ t \geq t_0,$$

and second-order multi-agent systems

$$\begin{cases} \dot{x}_i(t) = r_i(t), \\ \dot{r}_i(t) = u_i(t),\ i \in [n],\ t \geq t_0. \end{cases}$$

We assume that the communication network is undirected and all agents have limited communication radius. The problem to solve is to design distributed event-triggered controllers together with triggering laws such that a desired formation is achieved while connectivity is preserved.

1.4 Thesis outline and contributions

In this section, we provide the outline of the thesis and indicate the contributions of each chapter.

1.4.1 Outline and contributions

The main results of this thesis are presented in Chapters 3–10 and are categorized into three parts according to their topics.

Part I focuses on distributed nonconvex optimization problems, which are known to be hard in general. Normally only stationary points can be found, which could be local maxima or minima. In this part, we propose algorithms to solve these problems under different information feedback settings. Part I consists of Chapters 3–5 and an overview of this part is provided in Table 1.1. The rows list the problem settings and convergence results. Firstly, when full-information feedback is available, we show in Chapter 3 that a stationary point can be found by the proposed primal–dual first-order and ADMM algorithms with an $\mathcal{O}(1/T)$ convergence rate if each local cost function is smooth, and that a global optimum can be found with a linear convergence rate under an additional condition that the global cost function satisfies the Polyak-Łojasiewicz (P–Ł) condition. This condition is weaker than strong convexity, which is a standard condition for proving linear convergence of distributed optimization algorithms, and the global minimizer is not necessarily unique. Secondly, when stochastic gradient feedback is available, we show in Chapter 4 that the proposed primal–dual SGD algorithm achieves the linear speedup convergence rates $\mathcal{O}(1/\sqrt{nT})$ and $\mathcal{O}(1/(nT))$ without and with the P–Ł condition, respectively. Thirdly, when DZO oracle feedback is available, we show in Chapter 5 that the proposed primal-dual DZO algorithm achieves the same convergence results as its first-order counterpart in Chapter 3. Finally, when SZO oracle feedback is available, we show in Chapter 5 that the proposed primal-dual and dual SZO algorithms achieve the linear speedup convergence rates $\mathcal{O}(\sqrt{p/nT})$ and $\mathcal{O}(p/(nT))$ without and with the P–Ł condition, respectively.

Part II focuses on online convex optimization problems, which view optimization as a process or a repeated game. In this part, we propose algorithms to solve distributed online convex optimization problems with time-varying coupled inequality constraints under different settings on the information feedback. The main difference between Parts I and II is that in Part I the cost functions are fixed, while in Part II the cost and constraint functions are time-varying and revealed at the end of each time slot. In addition, Part II deals with constrained problems and they have to be convex. Part II consists of Chapters 6–7 and an overview of this part is provided in Table 1.2. The rows list the problem settings and convergence results. Firstly, when full-information feedback is available, we show

Table 1.1: Overview of Part I of this thesis.

	Chapter 3 (Algorithms 3.1–3.3)	Chapter 4 (Algorithm 4.1)	Chapter 5 (Algorithm 5.1)	Chapter 5 (Algorithms 5.2–5.3)
Considered problem	Distributed nonconvex optimization			
Communication network	Undirected connected			
Information feedback	Full-information	Stochastic gradient	DZO oracle	SZO oracle
Convergence rate with the smooth assumption	$O(1/T)$	$O(1/\sqrt{nT})$	$O(1/T)$	$O(\sqrt{p/(nT)})$
Convergence rate with the smooth and P–Ł condition assumptions	Linear	$O(1/(nT))$	Linear	$O(p/(nT))$

Table 1.2: Overview of Part II of this thesis.

	Chapter 6 (Algorithm 6.1)	Chapter 7 (Algorithm 7.1)	Chapter 7 (Algorithm 7.2)
Considered problem	Distributed online convex optimization with time-varying coupled inequality constraints		
Communication network	Time-varying, directed, uniformly jointly strongly connected		
Information feedback	Full-information	One-point bandit	Two-point bandit
Dynamic regret bound	$O(\max\{T^{\kappa}\sum_{t=1}^{T-1}\|x^*_{t+1}-x^*_t\|, T^{\max\{1-\kappa,\kappa\}}\})$, where $\kappa\in(0,1)$	$O(\max\{T^{\theta_1}\sum_{t=1}^{T-1}\|x^*_{t+1}-x^*_t\|, T^{\theta_1}\})$, where $\theta_1\in(3/4,5/6]$	$O(\max\{T^{\kappa}\sum_{t=1}^{T-1}\|x^*_{t+1}-x^*_t\|, T^{\max\{\kappa,1-\kappa\}}\})$
Constraint violation bound	$O(T^{1-\kappa/2})$; $O(T^{\max\{1-\kappa,\kappa\}})$ if Slater's condition holds	$O(T^{7/4-\theta_1})$	$O(T^{1-\kappa/2})$

in Chapter 6 that sublinear dynamic regret and constraint violation can be achieved by the proposed algorithm if the accumulated dynamic variation of the optimal sequence $\sum_{t=1}^{T-1}\|x^*_{t+1}-x^*_t\|$ grows sublinearly, where $\{x^*_t\}$ is the optimal dynamic decision sequence. Moreover, the constraint violation bound can be reduced when Slater's condition holds. Secondly, when one-point bandit feedback is available, we show in Chapter 7 that the proposed algorithm achieves larger dynamic regret and constraint violation bounds than the bounds achieved in Chapter 6, but they are still sublinear if the accumulated variation of the comparator sequence also grows sublinearly. Finally, when two-point bandit feedback is available, we show in Chapter 7 that the proposed algorithm achieves the same dynamic regret and constraint violation bounds as its full-information counterpart in Chapter 6.

Part III focuses on distributed event-triggered control problems. In this part, we propose distributed dynamic event-triggered control algorithms to solve consensus and formation problems for multi-agent systems under limited communication resources. Part III consists of Chapters 8–10 and an overview of this part is provided in Table 1.3. The rows list the problem settings and convergence results. In Chapter 8, we consider the average consensus problem for first-order multi-agent systems over undirected connected communication networks. In Chapter 9, we consider the global consensus problem for first-order multi-agent systems with input saturation over directed communication networks containing directed spanning trees. In Chapter 10, we consider the formation control problem for

Table 1.3: Overview of Part III of this thesis.

	Chapter 8	Chapter 9	Chapter 10
Considered problem	Average consensus for first-order multi-agent systems	Global consensus for multi-agent systems with input saturation	Formation control for multi-agent systems with connectivity preservation
Communication network	Undirected connected	Directed spanning tree	Undirected connected
Information type	Absolute		Relative
Algorithm	Distributed dynamic event-triggered control algorithms without Zeno behavior		
Convergence rate	Exponential		

first- and second-order multi-agent systems with connectivity preservation over undirected connected communication networks, and relative state information is used to design the control input. Distributed dynamic event-triggered control algorithms without Zeno behavior are proposed to solve these problems and exponential convergence rates are established.

The overall outline of the remainder of this thesis and its technical contributions are summarized in the following.

Chapter 2: Preliminaries

In Chapter 2, we list some essential elements of the background theory, including algebraic graph theory, convex functions, projections, smooth functions, Polyak–Łojasiewicz condition, Bregman divergence, gradient approximation, and some useful lemmas related to series, used in the thesis.

Chapter 3: Distributed primal–dual first-order and ADMM algorithms

In Chapter 3, we consider the distributed nonconvex optimization problem with full-information feedback. We propose three algorithms: a distributed primal–dual first-order algorithm, a distributed ADMM algorithm, and a distributed linearized ADMM algorithm. We show that each algorithm converges to a stationary point with an $O(1/T)$ convergence rate if each local cost function is smooth, where T is the total number of iterations, and to a global optimum with a linear convergence rate under an additional condition that the global cost function satisfies the P–Ł condition. This condition is weaker than strong convexity, which is a standard condition in the literature for proving linear convergence of distributed optimization algorithms, and the global minimizer is not necessarily unique or finite.

The covered material is based on the following contributions.

[C1] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Linear convergence for distributed optimization without strong convexity," *in IEEE Conference on Decision and Control*, 2020.

[J1] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Linear convergence of first- and zeroth-order primal–dual algorithms for distributed nonconvex optimization," *submitted to IEEE Transactions on Automatic Control.*

[M1] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Linear convergence of the alternating direction method of multipliers for distributed nonconvex optimization," *in preparation.*

Chapter 4: Distributed primal–dual SGD optimization algorithm

In Chapter 4, we consider the distributed nonconvex optimization problem with stochastic gradient feedback. We propose a distributed primal–dual SGD algorithm, suitable for arbitrarily connected communication networks and any smooth cost functions. We show that the proposed algorithm converges to a stationary point with the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ for smooth nonconvex cost functions, and to a global optimum with the linear speedup convergence rate $\mathcal{O}(1/(nT))$ when the global cost function satisfies the P–Ł condition in addition, where n and T are the number of agents and the total number of iterations, respectively. We also show that the output of the proposed algorithm with constant parameters linearly converges to a neighborhood of a global optimum.

The covered material is based on the following contribution.

[J2] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "A primal–dual SGD algorithm for distributed nonconvex optimization," *submitted to SIAM Journal on Control and Optimization.*

Chapter 5: Distributed zeroth-order optimization algorithms

In Chapter 5, we consider the distributed nonconvex optimization problem with ZO oracle feedback. We first consider the situation that DZO oracle feedback is available. We propose a distributed primal–dual DZO algorithm and show that it converges to a stationary point with an $\mathcal{O}(1/T)$ convergence rate for smooth nonconvex cost functions, and to a global optimum with a linear convergence rate when the global cost function satisfies the P–Ł condition in addition, where T is the total number of iterations. In other words, our proposed distributed DZO algorithm has the same convergence properties as its FO counterpart under the same conditions. We then consider the situation that SZO oracle feedback is available. We propose two distributed SZO algorithms: distributed primal–dual and dual SZO algorithms. We show that both SZO algorithms converge to a stationary point with the linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ for smooth nonconvex cost functions, and to a global optimum with the linear speedup convergence rate $\mathcal{O}(p/(nT))$ when the global cost function satisfies the P–Ł condition in addition, where p is the dimension of the decision variable. We also show that both SZO algorithms converge linearly when considering deterministic centralized optimization problems under the P–Ł condition.

The covered material is based on the following contribution.

[J1] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Linear convergence of first- and zeroth-order primal–dual algorithms for distributed nonconvex optimization," *submitted to IEEE Transactions on Automatic Control.*

[M2] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Zeroth-order algorithms for stochastic distributed nonconvex optimization," *in preparation*.

Chapter 6: Distributed online primal–dual optimization algorithm

In Chapter 6, we consider distributed online convex optimization with time-varying coupled inequality constraints. The global objective function is composed of local convex cost and regularization functions and the coupled constraint function is the sum of local convex functions. A distributed online primal–dual dynamic mirror descent algorithm is proposed to solve this problem, where the local cost, regularization, and constraint functions are held privately and revealed only after each time slot. Without assuming Slater's condition, we first derive regret and constraint violation bounds for the proposed algorithm and show how they depend on the stepsize sequences, the accumulated dynamic variation of the comparator sequence, the number of agents, and the network connectivity. As a result, under some natural decreasing stepsize sequences, we prove that the proposed algorithm achieves sublinear dynamic regret and constraint violation if the accumulated dynamic variation of the optimal sequence also grows sublinearly. In particular, we show that it achieves $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$ static regret and $\mathcal{O}(T^{1-\kappa/2})$ constraint violation bounds, where $\kappa \in (0, 1)$ is a user-defined trade-off parameter. Assuming Slater's condition, we show that the dynamic regret bound is similar to the bound without assuming Slater's condition, but the constraint violation bound can be reduced to $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$. Moreover, we show that both static regret and constraint violation bounds grow as $\mathcal{O}(\sqrt{T})$. In addition, smaller bounds on the static regret are achieved when the objective function is strongly convex.

The covered material is based on the following contributions.

[C2] X. Yi, X. Li, L. Xie, and K. H. Johansson, "A distributed algorithm for online convex optimization with time-varying coupled inequality constraints," *in IEEE Conference on Decision and Control*, 2019, pp. 555–560.

[J3] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *in IEEE Transactions on Signal Processing*, vol. 68, no. 2, pp. 731–746, 2020.

Chapter 7: Distributed bandit online primal–dual optimization algorithms

In Chapter 7, we consider distributed bandit online convex optimization with time-varying coupled inequality constraints, motivated by a repeated game between a group of learners and an adversary. The learners attempt to minimize a sequence of global loss functions and at the same time satisfy a sequence of coupled constraint functions, where the constraints are coupled across the distributed learners at each round. The global loss and the coupled constraint functions are the sum of local convex loss and constraint functions, respectively, which are adaptively generated by the adversary. The local loss and constraint functions are revealed in a bandit manner, i.e., only the values of loss and constraint functions are revealed to the learners at the sampling instance, and the revealed function values are held privately by each learner. Both one- and two-point bandit feedback are studied with the two corresponding distributed bandit online algorithms used by the learners. We show that sublinear expected dynamic regret and constraint violation are achieved by these two algorithms, if the accumulated variation of the comparator sequence also grows sublinearly. In particular, we show that $\mathcal{O}(T^{\theta_1})$ expected static regret and $\mathcal{O}(T^{7/4-\theta_1})$ constraint violation bounds are achieved in the one-point bandit feedback setting, and $\mathcal{O}(T^{\max\{\kappa,1-\kappa\}})$ expected static regret and $\mathcal{O}(T^{1-\kappa/2})$ constraint violation bounds in the two-point bandit feedback setting, where T is the total number of rounds and $\theta_1 \in (3/4, 5/6]$ and $\kappa \in (0, 1)$ are user-defined trade-off parameters.

The covered material is based on the following contributions.

[C3] X. Yi, X. Li, T. Yang, L. Xie, T. Chai, and K. H. Johansson, "A distributed primal–dual algorithm for bandit online convex optimization with time-varying coupled inequality constraints," *in American Control Conference*, 2020, pp. 327–332.

[J4] X. Yi, X. Li, T. Yang, L. Xie, T. Chai, and K. H. Johansson, "Distributed bandit online convex optimization with time-varying coupled inequality constraints," *submitted to IEEE Transactions on Automatic Control.*

Chapter 8: Distributed dynamic event-triggered control algorithms

In Chapter 8, we propose dynamic event-triggered approaches to solve the average consensus problem for first-order continuous-time multi-agent systems over undirected graphs. More specifically, two distributed dynamic triggering laws and one self-triggered algorithm are proposed to determine the triggering times. Compared with existing triggering laws, the proposed triggering laws involve internal dynamic variables which play an essential role in guaranteeing that the triggering time sequence does not exhibit Zeno behavior. Moreover, our dynamic triggering laws include some existing triggering laws as special cases. More importantly, continuous listening is avoided in our proposed self-triggered algorithm. The main idea is that each agent predicts its next triggering time and broadcasts it to its neighbors at the current triggering time. Thus each agent only needs to sense and broadcast at its triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. It is proven that the proposed laws ensure that the state of each agent converge exponentially to the average of the agents' initial states if and only if the underlying graph is connected.

The covered material is based on the following contributions.

[C4] X. Yi, K. Liu, D. V. Dimarogonas and K. H. Johansson, "Distributed dynamic event-triggered control for multi-agent systems," *in IEEE Conference on Decision and Control*, 2017, pp. 6683–6688.

[J5] X. Yi, K. Liu, D. V. Dimarogonas and K. H. Johansson, "Dynamic event-triggered and self-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3300–3307, 2019.

Chapter 9: Distributed event-triggered saturation control algorithms

In Chapter 9, we consider the global consensus problem for multi-agent systems with input saturation over directed graphs. It is shown that the underlying directed graph having a directed spanning tree is a necessary and sufficient condition for consensus; thus, this condition for consensus without input saturation extends to the case with saturation constraints. Moreover, in order to reduce the overall need of communication and system updates, we then propose an event-triggered consensus protocol and a triggering law, which do not require any a priori knowledge of global network parameters. Furthermore, in order to avoid continuous listening, we also propose a self-triggered algorithm. It is shown that Zeno behavior is excluded for these systems and that consensus is achieved, again, if and only if the underlying directed graph has a directed spanning tree. We use a new Lyapunov function to show the sufficient condition and it inspires the triggering law.

The covered material is based on the following contribution.

[J6] X. Yi, T. Yang, J. Wu, and K. H. Johansson, "Distributed event-triggered control for global consensus of multi-agent systems with input saturation," *Automatica*, vol 100, no. 2, pp. 1–9, 2019.

Chapter 10: Distributed event-triggered formation control algorithms

In Chapter 10, event- and self-triggered control algorithms are proposed to establish prespecified formations with connectivity preservation. Each agent only needs to update its control input by sensing the relative state to its neighbors and to broadcast its triggering information at its own triggering times. The agents listen to and receive neighbors' triggering information at their triggering times. Two types of system dynamics, single integrators and double integrators, are considered. It is shown that all agents converge to the prespecified formation exponentially with connectivity preservation and exclusion of Zeno behavior.

The covered material is based on the following contributions.

[C5] X. Yi, J. Wei, D. V. Dimarogonas, and K. H. Johansson, "Formation control for multi-agent systems with connectivity preservation and event-triggered controllers," *in IFAC World Congress*, 2017, pp. 9367–9373.

Chapter 11: Conclusions and future research

In Chapter 11, we present a summary of the results and discuss directions for future research.

The results presented in Part III in this thesis have previously appeared in the following thesis:

- X. Yi, *Resource-constrained multi-agent control systems: Dynamic event-triggering, input saturation, and connectivity preservation,* Licentiate thesis, KTH Royal Institute of Technology, 2017.

1.4.2 Contributions not covered in this thesis

The following works by the author are not covered in this thesis, but contain related material:

[J7] J. Wu, B. Mu, X. Yi, J. Wei, and K. H. Johansson, "Localizability with range-difference measurements," *submitted to IEEE/ACM Transactions on Networking*.

[J8] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *submitted to IEEE Transactions on Control of Network Systems*.

[J9] X. Li, X. Yi, and L. Xie, "Distributed online optimization for multi-agent networks with coupled inequality constraints," *IEEE Transactions on Automatic Control*, to appear.

[J10] T. Yang, J. George, J. Qin, X. Yi, and J. Wu, "Distributed least squares solver for network linear equations," *Automatica*, vol. 113, 2020.

[J11] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[J12] J. Wei, X. Yi, H. Sandberg, and K. H. Johansson, "Nonlinear consensus protocols with applications to quantized communication and actuation," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 598–608, 2018.

[C6] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Exponential convergence for distributed optimization under the restricted secant inequality condition," *in IFAC World Congress*, 2020.

[C7] X. Yi, S. Zhang, T. Yang, J. Wu, and K. H. Johansson, "Event-triggered control for consensus of multi-agent systems with nonlinear output and directed topologies," *in Chinese Control Conference*, 2019, pp. 5721–5726.

[C8] S. Zhang, X. Yi, J. George, and T. Yang, "Computational convergence analysis of distributed optimization algorithms for directed graphs," *in International Conference on Control and Automation*, 2019, pp. 1096–1101.

[C9] X. Yi, L. Yao, T. Yang, J. George, and K. H. Johansson "Distributed optimization for second-order multi-agent systems with dynamic event-triggered communication," *in IEEE Conference on Decision and Control*, 2018, pp. 3397–3402.

[C10] W. Du, X. Yi, J. George, K. H. Johansson, and T. Yang, "Distributed optimization with dynamic event-triggered mechanisms," *in IEEE Conference on Decision and Control*, 2018, pp. 969–974.

[C11] J. George, X. Yi, and T. Yang, "Distributed robust dynamic average consensus with dynamic event-triggered communication," *in IEEE Conference on Decision and Control*, 2018, pp. 434–439.

[C12] M. Jafarian, X. Yi, M. Pirani, H. Sandberg, and K. H. Johansson, "Synchronization of Kuramoto oscillators in a bidirectional frequency-dependent tree network," *in IEEE Conference on Decision and Control*, 2018, pp. 4505–4510.

[C13] X. Yi, T. Yang, J. Wu, and K. H. Johansson, "Event-triggered control for multi-agent systems with output saturation," *in Chinese Control Conference*, 2017, pp. 8431–8436.

[C14] J. Wei, X. Yi, H. Sandberg, and K. H. Johansson, "Nonlinear consensus protocols with applications to quantized systems," *in IFAC World Congress*, 2017, pp. 15440–15445.

[C15] X. Yi, J. Wei, and K. H. Johansson, "Self-triggered control for multi-agent systems with quantized communication or sensing," *in IEEE Conference on Decision and Control*, 2016, pp. 2227–2232.

Contribution by the Author

The order of authors reflects their contribution to each paper. The first author has the most important contribution, while the last author has taken a supervisory role. In all the listed publications, all the authors were actively involved in formulating the problems, developing the solutions, evaluating the results, and writing the paper.

Chapter 2

Preliminaries

This chapter gives some essential elements of the mathematical background to the results developed in the thesis, including algebraic graph theory, convex functions, projections, smooth functions, Polyak–Łojasiewicz condition, Bregman divergence, gradient approximation, and some useful lemmas related to series, used in the thesis. The related studies to each considered problem in this thesis are reviewed separately in the corresponding chapter.

2.1 Directed graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ denote a weighted directed graph (digraph), where $\mathcal{V} = [n]$ is the set of vertices (nodes), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (links), and $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ with nonnegative elements a_{ij} is the weighted adjacency matrix. An edge of $\mathcal{G}$ is denoted by $(i, j) \in \mathcal{E}$ if there is a directed link from vertex i to vertex j with weight $a_{ji} > 0$, i.e., vertex i can send information to vertex j. The adjacency elements associated with the edges of the graph are positive, i.e., $(i, j) \in \mathcal{E}$ if and only if $a_{ji} > 0$. It is assumed that $a_{ii} = 0,\ \forall i \in [n]$. The in-degree of vertex i is defined as $\deg_i^{\rm in} = \sum_{j=1}^{n} a_{ij}$. The in-degree matrix of $\mathcal{G}$ is defined as $\mathrm{Deg}^{\rm in} = \mathrm{Diag}([\deg_1^{\rm in}, \dots, \deg_n^{\rm in}])$. The (weighted) Laplacian matrix associated with $\mathcal{G}$ is defined as $L = \mathrm{Deg}^{\rm in} - A$. Let $\mathcal{N}_i^{\rm in} = \{j \in [n] \mid a_{ij} > 0\}$ and $\mathcal{N}_i^{\rm out} = \{j \in [n] \mid a_{ji} > 0\}$ denote the in- and out-neighbors of vertex i, respectively. The mixing matrix $W \in \mathbb{R}^{n\times n}$ associated with a digraph $\mathcal{G}$ fulfills $[W]_{ij} > 0$ if $(j, i) \in \mathcal{E}$ or $i = j$, and $[W]_{ij} = 0$ otherwise. When necessary, we use $\mathcal{E}(\mathcal{G})$, $L(\mathcal{G})$, $\mathcal{N}_i^{\rm in}(\mathcal{G})$, and $\mathcal{N}_i^{\rm out}(\mathcal{G})$ to highlight their connections with $\mathcal{G}$.

If a digraph is time-varying, then we use $\mathcal{G}_t$ to denote this time-varying digraph at time t. Similarly, let $\mathcal{E}_t$, A_t, L_t, W_t, $\mathcal{N}_i^{\rm in}(\mathcal{G}_t)$, and $\mathcal{N}_i^{\rm out}(\mathcal{G}_t)$ denote the edge set, the weighted adjacency matrix, the Laplacian matrix, the mixing matrix, the in-neighbors of vertex i, and out-neighbors of vertex i at time t, respectively.

A directed path from vertex i to vertex j is a directed subgraph of $\mathcal{G}$ with distinct vertices $i, i_1, \dots, i_k, j$ and edges $(i, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k), (i_k, j)$.

Definition 2.1 (Strongly connected digraph)**.** *A digraph $\mathcal{G}$ is strongly connected if there is* *at least one directed path from any vertex to any other vertex in the graph.*

$\mathcal{G}$ is strongly connected is equivalent to L is irreducible. Strong connectivity requires that any vertex is accessible to all other vertices, while the following weaker connectivity condition only requires that one vertex can access all other vertices.

Definition 2.2 (Directed spanning tree)**.** *A digraph $\mathcal{G}$ has a directed spanning tree if there exists one vertex such that there exists a directed path from this vertex to any other vertices.*

By Perron–Frobenius Theorem in [197], we have the following result (see [198] or [199] for a proof) for digraphs.

Lemma 2.1. *If L is the Laplacian matrix associated with a digraph $\mathcal{G}$ that has a directed spanning tree, then rank$(L) = n-1$, and zero is an algebraically simple eigenvalue of L, and there is a nonnegative vector $\xi = \mathrm{col}(\xi_1, \dots, \xi_n)$ such that $\xi^\top L = 0$ and $\sum_{i=1}^n \xi_i = 1$. Moreover, if $\mathcal{G}$ is strongly connected, then $\xi_i > 0,\ \forall i \in [n]$.*

The following result from [200] is also useful for our analysis.

Lemma 2.2. *Suppose that L is the Laplacian matrix associated with a digraph $\mathcal{G}$ that is strongly connected and ξ is the vector defined in Lemma 2.1. Let $\Xi = Diag(\xi)$, $U = \Xi - \xi\xi^\top$, and $R = \frac{1}{2}(\Xi L + L^\top \Xi)$. Then,*

$$R = \frac{1}{2}(UL + L^\top U),\ U \geq \frac{\rho_2(U)}{\rho(L^\top L)} L^\top L \geq 0,\ R \geq \frac{\rho_2(R)}{\rho(U)} U \geq 0. \tag{2.1}$$

By proper row and column permutations, any Laplacian matrix L can be written in Perron–Frobenius form (see Definition 2.3 in [201]):

$$L = \begin{bmatrix} L^{1,1} & L^{1,2} & \cdots & L^{1,M} \\ 0 & L^{2,2} & \cdots & L^{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L^{M,M} \end{bmatrix}, \tag{2.2}$$

where $L^{m,m}$ is a n_m-by-n_m matrix associated with the m-th strongly connected component (SCC) of $\mathcal{G}$, denoted by SCC_m, $m = 1, \dots, M$. Hence, a digraph $\mathcal{G}$ is strongly connected if and only if $M = 1$. In the following, without loss of generality, we assume that L has the form (2.2).

SCC_m is called closed if and only if there are no edges from vertices outside SCC_m to vertices inside SCC_m, i.e., $L^{m,q} = 0,\ \forall q > m$. The following result, which follows from Lemma 1 in [202], gives an equivalent description of a digraph that has a directed spanning tree.

Lemma 2.3. *The digraph $\mathcal{G}$ contains a directed spanning tree if and only if for each $m = 1, \dots, M-1$, SCC_m is not closed.*

Let us illustrate this construction with an example.

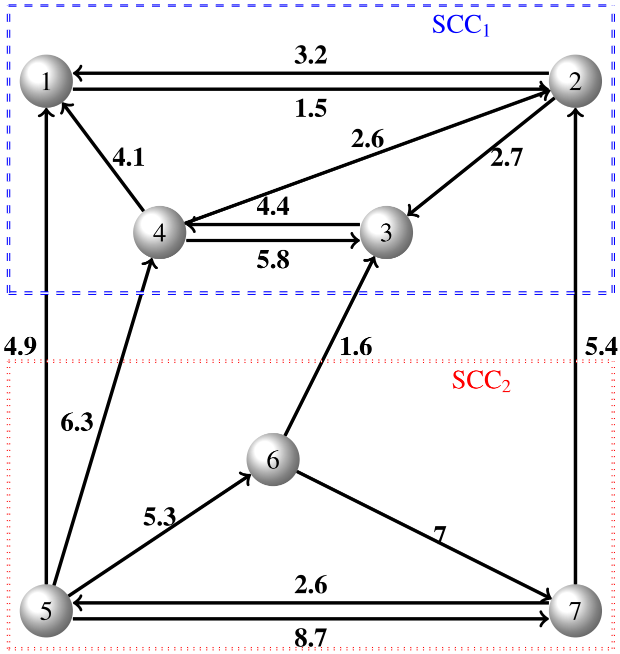


Figure 2.1: An example of a digraph which contains directed spanning trees.

Example 2.1. *Figure 2.1 shows a digraph of 7 vertices having multiple directed spanning trees. For example, edges* $(7,5),(5,6),(6,3),(3,4),(4,2),(2,1)$ *describe a directed spanning tree. This digraph can be divided into two strongly connected components. Specifically, the subgraph inside the dashed blue lines is the first strongly connected component, and the subgraph inside the dotted red lines is the second strongly connected component. The corresponding Laplacian matrix*

$$L = \left[\begin{array}{cccc|ccc} 12.2 & -3.2 & 0 & -4.1 & -4.9 & 0 & 0 \\ -1.5 & 9.5 & 0 & -2.6 & 0 & 0 & -5.4 \\ 0 & -2.7 & 10.1 & -5.8 & 0 & -1.6 & 0 \\ 0 & 0 & -4.4 & 10.7 & -6.3 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 2.6 & 0 & -2.6 \\ 0 & 0 & 0 & 0 & -5.3 & 5.3 & 0 \\ 0 & 0 & 0 & 0 & -8.7 & -7 & 15.7 \end{array}\right],$$

has the form (2.2).

For SCC_m with $m < M$, define an auxiliary matrix $\tilde{L}^{m,m} = [\tilde{L}_{ij}^{m,m}]_{i,j=1}^{n_m}$ as

$$\tilde{L}_{ij}^{m,m} = \begin{cases} L_{ij}^{m,m} & i \neq j, \\ -\sum_{r=1,r\neq i}^{n_m} L_{ir}^{m,m} & i = j. \end{cases}$$

Example 2.2. *In Example 2.1,*

$$\tilde{L}^{1,1} = \begin{bmatrix} 7.3 & -3.2 & 0 & -4.1 \\ -1.5 & 4.1 & 0 & -2.6 \\ 0 & -2.7 & 8.5 & -5.8 \\ 0 & 0 & -4.4 & 4.4 \end{bmatrix}.$$

Similar to Lemma 2.2, we have the following lemma.

Lemma 2.4. *Let $\xi^m = [\xi_1^m, \dots, \xi_{n_m}^m]^\top$ be the positive left eigenvector of the irreducible $\tilde{L}^{m,m}$ corresponding to the eigenvalue zero and the sum of its components is* 1. *Denote $\Xi^m = Diag(\xi^m)$, $Q^m = \frac{1}{2}[\Xi^m L^{m,m} + (\Xi^m L^{m,m})^\top]$, $\forall m \in [M]$, and $U^M = \Xi^M - \xi^M(\xi^M)^\top$. Then*

$$Q^m > 0,\ \forall m \in [M-1],\ Q^M \geq 0,\ U^M \geq 0,\ Q^M \geq \frac{\rho_2(Q^M)}{\rho(U^M)} U^M. \tag{2.3}$$

Proof. For the proof of $Q^m > 0$ for all $m < M$, see Lemma 3.1 in [203].

$Q^M \geq 0$ is straightforward since we can regard Q^M as the Laplacian matrix of a connected undirected graph.

$U^M \geq 0$ is also straightforward since we can regard U^M as the Laplacian matrix of a complete graph.

The idea of the proof of $Q^M \geq \frac{\rho_2(Q^M)}{\rho(U^M)} U^M$ follows a similar trend as the proof of (2.1), and it can be found in [200]. We thus omit the proof here. □

Let n_e denotes the number of edges in $\mathcal{G}$, i.e., $n_e = |\mathcal{E}(\mathcal{G})|$ and label the edges in $\mathcal{G}$ as $e_1, \dots, e_{n_e}$. Define $\Omega(\mathcal{G}) = \mathrm{Diag}([\omega(e_1), \cdots, \omega(e_{n_e})])$, where $\omega(e_k) = a_{ij}$ with e_k being the label of edge (i, j).

Definition 2.3 (Incidence matrix)**.** *The n-by-n_e incidence matrix $B(\mathcal{G}) = (B_{ij})$ is defined as*

$$\mathrm{B}_{ij} = \begin{cases} -1 & \textit{if vertex } i \textit{ is the tail of edge } e_j, \\ 1 & \textit{if vertex } i \textit{ is the head of edge } e_j, \\ 0 & \textit{otherwise.} \end{cases}$$

2.2 Undirected graphs

A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ is undirected if $A = A^\top$. In an undirected graph, a path of length k between vertex i and vertex j is a subgraph with distinct vertices $i_0 = i, \dots, i_k = j \in \mathcal{V}$ and edges $(i_j, i_{j+1}) \in \mathcal{E}$, $j = 0, \dots, k-1$.

Definition 2.4 (Connected undirected graph)**.** *An undirected graph is connected if there exists at least one path between any two vertices. An undirected graph is complete if any two distinct vertices are connected by an edge.*

Similar to the definition of SCC in digraphs, by proper row and column permutations, we can rewrite any Laplacian matrix L associated with undirected graphs in the following form

$$L = \begin{bmatrix} L^{1,1} & 0 & \cdots & 0 \\ 0 & L^{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L^{M,M} \end{bmatrix}, \tag{2.4}$$

where $L^{m,m}$ is a n_m-by-n_m matrix associated with the m-th connected component (CC) of $\mathcal{G}$, denoted by CC_m, $m = 1, \dots, M$. Hence, a disconnected graph has more than one CC and $L^{m,m}$ is the Laplacian matrix of CC_m.

Obviously, there is a one-to-one correspondence between a graph and its adjacency matrix or its Laplacian matrix. Denote $K_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, then we can treat K_n as the Laplacian matrix of a complete graph with n vertices and edge weight $\frac{1}{n}$.

For a connected graph we have the following results.

Lemma 2.5. *(Lemmas 1 and 2 in [204]) Let L be the Laplacian matrix of a connected undirected graph $\mathcal{G}$ and $K_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$. Then L and K_n are positive semi-definite,* $\mathrm{null}(L) = \mathrm{null}(K_n) = \mathbf{1}_n$*, $L \leq \rho(L)\boldsymbol{I}_n$, $\rho(K_n) = 1$,*

$$K_n L = L K_n = L, \tag{2.5}$$

$$0 \leq \rho_2(L) K_n \leq L \leq \rho(L) K_n. \tag{2.6}$$

Moreover, there exists an orthogonal matrix $[r\ R] \in \mathbb{R}^{n\times n}$ with $r = \frac{1}{\sqrt{n}}\mathbf{1}_n$ and $R \in \mathbb{R}^{n\times(n-1)}$ such that

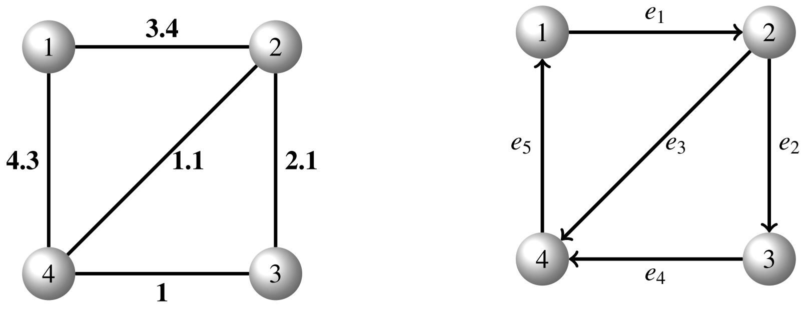
$$R\Lambda_1^{-1}R^\top L = LR\Lambda_1^{-1}R^\top = K_n, \tag{2.7}$$

$$\frac{1}{\rho(L)} K_n \leq R\Lambda_1^{-1}R^\top \leq \frac{1}{\rho_2(L)} K_n, \tag{2.8}$$

where $\Lambda_1 = Diag([\lambda_2, \dots, \lambda_n])$ with $0 < \lambda_2 \leq \cdots \leq \lambda_n$ being the eigenvalues of the Laplacian matrix L.

For undirected graphs, the incidence matrix can be defined after arbitrarily assigning a direction to each edge. The following results from [205] are also useful for our analysis.

Lemma 2.6. *For any undirected graph $\mathcal{G}$, $B(\mathcal{G})B(\mathcal{G})^\top$ is independent of the labels and orientations given to $\mathcal{G}$, and $B(\mathcal{G})\Omega(\mathcal{G})B(\mathcal{G})^\top = L$.*



(a) An example of an undirected graph $\mathcal{G}$.

(b) An example of assigning a direction to each edge of $\mathcal{G}$.

Figure 2.2: Illustration of assigning directions to an undirected graph.

Example 2.3. *Figure 2.2 (a) shows an undirected graph $\mathcal{G}$ and Figure 2.2 (b) shows an example of assigning a direction to each edge of $\mathcal{G}$. Then*

$$L = \begin{bmatrix} 3.4 & -3.4 & 0 & 0 \\ -3.4 & 9.8 & -2.1 & -4.3 \\ 0 & -2.1 & 3.2 & -1.1 \\ 0 & -4.3 & -1.1 & 5.4 \end{bmatrix}, \; B(\mathcal{G}) = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix},$$

$$\Omega(\mathcal{G}) = \begin{bmatrix} 3.4 & 0 & 0 & 0 & 0 \\ 0 & 2.1 & 0 & 0 & 0 \\ 0 & 0 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4.3 \end{bmatrix}.$$

Moreover, one can easily verify that $B(\mathcal{G})\Omega(\mathcal{G})B(\mathcal{G})^\top = L$.

2.3 Convex functions

Definition 2.5 (Convex set)**.** *A set $\mathbb{K} \subseteq \mathbb{R}^p$ is called convex if for any $x, y \in \mathbb{K}$ and $\alpha \in [0, 1]$ we have*

$$\alpha x + (1 - \alpha)y \in \mathbb{K}.$$

Definition 2.6 (Convex function)**.** *A function $f : \mathbb{R}^p \to \mathbb{R}$ is called convex on a convex set $\mathbb{K} \subseteq \mathbb{R}^p$ if for any $x, y \in \mathbb{K}$ and $\alpha \in [0, 1]$ we have*

$$f(\alpha x + (1 - \alpha)y) \le \alpha f(x) + (1 - \alpha)f(y).$$

From Theorems 2.1.2 and 2.1.3 in [62], we have the following results for convex functions.

Lemma 2.7. *Suppose function $f : \mathbb{R}^p \to \mathbb{R}$ is differentiable on $\mathbb{K}$ with $\mathbb{K} \subseteq \mathbb{R}^p$ being a convex set, then the following statements are equivalent.*

(i) *The function f is convex on $\mathbb{K}$.*

(ii) *For any $x, y \in \mathbb{K}$, it holds that*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

(iii) *For any $x, y \in \mathbb{K}$, it holds that*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0.$$

Definition 2.7 (Strong convexity)**.** *A function $f : \mathbb{R}^p \to \mathbb{R}$ is called strongly convex with convexity parameter $\mu > 0$ on a convex set $\mathbb{K} \subseteq \mathbb{R}^p$ if the function $f(x) - \frac{1}{2}\mu\|x\|^2$ is convex on $\mathbb{K}$.*

From Theorems 2.1.9 and 2.1.10 in [62], we have the following results for strongly convex functions.

Lemma 2.8. *Suppose function $f : \mathbb{R}^p \to \mathbb{R}$ is differentiable on $\mathbb{K}$ with $\mathbb{K} \subseteq \mathbb{R}^p$ being a convex set, then the following statements are equivalent.*

(i) *The function f is strongly convex with convexity parameter $\mu > 0$ on $\mathbb{K}$.*

(ii) *For any $x, y \in \mathbb{K}$, it holds that*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}\mu\|y - x\|^2.$$

(iii) *For any $x, y \in \mathbb{K}$, it holds that*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu\|x - y\|^2.$$

(iv) *For any $x, y \in \mathbb{K}$ and $\alpha \in [0, 1]$, it holds that*

$$\alpha f(x) + (1 - \alpha)f(y) \geq f(\alpha x + (1 - \alpha)y) + \frac{1}{2}\alpha(1 - \alpha)\mu\|x - y\|^2.$$

Lemma 2.9. *Suppose $f : \mathbb{R}^p \to \mathbb{R}$ is a differentiable and strongly convex function with convexity parameter $\mu > 0$, then for any $x, y \in \mathbb{R}^p$ we have*

$$\begin{aligned}
&f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2\mu}\|\nabla f(x) - \nabla f(y)\|^2, \\
&\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq \frac{1}{\mu}\|\nabla f(x) - \nabla f(y)\|^2, \\
&\mu\|x - y\|^2 \leq \|\nabla f(x) - \nabla f(y)\|^2.
\end{aligned}$$

Definition 2.8 (Subgradient)**.** *Let $f : \mathbb{D} \to \mathbb{R}$ be a function with $\mathbb{D} \subset \mathbb{R}^p$. A vector $g \in \mathbb{R}^p$ is called a subgradient of f at $x \in \mathbb{D}$ if*

$$f(y) \geq f(x) + \langle g, y - x \rangle, \ \forall y \in \mathbb{D}. \tag{2.9}$$

The set of all subgradients of f at x, denoted $\partial f(x)$, is called the subdifferential of f at x.

If the function f is convex and differentiable, then from Lemma 2.7 we know that its gradient at x is a subgradient, and from [206] we know that $\partial f(x)$ is a singleton. If f is a closed convex function, then $\partial f(x)$ is nonempty for any $x \in \mathbb{D}$ [207]. With a slight abuse of the notation, we use $\nabla f(x)$ to denote the subgradient of f at x also when f is not differentiable. Similarly, for a vector function $f = \mathrm{col}(f_1, \dots, f_m) : \mathbb{D} \to \mathbb{R}^m$, its subgradient at $x \in \mathbb{D}$ is denoted as

$$\nabla f(x) = \begin{bmatrix} (\nabla f_1(x))^\top \\ (\nabla f_2(x))^\top \\ \vdots \\ (\nabla f_m(x))^\top \end{bmatrix} \in \mathbb{R}^{m \times p}.$$

2.4 Projections

For a set $\mathbb{K} \subseteq \mathbb{R}^p$, let $\mathcal{P}_{\mathbb{K}}(\cdot)$ denotes the projection operator, i.e.,

$$\mathcal{P}_{\mathbb{K}}(y) \in \operatorname*{argmin}_{x \in \mathbb{K}} \|x - y\|^2, \ \forall y \in \mathbb{R}^p.$$

For simplicity, we use $[\cdot]_+$ to denote $\mathcal{P}_{\mathbb{K}}(\cdot)$ when $\mathbb{K} = \mathbb{R}^p_+$.

The projection operator has the following properties.

Lemma 2.10. *Let $\mathbb{K}$ be a nonempty closed convex subset of $\mathbb{R}^p$ and let $a,\ b,\ c$ be three vectors in $\mathbb{R}^p$. The following statements hold.*

(i) *For each $x \in \mathbb{R}^p$, $\mathcal{P}_{\mathbb{K}}(x)$ exists and is unique.*

(ii) *$\mathcal{P}_{\mathbb{K}}(x)$ is nonexpansive, i.e.,*

$$\|\mathcal{P}_{\mathbb{K}}(x) - \mathcal{P}_{\mathbb{K}}(y)\| \leq \|x - y\|, \ \forall x, y \in \mathbb{R}^p. \tag{2.10}$$

(iii) *If $a \leq b$, then*

$$\|[a]_+\| \leq \|b\|, \tag{2.11a}$$
$$[a]_+ \leq [b]_+. \tag{2.11b}$$

(iv) *If $x_1 = \mathcal{P}_{\mathbb{K}}(c - a)$, then*

$$2\langle x_1 - y, a \rangle \leq \|y - c\|^2 - \|y - x_1\|^2 - \|x_1 - c\|^2, \ \forall y \in \mathbb{K}. \tag{2.12}$$

Proof. The first two parts are from Theorem 1.5.5 in [208].

Substituting $x = a$ and $y = a - b$ into (2.10) with $\mathbb{K} = \mathbb{R}^p_+$ gives (2.11a). If $a \leq b$, then it is straightforward to see $[a]_+ \leq [b]_+$ since all inequalities are understood componentwise.

Denote $h(y) = \|c - y\|^2 + 2\langle a, y\rangle$. Then, $x_1 = \operatorname{argmin}_{y\in\mathbb{K}} h(y)$. This optimality condition implies that

$$\langle x_1 - y, \nabla h(x_1)\rangle \leq 0, \ \forall y \in \mathbb{K}.$$

Substituting $\nabla h(x_1) = 2x_1 - 2c + 2a$ into above inequality yields (2.12). □

2.5 Smooth functions

Definition 2.9 (Smooth function)**.** *A function $f : \mathbb{R}^p \mapsto \mathbb{R}$ is called smooth with constant $L_f > 0$ if it is differentiable and*

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f\|x - y\|, \ \forall x, y \in \mathbb{R}^p. \tag{2.13}$$

From Lemma 1.2.3 in [62], we know that (2.13) implies

$$|f(y) - f(x) - (y - x)^\top \nabla f(x)| \leq \frac{L_f}{2}\|y - x\|^2, \ \forall x, y \in \mathbb{R}^p, \tag{2.14}$$

which further implies

$$\|\nabla f(x)\|^2 \leq 2L_f(f(x) - f^*), \ \forall x, y \in \mathbb{R}^p, \tag{2.15}$$

where $f^* = \min_{x\in\mathbb{R}^p} f(x)$. Moreover, we have the following lemma.

Lemma 2.11. *If $f : \mathbb{R}^p \mapsto \mathbb{R}$ is smooth with constant $L_f > 0$, then, for any $a > L_f$, the function $g(x) = f(x) + \frac{a}{2}\|x\|^2$ is strongly convex with convex parameter $a - L_f$.*

Proof. From (2.13), we have

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq -\|\nabla f(x) - \nabla f(y)\|\|x - y\| \geq -L_f\|x - y\|^2.$$

Then,

$$\begin{aligned}\langle \nabla g(x) - \nabla g(y), x - y\rangle &= \langle \nabla f(x) + ax - \nabla f(y) - ay, x - y\rangle \\ &= \langle \nabla f(x) - \nabla f(y), x - y\rangle + a\|x - y\|^2 \\ &\geq (a - L_f)\|x - y\|^2.\end{aligned}$$

Then, from Lemma 2.8, we know that this lemma holds. □

2.6 Polyak–Łojasiewicz condition

Let $f(x) : \mathbb{R}^p \mapsto \mathbb{R}$ be a differentiable function. Let $\mathbb{X}^* = \operatorname{argmin}_{x\in\mathbb{R}^p} f(x)$ and $f^* = \min_{x\in\mathbb{R}^p} f(x)$. Moreover, we assume that $f^* > -\infty$.

Definition 2.10 (Polyak–Łojasiewicz condition)**.** *The function f satisfies the Polyak–Łojasiewicz (P–Ł) condition with constant $\nu > 0$ if*

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \nu(f(x) - f^*),\ \forall x \in \mathbb{R}^p. \tag{2.16}$$

From Lemma 2.9, it is straightforward to see that every strongly convex function satisfies the P–Ł condition. The P–Ł condition implies that every stationary point is a global minimizer, i.e., $\mathbb{X}^* = \{x \in \mathbb{R}^p : \nabla f(x) = \mathbf{0}_p\}$. But unlike strong convexity, the P–Ł condition (2.16) alone does not even imply the convexity of f. Moreover, it does not imply that $\mathbb{X}^*$ is a singleton or finite either.

Many practical applications, such as least squares, do not always have strongly convex cost functions. The cost function in least squares problems has the form

$$f(x) = \frac{1}{2}\|Ax - b\|^2,$$

where $A \in \mathbb{R}^{m\times p}$ and $b \in \mathbb{R}^m$. Note that if A has full column rank, then $f(x)$ is strongly convex. However, if A is rank deficient, then $f(x)$ is not strongly convex, but it is convex and satisfies the P–Ł condition. The function $f(x) = x^2 + 3\sin^2(x)$ is an example of a nonconvex function satisfying the P–Ł condition with $\nu = 1/32$ [209]. More examples of nonconvex functions which satisfy the P–Ł condition can be found in [209, 210].

Although it is difficult to precisely characterize the general class of functions for which the P–Ł condition is satisfied, in [209], one important special case was given as follows:

Lemma 2.12. *Let $f(x) = g(Ax)$, where $g : \mathbb{R}^p \to \mathbb{R}$ is a strongly convex function and $A \in \mathbb{R}^{p\times p}$ is a matrix, then f satisfies the P–Ł condition.*

In the literature, there are some conditions that are weaker than strong convexity but stronger than the P–Ł condition.

Definition 2.11 (Essential strong convexity)**.** *The function f is said to be essentially strongly convex with constant $\mu > 0$ if for all $x, y \in \mathbb{R}^p$ such that $\mathcal{P}_{\mathbb{X}^*}(x) = \mathcal{P}_{\mathbb{X}^*}(y)$, it holds that*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|x - y\|^2.$$

Definition 2.12 (Weak strong convexity)**.** *The function f is said to be weakly strongly convex with constant $\mu > 0$ if for all $x \in \mathbb{R}^p$, it holds that*

$$f^* \geq f(x) + \langle \nabla f(x), \mathcal{P}_{\mathbb{X}^*}(x) - x\rangle + \frac{\mu}{2}\|x - \mathcal{P}_{\mathbb{X}^*}(x)\|^2.$$

Definition 2.13 (Restricted secant inequality condition)**.** *The function f satisfies the restricted secant inequality (RSI) condition with constant $\nu > 0$ if for all $x \in \mathbb{R}^p$, it holds that*

$$(\nabla f(x) - \nabla f(\mathcal{P}_{\mathbb{X}^*}(x))^\top (x - \mathcal{P}_{\mathbb{X}^*}(x)) \geq \nu \|x - \mathcal{P}_{\mathbb{X}^*}(x)\|^2.$$

If the function f is also convex it is called restricted strong convexity.

The following lemma summarizes the relations between the function classes discussed above.

Lemma 2.13. *(Theorem 2 in [209]) Let $f(x) : \mathbb{R}^p \mapsto \mathbb{R}$ be a differentiable function. Then,* (Strong convexity) $\Rightarrow$ (Essential strong convexity) $\Rightarrow$ (Weak strong convexity) $\Rightarrow$ (RSI condition). *Moreover,* (RSI condition) $\Rightarrow$ (P–Ł condition) *if f is smooth.*

2.7 Bregman divergence

Let $\mathbb{K} \subseteq \mathbb{R}^p$ be a convex set. The Bregman divergence

$$\mathcal{D}_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y\rangle, \tag{2.17}$$

is to measure the distance between $x, y \in \mathbb{K}$, where $\psi : \mathbb{R}^p \to \mathbb{R}$ is a function which is differentiable and strongly convex with convexity parameter $\sigma > 0$ on $\mathbb{K}$. Then, from Lemma 2.8, we have $\psi(x) \geq \psi(y) + \langle \nabla\psi(y), x - y\rangle + \frac{\sigma}{2}\|x - y\|^2$. Thus,

$$\mathcal{D}_\psi(x, y) \geq \frac{\sigma}{2}\|x - y\|^2. \tag{2.18}$$

Hence, $\mathcal{D}_\psi(\cdot, y)$ is strongly convex with convexity parameter σ on $\mathbb{K}$ for all fixed $y \in \mathbb{K}$. Additionally, (2.17) implies that for all $x, y, z \in \mathbb{K}$,

$$\langle y - x, \nabla\psi(z) - \nabla\psi(y)\rangle = \mathcal{D}_\psi(x, z) - \mathcal{D}_\psi(x, y) - \mathcal{D}_\psi(y, z). \tag{2.19}$$

Two well-known examples of Bregman divergence are Euclidean distance $\mathcal{D}_\psi(x, y) = \|x - y\|^2$ (with $\mathbb{K}$ being an arbitrary convex and compact set in $\mathbb{R}^p$) generated from $\psi(x) = \|x\|^2$, and the Kullback–Leibler divergence $\mathcal{D}_\psi(x, y) = -\sum_{j=1}^p x_j \log\frac{y_j}{x_j}$ between two p-dimensional standard unit vectors (with $\mathbb{K}$ being the p-dimensional probability simplex in $\mathbb{R}^p$) generated from $\psi(x) = \sum_{j=1}^p (x_j \log x_j - x_j)$.

We have the following results on the regularized Bregman projection.

Lemma 2.14. *Suppose function $\psi : \mathbb{R}^p \to \mathbb{R}^p$ is differentiable and strongly convex with convexity parameter $\sigma > 0$ on $\mathbb{K}$, and function $h : \mathbb{R}^p \to \mathbb{R}$ is convex on $\mathbb{K}$, where $\mathbb{K} \subseteq \mathbb{R}^p$ is a convex and closed set. Moreover, assume that $\nabla h(x)$, $\forall x \in \mathbb{K}$, exists and there exists $G_h > 0$ such that $\|\nabla h(x)\| \leq G_h$, $\forall x \in \mathbb{K}$. Given $z \in \mathbb{K}$, the regularized Bregman projection*

$$y = \operatorname*{argmin}_{x\in\mathbb{K}}\{h(x) + \mathcal{D}_\psi(x, z)\}, \tag{2.20}$$

satisfies the following inequalities

$$\langle y - x, \nabla h(y)\rangle \le \mathcal{D}_\psi(x,z) - \mathcal{D}_\psi(x,y) - \mathcal{D}_\psi(y,z),\ \forall x \in \mathbb{K}, \tag{2.21}$$

$$\|y - z\| \le \frac{G_h}{\sigma}. \tag{2.22}$$

Proof. **(i)** Denote $\tilde{h}(x) = h(x) + \mathcal{D}_\psi(x,z)$. Then $\tilde{h}$ is convex on $\mathbb{K}$. Thus the optimality condition (2.20), i.e., $y = \operatorname{argmin}_{x\in\mathbb{K}} \tilde{h}(x)$, implies $\langle y - x, \nabla\tilde{h}(y)\rangle \le 0,\ \forall x \in \mathbb{K}$. Substituting $\nabla\tilde{h}(y) = \nabla h(y) + \nabla\psi(y) - \nabla\psi(z)$ into the above inequality yields

$$\begin{aligned}\langle y - x, \nabla h(y)\rangle &\le \langle y - x, \nabla\psi(z) - \nabla\psi(y)\rangle\\ &= \mathcal{D}_\psi(x,z) - \mathcal{D}_\psi(x,y) - \mathcal{D}_\psi(y,z),\ \forall x \in \mathbb{K},\end{aligned}$$

where the equality holds due to (2.19). Hence, (2.21) holds.
(ii) $\tilde{h}(x)$ is strongly convex with convexity parameter σ on $\mathbb{K}$ since $\mathcal{D}_\psi$ is strongly convex on $\mathbb{K}$. It is known that if $\tilde{h} : \mathbb{K} \to \mathbb{R}$ is a strongly convex function and is minimized at the point $x^{\min} \in \mathbb{K}$, then

$$\tilde{h}(x^{\min}) \le \tilde{h}(x) - \frac{\sigma}{2}\|x - x^{\min}\|^2,\ \forall x \in \mathbb{K}.$$

Thus the optimality condition of (2.20) implies

$$h(y) + \mathcal{D}_\psi(y,z) \le h(z) + \mathcal{D}_\psi(z,z) - \frac{\sigma}{2}\|z - y\|^2.$$

Noting that $\mathcal{D}_\psi(y,z) \ge \frac{\sigma}{2}\|z - y\|^2$ and $\mathcal{D}_\psi(z,z) = 0$, and rearranging the above inequality gives

$$\sigma\|z - y\|^2 \le \frac{\sigma}{2}\|z - y\|^2 + \mathcal{D}_\psi(y,z) \le h(z) - h(y). \tag{2.23}$$

From (2.9) and $\|\nabla h(x)\| \le G_h,\ \forall x \in \mathbb{K}$, we have

$$h(z) - h(y) \le \langle \nabla h(z), z - y\rangle \le G_h\|z - y\|. \tag{2.24}$$

Thus, combining (2.23) and (2.24) yields (2.22). □

Note that (2.21) extends Lemma 6 in [181] and (2.22) presents an upper bound on the deviation of the optimal point from a fixed point for the regularized Bregman projection.

To end this section, we introduce a generalized definition of strong convexity, which is Definition 2 in [211].

Definition 2.14 (Generalized strong convexity)**.** *Let $\mu > 0$ be a constant and $\mathbb{K} \subseteq \mathbb{R}^p$ be a convex set. Let $f : \mathbb{R}^p \to \mathbb{R}$ and $\psi : \mathbb{R}^p \to \mathbb{R}$ be two functions. Suppose that f is convex on $\mathbb{K}$, and ψ is differentiable and strongly convex on $\mathbb{K}$. The function f is said to be μ-strongly convex on $\mathbb{K}$ with respect to ψ if for all $x, y \in \mathbb{K}$,*

$$f(x) \ge f(y) + \langle x - y, \nabla f(y)\rangle + \mu\mathcal{D}_\psi(x,y).$$

This definition generalizes the usual definition of strong convexity by replacing the Euclidean distance with the Bregman divergence.

2.8 Random gradient estimators

In this section, we introduce one- and two-point sampling gradient estimators.

Let $f : \mathbb{K} \to \mathbb{R}$ be a function with $\mathbb{K} \subseteq \mathbb{R}^p$. We assume that $\mathbb{K}$ is convex and closed, and has a nonempty interior. Specifically, we assume that $\mathbb{K}$ contains the ball of radius $r(\mathbb{K})$ centered at the origin, i.e., $r(\mathbb{K})\mathbb{B}^p \subseteq \mathbb{K}$. The authors of [212] proposed the following gradient estimator:

$$\hat{\nabla}_1 f(x) = \hat{\nabla}_1 f(x, \delta, u) = \frac{p}{\delta} f(x + \delta u)u, \ \forall x \in (1 - \xi)\mathbb{K}, \tag{2.25}$$

where $u \in \mathbb{S}^p$ is a uniformly distributed random vector, $\delta \in (0, r(\mathbb{K})\xi]$ is an smoothing/exploration parameter, and $\xi \in (0, 1)$ is a shrinkage coefficient. The estimator $\hat{\nabla}_1 f$ only requires to sample the function value at one point, so it is a one-point sampling gradient estimator. The intuition for this estimator can be found in [212]. Different from [213], uniform distribution rather than Gaussian distribution is used to generate u in (2.25) since the later may generate unbounded u. The estimator $\hat{\nabla}_1 f$ is defined over the set $(1 - \xi)\mathbb{K}$ instead of $\mathbb{K}$, since otherwise the perturbations may move points outside $\mathbb{K}$. The feasibility of the perturbations is guaranteed by the following lemma.

Lemma 2.15. *(Observation 2 in [212]) For any $x \in (1 - \xi)\mathbb{K}$ and $u \in \mathbb{S}^p$, it holds that $x + \delta u \in \mathbb{K}$ for any $\delta \in (0, r(\mathbb{K})\xi]$.*

The two-point sampling gradient estimator is defined as

$$\hat{\nabla}_2 f(x) = \hat{\nabla}_2 f(x, \delta, u) = \frac{p}{\delta}(f(x + \delta u) - f(x))u, \ \forall x \in (1 - \xi)\mathbb{K}. \tag{2.26}$$

The intuition follows from directional derivatives [214].

Both estimators $\hat{\nabla}_1 f$ and $\hat{\nabla}_2 f$ are unbiased gradient estimators of f^s, where f^s is the uniformly smoothed version of f defined as

$$f^s(x) = f^s(x, \delta) = \mathbf{E}_{v\in\mathbb{B}^p}[f(x + \delta v)], \ \forall x \in (1 - \xi)\mathbb{K},$$

where the expectation is taken with respect to uniform distribution. Some properties of f^s, $\hat{\nabla}_1 f$, and $\hat{\nabla}_2 f$ are presented in the following lemma.

Lemma 2.16. **(i)** *The uniform smoothing f^s is differentiable on $(1 - \xi)\mathbb{K}$ even when f is not, and for all $x \in (1 - \xi)\mathbb{K}$,*

$$\nabla f^s(x) = \mathbf{E}_{u\in\mathbb{S}^p}[\hat{\nabla}_1 f(x)] = \mathbf{E}_{u\in\mathbb{S}^p}[\hat{\nabla}_2 f(x)]. \tag{2.27}$$

(ii) *If f is convex on $\mathbb{K}$, then f^s is convex on $(1 - \xi)\mathbb{K}$ and*

$$f(x) \le f^s(x), \ \forall x \in (1 - \xi)\mathbb{K}. \tag{2.28}$$

(iii) *If f is Lipschitz-continuous on $\mathbb{K}$ with constant $L_0(f) > 0$, then f^s and ∇f^s are Lipschitz-continuous on $(1 - \xi)\mathbb{K}$ with constants $L_0(f)$ and $pL_0(f)/\delta$, respectively. Moreover,*

$$|f^s(x) - f(x)| \le \delta L_0(f), \ \forall x \in (1 - \xi)\mathbb{K}. \tag{2.29}$$

(iv) *If f is bounded on $\mathbb{K}$, i.e., there exists $F_0(f) > 0$ such that $|f(x)| \leq F_0(f)$, $\forall x \in \mathbb{K}$, then*

$$|f^s(x)| \leq F_0(f),\ \|\hat{\nabla}_1 f(x)\| \leq \frac{pF_0(f)}{\delta},\ \forall x \in (1-\xi)\mathbb{K}. \tag{2.30}$$

(v) *If f is Lipschitz-continuous on $\mathbb{K}$ with constant $L_0(f) > 0$, then*

$$\|\hat{\nabla}_2 f(x)\| \leq pL_0(f),\ \forall x \in (1-\xi)\mathbb{K}. \tag{2.31}$$

(vi) *If f is smooth with constant $L_f > 0$, then*

$$\|\nabla f^s(x,\delta) - \nabla f(x)\| \leq \delta L_f, \tag{2.32a}$$

$$\mathbf{E}_{u\in\mathbb{S}^p}[\|\hat{\nabla}_2 f(x)\|^2] \leq 2p\|\nabla f(x)\|^2 + \frac{1}{2}p^2\delta^2 L_f^2. \tag{2.32b}$$

Proof. **(i)** From Lemma 1 in [212], we have $\nabla f^s(x) = \mathbf{E}_{u\in\mathbb{S}^p}[\hat{\nabla}_1 f(x)]$. Moreover, we have $\nabla f^s(x) = \mathbf{E}_{u\in\mathbb{S}^p}[\hat{\nabla}_2 f(x)]$ due to $\mathbf{E}_{u\in\mathbb{S}^p}[f(x)u] = f(x)\mathbf{E}_{u\in\mathbb{S}^p}[u] = \mathbf{0}_p$.
(ii) It is straightforward to see that $(1-\xi)\mathbb{K}$ is convex since $\mathbb{K}$ is convex.

For any $x, y \in (1-\xi)\mathbb{K}$ and $\alpha \in [0,1]$, then $\alpha x + (1-\alpha)y \in (1-\xi)\mathbb{K}$ since $(1-\xi)\mathbb{K}$ is convex and $\alpha x + (1-\alpha)y + \delta v \in \mathbb{K}$ due to Lemma 2.15. Moreover,

$$\begin{aligned} f^s(\alpha x + (1-\alpha)y) &= \mathbf{E}_{v\in\mathbb{B}^p}[f(\alpha x + (1-\alpha)y + \delta v)] \\ &\leq \mathbf{E}_{v\in\mathbb{B}^p}[\alpha f(x+\delta v) + (1-\alpha)f(y+\delta v)] \\ &= \alpha f^s(x) + (1-\alpha)f^s(y). \end{aligned}$$

Hence, f^s is convex on $(1-\xi)\mathbb{K}$.

From Lemma 2.15, we know that $(1-\xi)\mathbb{K}$ is a subset of the interior of $\mathbb{K}$. Then, for any $x \in (1-\xi)\mathbb{K}$, from Theorem 3.1.15 in [62], we know that $\nabla f(x)$ exists. Moreover,

$$f^s(x) = \mathbf{E}_{v\in\mathbb{B}^p}[f(x+\delta v)] \geq \mathbf{E}_{v\in\mathbb{B}^p}[f(x) + \delta\langle\nabla f(x), v\rangle] = f(x).$$

(iii) For any $x, y \in (1-\xi)\mathbb{K}$,

$$\begin{aligned} |f^s(x) - f^s(y)| &= |\mathbf{E}_{v\in\mathbb{B}^p}[f(x+\delta v) - f(y+\delta v)]| \\ &\leq \mathbf{E}_{v\in\mathbb{B}^p}[|f(x+\delta v) - f(y+\delta v)|] \\ &\leq \mathbf{E}_{v\in\mathbb{B}^p}[L_0(f)\|x-y\|] = L_0(f)\|x-y\|. \end{aligned}$$

Hence, f^s is Lipschitz-continuous on $(1-\xi)\mathbb{K}$ with constant $L_0(f)$.

Similarly,

$$\begin{aligned} \|\nabla f^s(x) - \nabla f^s(y)\| &= \frac{p}{\delta}\|\mathbf{E}_{u\in\mathbb{S}^p}[f(x+\delta u)u - f(y+\delta u)u]\| \\ &\leq \frac{p}{\delta}\mathbf{E}_{u\in\mathbb{S}^p}[|f(x+\delta u) - f(y+\delta u)|\|u\|] \\ &\leq \frac{p}{\delta}\mathbf{E}_{u\in\mathbb{S}^p}[L_0(f)\|x-y\|] = \frac{pL_0(f)}{\delta}\|x-y\|. \end{aligned}$$

Hence, ∇f^s is Lipschitz-continuous on $(1-\xi)\mathbb{K}$ with constant $pL_0(f)/\delta$.

For any $x \in (1-\xi)\mathbb{K}$,

$$\begin{aligned}|f^s(x) - f(x)| &= |\mathbf{E}_{v\in\mathbb{B}^p}[f(x+\delta v)] - \mathbf{E}_{v\in\mathbb{B}^p}[f(x)]| \\ &\le \mathbf{E}_{v\in\mathbb{B}^p}[|f(x+\delta v) - f(x)|] \le \mathbf{E}_{v\in\mathbb{B}^p}[\delta L_0(f)\|v\|] \\ &\le \mathbf{E}_{v\in\mathbb{B}^p}[\delta L_0(f)] = \delta L_0(f).\end{aligned}$$

(iv) For any $x \in (1-\xi)\mathbb{K}$ and $u \in \mathbb{S}^p$,

$$|f^s(x)| = |\mathbf{E}_{v\in\mathbb{B}^p}[f(x+\delta v)]| \le \mathbf{E}_{v\in\mathbb{B}^p}[|f(x+\delta v)|] \le F_0(f),$$

and

$$\|\hat{\nabla}_1 f(x)\| = \|\frac{p}{\delta} f(x+\delta u)u\| \le \frac{p}{\delta}|f(x+\delta u)|\|u\| \le \frac{pF_0(f)}{\delta}.$$

(v) For any $x \in (1-\xi)\mathbb{K}$ and $u \in \mathbb{S}^p$,

$$\begin{aligned}\|\hat{\nabla}_2 f(x)\| &= \|\frac{p}{\delta}(f(x+\delta u) - f(x))u\| \\ &\le \frac{pL_0(f)}{\delta}\|x+\delta u - x\|\|u\| = pL_0(f).\end{aligned}$$

(vi) From Lemma 5 in [151], we have (2.32a). From Proposition 7.6 in [215], we have (2.32b). □

Intuitively, the key idea of gradient-free optimization methods is using the smoothed function f^s to replace the original function f since they are close when δ is small as shown in (c) of Lemma 2.16. Moreover, the gradient of f^s can be estimated by the gradient estimators $\hat{\nabla}_1 f$ or $\hat{\nabla}_2 f$ as shown in (2.27). The main difference between these two gradient estimators is that the norm of $\hat{\nabla}_1 f$ is large when δ is small, while $\hat{\nabla}_2 f$ has a bounded norm, as shown in (2.30) and (2.31), respectively. This difference leads to improved results for the two-point sampling based algorithms.

2.9 Deterministic gradient estimators

Let $f(x) : \mathbb{R}^p \mapsto \mathbb{R}$ be a differentiable function. The authors of [164] proposed the following deterministic gradient estimator:

$$\hat{\nabla}_p f(x,\delta) = \frac{1}{\delta}\sum_{l=1}^{p}(f(x+\delta \boldsymbol{e}_l) - f(x))\boldsymbol{e}_l, \tag{2.33}$$

where $\delta > 0$ is an exploration parameter. This gradient estimator can be calculated by querying the function values of f at $p+1$ points. Another commonly used deterministic gradient estimator is

$$\hat{\nabla}_{2p} f(x,\delta) = \frac{1}{2\delta}\sum_{l=1}^{p}(f(x+\delta \boldsymbol{e}_l) - f(x-\delta \boldsymbol{e}_l))\boldsymbol{e}_l. \tag{2.34}$$

This gradient estimator can be viewed as a noise-free version of the classical Kiefer–Wolfowitz type method [216], and can be calculated by querying the function values of f at $2p$ points. Thus, when p is large, $\hat{\nabla}_p f$ is more favorable than $\hat{\nabla}_{2p} f$.

From equation (16) in [164] we know that $\hat{\nabla}_p f(x,\delta)$ and $\hat{\nabla}_{2p} f(x,\delta)$ are close to $\nabla f(x)$ when δ is small, which is summarized in the following lemma.

Lemma 2.17. *Suppose that f is smooth with constant $L_f > 0$, then*

$$\|\hat{\nabla}_p f(x,\delta) - \nabla f(x)\| \le \frac{\sqrt{p} L_f \delta}{2},\ \forall x \in \mathbb{R}^p,\ \forall \delta > 0, \tag{2.35a}$$

$$\|\hat{\nabla}_{2p} f(x,\delta) - \nabla f(x)\| \le \frac{\sqrt{p} L_f \delta}{2},\ \forall x \in \mathbb{R}^p,\ \forall \delta > 0. \tag{2.35b}$$

2.10 Useful lemmas on series

Lemma 2.18. *Let $a, b \in (0,1)$ be two constants, then*

$$\sum_{\tau=0}^{k} a^\tau b^{k-\tau} \le \begin{cases} \frac{a^{k+1}}{a-b}, & \text{if } a > b, \\ \frac{b^{k+1}}{b-a}, & \text{if } a < b, \\ \frac{c^{k+1}}{c-b}, & \text{if } a = b, \end{cases} \tag{2.36}$$

where c is any constant in $(a,1)$.

Proof. If $a > b$, then

$$\sum_{\tau=0}^{k} a^\tau b^{k-\tau} = a^k \sum_{\tau=0}^{k} \Big(\frac{b}{a}\Big)^{k-\tau} \le \frac{a^{k+1}}{a-b}.$$

Similarly, when $a < b$, we have

$$\sum_{\tau=0}^{k} a^\tau b^{k-\tau} = b^k \sum_{\tau=0}^{k} \Big(\frac{a}{b}\Big)^{\tau} \le \frac{b^{k+1}}{b-a}.$$

If $a = b$, then for any $c \in (a,1)$, we have

$$\sum_{\tau=0}^{k} a^\tau b^{k-\tau} \le \sum_{\tau=0}^{k} c^\tau b^{k-\tau} = c^k \sum_{\tau=0}^{k} \Big(\frac{b}{c}\Big)^{k-\tau} \le \frac{c^{k+1}}{c-b}.$$

Hence, this lemma holds. □

Lemma 2.19. *Let k and τ be two integers and δ be a constant. Suppose $k \ge \tau \ge 1$, then*

$$\sum_{l=\tau}^{k} l^\delta \le \begin{cases} \frac{(k+1)^{\delta+1}}{\delta+1}, & \text{if } \delta > -1, \\ \ln(k), & \text{if } \delta = -1, \\ \frac{-(\tau-1)^{\delta+1}}{\delta+1}, & \text{if } \delta < -1 \text{ and } \tau \ge 2. \end{cases} \tag{2.37}$$

Proof. If $\delta \geq 0$, then $h(t) = t^\delta$ is an increasing function in the interval $[1, +\infty)$. Hence,

$$\sum_{l=\tau}^{k} l^\delta \leq \int_{\tau}^{k+1} t^\delta dt = \frac{(k+1)^{\delta+1} - \tau^{\delta+1}}{\delta+1} \leq \frac{(k+1)^{\delta+1}}{\delta+1}. \tag{2.38}$$

If $\delta < 0$, then $h(t) = t^\delta$ is a decreasing function in the interval $[1, +\infty)$. Hence,

$$\begin{aligned}\sum_{l=\tau}^{k} l^\delta \leq \int_{\tau-1}^{k} t^\delta dt &= \begin{cases} \ln(\frac{k}{\tau-1}), & \text{if } \delta = -1, \\ \frac{k^{\delta+1}-(\tau-1)^{\delta+1}}{\delta+1}, & \text{if } -1 < \delta < 0, \\ \frac{k^{\delta+1}-(\tau-1)^{\delta+1}}{\delta+1}, & \text{if } \delta < -1 \text{ and } \tau \geq 2, \end{cases} \\ &\leq \begin{cases} \ln(k), & \text{if } \delta = -1, \\ \frac{(k+1)^{\delta+1}}{\delta+1}, & \text{if } -1 < \delta < 0, \\ \frac{-(\tau-1)^{\delta+1}}{\delta+1}, & \text{if } \delta < -1 \text{ and } \tau \geq 2. \end{cases} \end{aligned} \tag{2.39}$$

Finally, (2.38) and (2.39) yield (2.37). □

Lemma 2.20. *Let $\{z_k\}$, $\{r_{1,k}\}$, and $\{r_{2,k}\}$ be sequences. Suppose there exists $t_1 \in \mathbb{N}_+$ such that*

$$z_k \geq 0, \tag{2.40a}$$
$$z_{k+1} \leq (1 - r_{1,k})z_k + r_{2,k}, \tag{2.40b}$$
$$1 > r_{1,k} \geq \frac{a_1}{(k+t_1)^{\delta_1}}, \tag{2.40c}$$
$$r_{2,k} \leq \frac{a_2}{(k+t_1)^{\delta_2}}, \ \forall k \in \mathbb{N}_0, \tag{2.40d}$$

where $a_1 > 0$, $a_2 > 0$, $\delta_1 \in [0, 1]$, and $\delta_2 > \delta_1$ are constants.

(i) *If $\delta_1 \in (0, 1)$, then*

$$z_k \leq \phi_1(k, t_1, a_1, a_2, \delta_1, \delta_2, z_0), \ \forall k \in \mathbb{N}_+, \tag{2.41}$$

where

$$\begin{aligned}\phi_1(k, t_1, a_1, a_2, \delta_1, \delta_2, z_0) = &\frac{1}{s_1(k+t_1)}\Big(s_1(t_1)z_0 + \frac{[t_2 - 1 - t_1]_+ s_1(t_1+1)a_2}{t_1^{\delta_2}}\Big) \\ &+ \frac{a_2}{(k+t_1-1)^{\delta_2}} + \frac{\mathbf{1}_{(k+t_1-1\geq t_2)}(\frac{t_1+1}{t_1})^{\delta_2} a_2\delta_2}{a_1\delta_1(k+t_1)^{\delta_2-\delta_1}},\end{aligned} \tag{2.42}$$

$s_1(k) = e^{\frac{a_1}{1-\delta_1}k^{1-\delta_1}}$ *and* $t_2 = \lceil(\frac{\delta_2}{a_1})^{\frac{1}{1-\delta_1}}\rceil$.

(ii) *If $\delta_1 = 1$, then*

$$z_k \leq \phi_2(k, t_1, a_1, a_2, \delta_2, z_0), \ \forall k \in \mathbb{N}_+, \tag{2.43}$$

where

$$\phi_2(k,t_1,a_1,a_2,\delta_2,z_0)=\frac{t_1^{a_1}z_0}{(k+t_1)^{a_1}}+\frac{a_2}{(k+t_1-1)^{\delta_2}}+\Big(\frac{t_1+1}{t_1}\Big)^{\delta_2}a_2s_2(k+t_1), \quad (2.44)$$

and

$$s_2(k)=\begin{cases}\frac{1}{(a_1-\delta_2+1)k^{\delta_2-1}}, & \text{if } a_1-\delta_2>-1,\\ \frac{\ln(k-1)}{k^{a_1}}, & \text{if } a_1-\delta_2=-1,\\ \frac{-t_1^{a_1-\delta_2+1}}{(a_1-\delta_2+1)k^{a_1}}, & \text{if } a_1-\delta_2<-1.\end{cases}$$

(iii) *If* $\delta_1=0$, *then*

$$z_k\le\phi_3(k,t_1,a_1,a_2,\delta_2,z_0),\ \forall k\in\mathbb{N}_+, \quad (2.45)$$

where

$$\begin{aligned}&\phi_3(k,t_1,a_1,a_2,\delta_2,z_0)\\&=a_2(1-a_1)^{k+t_1-1}\Big([t_3-t_1]_+s_3(t_1)+([t_4-t_1]_+-[t_3-t_1]_+)s_3(t_4)\Big)\\&\quad+(1-a_1)^kz_0+\frac{\mathbf{1}_{(k+t_1-1\ge t_4)}2a_2}{-\ln(1-a_1)(k+t_1)^{\delta_2}(1-a_1)},\end{aligned} \quad (2.46)$$

$s_3(k)=\frac{1}{k^{\delta_2}(1-a_1)^k}$, $t_3=\lceil\frac{-\delta_2}{\ln(1-a_1)}\rceil$, *and* $t_4=\lceil\frac{-2\delta_2}{\ln(1-a_1)}\rceil$.

Proof. This proof is inspired by the proof of Lemma 25 in [217].
From (2.40a)–(2.40c), for any $k\in\mathbb{N}_+$, it holds that

$$z_k\le\prod_{\tau=0}^{k-1}(1-r_{1,\tau})z_0+r_{2,k-1}+\sum_{l=0}^{k-2}\prod_{\tau=l+1}^{k-1}(1-r_{1,\tau})r_{2,l}. \quad (2.47)$$

For any $t\in[0,1]$, it holds that $1-t\le e^{-t}$ since $s_4(t)=1-t-e^{-t}$ is a nonincreasing function in the interval $[0,1]$ and $s_4(0)=0$. Thus, for any $k>l\ge0$, it holds that

$$\prod_{\tau=l}^{k-1}(1-r_{1,\tau})\le e^{-\sum_{\tau=l}^{k-1}r_{1,\tau}}. \quad (2.48)$$

We also have

$$\begin{aligned}\sum_{\tau=l}^{k-1}r_{1,\tau}&\ge\sum_{\tau=l}^{k-1}\frac{a_1}{(\tau+t_1)^{\delta_1}}=\sum_{\tau=l+t_1}^{k-1+t_1}\frac{a_1}{\tau^{\delta_1}}\ge\int_{t=l+t_1}^{k+t_1}\frac{a_1}{t^{\delta_1}}dt\\&=\begin{cases}\frac{a_1}{1-\delta_1}((k+t_1)^{1-\delta_1}-(l+t_1)^{1-\delta_1}), & \text{if } \delta_1\in(0,1),\\ a_1\ln(\frac{k+t_1}{l+t_1}), & \text{if } \delta_1=1,\end{cases}\end{aligned} \quad (2.49)$$

where the first inequality holds due to (2.40c) and the second inequality holds since $s_5(t) = a_1/t^{\delta_1}$ is a decreasing function in the interval $[1, +\infty)$.

Hence, (2.48) and (2.49) yield

$$\prod_{\tau=l}^{k-1}(1-r_{1,\tau}) \le e^{-\sum_{\tau=l}^{k-1} r_{1,\tau}} \le \begin{cases} \frac{s_1(l+t_1)}{s_1(k+t_1)}, & \text{if } \delta_1 \in (0,1), \\ \frac{(l+t_1)^{a_1}}{(k+t_1)^{a_1}}, & \text{if } \delta_1 = 1. \end{cases} \tag{2.50}$$

(i) When $\delta_1 \in (0,1)$, from (2.50) and (2.40d), we have

$$\begin{aligned}
\sum_{l=0}^{k-2}\prod_{\tau=l+1}^{k-1}(1-r_{1,\tau})r_{2,l} &\le \sum_{l=0}^{k-2}\frac{s_1(l+t_1+1)}{s_1(k+t_1)}\frac{a_2}{(l+t_1)^{\delta_2}} \\
&= \frac{a_2}{s_1(k+t_1)}\sum_{l=0}^{k-2}\frac{s_1(l+t_1+1)}{(l+t_1)^{\delta_2}} \\
&\le \frac{a_2}{s_1(k+t_1)}\sum_{l=0}^{k-2}\frac{s_1(l+t_1+1)}{(\frac{t_1}{t_1+1}l+t_1)^{\delta_2}} \\
&= \frac{(\frac{t_1+1}{t_1})^{\delta_2}a_2}{s_1(k+t_1)}\sum_{l=0}^{k-2}\frac{s_1(l+t_1+1)}{(l+t_1+1)^{\delta_2}} \\
&= \frac{(\frac{t_1+1}{t_1})^{\delta_2}a_2}{s_1(k+t_1)}\sum_{l=t_1+1}^{k+t_1-1}\frac{s_1(l)}{l^{\delta_2}} \\
&= \frac{(\frac{t_1+1}{t_1})^{\delta_2}a_2}{s_1(k+t_1)}\Big(\sum_{l=t_1+1}^{t_2-1}\frac{s_1(l)}{l^{\delta_2}} + \sum_{l=t_2}^{k+t_1-1}\frac{s_1(l)}{l^{\delta_2}}\Big).
\end{aligned} \tag{2.51}$$

We know that $s_6(t) = s_1(t)/t^{\delta_2}$ is a decreasing function in the interval $[1, t_2-1]$ due to

$$\frac{ds_6(t)}{dt} = \Big(a_1 - \frac{\delta_2}{t^{1-\delta_1}}\Big)\frac{s_6(t)}{t^{\delta_1}} \le 0,\ \forall t \in \Big(0, \Big(\frac{\delta_2}{a_1}\Big)^{\frac{1}{1-\delta_1}}\Big].$$

Thus, for any $k \in [1, t_2-1]$, we have

$$\sum_{l=k}^{t_2-1}\frac{s_1(l)}{l^{\delta_2}} \le (t_2-k)\frac{s_1(k)}{k^{\delta_2}}. \tag{2.52}$$

Noting that $s_6(t) = s_1(t)/t^{\delta_2}$ is an increasing function in the interval $[t_2, +\infty)$, for any $k \ge t_2$, we have

$$\sum_{l=t_2}^{k}\frac{s_1(l)}{l^{\delta_2}} \le \int_{t_2}^{k+1}\frac{s_1(t)}{t^{\delta_2}}dt. \tag{2.53}$$

We have

$$\int_{t_2}^{k+1}\frac{s_1(t)}{t^{\delta_2}}dt = \int_{t_2}^{k+1}\frac{1}{a_1 t^{\delta_2-\delta_1}}ds_1(t)$$

$$
\begin{aligned}
&= \frac{s_1(k+1)}{a_1(k+1)^{\delta_2-\delta_1}} - \frac{s_1(t_2)}{a_1 t_2^{\delta_2-\delta_1}} + \int_{t_2}^{k+1} \frac{(\delta_2-\delta_1)s_1(t)}{a_1 t^{\delta_2-\delta_1+1}} dt \\
&\le \frac{s_1(k+1)}{a_1(k+1)^{\delta_2-\delta_1}} + \int_{t_2}^{k+1} \frac{(\delta_2-\delta_1)}{a_1 t^{1-\delta_1}} \frac{s_1(t)}{t^{\delta_2}} dt \\
&\le \frac{s_1(k+1)}{a_1(k+1)^{\delta_2-\delta_1}} + \frac{\delta_2-\delta_1}{a_1 t_2^{1-\delta_1}} \int_{t_2}^{k+1} \frac{s_1(t)}{t^{\delta_2}} dt \\
&\le \frac{s_1(k+1)}{a_1(k+1)^{\delta_2-\delta_1}} + \frac{\delta_2-\delta_1}{\delta_2} \int_{t_2}^{k+1} \frac{s_1(t)}{t^{\delta_2}} dt, \qquad (2.54)
\end{aligned}
$$

where the second inequality holds since $s_7(t) = 1/t^{1-\delta_1}$ is a decreasing function in the interval $[1, +\infty)$; and the last inequality holds due to $t_2^{1-\delta_1} \ge \frac{\delta_2}{a_1}$.

From (2.53) and (2.54), for any $k \ge t_2$, we have

$$
\sum_{l=t_2}^{k} \frac{s_1(l)}{l^{\delta_2}} \le \int_{t_2}^{k+1} \frac{s_1(t)}{t^{\delta_2}} dt \le \frac{\delta_2 s_1(k+1)}{a_1\delta_1(k+1)^{\delta_2-\delta_1}}. \qquad (2.55)
$$

From (2.51), (2.52), and (2.55), we have

$$
\sum_{l=0}^{k-2}\prod_{\tau=l+1}^{k-1}(1-r_{1,\tau})r_{2,l} \le \frac{(\frac{t_1+1}{t_1})^{\delta_2}a_2}{s_1(k+t_1)}\Big(\frac{[t_2-1-t_1]_+ s_1(t_1+1)}{(t_1+1)^{\delta_2}} + \frac{\mathbf{1}_{(k+t_1-1\ge t_2)}\delta_2 s_1(k+t_1)}{a_1\delta_1(k+t_1)^{\delta_2-\delta_1}}\Big). \qquad (2.56)
$$

Then, (2.47), (2.50), and (2.56) yield (2.41).

(ii) When $\delta_1 = 1$, from (2.50) and (2.40d), we have

$$
\begin{aligned}
\sum_{l=0}^{k-2}\prod_{\tau=l+1}^{k-1}(1-r_{1,\tau})r_{2,l} &\le \sum_{l=0}^{k-2} \frac{(l+t_1+1)^{a_1}}{(k+t_1)^{a_1}} \frac{a_2}{(l+t_1)^{\delta_2}} \\
&\le \sum_{l=0}^{k-2} \frac{(l+t_1+1)^{a_1}}{(k+t_1)^{a_1}} \frac{a_2}{(\frac{t_1}{t_1+1}l+t_1)^{\delta_2}} \\
&= \frac{(\frac{t_1+1}{t_1})^{\delta_2}a_2}{(k+t_1)^{a_1}} \sum_{l=0}^{k-2} \frac{(l+t_1+1)^{a_1}}{(l+t_1+1)^{\delta_2}} \\
&= \frac{(\frac{t_1+1}{t_1})^{\delta_2}a_2}{(k+t_1)^{a_1}} \sum_{l=t_1+1}^{k+t_1-1} l^{a_1-\delta_2}, \qquad (2.57)
\end{aligned}
$$

where the first inequality holds due to (2.50) and (2.40d).

From (2.47), (2.50), (2.57), and (2.37), we have (2.43).

(iii) Denote $a = 1 - a_1$. From (2.40c) and $\delta_1 = 0$, we know that $a_1 \in (0, 1)$. Thus, $a \in (0, 1)$.

From (2.40a)–(2.40d) and $\delta_1 = 0$, for any $k \in \mathbb{N}_+$, it holds that

$$
z_k \le (1-a_1)^k z_0 + \sum_{\tau=0}^{k-1}(1-a_1)^{k-1-\tau} r_{2,\tau} \le a^k z_0 + a_2 a^{k+t_1-1} \sum_{\tau=0}^{k-1} \frac{1}{(\tau+t_1)^{\delta_2} a^{\tau+t_1}}. \qquad (2.58)
$$

We have

$$\sum_{\tau=0}^{k-1}\frac{1}{(\tau+t_1)^{\delta_2}a^{\tau+t_1}}=\sum_{\tau=t_1}^{k+t_1-1}\frac{1}{\tau^{\delta_2}a^{\tau}}=\sum_{\tau=t_1}^{t_3-1}s_3(\tau)+\sum_{\tau=t_3}^{t_4-1}s_3(\tau)+\sum_{\tau=t_4}^{k+t_1-1}s_3(\tau). \tag{2.59}$$

We know that $s_3(t)=1/(t^{\delta_2}a^t)$ is decreasing and increasing in the intervals $[1, t_3-1]$ and $[t_3, +\infty)$, respectively, since

$$\begin{aligned}\frac{ds_3(t)}{dt}&=-s_3(t)\Big(\frac{\delta_2}{t}+\ln(a)\Big)\le 0,\ \forall t\in\Big(0,\frac{-\delta_2}{\ln(a)}\Big],\\ \frac{ds_3(t)}{dt}&=-s_3(t)\Big(\frac{\delta_2}{t}+\ln(a)\Big)\ge 0,\ \forall t\in\Big[\frac{-\delta_2}{\ln(a)},+\infty\Big).\end{aligned}$$

Thus, we have

$$\sum_{\tau=k_1}^{t_3-1}s_3(\tau)\le(t_3-k_1)s_3(k_1),\ \forall k_1\in[1,t_3-1], \tag{2.60a}$$

$$\sum_{\tau=k_2}^{t_4-1}s_3(\tau)\le(t_4-k_2)s_3(t_4),\ \forall k_2\in[t_3,t_4-1], \tag{2.60b}$$

$$\sum_{\tau=t_4}^{k_3}s_3(\tau)\le\int_{t_4}^{k_3+1}s_3(t)dt,\ \forall k_3\ge t_4. \tag{2.60c}$$

Denote $b=1/a$. We have

$$\begin{aligned}\int_{t_4}^{k_3+1}s_3(t)dt&=\int_{t_4}^{k_3+1}\frac{b^t}{t^{\delta_2}}dt=\int_{t_4}^{k_3+1}\frac{1}{\ln(b)t^{\delta_2}}db^t\\ &=\frac{b^{k_3+1}}{\ln(b)(k_3+1)^{\delta_2}}-\frac{b^{t_4}}{\ln(b)t_4^{\delta_2}}+\int_{t_4}^{k_3+1}\frac{\delta_2 b^t}{\ln(b)t^{\delta_2+1}}dt\\ &\le\frac{b^{k_3+1}}{\ln(b)(k_3+1)^{\delta_2}}+\int_{t_4}^{k_3+1}\frac{\delta_2}{\ln(b)t}s_3(t)dt\\ &\le\frac{b^{k_3+1}}{\ln(b)(k_3+1)^{\delta_2}}+\frac{\delta_2}{\ln(b)t_4}\int_{t_4}^{k_3+1}s_3(t)dt\\ &\le\frac{b^{k_3+1}}{\ln(b)(k_3+1)^{\delta_2}}+\frac{1}{2}\int_{t_4}^{k_3+1}s_3(t)dt,\end{aligned} \tag{2.61}$$

where the last inequality holds due to $t_4=\lceil -2\delta_2/\ln(1-a_1)\rceil\ge -2\delta_2/\ln(1-a_1)=2\delta_2/\ln(b)$.

From (2.60c) and (2.61), we have

$$\sum_{\tau=t_4}^{k_3}s_3(\tau)\le\frac{2}{-\ln(a)(k_3+1)^{\delta_2}a^{k_3+1}},\ \forall k_3\ge t_4. \tag{2.62}$$

From (2.58), (2.59), (2.60a), (2.60b), and (2.62), we get (2.45). □

Lemma 2.21. *Let $a \in (0,1)$ be a constant, then*

$$(1-a)^T \leq \frac{k!}{(aT)^k}, \ \forall k, T \in \mathbb{N}_0. \tag{2.63}$$

Proof. For any constant $a \in (0,1)$, we have $\ln(1-a) \leq -a$. Thus,

$$(1-a)^T \leq e^{-aT}, \ \forall T \in \mathbb{N}_0. \tag{2.64}$$

For any constant $x > 0$, we have $e^x > \frac{x^k}{k!}$, $\forall k \in \mathbb{N}_0$. This result together with (2.64) yields (2.63). □

Lemma 2.22. *For any constants $\theta \in [0,1]$ and $\kappa \in [0,1)$, it holds that*

$$(t+1)^\kappa \left(\frac{1}{t^\theta} - \frac{1}{(t+1)^\theta}\right) \leq \frac{1}{t}, \ \forall t \in \mathbb{N}_+. \tag{2.65}$$

Proof. Denote $h_t(\theta) = \frac{1}{t^\theta} - \frac{1}{(t+1)^\theta}$. Then, for any fixed $t \in \mathbb{N}_+$, $\max_{\theta \in [0,1]}\{h_t(\theta)\} = h_t(1)$ due to $\frac{dh_t(\theta)}{d\theta} \geq 0$, $\forall \theta \in [0,1]$. Hence, $(t+1)^\kappa h_t(\theta) \leq (t+1)^\kappa h_t(1) = \frac{(t+1)^\kappa}{t(t+1)} \leq \frac{1}{t}$, i.e., (2.65) holds. □

Part I

Distributed Nonconvex Optimization

Chapter 3

Distributed primal–dual first-order and ADMM algorithms

This and the following two chapters consider the distributed nonconvex optimization problem under different information feedback settings. In this chapter, we consider the full-information feedback setting, i.e., each agent knows the true gradient and even the explicit expression of its local cost function. We propose three algorithms: a distributed primal–dual first-order (FO) algorithm, a distributed alternating direction method of multipliers (ADMM) algorithm, and a distributed linearized ADMM (L-ADMM) algorithm. We show that each algorithm converges to a stationary point with an $\mathcal{O}(1/T)$ convergence rate if each local cost function is smooth, where T is the total number of iterations, and to a global optimum with a linear convergence rate under an additional condition that the global cost function satisfies the P–Ł condition. This condition is weaker than strong convexity, which is a standard condition in the literature for proving linear convergence of distributed optimization algorithms, and the global minimizer is not necessarily unique or finite. The theoretical results are illustrated by numerical simulations.

This chapter is organized as follows. Section 3.1 gives the background. Section 3.2 presents problem formulation and assumptions. Sections 3.3–3.5 provide the distributed primal–dual FO algorithm, the distributed ADMM algorithm, and the distributed linearized ADMM algorithm, respectively, and analyze their convergence properties. Simulations are given in Section 3.6. Concluding remarks are offered in Section 3.7. To improve the readability, all the proofs can be found in Section 3.8

3.1 Introduction

In the study of distributed optimization, a standard assumption for proving linear convergence of existing algorithms, such as [68–91], is strong convexity of the cost functions. Unfortunately, some practical applications, such as least squares, do not have strongly convex cost functions [156]. This situation has motivated researchers to consider alternatives to strong convexity. There are some results in centralized optimization. For instance, in [218], the authors derived linear convergence of several centralized first-order algorithms for smooth and constrained optimization problems when cost functions are convex and satisfy the quadratic functional growth condition; and in [209], the authors showed linear convergence of centralized gradient algorithms for smooth optimization problems when cost functions satisfy the P–Ł condition which is weaker than the conditions assumed in [218].

There also are some results in distributed optimization [204, 219–223]. Specifically, in [219], the authors proposed the distributed exact first-order algorithm (EXTRA) to solve the smooth convex optimization and proved linear convergence under the conditions that the global cost function is restricted strongly convex and the optimal set is a singleton, which are stronger than the P–Ł condition. The authors of [220, 221] later extended the results in [219] to directed graphes. In [204], the authors proposed a continuous-time distributed heavy-ball algorithm with event-triggered communication to solve the smooth convex optimization and proved exponential convergence under the same conditions as that assumed in [219]. In [222], the authors established linear convergence of the distributed primal–dual gradient descent algorithm for solving the smooth convex optimization under the condition that the primal–dual gradient map is metrically subregular, which is different from the P–Ł condition but weaker than strong convexity. In [223], the authors proposed a distributed primal–dual gradient descent algorithm to solve the smooth optimization problem and established linear convergence under the assumptions that the global cost function satisfies the restricted secant inequality (RSI) condition and the gradients of each local cost function at optimal points are the same, which are also stronger than the P–Ł condition.

Among existing optimization algorithms, ADMM is very effective at numerically solving many practical convex and nonconvex optimization problems [93,224,225] and has wide applications in areas such as signal processing [226], power systems [227], optimal control [228], and computer version [229]. This has motivated researchers to consider distributed ADMM algorithms. If cost functions are convex, many distributed ADMM algorithms have been proposed, e.g., [71–73,76,90,230–237]. The convergence property of these algorithms has also been analyzed, for instance, the $\mathcal{O}(1/T)$ and linear convergence rates were established in [230, 231, 234] and [71–73, 76, 90, 232], respectively, where T is the total number of iterations.

However, when cost functions are nonconvex, existing distributed ADMM algorithms with provable convergence analysis normally require that the communication network is a star graph, i.e., hub–leaf topology. For instance, the authors of [238–240] proposed star graph based distributed ADMM algorithms and proved that first-order stationary points can be found with an $\mathcal{O}(1/T)$ convergence rate when each local cost function is smooth. One advantage of these algorithms is that they are asynchronous. However, in addition to the star graph restriction, the algorithms proposed in [238, 239] require that each leaf agent communicates both primal and dual variables to the hub agent. Moreover, the algorithm proposed in [240] is based on the standard master–worker architecture. Specifically, the master (hub agent) executes all of the updatings, while each worker (leaf agent) only computes the gradient of its own local cost function and sends it to the master. In other words, all decisions are made by a single agent, the master, which suffers from a single point of failure, high communication and computation cost, etc. To the best of our knowledge, the distributed proximal primal–dual algorithm (Prox-PDA) proposed in [112], which is a generalization of the distributed ADMM algorithms proposed in [71, 90], is the only distributed ADMM algorithm with provable convergence analysis to solve nonconvex optimization problem when communication network is arbitrarily connected. Through a lower bounded potential function, it was shown that the Prox-PDA algorithm finds a first-order stationary point with an $\mathcal{O}(1/T)$ convergence rate when each local cost function is smooth. To the best of our knowledge, there are no results to guarantee a global optimum can be linearly found by distributed ADMM algorithms when cost functions are nonconvex.

Noting above, two core theoretical questions with important practical relevance arise.

(Q3.1) As shown in [209], when strong convexity is replaced by the P–Ł condition, centralized FO algorithms still can linearly find a global optimum. Does this hold for distributed FO algorithms?

(Q3.2) Are there any distributed ADMM algorithms that not only are suitable for arbitrarily connected communication networks, but also linearly find a global optimum when the P–Ł condition holds in addition?

This chapter provides positive answers to the above two questions. We first propose a distributed primal–dual FO algorithm (Algorithm 3.1) and have the following contributions.

(C3.1) When each local cost function is smooth, we appropriately chose the algorithm parameters and construct a Lyapunov function for the proposed algorithm. With this Lyapunov function, we show in Theorem 3.1 that the proposed distributed FO algorithm finds a first-order stationary point with an $\mathcal{O}(1/T)$ convergence rate and that the cost difference between the global optimum and the resulting stationary point is bounded.

(C3.2) With the same Lyapunov function, we show in Theorem 3.2 that not only the proposed algorithm can find a global optimum but also the convergence rate is linear under an additional assumption that the global cost function satisfies the P–Ł condition, thus (Q3.1) is answered. The P–Ł condition is weaker than the (restrict) strong convexity condition assumed in [68–74, 76–91, 204, 219–221, 223] since it does not require convexity and the global minimizer is not necessarily unique. This condition is also different from the metric subregularity criterion assumed in [222]. In other words, we show that for a larger class of cost functions than strongly convex functions, the global optimum can be founded linearly by the proposed distributed algorithm.

Motivated from the classic ADMM algorithm, we then propose a distributed ADMM algorithm (Algorithm 3.2). We have the following contributions.

(C3.3) The proposed distributed ADMM algorithm is suitable for arbitrarily connected communication networks, not necessarily a star graph.

(C3.4) With another Lyapunov function, we show that it has the same theoretical convergence properties as our distributed FO algorithm under the same conditions. Specifically, we show in Theorems 3.3 and 3.4 that the proposed distributed ADMM algorithm converges to a first-order stationary point with an $\mathcal{O}(1/T)$ convergence rate if each local cost function is smooth and to a global optimum with linear convergence rate when the global cost function satisfies the P–Ł condition in addition, thus (Q3.2) is answered.

(C3.5) In order to reduce computation burden on each agent from solving an local optimization problem at each iteration, we also propose a distributed L-ADMM algorithm (Algorithm 3.3), derived from the proposed distributed ADMM algorithm by linearizing the local cost function at each iteration. We show in Theorems 3.5 and 3.6 that the proposed distributed L-ADMM algorithm has the same theoretical convergence properties as the proposed distributed ADMM algorithm under the same conditions.

Table 3.1 compares this chapter with other algorithms that obtain linear convergence for distributed optimization. Table 3.2 summarizes the comparison on distributed nonconvex optimization.

3.2 Distributed nonconvex optimization with full-information feedback

Consider a network of n agents, each of which has a local cost function $f_i : \mathbb{R}^p \to \mathbb{R}$. All agents collaborate to solve the optimization problem

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{3.1}$$

In this chapter, we consider the full-information feedback setting. In other words, each agent i knows the true gradient $\nabla f_i(x)$ and even the explicit expression of $f_i(x)$.

Based on the definitions introduced in Chapter 2, the following assumptions are made.

Assumption 3.1. *The communication among agents is described by a weighted undirected connected graph $\mathcal{G}$.*

Assumption 3.2. *The set $\mathbb{X}^*$ is nonempty and $f^* > -\infty$, where $\mathbb{X}^*$ and f^* denote the optimal set and the minimum function value of the optimization problem* (3.1)*, respectively.*

Assumption 3.3. *Each local cost function $f_i(x)$ is smooth with constant $L_f > 0$.*

Assumption 3.4. *The global cost function $f(x)$ satisfies the P–Ł condition with constant $\nu > 0$.*

Remark 3.1. *Assumptions 3.1–3.3 are common in the literature, e.g., [71, 219]. Assumption 3.4 is weaker than the assumption that the global or each local cost function is*

Table 3.1: Comparison of Chapter 3 to some related distributed optimization algorithms obtaining linear convergence.

Reference	Cost function	Communication strategy	Communication type
[68]	Strongly convex f_i, locally Lipschitz $\nabla^2 f$	Connected undirected, one variable	Continuous-time
[69]	Strongly convex and smooth f_i	Connected undirected, one variable	Event-triggered
[70–75]	Strongly convex and smooth f_i	Connected undirected, one variable	Discrete-time
[76]	Strongly convex and smooth f_i, Lipschitz $\nabla^2 f$	Connected undirected, one variable	Discrete-time
[77]	Strongly convex and smooth f_i	Uniformly jointly strongly connected, two variables	Discrete-time
[78, 79]	Strongly convex and smooth f_i	Connected undirected, three variables	Discrete-time
[80–82]	Strongly convex and smooth f_i	Connected undirected, two variables	Discrete-time
[83–86]	Strongly convex and smooth f_i	Strongly connected, three variables	Discrete-time
[87]	Strongly convex and smooth f_i	Undirected stochastic graphs with random failures, two variables	Discrete-time
[88]	Convex and smooth f_i, strongly convex f	Connected undirected, four variables	Discrete-time
[89]	Convex and smooth f_i, strongly convex f	Uniformly jointly strongly connected, two variables	Discrete-time
[90]	Convex and smooth f_i, strongly convex f	Connected undirected, one variable	Discrete-time
[91]	Smooth f_i, strongly convex f	Uniformly jointly strongly connected with delays, five variables	Discrete-time
[219]	Convex and smooth f_i, restricted strongly convex f, unique x^*	Connected undirected, one variable	Discrete-time
[220, 221]	Convex and smooth f_i, restricted strongly convex f, unique x^*	Strongly connected, two variables	Discrete-time
[204]	Convex and smooth f_i, restricted strongly convex f, unique x^*	Connected undirected, one variable	Event-triggered
[222]	Convex and smooth f_i, the primal–dual gradient map is metric subregularity	Connected undirected, two variables	Discrete-time
[223]	Smooth f_i, f satisfies the RSI condition, $\{\nabla f_i(x^*)\}$ is a singleton	Connected undirected, one variable	Discrete-time
[232]	Convex f_i, unique x^*, $\nabla^2 f(x^*) > 0$	Connected undirected, one variable	Discrete-time
This chapter	Smooth f_i, f satisfies the P–Ł condition	Connected undirected, one variable	Discrete-time

strongly convex. It should be highlighted that the convexity of the cost functions and the boundedness of their gradients are not assumed. Moreover, we do not assume that $\mathbb{X}^*$ is a singleton or finite set either.

Our goal in this chapter is to answer (Q3.1) and (Q3.2), i.e., solve the following problem.

Problem 3.1. *Propose distributed FO and ADMM algorithms for the nonconvex optimization problem* (3.1) *such that the global optimum can be linearly found.*

Table 3.2: Comparison of Chapter 3 to some related distributed nonconvex optimization algorithms.

Reference	Cost function	Communication strategy	Convergence rate
[111]	Lipschitz and smooth f_i, the set of stationary points is a union of finitely many connected components, no saddle points	Uniformly jointly strongly connected, two variables	$O(1/T)$ to a local optimum
[110]	Lipschitz f_i	Connected undirected, one variable	Asymptotic
[112–115]	Smooth f_i	Connected undirected, one variable	$O(1/T)$
[117]	Smooth f_i, Lipschitz $\nabla^2 f$, f satisfies the K–Ł condition, $p = 1$	Connected undirected, one variable	Almost surely to an SOS solution
[116]	Smooth f_i	Strongly connected, two variables	$O(1/T)$
	Smooth f_i, special initialization	Connected undirected or strongly connected with $p = 1$, special weight matrices, two variables	Almost surely to an SOS solution
This chapter	Smooth f_i	Connected undirected, one variable	$O(1/T)$
	Smooth f_i, f satisfies the P–Ł condition		Linearly to a global optimum

3.3 Distributed primal–dual FO algorithm

In this section, we consider the situation that agent i knows the true gradient $\nabla f_i(x)$. We propose a distributed primal–dual FO algorithm to solve the optimization problem (3.1) and analyze its convergence rates under different conditions.

3.3.1 Algorithm description

In this section, we present the derivation of our proposed algorithm.

For simplicity, denote $\boldsymbol{x} = \mathrm{col}(x_1, \dots, x_n)$, $\tilde{f}(\boldsymbol{x}) = \sum_{i=1}^n f_i(x_i)$, and $\boldsymbol{L} = L \otimes \boldsymbol{I}_p$, where L is the Laplacian matrix of the communication graph $\mathcal{G}$. The optimization problem (3.1) is equivalent to the constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^{np}} \quad & \tilde{f}(\boldsymbol{x}) = \sum_{i=1}^n f_i(x_i) \\ \text{s.t.} \quad & x_i = x_j, \ \forall i, j \in [n]. \end{aligned} \tag{3.2}$$

Noting that the Laplacian matrix L is positive semi-definite and $\mathrm{null}(L) = \{\mathbf{1}_n\}$ when $\mathcal{G}$ is connected, we know that the optimization problem (3.2) is equivalent to the constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^{np}} \quad & \tilde{f}(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{L}^{1/2}\boldsymbol{x} = \mathbf{0}_{np}. \end{aligned} \tag{3.3}$$

Here, we use $\boldsymbol{L}^{1/2}\boldsymbol{x} = \mathbf{0}_{np}$ rather than $\boldsymbol{L}\boldsymbol{x} = \mathbf{0}_{np}$ as the constraint since they are both equivalent to $\boldsymbol{x} = \mathbf{1}_n \otimes x$ but the first has a particular property which will be discussed in Remark 3.6.

Let $\boldsymbol{u} \in \mathbb{R}^{np}$ denote the dual variable, then the augmented Lagrangian function associated with (3.3) is

$$\mathcal{A}(\boldsymbol{x},\boldsymbol{u}) = \tilde{f}(\boldsymbol{x}) + \frac{\alpha}{2}\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} + \beta\boldsymbol{u}^\top \boldsymbol{L}^{1/2}\boldsymbol{x}, \tag{3.4}$$

where $\alpha > 0$ and $\beta > 0$ are the regularization parameters.

Based on the primal–dual gradient method, a distributed FO algorithm to solve (3.3) is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k + \beta\boldsymbol{L}^{1/2}\boldsymbol{u}_k + \nabla\tilde{f}(\boldsymbol{x}_k)), \tag{3.5a}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \eta\beta\boldsymbol{L}^{1/2}\boldsymbol{x}_k,\ \forall\boldsymbol{x}_0,\ \boldsymbol{u}_0 \in \mathbb{R}^{np}, \tag{3.5b}$$

where $\eta > 0$ is a fixed stepsize. Denote $\boldsymbol{v}_k = \mathrm{col}(v_{1,k},\dots,v_{n,k}) = \boldsymbol{L}^{1/2}\boldsymbol{u}_k$, then the algorithm (3.5) can be rewritten as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k + \beta\boldsymbol{v}_k + \nabla\tilde{f}(\boldsymbol{x}_k)), \tag{3.6a}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \eta\beta\boldsymbol{L}\boldsymbol{x}_k,\ \forall\boldsymbol{x}_0 \in \mathbb{R}^{np},\ \sum_{j=1}^{n} v_{j,0} = \boldsymbol{0}_p. \tag{3.6b}$$

The initialization condition $\sum_{j=1}^{n} v_{j,0} = \boldsymbol{0}_p$ is derived from $\boldsymbol{v}_0 = \boldsymbol{L}^{1/2}\boldsymbol{u}_0$, and it is easy to be satisfied, for example $v_{i,0} = \boldsymbol{0}_p,\ \forall i \in [n]$ or $v_{i,0} = \sum_{j=1}^{n} L_{ij}x_{i,0},\ \forall i \in [n]$. It is straightforward to verify that the algorithm (3.6) is equivalent to the EXTRA algorithm proposed in [219] with mixing matrices $\boldsymbol{W} = \boldsymbol{I}_{np} - \eta\alpha\boldsymbol{L}$ and $\tilde{\boldsymbol{W}} = \boldsymbol{I}_{np} - \eta\alpha\boldsymbol{L} + \eta^2\beta^2\boldsymbol{L}$. Note that the distributed algorithm (3.6) can also be written agent-wise as

$$x_{i,k+1} = x_{i,k} - \eta\Big(\alpha\sum_{j=1}^{n} L_{ij}x_{j,k} + \beta v_{i,k} + \nabla f_i(x_{i,k})\Big), \tag{3.7a}$$

$$v_{i,k+1} = v_{i,k} + \eta\beta\sum_{j=1}^{n} L_{ij}x_{j,k},\ \forall x_{i,0} \in \mathbb{R}^p,\ \sum_{j=1}^{n} v_{j,0} = \boldsymbol{0}_p,\ \forall i \in [n]. \tag{3.7b}$$

We present the distributed primal–dual FO algorithm (3.7) in pseudo-code as Algorithm 3.1.

Remark 3.2. *In the literature, various distributed first-order algorithms have been proposed to solve the nonconvex optimization problem* (3.1)*. For example, the distributed gradient descent algorithm was proposed in [110, 116]; the distributed gradient tracking algorithm was proposed in [116]; and a distributed algorithm based on a novel approximate filtering-then-predict and tracking (xFILTER) strategy was proposed in [114]. Compared with the proposed distributed algorithm* (3.7)*, these algorithms have some potential drawbacks. For the distributed gradient algorithm, existing studies, such as [110, 116], only showed that the output of the algorithm converges to a neighborhood of a stationary point unless additional assumptions, such as the boundedness of the gradients of cost functions, are assumed. In the distributed gradient tracking algorithm [116], at*

Algorithm 3.1 Distributed Primal–Dual FO Algorithm

Input: parameters $\alpha > 0, \beta > 0$, and $\eta > 0$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p, \ \forall i \in [n]$.
for $k = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, n$ in parallel **do**
    Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
    Update $x_{i,k+1}$ by (3.7a);
    Update $v_{i,k+1}$ by (3.7b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

each iteration each agent i needs to communicate one additional p-dimensional variables besides the communication of $x_{i,k}$ with its neighbors. The xFILTER algorithm proposed in [114] is a double-loop algorithm and thus at each iteration it requires more communication and computation than the proposed distributed algorithm (3.7).

3.3.2 Convergence analysis

In this section, we provide convergence analysis for Algorithm 3.1.

Find stationary points

Let us consider the case when Algorithm 5.1 is able to find stationary points. We have the following convergence results.

Theorem 3.1. *Suppose that Assumptions 3.1–3.3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.1 with $\alpha \in (\beta + \kappa_1, \kappa_2\beta]$, $\beta > c_\beta$, and $\eta \in (0, c_\eta)$, where κ_1, κ_2, c_β, and c_η are constants given in Section 3.8.1. Then, for any $T \in \mathbb{N}_+$,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 = \mathcal{O}(\frac{1}{T}), \tag{3.8}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\nabla f(\bar{x}_k)\|^2 = \mathcal{O}(\frac{1}{T}), \tag{3.9}$$

$$f(\bar{x}_T) - f^* = \mathcal{O}(1), \tag{3.10}$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$.

Proof. The explicit expressions of the right-hand sides of (3.8)–(3.10) and the proof are given in Section 3.8.1. □

Remark 3.3. *This same convergence rate as stated in* (3.9) *has also been achieved by the distributed gradient tracking algorithm proposed in [116] and the xFILTER algorithm* *proposed in [114] under the same assumptions on the cost functions. However, as discussed in Remark 3.2, at each iteration, the distributed gradient tracking algorithm requires double amount of communication and the xFILTER algorithm requires more communication as well as more computation.*

Find global optima

Let us next consider the case when Algorithm 3.1 finds global optima. We have the following convergence results.

Theorem 3.2. *Suppose that Assumptions 3.1–3.4 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.1 with the same α, β, and η used in Theorem 3.1, then*

$$\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2 + f(\bar{x}_k) - f^* \leq \epsilon_0\epsilon^k,\ \forall k \in \mathbb{N}_0, \tag{3.11}$$

where $\epsilon_0 > 0$ and $\epsilon \in (0,1)$ given in Section 3.8.2.

Proof. The proof is given in Section 3.8.2. □

Remark 3.4. *The proofs of Theorems 3.1 and 3.2 are based on the same appropriately designed Lyapunov function given in Lemma 3.1 in Section 3.8.1. In the literature that considered distributed nonconvex optimization, e.g., [112–115, 117], the lower bounded potential functions (which may be negative) are commonly used to analyze the convergence properties of the proposed algorithms. So the analysis in those studies cannot be extended to show linear convergence when the P–Ł condition holds since the lower bounded potential functions may not be Lyapunov functions. In the literature that obtained linear convergence for distributed optimization, e.g., [68–91, 204, 219–223, 232], the convexity and/or the uniqueness of the global minimizer are the key in the analysis. So the analysis in those studies cannot be extended to show linear convergence when strong convexity is relaxed by the P–Ł condition since the later does not imply convexity of cost functions and uniqueness of global minimizers.*

Remark 3.5. *The distributed first-order algorithms proposed in [68–91, 204, 219–223] also established linear convergence. However, in [68–87], it was assumed that each local cost function is strongly convex. In [88–90], it was assumed that each local cost function is convex and the global cost function is strongly convex. In [91], it was assumed that the global cost function is strongly convex. In [204, 219], it was assumed that each local cost function is convex, the global cost function is restricted strongly convex, and $\mathbb{X}^*$ is a singleton. In [220, 221], it was assumed that each local cost function is restricted strongly convex and the optimal set $\mathbb{X}^*$ is a singleton. In [222], it was assumed that each local cost function is convex and the primal–dual gradient map is metrically subregular. In [223], it was assumed that the global cost function satisfies the restricted secant inequality condition and the gradients of each local cost function at optimal points are the same. In contrast, the linear convergence result established in Theorem 3.2 only requires that* *the global cost function satisfies the P–Ł condition, but the convexity assumption on cost functions and the singleton assumption on the optimal set and the set of the gradients of each local cost function at optimal points are not required. Moreover, it should be highlighted that when executing Algorithm 3.1 the P–Ł constant ν is not needed. Compared with some of the aforementioned studies, one potential drawback is that we assume the communication graph is static and undirected. We leave the extension to time-varying directed graph for future work.*

Remark 3.6. *If we use $\boldsymbol{L}\boldsymbol{x} = \mathbf{0}_{np}$ as the constraint in* (3.3)*, then we could construct an alternative distributed primal–dual FO algorithm*

$$x_{i,k+1} = x_{i,k} - \eta\Big(\sum_{j=1}^{n} L_{ij}(\alpha x_{j,k} + \beta v_{j,k}) + \nabla f_i(x_{i,k})\Big), \tag{3.12a}$$

$$v_{i,k+1} = v_{i,k} + \eta\beta \sum_{j=1}^{n} L_{ij} x_{j,k}, \ \forall x_{i,0}, \ v_{i,0} \in \mathbb{R}^p. \tag{3.12b}$$

Similar results as shown in Theorems 3.1 and 3.2 can be obtained. We omit the details due to the similarity. Different from the requirement that $v_{i,0} = \mathbf{0}_p$ in the algorithm (3.7)*, $v_{i,0}$ can be arbitrarily chosen in the algorithm* (3.12)*. In other words, the algorithm* (3.12) *is robust to the initial condition $v_{i,0}$. However, it requires additional communication of $v_{j,k}$ in* (3.12a)*, compared to* (3.7)*.*

3.4 Distributed ADMM algorithm

In this section, we consider the situation that each agent i knows the explicit expression of $f_i(x)$. We propose a distributed ADMM algorithm and analyze its convergence properties under different conditions.

3.4.1 Algorithm description

Note that the optimization problem (3.1) is equivalent to the constrained problem

$$\begin{aligned} \min_{x_i,\, x_0 \in \mathbb{R}^p} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} \quad & \beta x_i = \beta x_0, \ \forall i \in [n], \end{aligned} \tag{3.13}$$

where $\beta > 0$ is a constant.

If there exists a virtual agent, denoted as agent 0, which can communicate with all of the n agents[1], then the optimization problem (3.13) can be efficiently solved by the classic ADMM algorithm [93, 224]. Specifically, the classic ADMM algorithm to solve (3.13) is

$$x_{0,k+1} = \frac{1}{n}\sum_{i=1}^{n}(x_{j,k} + \frac{\beta}{\gamma} v_{j,k}), \tag{3.14a}$$

[1] This corresponds to that the communication graph $\mathcal{G}$ of the n agents is a star graph.

$$x_{i,k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^p} f_i(x) + \beta\langle v_{i,k}, x\rangle + \frac{\gamma}{2}\|x - x_{0,k+1}\|^2, \tag{3.14b}$$

$$v_{i,k+1} = v_{i,k} + \frac{\gamma}{\beta}(x_{i,k+1} - x_{0,k+1}),\ \forall i \in [n], \tag{3.14c}$$

where $\gamma > 0$ is the penalty parameter. It has been shown in [238–240] that the classic ADMM algorithm (3.14) can find first-order stationary points of the optimization problem (3.13) with an $\mathcal{O}(1/k)$ convergence rate if γ is large enough, $\beta = 1$, and Assumptions 3.2 and 3.3 hold. If the communication graph $\mathcal{G}$ is a general connected graph, then each agent i cannot execute (3.14b) and (3.14c) since $x_{0,k+1}$ is not available in this case. Thus, the classic ADMM algorithm (3.14) is restricted to a star graph. In order to remove this restriction, we modify the classic ADMM algorithm (3.14) as follows

$$x_{i,k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^p} f_i(x) + \beta\langle v_{i,k}, x\rangle + \frac{\gamma}{2}\Big\|x - x_{i,k} + \frac{\alpha}{\gamma}\sum_{j=1}^{n} L_{ij}x_{j,k}\Big\|^2, \tag{3.15a}$$

$$v_{i,k+1} = v_{i,k} + \frac{\beta}{\gamma}\sum_{j=1}^{n} L_{ij}x_{j,k+1},\ \forall x_{i,0} \in \mathbb{R}^p,\ \sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p,\ \forall i \in [n], \tag{3.15b}$$

where $\alpha > 0$ is a constant.

Remark 3.7. *The intuition of the modification from* (3.14) *to* (3.15) *is as follows. When γ is large enough, then from* (3.14a)*, we know $x_{0,k+1} \approx \frac{1}{n}\sum_{i=1}^{n} x_{j,k}$. In multi-agent systems, for each agent i, $\frac{1}{n}\sum_{i=1}^{n} x_{j,k}$ can be estimated by $x_{i,k} - b\sum_{j=1}^{n} L_{ij}x_{j,k}$ with some positive gains b. Thus, replacing $x_{0,k+1}$ in* (3.14b) *by its estimation $x_{i,k} - \frac{\alpha}{\gamma}\sum_{j=1}^{n} L_{ij}x_{j,k}$ gives* (3.15a)*. Then, each $x_{i,k+1}$ is available to each agent i and through communication it is also available to agent j if $j \in \mathcal{N}_i$. Thus, replacing $x_{0,k+1}$ in* (3.14c) *by its estimation $x_{i,k+1} - \frac{\beta^2}{\gamma^2}\sum_{j=1}^{n} L_{ij}x_{j,k+1}$ gives* (3.15b)*. Here, we used different gains $\frac{\alpha}{\gamma}$ and $\frac{\beta^2}{\gamma^2}$ since such a setting facilitates the convergence analysis. Moreover, the extra initialization condition $\sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p$ is also used to facilitate the convergence analysis. This initialization condition is easy to be satisfied, for example, $v_{i,0} = \mathbf{0}_p,\ \forall i \in [n]$, or $v_{i,0} = \sum_{j=1}^{n} L_{ij}x_{j,0},\ \forall i \in [n]$.*

Remark 3.8. *The objective function in subproblem* (3.15a) *may be not convex since each f_i is possibly nonconvex. However, if Assumption 3.3 holds and $\gamma > L_f$, then from Lemma 2.11, we know that the objective function is strongly convex with convexity parameter $\gamma - L_f > 0$. Hence, the subproblem* (3.15a) *is solvable.*

We write the distributed ADMM algorithm (3.15) in pseudo-code as Algorithm 3.2.

3.4.2 Convergence analysis

In this section, we provide convergence analysis for Algorithm 3.2.

Find stationary points

Let us consider the case when Algorithm 3.2 is able to find stationary points. We have the following convergence results.

Algorithm 3.2 Distributed ADMM Algorithm

Input: constants $\alpha > 0, \beta > 0$, and $\gamma > 0$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p,\ \forall i \in [n]$.
Broadcast $x_{i,0}$ to $\mathcal{N}_i$ and receive $x_{j,0}$ from $j \in \mathcal{N}_i$;
for $k = 0, 1, \dots$ **do**
  for $i = 1, \dots, n$ in parallel **do**
    Update $x_{i,k+1}$ by (3.15a);
    Broadcast $x_{i,k+1}$ to $\mathcal{N}_i$ and receive $x_{j,k+1}$ from $j \in \mathcal{N}_i$;
    Update $v_{i,k+1}$ by (3.15b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

Theorem 3.3. *Suppose Assumptions 3.1–3.3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.2 with $\alpha \in (\frac{1}{\rho_2(L)}(\rho(L)\beta + \chi_1), \chi_2\beta]$, $\beta > \hat{c}_\beta$, and $\gamma > \hat{c}_\gamma$, where χ_1, χ_2, $\hat{c}_\beta$, and $\hat{c}_\gamma$ are constants given in Section 3.8.3. Then, for any $T \in \mathbb{N}_+$,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 = \mathcal{O}(\frac{1}{T}), \tag{3.16}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\nabla f(\bar{x}_k)\|^2 = \mathcal{O}(\frac{1}{T}), \tag{3.17}$$

$$f(\bar{x}_T) - f^* = \mathcal{O}(1), \tag{3.18}$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$.

Proof. The explicit expressions of the right-hand sides of (3.16)–(3.18) and the proof are given in Section 3.8.3. □

Remark 3.9. *This same convergence rate as stated in* (3.17) *has also been achieved by the Prox-PDA proposed in [112] under the same conditions. Same convergence rate has also achieved by ADMM algorithms proposed in [117, 238–240]. However, these algorithms are restricted to a star graph. Moreover, the algorithms proposed in [117,238,239] require that each leaf agent has to communicate both primal and dual variables to the hub agent, and the algorithm proposed in [240] is based on the standard master–worker architecture. Compared with these algorithms, the advantages of Algorithm 3.2 are that it is suitable for general connected graphs and each agent only needs to communicate the primal variable with its neighbors, while one potential drawback is that our algorithm is synchronous. It is unclear how to analyze the convergence rate for the proposed distributed ADMM algorithm under the asynchronous communication, so we leave this for future studies.*

Remark 3.10. *The settings on α, β, and γ in Theorem 3.3 are instrumental in the convergence analysis of Algorithm 3.2. They are just sufficient conditions. In other words,* *the lower bounds for α, β, and γ are not tight. We numerically observed that smaller α, β, and γ still guarantee the same convergence rate and even lead to faster convergence in some simulation examples. It remains an open question to analyze the convergence rate under smaller α, β, and γ.*

Find global optima

Let us next consider the case when Algorithm 3.2 finds global optima. We have the following convergence results.

Theorem 3.4. *Suppose Assumptions 3.1–3.4 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.2 with the same α, β, and γ used in Theorem 3.3, then*

$$\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2+f(\bar{x}_k)-f^*\leq\varepsilon_0\varepsilon^k,\ \forall k\in\mathbb{N}_0, \tag{3.19}$$

where $\varepsilon_0>0$ and $\varepsilon\in(0,1)$ given in Section 3.8.4.

Proof. The proof is given in Section 3.8.4. □

Remark 3.11. *Among existing literature, to the best of our knowledge, the Prox-PDA algorithm proposed in [112] is the only distributed ADMM algorithm with provable convergence analysis when cost functions are nonconvex and the communication network is arbitrarily connected. The proposed distributed ADMM algorithm* (3.15) *is closely related to the Prox-PDA algorithm. The key differences between them is on the stepsize for the dual variable updating, which facilitate us to show explicit convergence rates for our distributed ADMM algorithm by a appropriately designed Lyapunov function given in Lemma 3.2, which is modified from the one used in the proofs of Theorems 3.1 and 3.2 given in Lemma 3.1. With this Lyapunov function, we prove Theorems 3.3 and 3.4. In contrast, a lower bounded potential functions (which may be negative) was used in [112] to analyze the convergence properties of the Prox-PDA algorithm. So the analysis in [112] cannot be extended to show linear convergence when the P–Ł condition holds since the lower bounded potential functions may not be Lyapunov functions.*

Remark 3.12. *Linear convergence was also established by the distributed ADMM algorithms proposed in [71, 73, 90, 232]. However, they all assumed that each local cost function is convex. Moreover, in [71, 73], it was assumed that each local cost function is strongly convex. In [232], it was assumed that the optimal set $\mathbb{X}^*$ is a singleton and the global cost function is locally strongly convex. In [90], it was assumed that the global cost function is strongly convex. In contrast, the linear convergence result established in Theorem 3.4 only requires that assumption that the global cost function satisfies the P–Ł condition, but the convexity assumption on cost functions and the singleton assumption on the optimal set are not required. Compared with the results established in [71, 73, 90, 232], one potential drawback of our results is that we need to use some global information, such as the eigenvalues of the Laplacian matrix associated with the communication graph. It is* *unclear how to overcome this drawback. This drawback may be overcome with the studies on estimating the second smallest eigenvalue (the connectivity) of the Laplacian matrix associated with the communication graph [241, 242].*

3.5 Distributed linearized ADMM algorithm

Same as existing distributed ADMM algorithms, such as [71, 73, 90, 230–233, 235–240], one potential limitation of Algorithm 3.2 is the requirement that at each iteration each subproblem (3.15a) needs to be solved exactly, which normally has no explicit solution, and thus results in high computation burden to each agent. To over come this, in this section, we propose a distributed linearized ADMM (L-ADMM) algorithm and analyze its convergence rate under different conditions.

3.5.1 Algorithm description

In this section, we present the modification of (3.15a). The main idea is that instead of minimizing exactly with respect to x we take an inexact minimization in which the function $f_i(x)$ is replaced by a linearized approximation centered at the current iteration. Specifically, replacing the function $f_i(x)$ with $f_i(x_{i,k}) + \langle \nabla f_i(x_{i,k}), x - x_{i,k}\rangle$ in (3.15a) gives the inexact update for $x_{i,k+1}$ as follows

$$x_{i,k+1} = \operatorname*{argmin}_{x\in\mathbb{R}^p} f_i(x_{i,k}) + \langle \nabla f_i(x_{i,k}), x - x_{i,k}\rangle + \beta\langle v_{i,k}, x\rangle + \frac{\gamma}{2}\Big\| x - x_{i,k} + \frac{\alpha}{\gamma}\sum_{j=1}^{n} L_{ij}x_{j,k}\Big\|^2. \tag{3.20}$$

Noting that the objective function in the subproblem (3.20) is strongly convex, from the first-order optimality conditions for convex optimization problems, we can compute the explicit expression of $x_{i,k+1}$. Hence, we get the distributed L-ADMM algorithm

$$x_{i,k+1} = x_{i,k} - \frac{1}{\gamma}\Big(\alpha\sum_{j=1}^{n} L_{ij}x_{j,k} + \beta v_{i,k} + \nabla f_i(x_{i,k})\Big), \tag{3.21a}$$

$$v_{i,k+1} = v_{i,k} + \frac{\beta}{\gamma}\sum_{j=1}^{n} L_{ij}x_{j,k+1},\ \forall x_{i,0}\in\mathbb{R}^p,\ \sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p,\ \forall i\in[n], \tag{3.21b}$$

We write the distributed L-ADMM algorithm (3.21) in pseudo-code as Algorithm 3.3.

Remark 3.13. *It is straightforward to see that the distributed L-ADMM algorithm* (3.21) *is similar to the distributed primal–dual FO algorithm* (3.7)*. The main difference between them is the updating of the local dual variable $v_{i,k+1}$. In* (3.21b)*, $\{x_{j,k+1}\}$ are used, while in* (3.7b)*, $\{x_{j,k}\}$ are used. This difference results in different designs of algorithm parameters and Lyapunov functions to analyze convergence rates, although they have the same convergence properties.*

Algorithm 3.3 Distributed L-ADMM Algorithm

Input: constants $\alpha > 0, \beta > 0$, and $\gamma > 0$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p,\ \forall i \in [n]$.
Broadcast $x_{i,0}$ to $\mathcal{N}_i$ and receive $x_{j,0}$ from $j \in \mathcal{N}_i$;
for $k = 0, 1, \dots$ **do**
  for $i = 1, \dots, n$ in parallel **do**
    Update $x_{i,k+1}$ by (3.21a);
    Broadcast $x_{i,k+1}$ to $\mathcal{N}_i$ and receive $x_{j,k+1}$ from $j \in \mathcal{N}_i$;
    Update $v_{i,k+1}$ by (3.21b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

3.5.2 Convergence analysis

In this section, we provide convergence analysis for Algorithm 3.3.

Find stationary points

Similar to Theorems 3.1 and 3.3, we have the following convergence result.

Theorem 3.5. *Suppose Assumptions 3.1–3.3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.3 with $\alpha \in (\frac{1}{\rho_2(L)}(\rho(L)\beta + \breve{\chi}_1), \chi_2\beta]$, $\beta > \check{c}_\beta$, and $\gamma > \check{c}_\gamma$, where $\breve{\chi}_1$, $\check{c}_\beta$, and $\check{c}_\gamma$ and χ_2 are constants given in Sections 3.8.5 and 3.8.3, respectively. Then, for any $T \in \mathbb{N}_+$,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 = \mathcal{O}(\frac{1}{T}), \tag{3.22}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\nabla f(\bar{x}_k)\|^2 = \mathcal{O}(\frac{1}{T}), \tag{3.23}$$

$$f(\bar{x}_T) - f^* = \mathcal{O}(1). \tag{3.24}$$

Proof. The explicit expressions of the right-hand sides of (3.22)–(3.24) and the proof are given in Section 3.8.5. □

Remark 3.14. *The same convergence rate as stated in* (3.23) *has also been achieved by the linearized version of the Prox-PDA algorithm, the distributed proximal gradient primal–dual algorithm (Prox-GPDA), proposed in [112] under the same conditions.*

Find global optima

When Assumption 3.4 also holds, similar to Theorems 3.2 and 3.4 we have the following results.

Theorem 3.6. *Suppose Assumptions 3.1–3.4 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.3 with the same α, β, and γ used in Theorem 3.5, then*

$$\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2+f(\bar{x}_k)-f^*\le\breve{\varepsilon}_0\breve{\varepsilon}^k,\ \forall k\in\mathbb{N}_0, \tag{3.25}$$

where $\breve{\varepsilon}_0>0$ and $\breve{\varepsilon}\in(0,1)$ given in Section 3.8.4.

Proof. The proof is given in Section 3.8.6. □

Remark 3.15. *Linear convergence was also established by the distributed L-ADMM algorithm proposed in [72]. However, in [72], it was assumed that each local cost function is strongly convex, while we assume that the global cost function satisfies the P–Ł condition, which is much weaker. Same as the analysis in Remark 3.12, compared with the results established in [72], one potential drawback of our results is that we need to use some global information, such as the eigenvalues of the Laplacian matrix associated with the communication graph.*

Remark 3.16. *By comparing Theorems 3.3 and 3.4 with Theorems 3.5 and 3.6, respectively, we see that, in theory, under the same conditions the distributed L-ADMM algorithm* (3.21) *has the same convergence properties as the distributed ADMM algorithm* (3.15). *However, in numerical simulations, the distributed ADMM algorithm* (3.15) *normally requires less iterations than the distributed L-ADMM algorithm* (3.21) *to reach the same error bound at a cost of more computation resource being needed by each agent to solve the local optimization problem.*

3.6 Simulations

3.6.1 Distributed regularized logistic regression

This section evaluates the performance of Algorithm 3.1 in solving the nonconvex distributed regularized logistic regression problem with each component function f_i described in (1.2), i.e.,

$$f_i(x)=\frac{n}{m}\sum_{l=1}^{m_i}(y_{il}\log(1+\exp(-x^\top z_{il}))+(1-y_{il})\log(1+\exp(x^\top z_{il})))+\sum_{s=1}^{p}\frac{\lambda\mu[x]_s^2}{1+\mu[x]_s^2}.$$

In this simulation, all settings for cost functions and the communication graph are the same as those described in [114]. Specifically, $n=20$, $p=50$, $m_i=200$, $\lambda=0.001$, and $\mu=1$. The graph used in the simulation is the random geometric graph and the graph parameter is set to be 0.5. We independently and randomly generate nm data points with dimension p and each agent contains m data points.

We compare Algorithm 3.1 with state-of-the-art algorithms: distributed gradient descent (DGD) with diminishing stepsizes [110, 116], distributed gradient tracking algorithm (DGTA) [80, 116], xFILTER [114], Prox-GPDA [112], and D-GPDA [113]. Figure 3.1

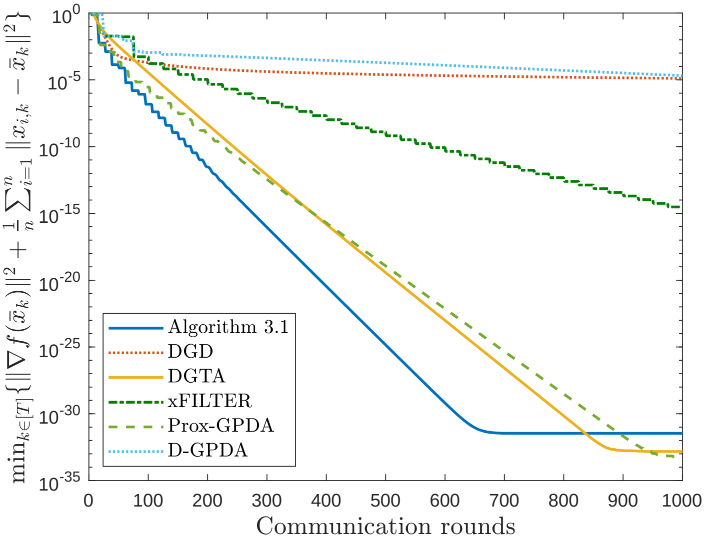


Figure 3.1: Performance of distributed FO optimization algorithms in the nonconvex distributed regularized logistic regression problem: Evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^n \|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of communication rounds.

illustrates the evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^n \|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of communication rounds for these algorithms with the same initial condition. It can be seen that our primal–dual FO algorithm (Algorithm 3.1) gives the best performance in general.

3.6.2 Distributed phase retrieval

This section evaluates the performance of Algorithms 3.2 and 3.3 in solving the distributed phase retrieval problem with each component function f_i described in (1.3), i.e.,

$$f_i(x) = \frac{n}{m}\sum_{l=1}^{m_i}(y_{il}^2 - |b_{il}^\top x|^2)^2 = \frac{n}{m}\sum_{l=1}^{m_i}(y_{il}^2 - (x^\top b_{il}^R)^2 - (x^\top b_{il}^I)^2)^2.$$

In this simulation, all settings for cost functions and the communication graph are the same as those described in [243]. Specifically, $n = 50$, $p = 64$, and $m_i = 30$. We independently and randomly generate the vectors b_{il}^R and b_{il}^I such that $(b_{il}^R, b_{il}^I) \sim \mathcal{N}(\mathbf{0}_{2p}, \frac{1}{2}\boldsymbol{I}_{2p})$. The scalars y_{il} are generated by $y_{il} = |b_{il}^\top y_0| + \varepsilon_{i,l}$, where $y_0 = (1, 0, \ldots, 0)^\top$ and $\varepsilon_{i,l} \sim \mathcal{N}(0, 0.01^2)$ are independent Gaussian noise. The graph used in the simulation is generated by uniformly randomly sampling n points on $\mathbb{S}^2$, and then connecting pairs of points with spherical distances less than $\pi/4$.

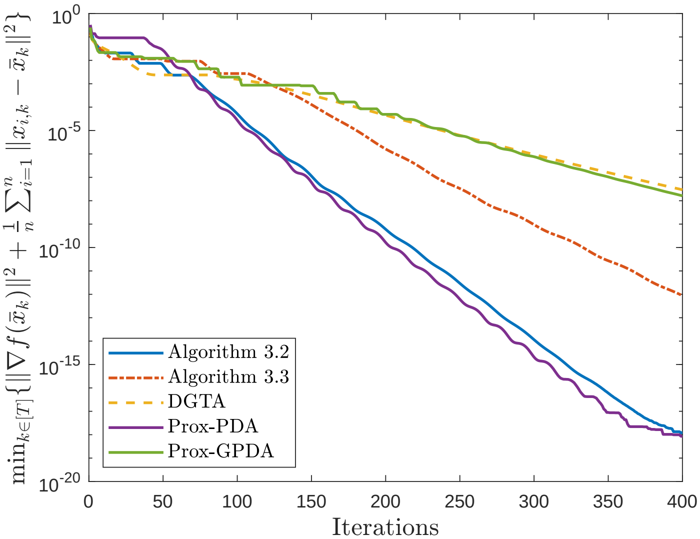


Figure 3.2: Performance of distributed ADMM optimization algorithms in the distributed phase retrieval problem: Evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^n \|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of iterations.

We compare Algorithms 3.2 and 3.3 with state-of-the-art algorithms: distributed gradient tracking algorithm (DGTA) [80, 243], Prox-PDA (which is a distributed ADMM algorithm) and its linearized version (Prox-GPDA) [112]. Figure 3.2 illustrates the evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^n \|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of iterations for these algorithms with the same initial condition. It can be seen that, in this numerical example, both distributed ADMM algorithms (Algorithms 3.2 and Prox-PDA) give almost the same performance and are better than the rest algorithms. By comparing the two distributed L-ADMM algorithms (Algorithm 3.3 and Prox-GPDA), we see that Algorithm 3.3 converges faster. Moreover, Algorithm 3.3 also converges faster than DGTA.

3.7 Summary

In this chapter, we studied distributed nonconvex optimization with full-information feedback. We proposed three distributed algorithms: a distributed primal–dual FO algorithm, a distributed ADMM algorithm, and a distributed L-ADMM algorithm. We derived their convergence properties under different conditions. Particularly, linear convergence was established when the global cost function satisfies the P–Ł condition. This relaxes the standard strong convexity condition in the literature. Interesting directions for future work include proving linear convergence rate for larger stepsizes, considering time-varying graphs, and studying constraints.

3.8 Proofs

3.8.1 Proof of Theorem 3.1

Denote $K_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, $\boldsymbol{K} = K_n \otimes \boldsymbol{I}_p$, $\boldsymbol{H} = \frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)$, $\bar{x}_k = \frac{1}{n}(\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)\boldsymbol{x}_k$, $\bar{\boldsymbol{x}}_k = \mathbf{1}_n \otimes \bar{x}_k$, $\boldsymbol{g}_k = \nabla \tilde{f}(\boldsymbol{x}_k)$, $\bar{\boldsymbol{g}}_k = \boldsymbol{H}\boldsymbol{g}_k$, $\boldsymbol{g}_k^0 = \nabla \tilde{f}(\bar{\boldsymbol{x}}_k)$, and $\bar{\boldsymbol{g}}_k^0 = \boldsymbol{H}\boldsymbol{g}_k^0 = \frac{1}{n}(\mathbf{1}_n \otimes \nabla f(\bar{x}_k))$.

We also denote the following notations.

$$
\begin{aligned}
&c_\beta = \max\Big\{\frac{\kappa_1}{\kappa_2 - 1},\ \kappa_3,\ \kappa_4\Big\}, c_\eta = \min\Big\{\frac{\epsilon_1}{\epsilon_2},\ \frac{\epsilon_3}{\epsilon_4},\ \frac{\epsilon_5}{\epsilon_6}\Big\},\ \kappa_1 = \frac{1}{2\rho_2(L)}(2 + 3L_f^2),\ \kappa_2 > 1,\\
&\kappa_3 = \frac{1}{4}\Big(1 + \Big(1 + 8\kappa_2 + \frac{8}{\rho_2(L)}\Big)^{\frac{1}{2}}\Big),\ \kappa_4 = \Big(\kappa_2 + \frac{1}{\rho_2(L)}\Big)L_f^2 + \Big(\Big(\kappa_2 + \frac{1}{\rho_2(L)}\Big)^2 L_f^2 + 2\Big)^{\frac{1}{2}} L_f,\\
&\epsilon_1 = (\alpha - \beta)\rho_2(L) - \frac{1}{2}(2 + 3L_f^2),\ \epsilon_2 = \beta^2\rho(L) + (2\alpha^2 + \beta^2)\rho^2(L) + \frac{5}{2}L_f^2,\\
&\epsilon_3 = \beta - \frac{1}{2} - \frac{\alpha}{2\beta^2} - \frac{1}{2\beta\rho_2(L)},\ \epsilon_4 = 2\beta^2 + \frac{1}{2},\ \epsilon_5 = \frac{1}{4} - \frac{1}{2\beta}\Big(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2,\\
&\epsilon_6 = \frac{1}{\beta^2}\Big(1 + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2 + \frac{L_f(1 + L_f)}{2},\ \epsilon_7 = \eta\min\Big\{\epsilon_1 - \eta\epsilon_2,\ \epsilon_3 - \eta\epsilon_4,\ \epsilon_5 - \eta\epsilon_6,\ \frac{1}{4}\Big\},\\
&\epsilon_8 = \frac{\alpha + \beta}{2\beta} + \frac{1}{2\rho_2(L)}\ \epsilon_9 = \min\Big\{\frac{1}{2\rho(L)},\ \frac{\alpha - \beta}{2\alpha}\Big\}.
\end{aligned}
$$

To prove Theorem 3.1, the following lemma is used, which presents the general relations of two consecutive outputs of Algorithm 3.1.

Lemma 3.1. *Suppose Assumptions 3.1–3.3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.1 with $\alpha > \beta$. Then,*

$$
V_{k+1} \le V_k - \|\boldsymbol{x}_k\|^2_{\eta_k(\epsilon_1 - \eta_k\epsilon_2)\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\epsilon_3 - \eta_k\epsilon_4)\boldsymbol{K}} - \eta_k(\epsilon_5 - \eta_k\epsilon_6)\|\bar{\boldsymbol{g}}_k\|^2 - \frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k^0\|^2, \tag{3.26}
$$

where

$$
\begin{aligned}
&V_k = \sum_{i=1}^4 V_{i,k},\ V_{1,k} = \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}},\ V_{2,k} = \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{Q} + \frac{\alpha}{\beta}\boldsymbol{K}},\\
&V_{3,k} = \boldsymbol{x}_k^\top \boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big),\ V_{4,k} = n(f(\bar{x}_k) - f^*) = \tilde{f}(\bar{\boldsymbol{x}}_k) - \tilde{f}^*,
\end{aligned}
$$

and $\boldsymbol{Q} = R\Lambda_1^{-1}R^\top \otimes \boldsymbol{I}_p$ with matrices R and Λ_1^{-1} given in Lemma 2.5.

Proof. We first note that $V_{4,k}$ is well defined due to $f^* > -\infty$ as assumed in Assumption 3.2. Thus, V_k is well defined.

Denote $\bar{v}_k = \frac{1}{n}(\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)\boldsymbol{v}_k$. Then, from (3.7b), we know that

$$
\bar{v}_{k+1} = \bar{v}_k. \tag{3.27}
$$

Then, from (3.27) and $\sum_{i=1}^{n} v_{i,0} = \mathbf{0}_p$, we know that

$$\bar{v}_k = \mathbf{0}_p, \tag{3.28}$$

Then, from (3.28) and (3.7a), we know that

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_k - \eta\bar{\boldsymbol{g}}_k. \tag{3.29}$$

Noting that $\nabla\tilde{f}$ is Lipschitz-continuous with constant $L_f > 0$ as assumed in Assumption 3.3, we have

$$\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k\|^2 = \|\nabla\tilde{f}(\bar{\boldsymbol{x}}_k) - \nabla\tilde{f}(\boldsymbol{x}_k)\|^2 \le L_f^2\|\bar{\boldsymbol{x}}_k - \boldsymbol{x}_k\|^2 = L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}. \tag{3.30}$$

Then, from (3.30) and $\rho(\boldsymbol{H}) = 1$, we have

$$\|\bar{\boldsymbol{g}}_k^0 - \bar{\boldsymbol{g}}_k\|^2 = \|\boldsymbol{H}(\boldsymbol{g}_k^0 - \boldsymbol{g}_k)\|^2 \le \|\boldsymbol{g}_k^0 - \boldsymbol{g}_k\|^2 \le L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}. \tag{3.31}$$

From $\nabla\tilde{f}$ is Lipschitz-continuous and (3.29), we have

$$\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 \le L_f^2\|\bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k\|^2 = \eta^2L_f^2\|\bar{\boldsymbol{g}}_k\|^2. \tag{3.32}$$

We have

$$\begin{aligned}
V_{1,k+1} &= \frac{1}{2}\|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}} = \frac{1}{2}\|\boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k + \beta\boldsymbol{v}_k + \boldsymbol{g}_k)\|^2_{\boldsymbol{K}}\\
&= \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \eta\alpha\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}} + \frac{\eta^2\alpha^2}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2} - \eta\beta\boldsymbol{x}_k^\top(\boldsymbol{I}_{np} - \eta\alpha\boldsymbol{L})\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k\Big)\\
&\quad + \frac{\eta^2\beta^2}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k\Big\|^2_{\boldsymbol{K}}\\
&= V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L} - \frac{\eta^2\alpha^2}{2}\boldsymbol{L}^2} - \eta\beta\boldsymbol{x}_k^\top(\boldsymbol{I}_{np} - \eta\alpha\boldsymbol{L})\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0 + \frac{1}{\beta}\boldsymbol{g}_k - \frac{1}{\beta}\boldsymbol{g}_k^0\Big)\\
&\quad + \frac{\eta^2\beta^2}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0 + \frac{1}{\beta}\boldsymbol{g}_k - \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\\
&\le V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L} - \frac{\eta^2\alpha^2}{2}\boldsymbol{L}^2} - \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \frac{\eta}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{\eta}{2}\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2\\
&\quad + \frac{\eta^2\alpha^2}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2} + \frac{\eta^2\beta^2}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \frac{\eta^2\alpha^2}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2} + \frac{\eta^2}{2}\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2\\
&\quad + \eta^2\beta^2\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \eta^2\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2\\
&= V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L} - \frac{\eta}{2}\boldsymbol{K} - \frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2} + \frac{\eta}{2}(1 + 3\eta)\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2\\
&\quad - \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}} \qquad (3.33)\\
&\le V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L} - \frac{\eta}{2}\boldsymbol{K} - \frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2 - \frac{\eta}{2}(1+3\eta)L_f^2\boldsymbol{K}} - \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}},
\end{aligned}$$
$$(3.34)$$

where the second equality holds due to (3.6a); the third equality holds due toe (2.5); the first inequality holds due to the Cauchy-Schwarz inequality and $\rho(\boldsymbol{K}) = 1$; and the last inequality holds due to (3.30).

We have

$$
\begin{aligned}
V_{2,k+1} &= \frac{1}{2}\Big\|\boldsymbol{v}_{k+1} + \frac{1}{\beta}\boldsymbol{g}^0_{k+1}\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} = \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k + \eta\beta\boldsymbol{L}\boldsymbol{x}_k + \frac{1}{\beta}(\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k)\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} \\
&= \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big) + \|\boldsymbol{x}_k\|^2_{\frac{\eta^2\beta}{2}(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\
&\quad + \frac{1}{2\beta^2}\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \frac{1}{\beta}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k + \eta\beta\boldsymbol{L}\boldsymbol{x}_k\Big)^\top\Big(\boldsymbol{Q} + \frac{\alpha}{\beta}\boldsymbol{K}\Big)(\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k) \\
&\le V_{2,k} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big) + \|\boldsymbol{x}_k\|^2_{\frac{\eta^2\beta}{2}(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\
&\quad + \frac{1}{2\beta^2}\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \frac{\eta}{2\beta}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \frac{1}{2\eta\beta}\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} \\
&\quad + \frac{\eta^2\beta^2}{2}\|\boldsymbol{L}\boldsymbol{x}_k\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \frac{1}{2\beta^2}\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} \\
&= V_{2,k} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big) + \|\boldsymbol{x}_k\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\
&\quad + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + \Big(\frac{1}{\beta^2} + \frac{1}{2\eta\beta}\Big)\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} \\
&\le V_{2,k} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big) + \|\boldsymbol{x}_k\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\
&\quad + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + \Big(\frac{1}{\beta^2} + \frac{1}{2\eta\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2 \qquad (3.35) \\
&\le V_{2,k} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big) + \|\boldsymbol{x}_k\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\
&\quad + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + \eta\Big(\frac{\eta}{\beta^2} + \frac{1}{2\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2\|\bar{\boldsymbol{g}}_k\|^2, \qquad (3.36)
\end{aligned}
$$

where the second equality holds due to (3.6b); the third equality holds due to (2.5) and (2.7); the first inequality holds due to the Cauchy-Schwarz inequality; the last equality holds due to (2.5) and (2.7); the second inequality holds due to $\rho(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}) \le \rho(\boldsymbol{Q}) + \frac{\alpha}{\beta}\rho(\boldsymbol{K})$, (2.8), $\rho(\boldsymbol{K}) = 1$; and the last inequality holds due to (3.32).

We have

$$
\begin{aligned}
V_{3,k+1} &= \boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1} + \frac{1}{\beta}\boldsymbol{g}^0_{k+1}\Big) \\
&= (\boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k + \beta\boldsymbol{v}_k + \boldsymbol{g}^0_k + \boldsymbol{g}_k - \boldsymbol{g}^0_k))^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k + \eta\beta\boldsymbol{L}\boldsymbol{x}_k + \frac{1}{\beta}(\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k)\Big) \\
&= \boldsymbol{x}_k^\top(\boldsymbol{K} - \eta(\alpha + \eta\beta^2)\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big) + \|\boldsymbol{x}_k\|^2_{\eta\beta(\boldsymbol{L}-\eta\alpha\boldsymbol{L}^2)}
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{\beta}\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta\alpha\boldsymbol{L})(\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0)-\eta\beta\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2-\eta\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)^\top\boldsymbol{K}(\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0)\\
&-\eta(\boldsymbol{g}_k-\boldsymbol{g}_k^0)^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0+\eta\beta\boldsymbol{L}\boldsymbol{x}_k+\frac{1}{\beta}(\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0)\Big)\\
\le{}&\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta\alpha\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)+\frac{\eta^2\beta^2}{2}\|\boldsymbol{L}\boldsymbol{x}_k\|^2+\frac{\eta^2\beta^2}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2+\|\boldsymbol{x}_k\|_{\eta\beta(\boldsymbol{L}-\eta\alpha\boldsymbol{L}^2)}^2\\
&+\frac{\eta}{2}\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2+\Big(\frac{1}{2\eta\beta^2}+\frac{1}{2\beta^2}\Big)\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2+\frac{\eta^2\alpha^2}{2}\|\boldsymbol{L}\boldsymbol{x}_k\|^2-\eta\beta\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2\\
&+\frac{\eta^2}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2+\frac{1}{2}\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2+\frac{\eta}{2}\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2+\frac{\eta}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2\\
&+\frac{\eta^2}{2}\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2+\frac{\eta^2\beta^2}{2}\|\boldsymbol{L}\boldsymbol{x}_k\|^2+\frac{\eta^2}{2}\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2+\frac{1}{2\beta^2}\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2\\
={}&\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta\alpha\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)+\frac{\eta}{2}(1+2\eta)\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2+\|\boldsymbol{x}_k\|_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{\alpha^2}{2}-\alpha\beta+\beta^2)\boldsymbol{L}^2}^2\\
&+\Big(\frac{1}{2\eta\beta^2}+\frac{1}{\beta^2}+\frac{1}{2}\Big)\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\eta(\beta-\frac{1}{2}-\frac{\eta}{2}-\frac{\eta\beta^2}{2})\boldsymbol{K}}^2 &&(3.37)\\
\le{}&V_{3,k}-\eta\alpha\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)+\|\boldsymbol{x}_k\|_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{\alpha^2}{2}-\alpha\beta+\beta^2)\boldsymbol{L}^2+\frac{\eta}{2}(1+2\eta)L_f^2\boldsymbol{K}}^2\\
&+\eta\Big(\frac{1}{2\beta^2}+\frac{\eta}{\beta^2}+\frac{\eta}{2}\Big)L_f^2\|\bar{\boldsymbol{g}}_k\|^2-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\eta(\beta-\frac{1}{2}-\frac{\eta}{2}-\frac{\eta\beta^2}{2})\boldsymbol{K}}^2, &&(3.38)
\end{aligned}
$$

where the second equality holds due to (3.6); the third equality holds due to (2.5); the first inequality holds due to the Cauchy-Schwarz inequality, (2.5), and $\rho(\boldsymbol{K})=1$; and the last inequality holds due to (3.32) and (3.30).

We have

$$
\begin{aligned}
V_{4,k+1}&=n(f(\bar{x}_k)-f^*)=\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*+\tilde{f}(\bar{\boldsymbol{x}}_{k+1})-\tilde{f}(\bar{\boldsymbol{x}}_k)\\
&\le\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*-\eta\bar{\boldsymbol{g}}_k^\top\boldsymbol{g}_k^0+\frac{\eta^2L_f}{2}\|\bar{\boldsymbol{g}}_k\|^2=V_{4,k}-\eta\bar{\boldsymbol{g}}_k^\top\bar{\boldsymbol{g}}_k^0+\frac{\eta^2L_f}{2}\|\bar{\boldsymbol{g}}_k\|^2\\
&=V_{4,k}-\frac{\eta}{2}\bar{\boldsymbol{g}}_k^\top(\bar{\boldsymbol{g}}_k+\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k)-\frac{\eta}{2}(\bar{\boldsymbol{g}}_k-\bar{\boldsymbol{g}}_k^0+\bar{\boldsymbol{g}}_k^0)^\top\bar{\boldsymbol{g}}_k^0+\frac{\eta^2L_f}{2}\|\bar{\boldsymbol{g}}_k\|^2\\
&\le V_{4,k}-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k\|^2+\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k\|^2-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2+\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k\|^2+\frac{\eta^2L_f}{2}\|\bar{\boldsymbol{g}}_k\|^2\\
&=V_{4,k}-\frac{\eta}{4}(1-2\eta L_f)\|\bar{\boldsymbol{g}}_k\|^2+\frac{\eta}{2}\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k\|^2-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 &&(3.39)\\
&\le V_{4,k}-\frac{\eta}{4}(1-2\eta L_f)\|\bar{\boldsymbol{g}}_k\|^2+\|\boldsymbol{x}_k\|_{\frac{\eta}{2}L_f^2\boldsymbol{K}}^2-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2, &&(3.40)
\end{aligned}
$$

where the first inequality holds since that $\tilde{f}$ is smooth, (2.14) and (3.29); the third equality holds due to $\bar{\boldsymbol{g}}_k^\top\boldsymbol{g}_k^0=\boldsymbol{g}_k^\top\boldsymbol{H}\boldsymbol{g}_k^0=\boldsymbol{g}_k^\top\boldsymbol{H}\boldsymbol{H}\boldsymbol{g}_k^0=\bar{\boldsymbol{g}}_k^\top\bar{\boldsymbol{g}}_k^0$; the second inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds due to (3.31).

From (3.34), (3.36), (3.38), and (3.40), we have

$$\begin{aligned}
V_{k+1} \le{}& V_k - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2-\frac{\eta}{2}(1+3\eta)L_f^2\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}} \\
&+ \|\boldsymbol{x}_k\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + \eta\Big(\frac{\eta}{\beta^2} + \frac{1}{2\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2\|\bar{\boldsymbol{g}}_k\|^2 \\
&+ \|\boldsymbol{x}_k\|^2_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{\alpha^2}{2}-\alpha\beta+\beta^2)\boldsymbol{L}^2+\frac{\eta}{2}(1+2\eta)L_f^2\boldsymbol{K}} + \eta\Big(\frac{1}{2\beta^2} + \frac{\eta}{\beta^2} + \frac{\eta}{2}\Big)L_f^2\|\bar{\boldsymbol{g}}_k\|^2 \\
&- \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|^2_{\eta(\beta-\frac{1}{2}-\frac{\eta}{2}-\frac{\eta\beta^2}{2})\boldsymbol{K}} - \frac{\eta}{4}(1-2\eta L_f)\|\bar{\boldsymbol{g}}_k\|^2 + \|\boldsymbol{x}_k\|^2_{\frac{\eta}{2}L_f^2\boldsymbol{K}} - \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 \\
={}& V_k - \|\boldsymbol{x}_k\|^2_{\eta\boldsymbol{M}_1-\eta^2\boldsymbol{M}_2} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta\boldsymbol{M}_3-\eta^2\boldsymbol{M}_4} - \eta(\epsilon_5 - \eta\epsilon_6)\|\bar{\boldsymbol{g}}_k\|^2 - \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2, \qquad (3.41)
\end{aligned}$$

where

$$\boldsymbol{M}_1 = (\alpha-\beta)\boldsymbol{L} - \frac{1}{2}(2+3L_f^2)\boldsymbol{K},\ \boldsymbol{M}_2 = \beta^2\boldsymbol{L} + (2\alpha^2+\beta^2)\boldsymbol{L}^2 + \frac{5}{2}L_f^2\boldsymbol{K},$$
$$\boldsymbol{M}_3 = \Big(\beta - \frac{1}{2} - \frac{\alpha}{2\beta^2}\Big)\boldsymbol{K} - \frac{1}{2\beta}\boldsymbol{Q},\ \boldsymbol{M}_4 = \Big(2\beta^2 + \frac{1}{2}\Big)\boldsymbol{K}.$$

From $\alpha > \beta$, (2.6), and (2.8), we have

$$\boldsymbol{M}_1 = (\alpha-\beta)\boldsymbol{L} - \frac{1}{2}(2+3L_f^2)\boldsymbol{K} \ge (\alpha-\beta)\rho_2(L)\boldsymbol{K} - \frac{1}{2}(2+3L_f^2)\boldsymbol{K} = \epsilon_1\boldsymbol{K}, \qquad (3.42)$$

$$\boldsymbol{M}_2 = \beta^2\boldsymbol{L} + (2\alpha^2+\beta^2)\boldsymbol{L}^2 + \frac{5}{2}L_f^2\boldsymbol{K} \le \epsilon_2\boldsymbol{K}, \qquad (3.43)$$

$$\boldsymbol{M}_3 = \Big(\beta - \frac{1}{2} - \frac{\alpha}{2\beta^2}\Big)\boldsymbol{K} - \frac{1}{2\beta}\boldsymbol{Q} \ge \Big(\beta - \frac{1}{2} - \frac{\alpha}{2\beta^2}\Big)\boldsymbol{K} - \frac{1}{2\beta\rho_2(L)}\boldsymbol{K} = \epsilon_3\boldsymbol{K}. \qquad (3.44)$$

From (3.41) and (3.42)–(3.44), we know that (3.26) holds.

□

We are now ready to prove Theorem 3.1.
Denote

$$\hat{V}_k = \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + n(f(\bar{x}_k) - f^*). \qquad (3.45)$$

We know

$$\begin{aligned}
V_k ={}& \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + V_{4,k} \\
\ge{}& \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{2}\Big(\frac{1}{\rho(L)} + \frac{\alpha}{\beta}\Big)\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} - \frac{\beta}{2\alpha}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{\alpha}{2\beta}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + V_{4,k} \\
\ge{}& \epsilon_9\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\Big) + V_{4,k} \qquad (3.46)
\end{aligned}$$

$$\geq \epsilon_9 \hat{V}_k \geq 0, \tag{3.47}$$

where the first inequality holds due to (2.8) and the Cauchy-Schwarz inequality; and the last inequality holds due to $0 < \epsilon_9 < 1$. Similarly, we have

$$V_k \leq \epsilon_8 \hat{V}_k. \tag{3.48}$$

From $\beta + \kappa_1 < \alpha$ and $\kappa_1 = \frac{1}{2\rho_2(L)}(2 + 3L_f^2)$, we have

$$\epsilon_1 = (\alpha - \beta)\rho_2(L) - \frac{1}{2}(2 + 3L_f^2) > 0. \tag{3.49}$$

From $\alpha \leq \kappa_2\beta$ and $\beta > \kappa_3$, we have

$$\epsilon_3 \geq \Big(\beta - \frac{1}{2} - \frac{\kappa_2}{2\beta}\Big) - \frac{1}{2\beta\rho_2(L)} > 0. \tag{3.50}$$

From $\alpha \leq \kappa_2\beta$ and $\beta > \kappa_4$, we have

$$\epsilon_5 = \frac{1}{4} - \frac{1}{2\beta}\Big(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2 \geq \frac{1}{4} - \frac{1}{2\beta}\Big(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \kappa_2\Big)L_f^2 > 0. \tag{3.51}$$

From (3.49)–(3.51), and $0 < \eta < \min\{\frac{\epsilon_1}{\epsilon_2}, \frac{\epsilon_3}{\epsilon_4}, \frac{\epsilon_5}{\epsilon_6}\}$, we have

$$\eta(\epsilon_1 - \eta\epsilon_2) > 0, \tag{3.52}$$

$$\eta(\epsilon_3 - \eta\epsilon_4) > 0, \tag{3.53}$$

$$\eta(\epsilon_5 - \eta\epsilon_6) > 0. \tag{3.54}$$

Then, rom (3.52)–(3.54), we have

$$\epsilon_7 > 0 \tag{3.55}$$

From (3.26), we have

$$\begin{aligned} \sum_{k=0}^{T} V_{k+1} \leq \sum_{k=0}^{T} V_k - \sum_{k=0}^{T} \|\boldsymbol{x}_k\|^2_{\eta(\epsilon_1-\eta\epsilon_2)\boldsymbol{K}} - \sum_{k=0}^{T} \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta(\epsilon_3-\eta\epsilon_4)\boldsymbol{K}} \\ - \sum_{k=0}^{T} \eta(\epsilon_5 - \eta\epsilon_6)\|\bar{\boldsymbol{g}}_k\|^2 - \sum_{k=0}^{T} \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2. \end{aligned} \tag{3.56}$$

Hence, from (3.56), we have

$$V_{T+1} + \epsilon_7 \sum_{k=0}^{T} \Big(\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}_k\|^2 + \|\bar{\boldsymbol{g}}_k^0\|^2\Big) \leq V_0. \tag{3.57}$$

From (3.57), (3.55), and (3.47), we know that

$$\frac{\sum_{k=0}^{T}(\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}_k\|^2 + \|\bar{\boldsymbol{g}}_k^0\|^2)}{T+1} \leq \frac{V_0}{\epsilon_7(T+1)}, \ \forall T \in \mathbb{N}_0, \tag{3.58}$$

which yields (3.8) and (3.9).

From (3.57), (3.55), and (3.46), we know that

$$f(\bar{x}_{T+1}) - f^* \leq \frac{V_0}{n}, \ \forall T \in \mathbb{N}_0, \tag{3.59}$$

which gives (3.10).

3.8.2 Proof of Theorem 3.2

In addition to the notations defined in Section 3.8.1, we also denote the following notations.

$$\epsilon_0 = \frac{V_0}{\epsilon_9}, \ \epsilon = 1 - \frac{\epsilon_{10}}{\epsilon_8}, \ \epsilon_{10} = \eta \min\Big\{\epsilon_1 - \eta\epsilon_2, \ \epsilon_3 - \eta\epsilon_4, \ \frac{\nu}{2}\Big\}.$$

From (3.47), we have

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) = \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + n(f(\bar{x}_k) - f^*) \leq \hat{V}_k \leq \frac{V_k}{\epsilon_9}. \tag{3.60}$$

From Assumptions 3.2 and 3.4 as well as (2.16), we have that

$$\|\bar{\boldsymbol{g}}_k^0\|^2 = n\|\nabla f(\bar{x}_k)\|^2 \geq 2\nu n(f(\bar{x}_k) - f^*). \tag{3.61}$$

From (3.52) and (3.53), we have

$$\epsilon_{10} > 0, \ \frac{\epsilon_{10}}{\epsilon_8} > 0. \tag{3.62}$$

Noting that $\epsilon_3 < \beta$, $\epsilon_4 > 2\beta^2$, and $\epsilon_8 > \frac{\alpha+\beta}{2\beta} > 1$, we have

$$0 < \frac{\epsilon_{10}}{\epsilon_8} \leq \frac{\eta(\epsilon_3 - \eta\epsilon_4)}{\epsilon_8} \leq \frac{\epsilon_3^2}{4\epsilon_4\epsilon_8} < \frac{1}{8}. \tag{3.63}$$

Then, from (3.26), (3.54), and (3.61), we have

$$V_{k+1} \leq V_k - \hat{V}_k\eta_k \min\Big\{\epsilon_1 - \eta_k\epsilon_2, \ \epsilon_3 - \eta_k\epsilon_4, \ \frac{\nu}{2}\Big\}. \tag{3.64}$$

From (3.64), (3.62), and (3.48), we have

$$V_{k+1} \leq V_k - \epsilon_{10}\hat{V}_k \leq V_k - \frac{\epsilon_{10}}{\epsilon_8}V_k. \tag{3.65}$$

From (3.65) and (3.63), we have

$$V_{k+1} \leq \Big(1 - \frac{\epsilon_{10}}{\epsilon_8}\Big)V_k \leq \Big(1 - \frac{\epsilon_{10}}{\epsilon_8}\Big)^{k+1} V_0. \tag{3.66}$$

Hence, (3.66) and (3.60) give

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) \leq \frac{V_0}{\epsilon_9}\Big(1 - \frac{\epsilon_{10}}{\epsilon_8}\Big)^k, \ \forall k \in \mathbb{N}_0, \tag{3.67}$$

which yields (3.11).

3.8.3 Proof of Theorem 3.3

In addition to the notations introduced in Section 3.8.1, we also denote the following notations.

$$
\begin{aligned}
&\hat{c}_\beta = \max\Big\{\frac{\chi_1}{\chi_2\rho_2(L)-\rho(L)},\ \chi_3,\ \chi_4\Big\},\ \hat{c}_\gamma = \max\Big\{\frac{\varepsilon_4}{\varepsilon_3},\ \frac{\varepsilon_6}{\varepsilon_5},\ \frac{\varepsilon_8+\varepsilon_9+\varepsilon_{10}}{\varepsilon_7},\ \frac{1}{\varepsilon_{15}}\Big\},\\
&\chi_1 = 2L_f^2+2,\ \chi_2 > \frac{\rho(L)}{\rho_2(L)},\ \chi_3 = \frac{1}{4}\Big(1+\Big(1+8\chi_2+\frac{8}{\rho_2(L)}\Big)^{\frac{1}{2}}\Big),\\
&\chi_4 = \Big(\chi_2+\frac{1}{\rho_2(L)}\Big)L_f^2+\Big(\Big(\chi_2+\frac{1}{\rho_2(L)}\Big)^2L_f^4+2L_f^2\Big)^{\frac{1}{2}},\ \varepsilon_1 = \frac{3}{2}+2L_f^2+\beta\rho(L),\\
&\varepsilon_2 = (2+\rho(L^2))3L_f^2+\beta^2\rho(L)+\alpha\beta\rho(L^2),\ \varepsilon_3 = \alpha\rho_2(L)-\frac{1}{2}-\varepsilon_1,\\
&\varepsilon_4 = \frac{3}{2}(3+\rho(L^2))\alpha^2\rho(L^2)+\varepsilon_2,\ \varepsilon_5 = \beta-\frac{1}{2}-\frac{\alpha}{2\beta^2}-\frac{1}{2\beta\rho_2(L)},\\
&\varepsilon_6 = \frac{1}{2}(\alpha^2+(7+3\rho(L^2))\beta^2),\ \varepsilon_7 = \frac{1}{4}-\frac{1}{2\beta}\Big(\frac{1}{\rho_2(L)}+\frac{\alpha+1}{\beta}\Big)L_f^2,\\
&\varepsilon_8 = \Big(\frac{1}{2}+\frac{1}{\beta^2}\Big(\frac{1}{\rho_2(L)}+\frac{\alpha}{\beta}\Big)L_f\Big)L_f,\ \varepsilon_9 = 3L_f^2,\ \varepsilon_{10} = 3(2+\rho(L^2))L_f^2, \varepsilon_{11} = \frac{1}{2}-\frac{1}{\gamma}\varepsilon_1-\frac{1}{\gamma^2}\varepsilon_2,\\
&\varepsilon_{12} = \frac{1}{2}\Big(\frac{1}{\rho(L)}+\frac{\alpha}{\beta}\Big),\ \varepsilon_{13} = \frac{1}{2}\Big(\varepsilon_{11}-\varepsilon_{12}+((\varepsilon_{11}-\varepsilon_{12})^2+1)^{\frac{1}{2}}\Big),\ \varepsilon_{14} = \frac{\alpha+\beta}{2\beta}+\frac{1}{2\rho_2(L)},\\
&\varepsilon_{15} = \frac{1}{2\varepsilon_2}\Big(-\varepsilon_1+\Big(\varepsilon_1^2+2-\frac{1}{\varepsilon_{12}}\Big)^{\frac{1}{2}}\Big),\ \varepsilon_{16} = \frac{1}{\gamma}\min\Big\{\varepsilon_3-\frac{1}{\gamma}\varepsilon_4,\ \varepsilon_5-\frac{1}{\gamma}\varepsilon_6,\ \frac{1}{4}\Big\}.
\end{aligned}
$$

To prove Theorem 3.3, the following lemma is used, which presents the general relations of two consecutive outputs of Algorithm 3.2.

Lemma 3.2. *Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.2. If Assumptions 3.1–3.3 hold and $\gamma > L_f$, then*

$$
\begin{aligned}
\tilde{V}_{k+1} \le \tilde{V}_k &- \|\boldsymbol{x}_k\|^2_{\frac{1}{\gamma}(\varepsilon_3-\frac{1}{\gamma}\varepsilon_4)\boldsymbol{K}} - \Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{1}{\gamma}(\varepsilon_5-\frac{1}{\gamma}\varepsilon_6)\boldsymbol{K}} - \frac{1}{4\gamma}\|\bar{\boldsymbol{g}}_k^0\|^2\\
&- \frac{1}{\gamma}\Big(\varepsilon_7-\frac{1}{\gamma}\varepsilon_8-\frac{1}{\gamma^2}\varepsilon_9-\frac{1}{\gamma^3}\varepsilon_{10}\Big)\|\bar{\boldsymbol{g}}_{k+1}\|^2,\ \forall k\in\mathbb{N}_0, \qquad (3.68)
\end{aligned}
$$

where

$$
\begin{aligned}
&\tilde{V}_k = V_k - \|\boldsymbol{x}_k\|^2_{\frac{1}{\gamma}(\varepsilon_1+\frac{1}{\gamma}\varepsilon_2)\boldsymbol{K}},\ V_k = \sum_{i=1}^{4} V_{i,k},\ V_{1,k} = \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}},\ V_{2,k} = \frac{1}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}},\\
&V_{3,k} = \boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big),\ V_{4,k} = n(f(\bar{x}_k)-f^*) = \tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*,
\end{aligned}
$$

and $\boldsymbol{Q} = R\Lambda_1^{-1}R^\top\otimes\boldsymbol{I}_p$ with matrices R and Λ_1^{-1} given in Lemma 2.5.

Proof. Noting $\gamma > L_f$, from Remark 3.8, we know that the subproblem (3.15a) is solvable and $x_{i,k+1}$ is unique. Then noting first order optimality conditions for convex optimization problems, we know that the algorithm (3.15) can be rewritten as

$$x_{i,k+1} = x_{i,k} - \eta\Big(\alpha \sum_{j=1}^{n} L_{ij}x_{j,k} + \beta v_{i,k} + \nabla f_i(x_{i,k+1})\Big), \tag{3.69a}$$

$$v_{i,k+1} = v_{i,k} + \eta\beta \sum_{j=1}^{n} L_{ij}x_{j,k+1},\ \forall x_{i,0} \in \mathbb{R}^p,\ \sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p, \tag{3.69b}$$

where $\eta = \frac{1}{\gamma}$. We write (3.69) in a compact form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta(\alpha \boldsymbol{L}\boldsymbol{x}_k + \beta \boldsymbol{v}_k + \boldsymbol{g}_{k+1}), \tag{3.70a}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \eta\beta \boldsymbol{L}\boldsymbol{x}_{k+1},\ \forall \boldsymbol{x}_0 \in \mathbb{R}^{np},\ \sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p. \tag{3.70b}$$

From (3.70), similar to the way to get (3.29), we know that

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_k - \eta \bar{\boldsymbol{g}}_{k+1}. \tag{3.71}$$

We have

$$\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2 = \|\nabla \tilde{f}(\bar{\boldsymbol{x}}_{k+1}) - \nabla \tilde{f}(\bar{\boldsymbol{x}}_k)\|^2 \le L_f^2 \|\bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k\|^2 = \eta^2 L_f^2 \|\bar{\boldsymbol{g}}_{k+1}\|^2, \tag{3.72}$$

where the first inequality holds since $\nabla \tilde{f}$ is Lipschitz-continuous; and the last equality holds due to (3.71). Then, we have

$$\begin{aligned}\|\boldsymbol{g}^0_k - \boldsymbol{g}_{k+1}\|^2 &= \|\boldsymbol{g}^0_k - \boldsymbol{g}^0_{k+1} + \boldsymbol{g}^0_{k+1} - \boldsymbol{g}_{k+1}\|^2 \le 2\|\boldsymbol{g}^0_k - \boldsymbol{g}^0_{k+1}\|^2 + 2\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}_{k+1}\|^2 \\ &\le 2\eta^2 L_f^2 \|\bar{\boldsymbol{g}}_{k+1}\|^2 + 2L_f^2 \|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}}, \end{aligned} \tag{3.73}$$

where the first inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds due to (3.30) and (3.72). Then, we have

$$\|\bar{\boldsymbol{g}}^0_k - \bar{\boldsymbol{g}}_{k+1}\|^2 = \|\boldsymbol{H}(\boldsymbol{g}^0_k - \boldsymbol{g}_{k+1})\|^2 \le \|\boldsymbol{g}^0_k - \boldsymbol{g}_{k+1}\|^2 \le 2\eta^2 L_f^2 \|\bar{\boldsymbol{g}}_{k+1}\|^2 + 2L_f^2 \|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}}, \tag{3.74}$$

where the first inequality holds due to $\rho(\boldsymbol{H}) = 1$; and the last inequality holds due to (3.73). Then, we have

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2_{\boldsymbol{K}} &= \eta^2 \|\alpha \boldsymbol{L}\boldsymbol{x}_k + \beta \boldsymbol{v}_k + \boldsymbol{g}_{k+1}\|^2_{\boldsymbol{K}} = \eta^2 \|\alpha \boldsymbol{L}\boldsymbol{x}_k + \beta \boldsymbol{v}_k + \boldsymbol{g}^0_k + \boldsymbol{g}_{k+1} - \boldsymbol{g}^0_k\|^2_{\boldsymbol{K}} \\ &\le 3\eta^2 (\|\alpha \boldsymbol{L}\boldsymbol{x}_k\|^2 + \|\beta \boldsymbol{v}_k + \boldsymbol{g}^0_k\|^2_{\boldsymbol{K}} + \|\boldsymbol{g}_{k+1} - \boldsymbol{g}^0_k\|^2) \\ &\le 3\eta^2 (\alpha^2 \rho(L^2) \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \|\beta \boldsymbol{v}_k + \boldsymbol{g}^0_k\|^2_{\boldsymbol{K}} + 2\eta^2 L_f^2 \|\bar{\boldsymbol{g}}_{k+1}\|^2 + 2L_f^2 \|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}}) \\ &= \|\boldsymbol{x}_k\|^2_{3\eta^2\alpha^2\rho(L^2)\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{3\eta^2\beta^2\boldsymbol{K}} + 6\eta^4 L_f^2 \|\bar{\boldsymbol{g}}_{k+1}\|^2 + \|\boldsymbol{x}_{k+1}\|^2_{6\eta^2 L_f^2 \boldsymbol{K}}, \end{aligned} \tag{3.75}$$

where the first equality holds due to (3.70a); the first inequality holds due to the Cauchy-Schwarz inequality, (2.5), and $\rho(\boldsymbol{K}) = 1$; and the last inequality holds due to (2.6) and (3.73).

From (3.70a), similar to the way to get (3.33), we have

$$V_{1,k+1} \le V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2} + \frac{\eta}{2}(1+3\eta)\|\boldsymbol{g}_{k+1}-\boldsymbol{g}_k^0\|^2$$
$$- \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}}. \tag{3.76}$$

Then, from (3.76), the Cauchy-Schwarz inequality, (2.6), and (3.73), we have

$$\begin{aligned}V_{1,k+1} &\le V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2} + \frac{\eta}{2}(1+3\eta)\|\boldsymbol{g}_{k+1}-\boldsymbol{g}_k^0\|^2\\
&\quad - \eta\beta(\boldsymbol{x}_k - \boldsymbol{x}_{k+1} + \boldsymbol{x}_{k+1})^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}}\\
&\le V_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\rho_2(L)\boldsymbol{K}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\rho(L^2)\boldsymbol{K}} + \|\boldsymbol{x}_{k+1}\|^2_{\eta(1+3\eta)L_f^2\boldsymbol{K}} + \eta^3(1+3\eta)L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2\\
&\quad - \eta\beta\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \frac{1}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{2\eta^2\beta^2\boldsymbol{K}}.\end{aligned} \tag{3.77}$$

From (3.70b), similar to the way to get (3.35), we have

$$\begin{aligned}V_{2,k+1} &\le V_{2,k} + \eta\boldsymbol{x}_{k+1}^\top(\beta\boldsymbol{K}+\alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \|\boldsymbol{x}_{k+1}\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)}\\
&\quad + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + \Big(\frac{1}{\beta^2}+\frac{1}{2\eta\beta}\Big)\Big(\frac{1}{\rho_2(L)}+\frac{\alpha}{\beta}\Big)\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2.\end{aligned} \tag{3.78}$$

Then, from (3.78), (2.6), (2.8), and (3.72), we have

$$\begin{aligned}V_{2,k+1} &\le V_{2,k} + \eta\boldsymbol{x}_{k+1}^\top(\beta\boldsymbol{K}+\alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \|\boldsymbol{x}_{k+1}\|^2_{\eta^2\beta(\beta\rho(L)+\alpha\rho(L^2))\boldsymbol{K}}\\
&\quad + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{\eta}{2\beta}(\frac{1}{\rho_2(L)}+\frac{\alpha}{\beta})\boldsymbol{K}} + \eta\Big(\frac{\eta}{\beta^2}+\frac{1}{2\beta}\Big)\Big(\frac{1}{\rho_2(L)}+\frac{\alpha}{\beta}\Big)L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2.\end{aligned} \tag{3.79}$$

We have

$$\begin{aligned}V_{3,k+1} &= \boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1} + \frac{1}{\beta}\boldsymbol{g}_{k+1}^0\Big) = \boldsymbol{x}_{k+1}^\top\boldsymbol{K}(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0 + \eta\beta\boldsymbol{L}\boldsymbol{x}_{k+1} + \frac{1}{\beta}(\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0))\\
&= (\boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k + \beta\boldsymbol{v}_k + \boldsymbol{g}_k^0 + \boldsymbol{g}_{k+1} - \boldsymbol{g}_k^0))^\top\boldsymbol{K}(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0)\\
&\quad + \boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\eta\beta\boldsymbol{L}\boldsymbol{x}_{k+1} + \frac{1}{\beta}(\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0)\Big)\\
&= \boldsymbol{x}_k^\top(\boldsymbol{K}-\eta\alpha\boldsymbol{L})(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0) - \eta\beta\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} - \eta(\boldsymbol{g}_{k+1}-\boldsymbol{g}_k^0)^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big)\\
&\quad + \eta\beta\boldsymbol{x}_{k+1}^\top\boldsymbol{L}\boldsymbol{x}_{k+1} + \frac{1}{\beta}\boldsymbol{x}_{k+1}^\top\boldsymbol{K}(\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0)\end{aligned}$$

$$
\begin{aligned}
&\leq \boldsymbol{x}_k^\top(\boldsymbol{K}-\eta\alpha\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)-\eta\beta\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2+\frac{\eta}{2}\|\boldsymbol{g}_{k+1}-\boldsymbol{g}_k^0\|^2+\frac{\eta}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2\\
&\quad+\eta\beta\boldsymbol{x}_{k+1}^\top\boldsymbol{L}\boldsymbol{x}_{k+1}+\frac{\eta}{2}\|\boldsymbol{x}_{k+1}\|_{\boldsymbol{K}}^2+\frac{1}{2\eta\beta^2}\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2\\
&\leq \boldsymbol{x}_k^\top\boldsymbol{K}(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0)-\eta\alpha(\boldsymbol{x}_k-\boldsymbol{x}_{k+1}+\boldsymbol{x}_{k+1})^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\eta(\beta-\frac{1}{2})\boldsymbol{K}}^2\\
&\quad+\eta^3L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2+\eta L_f^2\|x_{k+1}\|_{\boldsymbol{K}}^2+\|\boldsymbol{x}_{k+1}\|_{\frac{\eta}{2}\boldsymbol{K}+\eta\beta\boldsymbol{L}}^2+\frac{\eta L_f^2}{2\beta^2}\|\bar{\boldsymbol{g}}_{k+1}\|^2\\
&\leq V_{3,k}-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\eta(\beta-\frac{1}{2})\boldsymbol{K}}^2+\frac{\eta^2\alpha^2}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2+\frac{\rho(L^2)}{2}\|\boldsymbol{x}_k-\boldsymbol{x}_{k+1}\|_{\boldsymbol{K}}^2\\
&\quad-\eta\alpha\boldsymbol{x}_{k+1}^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)+\Big(\eta^3+\frac{\eta}{2\beta^2}\Big)L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2+\|\boldsymbol{x}_{k+1}\|_{\eta(\frac{1}{2}+L_f^2)\boldsymbol{K}+\eta\beta\boldsymbol{L}}^2\\
&= V_{3,k}-\eta\alpha\boldsymbol{x}_{k+1}^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big)-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\eta(\beta-\frac{1}{2})\boldsymbol{K}-\frac{\eta^2\alpha^2}{2}\boldsymbol{K}}^2\\
&\quad+\frac{\rho(L^2)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|_{\boldsymbol{K}}^2+\Big(\eta^3+\frac{\eta}{2\beta^2}\Big)L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2+\|\boldsymbol{x}_{k+1}\|_{\eta(\frac{1}{2}+L_f^2)\boldsymbol{K}+\eta\beta\rho(L)\boldsymbol{K}}^2,
\end{aligned}
\tag{3.80}
$$

where the second equality holds due to (3.70b); the third equality holds due to (3.70a); the forth equality holds due to (2.5); the first inequality holds due to the Cauchy-Schwarz inequality, (2.5), and $\rho(\boldsymbol{K}) = 1$; the second inequality holds due to (3.72) and (3.73); the third inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds due to (2.6).

From $\tilde{f}$ is smooth and (3.71), similar to the way to get (3.39), we have

$$
V_{4,k+1}\leq V_{4,k}-\frac{\eta}{4}(1-2\eta L_f)\|\bar{\boldsymbol{g}}_{k+1}\|^2+\frac{\eta}{2}\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_{k+1}\|^2-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2. \tag{3.81}
$$

Then, from (3.81) and (3.74), we have

$$
V_{4,k+1}\leq V_{4,k}-\frac{\eta}{4}(1-2\eta L_f-4\eta^2L_f^2)\|\bar{\boldsymbol{g}}_{k+1}\|^2+\|\boldsymbol{x}_{k+1}\|_{\eta L_f^2\boldsymbol{K}}^2-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2. \tag{3.82}
$$

Then, we have

$$
\begin{aligned}
V_{k+1}&\leq V_k-\|\boldsymbol{x}_k\|_{\eta\alpha\rho_2(L)\boldsymbol{K}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\rho(L^2)\boldsymbol{K}}^2+\|\boldsymbol{x}_{k+1}\|_{\eta(1+3\eta)L_f^2\boldsymbol{K}}^2+\eta^3(1+3\eta)L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2\\
&\quad+\frac{1}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|_{\boldsymbol{K}}^2+\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{2\eta^2\beta^2\boldsymbol{K}}^2+\|\boldsymbol{x}_{k+1}\|_{\eta^2\beta(\beta\rho(L)+\alpha\rho(L^2))\boldsymbol{K}}^2\\
&\quad+\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\frac{\eta}{2\beta}(\frac{1}{\rho_2(L)}\boldsymbol{K}+\frac{\alpha}{\beta}\boldsymbol{K})}^2+\eta\Big(\frac{\eta}{\beta^2}+\frac{1}{2\beta}\Big)\Big(\frac{1}{\rho_2(L)}+\frac{\alpha}{\beta}\Big)L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2\\
&\quad-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|_{\eta(\beta-\frac{1}{2})\boldsymbol{K}-\frac{\eta^2\alpha^2}{2}\boldsymbol{K}}^2+\frac{\rho(L^2)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|_{\boldsymbol{K}}^2+(\eta^3+\frac{\eta}{2\beta^2})L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2\\
&\quad+\|\boldsymbol{x}_{k+1}\|_{\eta(\frac{1}{2}+L_f^2)\boldsymbol{K}+\eta\beta\rho(L)\boldsymbol{K}}^2-\frac{\eta}{4}(1-2\eta L_f-4\eta^2L_f^2)\|\bar{\boldsymbol{g}}_{k+1}\|^2+\|\boldsymbol{x}_{k+1}\|_{\eta L_f^2\boldsymbol{K}}^2-\frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2
\end{aligned}
$$

$$
\begin{aligned}
&= V_k - \|\boldsymbol{x}_k\|^2_{\eta(\alpha\rho_2(L)-\frac{1}{2})\boldsymbol{K}-\eta^2\frac{3\alpha^2}{2}\rho(L^2)\boldsymbol{K}} + \|\boldsymbol{x}_{k+1}\|^2_{\eta\varepsilon_1\boldsymbol{K}+\eta^2(3L_f^2+\beta^2\rho(L)+\alpha\beta\rho(L^2))\boldsymbol{K}} \\
&\quad - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta\varepsilon_5\boldsymbol{K}-\eta^2(\frac{\alpha^2}{2}+2\beta^2)\boldsymbol{K}} + \frac{1+\rho(L^2)}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 \\
&\quad - \eta(\varepsilon_7 - \eta\varepsilon_8 - 3\eta^2 L_f^2(1+\eta))\|\bar{\boldsymbol{g}}_{k+1}\|^2 \\
&\le V_k - \|\boldsymbol{x}_k\|^2_{\eta(\alpha\rho_2(L)-\frac{1}{2})\boldsymbol{K}-\eta^2\frac{3\alpha^2}{2}\rho(L^2)\boldsymbol{K}} + \|\boldsymbol{x}_{k+1}\|^2_{\eta\varepsilon_1\boldsymbol{K}+\eta^2(3L_f^2+\beta^2\rho(L)+\alpha\beta\rho(L^2))\boldsymbol{K}} - \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 \\
&\quad + \frac{1+\rho(L^2)}{2}\Big(\|\boldsymbol{x}_k\|^2_{3\eta^2\alpha^2\rho(L^2)\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{3\eta^2\beta^2\boldsymbol{K}} + 6\eta^4L_f^2\|\bar{\boldsymbol{g}}_{k+1}\|^2 + \|\boldsymbol{x}_{k+1}\|^2_{6\eta^2L_f^2\boldsymbol{K}}\Big) \\
&\quad - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta\varepsilon_5\boldsymbol{K}-\eta^2(\frac{\alpha^2}{2}+2\beta^2)\boldsymbol{K}} - \eta(\varepsilon_7 - \eta\varepsilon_8 - 3\eta^2L_f^2(1+\eta))\|\bar{\boldsymbol{g}}_{k+1}\|^2,
\end{aligned}
\tag{3.83}
$$

where the first inequality holds due to (3.77), (3.79), (3.79), and (3.82); and the second inequality holds due to (3.75).

Combining and rearranging terms in (3.83) gives (3.68). □

We are now ready to prove Theorem 3.3.

From $\frac{1}{\rho_2(L)}(\rho(L)\beta + \chi_1) < \alpha$, we have $\frac{\alpha}{\beta} > \frac{\rho(L)}{\rho_2(L)} \ge 1$. Then, we know $\varepsilon_{12} > \frac{1}{2}$. Thus, $2 - \frac{1}{\varepsilon_{12}} > 0$. Hence,

$$\varepsilon_{15} > 0. \tag{3.84}$$

Then, from $0 < \eta = \frac{1}{\gamma} < \varepsilon_{15}$, we have $4\varepsilon_{11}\varepsilon_{12} > 1$. Hence,

$$\frac{1}{2} > \varepsilon_{11} - \varepsilon_{13} > 0. \tag{3.85}$$

From $\frac{1}{\rho_2(L)}(\rho(L)\beta + \chi_1) < \alpha$, we have

$$\varepsilon_3 = \alpha\rho_2(L) - \beta\rho(L) - 2L_f^2 - 2 > \chi_1 - 2L_f^2 - 2 = 0. \tag{3.86}$$

Hence, from $0 < \eta < \frac{\varepsilon_3}{\varepsilon_4}$ and (3.86), we have

$$\eta(\varepsilon_3 - \eta\varepsilon_4) > 0. \tag{3.87}$$

From $\alpha \le \chi_2\beta$ and $\beta > \chi_3$, we have

$$\varepsilon_5 \ge \Big(\beta - \frac{1}{2} - \frac{\chi_2}{2\beta}\Big) - \frac{1}{2\beta\rho_2(L)} > 0. \tag{3.88}$$

Hence, from $0 < \eta < \frac{\varepsilon_5}{\varepsilon_6}$ and (3.88), we have

$$\eta(\varepsilon_5 - \eta\varepsilon_6) > 0. \tag{3.89}$$

From (3.87) and (3.89), we have

$$\varepsilon_{16} > 0. \tag{3.90}$$

From $\alpha \le \chi_2\beta$ and $\beta > \chi_4$, we have

$$\varepsilon_7 \ge \frac{1}{4} - \frac{1}{2\beta}\Big(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \chi_2\Big)L_f^2 > 0. \tag{3.91}$$

From $\chi_2 > 1$, we have $\chi_3 > 1$. Thus, $\beta > 1$. Thus, $\eta < \frac{\varepsilon_5}{\varepsilon_6} < \frac{2}{7\beta} < \frac{2}{7}$. Hence, from $0 < \eta < \frac{\varepsilon_7}{\varepsilon_8+\varepsilon_9+\varepsilon_{10}}$ and (3.91), we have

$$\eta(\varepsilon_7 - \varepsilon_8\eta - \varepsilon_9\eta^2 - \varepsilon_{10}\eta^3) > \eta(\varepsilon_7 - \varepsilon_8\eta - \varepsilon_9\eta - \varepsilon_{10}\eta) > 0. \tag{3.92}$$

Noting that $\beta > \chi_4 > \sqrt{2}L_f$ and $0 < \varepsilon_5 < \beta$, we know $\gamma > \frac{\varepsilon_6}{\varepsilon_5} > \frac{\varepsilon_6}{\beta} > \frac{7\beta}{2} > \frac{7\sqrt{2}L_f}{2} > L_f$. Thus, the conditions needed in Lemma 3.2 are all satisfied. Thus, (3.68) holds.

We know

$$\begin{aligned}
\tilde{V}_k &= \Big(\frac{1}{2} - \varepsilon_1\eta - \varepsilon_2\eta^2\Big)\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K}} + \boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + V_{4,k} \\
&\ge \varepsilon_{11}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \varepsilon_{12}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} - \varepsilon_{13}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{4\varepsilon_{13}}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + V_{4,k} \\
&= (\varepsilon_{11} - \varepsilon_{13})\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\Big) + V_{4,k} \qquad (3.93) \\
&\ge (\varepsilon_{11} - \varepsilon_{13})\hat{V}_k \ge 0, \qquad (3.94)
\end{aligned}$$

where $\hat{V}_k$ is defined in (3.45); the first inequality holds due to (2.8) and the Cauchy-Schwarz inequality; the second equality holds due to $\varepsilon_{11} - \varepsilon_{13} = \varepsilon_{12} - \frac{1}{4\varepsilon_{13}}$; and the last inequality holds due to (3.85). Similarly, we have

$$\tilde{V}_k \le V_k \le \varepsilon_{14}\hat{V}_k. \tag{3.95}$$

From (3.68), (3.92) and $\boldsymbol{K} \ge 0$, we know that

$$\begin{aligned}
\tilde{V}_{k+1} &\le \tilde{V}_k - \|\boldsymbol{x}_k\|^2_{\eta(\varepsilon_1-\varepsilon_3-\eta(\varepsilon_2+\varepsilon_4))\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta(\varepsilon_5-\eta\varepsilon_6)\boldsymbol{K}} - \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 \\
&\le \tilde{V}_k - \varepsilon_{16}\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}_k^0\|^2\Big). \qquad (3.96)
\end{aligned}$$

Then, (3.96) yields

$$\sum_{k=0}^{T}\tilde{V}_{k+1} \le \sum_{k=0}^{T}\tilde{V}_k - \varepsilon_{16}\sum_{k=0}^{T}\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}_k^0\|^2\Big). \tag{3.97}$$

Then, (3.97) yields

$$\tilde{V}_{T+1} + \varepsilon_{16}\sum_{k=0}^{T}\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}_k^0\|^2\Big) \le \tilde{V}_0. \tag{3.98}$$

From (3.98), (3.90), and (3.94) we know that

$$\frac{\sum_{k=0}^{T}(\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|_{\boldsymbol{K}}^2 + \|\bar{\boldsymbol{g}}_k^0\|^2)}{T+1} \le \frac{\tilde{V}_0}{\varepsilon_{16}(T+1)},\ \forall T \in \mathbb{N}_0, \tag{3.99}$$

which yields (3.16) and (3.17).

From (3.98), (3.85), (3.90), and (3.93) we know that

$$f(\bar{x}_{T+1}) - f^* \le \frac{\tilde{V}_0}{n},\ \forall T \in \mathbb{N}_0, \tag{3.100}$$

which gives (3.18).

3.8.4 Proof of Theorem 3.4

In addition to the notations defined in Sections 3.8.1 and 3.8.3, we also denote the following notations.

$$\varepsilon_0 = \frac{\tilde{V}_0}{\varepsilon_{11} - \varepsilon_{13}},\ \varepsilon = 1 - \frac{\varepsilon_{17}}{\varepsilon_{14}},\ \varepsilon_{17} = \frac{1}{\gamma}\min\Big\{\varepsilon_3 - \frac{1}{\gamma}\varepsilon_4,\ \varepsilon_5 - \frac{1}{\gamma}\varepsilon_6,\ \frac{\nu}{2}\Big\}.$$

From (3.87) and (3.89), we have

$$\varepsilon_{17} > 0. \tag{3.101}$$

Then, from (3.92), (3.45), (3.61), (3.101), and (3.95) we have

$$\tilde{V}_{k+1} \le \tilde{V}_k - \varepsilon_{17}\hat{V}_k \le \tilde{V}_k - \frac{\varepsilon_{17}}{\varepsilon_{14}}\tilde{V}_k. \tag{3.102}$$

Noting that $\varepsilon_5 < \beta$, $\varepsilon_6 > \frac{7}{2}\beta^2$, and $\varepsilon_{14} > \frac{\alpha+\beta}{2\beta} > 1$, we have

$$0 < \varepsilon_{17} \le \eta(\varepsilon_5 - \eta\varepsilon_6) \le \frac{\varepsilon_5^2}{4\varepsilon_6} < \frac{1}{14}. \tag{3.103}$$

From (3.102), (3.94), and (3.103), we have

$$\tilde{V}_{k+1} \le \Big(1 - \frac{\varepsilon_{17}}{\varepsilon_{14}}\Big)\tilde{V}_k \le \Big(1 - \frac{\varepsilon_{17}}{\varepsilon_{14}}\Big)^{k+1}\tilde{V}_0. \tag{3.104}$$

Hence, from (3.94) and (3.85), we have

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) = \|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + n(f(\bar{x}_k) - f^*) \le \hat{V}_k \le \frac{\tilde{V}_k}{\varepsilon_{11} - \varepsilon_{13}}. \tag{3.105}$$

Hence, (3.104) and (3.105) give

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) \le \frac{\tilde{V}_0}{\varepsilon_{11} - \varepsilon_{13}}\Big(1 - \frac{\varepsilon_{17}}{\varepsilon_{14}}\Big)^k,\ \forall k \in \mathbb{N}_0, \tag{3.106}$$

which yields (3.19).

3.8.5 Proof of Theorem 3.5

In addition to the notations defined in Sections 3.8.1 and 3.8.3, we also denote the following notations.

$$\breve{c}_\beta = \max\Big\{\frac{\breve{\chi}_1}{\chi_2\rho_2(L)-\rho(L)},\ \chi_3,\ \chi_4\Big\},\ \breve{c}_\gamma = \max\Big\{\frac{\breve{\varepsilon}_4}{\breve{\varepsilon}_3},\ \frac{\varepsilon_6}{\varepsilon_5},\ \frac{\varepsilon_8}{\varepsilon_7},\ \frac{1}{\breve{\varepsilon}_{15}}\Big\},$$
$$\breve{\chi}_1 = \frac{3}{2}L_f^2+1,\ \breve{\varepsilon}_{11} = \frac{1}{2}-\frac{1}{\gamma}\breve{\varepsilon}_1-\frac{1}{\gamma^2}\breve{\varepsilon}_2,\ \breve{\varepsilon}_{13} = \frac{1}{2}\Big(\breve{\varepsilon}_{11}-\varepsilon_{12}+((\breve{\varepsilon}_{11}-\varepsilon_{12})^2+1)^{\frac{1}{2}}\Big),$$
$$\breve{\varepsilon}_{15} = \frac{1}{2\breve{\varepsilon}_2}\Big(-\breve{\varepsilon}_1+\Big(\breve{\varepsilon}_1^2+2-\frac{1}{\varepsilon_{12}}\Big)^{\frac{1}{2}}\Big),\ \breve{\varepsilon}_{16} = \frac{1}{\gamma}\min\Big\{\breve{\varepsilon}_3-\frac{1}{\gamma}\breve{\varepsilon}_4,\ \varepsilon_5-\frac{1}{\gamma}\varepsilon_6,\ \frac{1}{4}\Big\}.$$

Similar to Lemma 3.2, we have the following lemma, which presents the general relations of two consecutive outputs of Algorithm 3.3.

Lemma 3.3. *Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 3.3. If Assumptions 3.1–3.3 hold, then*

$$\breve{V}_{k+1} \le \breve{V}_k - \|\boldsymbol{x}_k\|^2_{\frac{1}{\gamma}(\breve{\varepsilon}_3-\frac{1}{\gamma}\breve{\varepsilon}_4)\boldsymbol{K}} - \Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{1}{\gamma}(\varepsilon_5-\frac{1}{\gamma}\varepsilon_6)\boldsymbol{K}} - \frac{1}{4\gamma}\|\bar{\boldsymbol{g}}_k^0\|^2 - \frac{1}{\gamma}\Big(\varepsilon_7-\frac{1}{\gamma}\varepsilon_8\Big)\|\bar{\boldsymbol{g}}_k\|^2, \tag{3.107}$$

where

$$\breve{V}_k = V_k - \|\boldsymbol{x}_k\|^2_{\frac{1}{\gamma}(\breve{\varepsilon}_1+\frac{1}{\gamma}\breve{\varepsilon}_2)\boldsymbol{K}},\ \breve{\varepsilon}_1 = \frac{1}{2}+\beta\rho(L),\ \breve{\varepsilon}_2 = \beta^2\rho(L)+\alpha\beta\rho(L^2),$$
$$\breve{\varepsilon}_3 = \frac{1}{2}(2\alpha\rho_2(L)-1-3L_f^2)-\breve{\varepsilon}_1,\ \breve{\varepsilon}_4 = \frac{3}{2}(2+\rho(L^2))(\alpha^2\rho(L^2)+L_f^2)+\breve{\varepsilon}_2.$$

Proof. This proof is similar to the proof of Lemma 3.2 with some modifications.

We rewrite the distributed linearized ADMM algorithm (3.21) into the compact form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k+\beta\boldsymbol{v}_k+\boldsymbol{g}_k), \tag{3.108a}$$
$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \eta\beta\boldsymbol{L}\boldsymbol{x}_{k+1},\ \forall\boldsymbol{x}_0\in\mathbb{R}^{np},\ \sum_{j=1}^n v_{j,0} = \boldsymbol{0}_p. \tag{3.108b}$$

From (3.108), we know that (3.29)–(3.31) still hold. Thus, (3.40) also holds.

From (3.30) and $\rho(\boldsymbol{H}) = 1$, similar to the way to get (3.75), we have

$$\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2_{\boldsymbol{K}} \le \|\boldsymbol{x}_k\|^2_{3\eta^2(\alpha^2\rho(L^2)+L_f^2)\boldsymbol{K}} + \Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{3\eta^2\beta^2\boldsymbol{K}}. \tag{3.109}$$

From (3.108a) and (3.30), similar to the way to get (3.77), we have

$$V_{1,k+1} \le V_{1,k} - \|\boldsymbol{x}_k\|^2_{\frac{\eta}{2}(2\alpha\rho_2(L)-1-L_f^2)\boldsymbol{K}-\frac{3\eta^2}{2}(\alpha^2\rho(L^2)+L_f^2)\boldsymbol{K}}$$
$$- \eta\beta\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \frac{1}{2}\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{2\eta^2\beta^2\boldsymbol{K}}. \tag{3.110}$$

From (3.108b) and (3.32), similar to the way to get (3.79), we have

$$V_{2,k+1} \le V_{2,k} + \eta \boldsymbol{x}_{k+1}^\top(\beta \boldsymbol{K} + \alpha \boldsymbol{L})(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0) + \|\boldsymbol{x}_{k+1}\|^2_{\eta^2\beta(\beta\rho(L)+\alpha\rho(L^2))\boldsymbol{K}}$$
$$+ \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|^2_{\frac{\eta}{2\beta}(\frac{1}{\rho_2(L)}+\frac{\alpha}{\beta})\boldsymbol{K}} + \eta(\frac{\eta}{\beta^2} + \frac{1}{2\beta})(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta})L_f^2\|\bar{\boldsymbol{g}}_k\|^2. \quad (3.111)$$

From (3.108), (3.29), (3.30), and (3.32), similar to the way to get (3.80), we have

$$V_{3,k+1} \le V_{3,k} - \eta\alpha \boldsymbol{x}_{k+1}^\top \boldsymbol{L}(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0) - \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|^2_{\eta(\beta-\frac{1}{2})\boldsymbol{K}-\frac{\eta^2\alpha^2}{2}\boldsymbol{K}}$$
$$+ \frac{\rho(L^2)}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \|x_k\|^2_{\frac{\eta}{2}L_f^2\boldsymbol{K}} + \|\boldsymbol{x}_{k+1}\|^2_{\frac{\eta}{2}(1+2\beta\rho(L))\boldsymbol{K}} + \frac{\eta L_f^2}{2\beta^2}\|\bar{\boldsymbol{g}}_k\|^2. \quad (3.112)$$

From (3.109)–(3.112) and (3.40), similar to the way to get (3.68), we know that (3.107) holds. □

Finally, similar to the way to get (3.99) and (3.100), we have

$$\frac{\sum_{k=0}^{T}(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}_k^0\|^2)}{T+1} \le \frac{\breve{V}_0}{\breve{\varepsilon}_{16}(T+1)}, \ \forall T \in \mathbb{N}_0, \quad (3.113)$$
$$f(\bar{x}_{T+1}) - f^* \le \frac{\breve{V}_0}{n}, \ \forall T \in \mathbb{N}_0. \quad (3.114)$$

From (3.113), we have (3.22) and (3.23).
From (3.114), we have (3.24).

3.8.6 Proof of Theorem 3.6

In addition to the notations defined in Sections 3.8.1, 3.8.3, and 3.8.5, we also denote the following notations.

$$\breve{\varepsilon}_0 = \frac{\breve{V}_0}{\breve{\varepsilon}_{11} - \breve{\varepsilon}_{13}}, \ \breve{\varepsilon} = 1 - \frac{\breve{\varepsilon}_{17}}{\varepsilon_{14}}, \ \breve{\varepsilon}_{17} = \frac{1}{\gamma}\min\Big\{\breve{\varepsilon}_3 - \frac{1}{\gamma}\breve{\varepsilon}_4, \ \varepsilon_5 - \frac{1}{\gamma}\varepsilon_6, \ \frac{\nu}{2}\Big\}.$$

Similar to they way to get (3.106), we have

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) \le \frac{\breve{V}_0}{\breve{\varepsilon}_{11} - \breve{\varepsilon}_{13}}\Big(1 - \frac{\breve{\varepsilon}_{17}}{\varepsilon_{14}}\Big)^k, \ \forall k \in \mathbb{N}_0, \quad (3.115)$$

which yields (3.25)

Chapter 4

Distributed primal–dual SGD optimization algorithm

In this chapter, we consider the distributed nonconvex optimization problem with stochastic gradient feedback, i.e., each agent is only able to collect stochastic gradients of its local cost function. We propose a distributed primal–dual stochastic gradient descent (SGD) algorithm, suitable for arbitrarily connected communication networks and any smooth cost functions. We show that the proposed algorithm converges to a stationary point with the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ for smooth nonconvex cost functions, and to a global optimum with the linear speedup convergence rate $\mathcal{O}(1/(nT))$ when the global cost function satisfies the P–Ł condition in addition, where n and T are the number of agents and the total number of iterations, respectively. We also show that the output of the proposed algorithm with constant parameters linearly converges to a neighborhood of a global optimum. We finally demonstrate through numerical simulations the efficiency of our algorithm in comparison with the baseline centralized SGD and recently proposed distributed SGD algorithms.

This chapter is organized as follows. Section 4.1 gives the background. Section 4.2 introduces the problem formulation and assumptions. Section 4.3 presents the distributed primal–dual SGD algorithm and analyzes its convergence properties. Simulations are given in Section 4.4. Concluding remarks are offered in Section 4.5. To improve the readability, all the proofs can be found in Section 4.6

4.1 Introduction

In general, SGD algorithms are suitable for scenarios where explicit expressions of the gradients are unavailable or at least difficult to obtain. For example, in empirical risk minimization, the actual gradient hass to be calculated from the entire data set, which results in a heavy computational burden. A stochastic gradient can be calculated from a randomly selected subset of the data and is often an efficient way to replace the actual gradient. Other examples when SGD algorithms are suitable include scenarios where data are arriving sequentially such as in online learning [244].

When the communication network is a star graph, various parallel SGD algorithms have been proposed. A potential performance bottleneck of such algorithms lies on the communication burden of the master. To overcome this issue, a promising strand of research is combining parallel SGD algorithms with communication reduction approaches, e.g., asynchronous parallel SGD algorithms [245–249], gradient compression based parallel SGD algorithms [246,250–253], periodic averaging based parallel SGD algorithms [33, 251, 252, 254–257], and parallel SGD algorithm with dynamic batch sizes [258]. Convergence properties of these algorithms have been analyzed in detail. In particular, in [33,251,255,258], the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ has been established for smooth nonconvex cost functions, where n and T are the number of agents and the total number of iterations, respectively. In [257,258], the convergence rate has been improved to $\mathcal{O}(1/(nT))$ when the global cost function satisfies the P–Ł condition, which also achieves a linear speedup. In addition to the star architecture restriction, aforementioned parallel SGD algorithms require certain restrictions on the cost functions, such as bounded gradients of the local cost functions or bounded difference between the gradients of the local and global cost functions.

Distributed algorithms executed over arbitrarily connected communication networks have been suggested to overcome communication bottlenecks for parallel SGD algorithms. Various distributed SGD algorithms have been proposed, e.g., synchronous distributed SGD algorithms [31, 33, 129, 130], asynchronous distributed SGD algorithms [131, 132], compression based distributed SGD algorithms [133–136], and periodic averaging based distributed SGD algorithm [137]. Convergence properties of these algorithms have been analyzed and the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ has been established for smooth nonconvex cost functions in [31, 33, 132, 133, 135–137]. However, similar to aforementioned parallel SGD algorithms, these distributed algorithms require restrictive assumptions on the cost functions. In order to remove these restrictions, the authors of [32] proposed a variant of the distributed SGD algorithm proposed in [31], named D^2, in which each agent stores the stochastic gradient and its local model in last iteration and linearly combines them with the current stochastic gradient and local model. For this algorithm the authors established the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$, but they required that the eigenvalues of the mixing matrix associated with the communication network are strictly greater than $-1/3$. The authors of [138, 139] proposed distributed stochastic gradient tracking algorithms suitable for arbitrarily connected communication networks. However, these algorithms only achieve an $\mathcal{O}(1/\sqrt{T})$ convergence rate, which is not a speedup. Moreover, gradient tracking algorithms have the common potential drawback that in order to track the global gradient, at each iteration each agent needs to communicate one additional p-dimensional variable with its neighbors. This results in heavy communication burden when p is large. Note that all aforementioned distributed SGD algorithms converge to stationary points, which may be local or global optima, or saddle points. None of existing studies on distributed SGD algorithms consider finding global optima when the global cost function satisfies some additional properties, such as the P–Ł condition studied for the parallel algorithms in [257, 258].

Noting above, two core theoretical questions with important practical relevance arise.

(Q4.1) Are there any distributed SGD algorithms that not only are suitable for arbitrarily connected communication networks and any smooth cost functions, but also find stationary points with the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$?

(Q4.2) If the P–Ł condition holds in addition, can the above SGD algorithms find global optima with the linear speedup convergence rate $\mathcal{O}(1/(nT))$?

This chapter provides positive answers to the above two questions. More specifically, the contributions of this chapter are summarized as follows.

(C4.1) We propose a distributed primal–dual SGD algorithm (Algorithm 4.1), which is suitable for arbitrarily connected communication networks and any smooth (possibly nonconvex) cost functions.

(C4.2) We show in Corollary 4.1 that our algorithm finds a stationary point with the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ for smooth nonconvex cost functions, thus (Q5.1) is answered. Compared with [31–33, 132, 133, 135–137, 251, 255, 258], we achieve the same convergence rate but under weaker assumptions related to network architectures and/or cost functions, and compared with [138, 139], we not only achieve linear speedup but also just use half communication in each iteration.

(C4.3) We show in Theorem 4.3 that our algorithm finds a global optimum with the linear speedup convergence rate $\mathcal{O}(1/(nT))$ when the global cost function satisfies the P–Ł condition, thus (Q5.2) is answered. Compared with [136, 140, 141, 257–259] , we achieve the same convergence rate but under weaker assumptions related to network architectures and/or cost functions, and compared with [129,252,260–264], we not only establish linear speedup but also relax the strong convexity by the P–Ł condition.

(C4.4) We show in Theorems 4.4 and 4.5 that the output of our algorithm with constant parameters linearly converges to a neighborhood of a global optimum when the global cost function satisfies the P–Ł condition. Compared with [129, 264–267], which used the strong convexity assumption, we achieve the similar convergence result under weaker assumptions on the cost function.

Table 4.1 compares this chapter with other SGD optimization algorithms.

4.2 Distributed nonconvex optimization with stochastic gradient feedback

Consider a network of n agents, each of which has a local cost function $f_i : \mathbb{R}^p \to \mathbb{R}$. All agents collaborate to solve the optimization problem

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{4.1}$$

Table 4.1: Comparison of Chapter 4 to some related SGD optimization algorithms.

Reference	Problem type	Extra assumption	Communication strategy	Communication rounds	Convergence rate
[251]	Nonconvex	Bounded $\Vert\nabla f_i - \nabla f\Vert$	Star graph, one quantized variable	$O(n^{5/4}T^{3/4})$	$O(1/\sqrt{nT})$
[252]	Nonconvex	Identical ∇f_i	Star graph, one quantized variable	$O(T)$	$O(1/\sqrt{T})$
	Strongly convex				$O(1/T)$
[255]	Nonconvex	Bounded $\Vert\nabla f_i\Vert$	Star graph, one variable	$O(n^{3/4}T^{3/4})$	$O(1/\sqrt{nT})$
[33]	Nonconvex	Bounded $\Vert\nabla f_i - \nabla f\Vert$	Star graph, two variables	$O(n^{3/4}T^{3/4})$	$O(1/\sqrt{nT})$
			Connected graph, two variables	$O(T)$	
[257]	P–Ł condition	Identical ∇f_i	Star graph, one variable	$O((nT)^{1/3})$	$O(1/(nT))$
[258]	Nonconvex	Identical ∇f_i, exponentially increasing batch size	Star graph, one variable	$O(\sqrt{nT}\log(T/n))$	$O(1/\sqrt{nT})$
	P–Ł condition			$O(\log(T))$	$O(1/(nT))$
[129]	Nonconvex	Bounded $\Vert\nabla f_i\Vert$	Connected graph, one variable	$O(T)$	$O(1/T^{\theta}),\ \forall\theta\in(0,0.5)$
	Strongly convex				$O(1/T)$; linearly to a neighbor
[31]	Nonconvex	Bounded $\Vert\nabla f_i - \nabla f\Vert$	Connected graph, one variable	$O(T)$	$O(1/\sqrt{nT})$
[132]	Nonconvex	Bounded $\Vert\nabla f_i - \nabla f\Vert$	Uniformly jointly strongly connected digraph, one variable	$O(T)$	$O(1/\sqrt{nT})$
[133]	Nonconvex	Bounded $\Vert\nabla f_i - \nabla f\Vert$	Connected graph, one compressed variable	$O(T)$	$O(1/\sqrt{nT})$
[135]	Nonconvex	Bounded $\Vert\nabla f_i\Vert$	Strongly connected digraph, one quantized variable	$O(T)$	$O(1/\sqrt{nT})$
[136]	Nonconvex	Bounded $\Vert\nabla f_i\Vert$	Connected graph, one compressed variable	Event-triggered	$O(1/\sqrt{nT})$
	Strongly convex				$O(1/(nT))$
[137]	Nonconvex	Identical ∇f_i	Connected graph, one variable	$O(n^{3/2}\sqrt{T})$	$O(1/\sqrt{nT})$
[32]	Nonconvex	The eigenvalues of the mixing matrix are strictly greater than $-1/3$	Connected graph, one variable	$O(T)$	$O(1/\sqrt{nT})$
[138, 139]	Nonconvex	No	Connected graph, two variables	$O(T)$	$O(1/\sqrt{T})$
[259]	Strongly convex	Bounded $\Vert\nabla f_i\Vert$	Star graph, one variable	$O(\sqrt{T/n})$	$O(1/(nT))$
[140]	Strongly convex	Bounded $\Vert\nabla f_i\Vert$	Connected graph, one compressed variable	$O(T)$	$O(1/(nT))$
[141]	Strongly convex	No	Connected graph, two variables	$O(T)$	$O(1/(nT))$
[260]	Strongly convex	Identical ∇f_i	Connected graph, one variable	$O(T)$	$O(1/T)$
[261]	Strongly convex	No	Connected graph, one variable	$O(\sqrt{T})$	$O(1/T)$
[262]	Strongly convex	Bounded $\Vert\nabla f_i\Vert$	Uniformly jointly strongly connected digraph, one variable	$O(T)$	$O(1/T)$
[263]	Strongly convex	No	Connected graph in expectation, one variable	$O(T)$	$O(1/T)$
[264]	Strongly convex	No	Connected graph, one variable	$O(T)$	$O(1/T)$; linearly to a neighbor
[265]	Strongly convex	No	Connected graph, one variable	$O(T)$	Linearly to a neighbor
[266]	Strongly convex	No	Connected graph, two variables	$O(T)$	Linearly to a neighbor
[267]	Strongly convex	No	Strongly connected digraph, two variables	$O(T)$	Linearly to a neighbor
This chapter	Nonconvex	No	Connected graph, one variable	$O(T)$	$O(1/\sqrt{nT})$
	P–Ł condition				$O(1/(T^{\theta})),\ \forall\theta\in(0,1)$; linearly to a neighbor
					$O(1/(nT))$

This is the same as the distributed nonconvex optimization problem (3.1). However, in this chapter, we consider the case where each agent is only able to collect the stochastic gradients rather than the actual gradient of its local cost function. Specifically, at each iteration k and given any $x \in \mathbb{R}^p$, each agent i knows $g_i(x, \xi_{i,k})$ which is a stochastic estimation of $\nabla f_i(x)$, where $\xi_{i,k}$ is a random variable.

Based on the definitions introduced in Chapter 2, the following assumptions are made.

Assumption 4.1. *The communication among agents is described by a weighted undirected connected graph $\mathcal{G}$.*

Assumption 4.2. *The set $\mathbb{X}^*$ is nonempty and $f^* > -\infty$, where $\mathbb{X}^*$ and f^* denote the optimal set and the minimum function value of the optimization problem* (4.1)*, respectively.*

Assumption 4.3. *Each local cost function $f_i(x)$ is smooth with constant $L_f > 0$.*

Assumption 4.4. *The global cost function $f(x)$ satisfies the P–Ł condition with constant $\nu > 0$.*

Assumption 4.5. *The random variables $\{\xi_{i,k},\ i \in [n],\ k \in \mathbb{N}_0\}$ are independent of each other.*

Assumption 4.6. *The stochastic gradient $g_i(x, \xi_{i,k})$ is unbiased, i.e., for all $i \in [n]$, $k \in \mathbb{N}_0$, and $x \in \mathbb{R}^p$,*

$$\mathbf{E}_{\xi_{i,k}}[g_i(x, \xi_{i,k})] = \nabla f_i(x). \tag{4.2}$$

Assumption 4.7. *The stochastic gradient $g_i(x, \xi_{i,k})$ has bounded variance, i.e., there exists a constant σ such that for all $i \in [n]$, $k \in \mathbb{N}_0$, and $x \in \mathbb{R}^p$,*

$$\mathbf{E}_{\xi_{i,k}}[\|g_i(x, \xi_{i,k}) - \nabla f_i(x)\|^2] \leq \sigma^2. \tag{4.3}$$

Remark 4.1. *Assumptions 4.5 and 4.6 are standard in the study of using SGD methods to solve optimization problems. The bounded variance assumption (Assumption 4.7) is weaker than the bounded second moment (or bounded gradient) assumption made in [129, 130, 135, 136, 140, 143, 245–247, 249, 253, 255, 259, 262]. Moreover, note that we make no assumption on the boundedness of the deviation between the gradients of local cost functions. In other words, we do not assume that $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x) - \nabla f(x)\|^2$ is uniformly bounded, which is commonly done in studies of deep learning, e.g., [31, 33, 131–133, 251, 255]. Also, we do not assume that the mean of each local stochastic gradient is the gradient of the global cost function, i.e., $\mathbf{E}_{\xi}[g_i(x, \xi)] = \nabla f(x)$, $\forall x \in \mathbb{R}^p$, $\forall i \in [n]$, which is commonly assumed in studies of empirical risk minimization and stochastic optimization, e.g., [134, 137, 248, 250, 252, 254, 256–258, 260].*

Our goal in this chapter is to answer (Q4.1) and (Q4.2), i.e., solve the following problem.

Problem 4.1. *Propose a distributed SGD algorithm for the nonconvex optimization problem* (4.1) *such that stationary points or global optima can be found at linear speedup convergence rates.*

Algorithm 4.1 Distributed Primal–Dual SGD Algorithm

Input: parameters $\{\alpha_k\}$, $\{\beta_k\}$, $\{\eta_k\} \subseteq (0, +\infty)$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p, \ \forall i \in [n]$.
for $k = 0, 1, \ldots$ **do**
    for $i = 1, \ldots, n$ in parallel **do**
        Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
        Sample stochastic gradient $g_i(x_{i,k}, \xi_{i,k})$;
        Update $x_{i,k+1}$ by (4.4a);
        Update $v_{i,k+1}$ by (4.4b).
    end for
end for
Output: $\{\boldsymbol{x}_k\}$.

4.3 Distributed primal–dual SGD algorithm

In this section, we propose a distributed SGD algorithm and analyze its convergence properties.

4.3.1 Algorithm description

Based on the distributed primal–dual FO algorithm (3.7), we propose the distributed primal–dual SGD algorithm

$$x_{i,k+1} = x_{i,k} - \eta_k\Big(\alpha_k \sum_{j=1}^{n} L_{ij} x_{j,k} + \beta_k v_{i,k} + g^u_{i,k}\Big), \tag{4.4a}$$

$$v_{i,k+1} = v_{i,k} + \eta_k\beta_k \sum_{j=1}^{n} L_{ij} x_{j,k}, \ \forall x_{i,0} \in \mathbb{R}^p, \ v_{i,0} = \mathbf{0}_p, \ \forall i \in [n], \tag{4.4b}$$

where $\eta_k > 0$ is the stepsize at iteration k, $\alpha_k > 0$ and $\beta_k > 0$ are the values of the parameters α and β at iteration k, respectively, $g^u_{i,k} = g_i(x_{i,k}, \xi_{i,k})$ is the stochastic gradient of f_i at $x_{i,k}$, and $\xi_{i,k}$ is a random variable.

We present the distributed stochastic gradient primal–dual algorithm (4.4) in pseudocode as Algorithm 4.1.

It should be pointed out that $\{\alpha_k\}$, $\{\beta_k\}$, $\{\eta_k\}$, $x_{i,0}$, $v_{i,0}$, and $v_{i,1}$ in Algorithm 4.1 are deterministic, while $\{x_{i,k}\}_{k\geq 1}$ and $\{v_{i,k}\}_{k\geq 2}$ are random variables generated by Algorithm 4.1. Let $\mathfrak{F}_k$ denote the σ-algebra generated by the random variables $\xi_{1,k}, \ldots, \xi_{n,k}$ and let $\mathcal{F}_k = \bigcup_{s=1}^{k} \mathfrak{F}_s$. It is straightforward to see that $x_{i,k}$ and $v_{i,k+1}, \ i \in [p]$ depend on $\mathcal{F}_{k-1}$ and are independent of $\mathfrak{F}_s$ for all $s \geq k$.

4.3.2 Convergence analysis

In this section, we analyze the convergence rate of Algorithm 4.1.

Find stationary points

Let us consider the case when Algorithm 4.1 is able to find stationary points. We have the following convergence results.

Theorem 4.1. *Suppose Assumptions 4.1–4.3 and 4.5–4.7 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 4.1 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \beta,\ \eta_k = \frac{\kappa_2}{\beta_k},\ \forall k \in \mathbb{N}_0, \tag{4.5}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, and $\beta \geq c_0(\kappa_1,\kappa_2)$ with $c_0(\kappa_1,\kappa_2)$, c_1, $c_2(\kappa_1) > 0$ defined in Appendix 4.6.2. Then, for any $T \in \mathbb{N}_+$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big] = \mathcal{O}(\frac{1}{T}) + \mathcal{O}(\frac{1}{\beta^2}), \tag{4.6a}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \mathcal{O}(\frac{\beta}{\kappa_2 T}) + \mathcal{O}(\frac{\kappa_2}{n\beta}) + \mathcal{O}(\frac{1}{T}) + \mathcal{O}(\frac{1}{\beta^2}), \tag{4.6b}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1) + \mathcal{O}(\frac{T}{n\beta^2}) + \mathcal{O}(\frac{T}{\beta^3}), \tag{4.6c}$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$.

Proof. The explicit expressions of the right-hand sides of (4.6a)–(4.6c) and the proof are given in Appendix 4.6.2. It should be highlighted that the omitted constants in the first two terms in the right-hand side of (4.6b) do not depend on any parameters related to the communication network. □

Noting the right-hand side of (4.6b), the linear speedup in the number of agents can be established if we set $\beta = \kappa_2\sqrt{T}/\sqrt{n}$, as shown in the following.

Corollary 4.1 (Linear speedup)**.** *Under the same assumptions as in Theorem 4.1, let $\beta = \kappa_2\sqrt{T}/\sqrt{n}$. Then, for any $T > \max\{n(c_0(\kappa_1,\kappa_2)/\kappa_2)^2,\ n^3\}$,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big] = \mathcal{O}(\frac{n}{T}), \tag{4.7a}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \mathcal{O}(\frac{1}{\sqrt{nT}}) + \mathcal{O}(\frac{n}{T}), \tag{4.7b}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1). \tag{4.7c}$$

Remark 4.2. *It should be highlighted that the omitted constants in the first term in the right-hand side of* (4.7b) *do not depend on any parameters related to the communication network. The same linear speedup result as in* (4.7b) *was also established by the SGD algorithms proposed in [31–33, 132, 133, 135–137, 251, 255, 258]. However, in*

[31, 33, 132, 133, 251], the additional assumption that the deviation between the gradients of local cost functions is bounded was made; in [135, 136, 255], it was required that each local stochastic gradient has bounded second moment; in [137, 258], it was assumed that the mean of each local stochastic gradient is the gradient of the global cost function; and in [32], it was required that the eigenvalues of the mixing matrix are strictly greater than $-1/3$. *Moreover, the algorithms proposed in [251, 258] are restricted to a star graph; the distributed momentum SGD algorithm proposed in [33] requires each agent* i *to communicate one additional* p*-dimensional variable besides the communication of* $x_{i,k}$ *with its neighbors at each iteration; and the algorithm proposed in [258] requires an exponentially increasing batch size, which is not favorable in practice. Under the same conditions, the well-known* $\mathcal{O}(1/\sqrt{T})$ *convergence rate, which is not a speedup, was achieved by the distributed stochastic gradient tracking algorithm proposed in [138, 139]. Moreover, similar to the distributed momentum SGD algorithm proposed in [33], one potential drawback of the distributed stochastic gradient tracking algorithms is that at each iteration each agent needs to communicate one additional variable. The potential drawbacks of the results stated in Corollary 4.1 are that (i) we do not consider communication efficiency, which was considered in [133, 135–137, 251, 255, 258]; and (ii) we use time-invariant undirected graphs rather than directed graphs as considered in [132, 135]. We leave the extension to the time-varying directed graphs with communication efficiency as future research directions.*

Find global optima

Let us next consider cases when Algorithm 4.1 finds global optima. We have the following global convergence results.

Theorem 4.2. *Suppose Assumptions 4.1–4.7 hold. For any given* $T \geq (c_0(\kappa_1, \kappa_2)/\kappa_2)^{1/\theta}$, *let* $\{\boldsymbol{x}_k,\ k \in [T]\}$ *be the output generated by Algorithm 4.1 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \kappa_2(T+1)^{\theta},\ \eta_k = \frac{\kappa_2}{\beta_k},\ \forall k \leq T, \tag{4.8}$$

where $\theta \in (0,1)$, $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$. *Then,*

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{1}{T^{2\theta}}), \tag{4.9a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{1}{nT^{\theta}}) + \mathcal{O}(\frac{1}{T^{2\theta}}). \tag{4.9b}$$

Proof. The explicit expressions of the right-hand sides of (4.9a) and (4.9b), and the proof are given in Appendix 4.6.3. It should be highlighted that the omitted constants in the first term in the right-hand side of (4.9b) do not depend on any parameters related to the communication network. □

From Theorem 4.2, we see that the convergence rate is strictly greater than $\mathcal{O}(1/(nT))$. In the following we show that the linear speedup convergence rate $\mathcal{O}(1/(nT))$ can be achieved if the P–Ł constant ν is known in advance and each $f_i^* > -\infty$, where $f_i^* = \min_{x\in\mathbb{R}^p} f_i(x)$. The total number of iterations T is not needed.

Theorem 4.3 (Linear speedup)**.** *Suppose Assumptions 4.1–4.7 hold, and the P–Ł constant ν is known in advance, and each $f_i^* > -\infty$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 4.1 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \kappa_0(k+t_1),\ \eta_k = \frac{\kappa_2}{\beta_k},\ \forall k \in \mathbb{N}_0, \tag{4.10}$$

where $\kappa_0 \in [\hat{c}_0\nu\kappa_2/4, \nu\kappa_2/4)$, $\kappa_1 > c_1$, $\kappa_2 \in (0, \hat{c}_2(\kappa_1))$, and $t_1 > \hat{c}_3(\kappa_0,\kappa_1,\kappa_2)$ with $\hat{c}_0 \in (0,1)$ being a constant, $\hat{c}_2(\kappa_1)$ and $\hat{c}_3(\kappa_0,\kappa_1,\kappa_2)$ defined in Appendix 4.6.4. Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T}-\bar{x}_T\|^2\Big] = \mathcal{O}(\frac{1}{T^2}), \tag{4.11a}$$

$$\mathbf{E}[f(\bar{x}_T)-f^*] = \mathcal{O}(\frac{1}{nT}) + \mathcal{O}(\frac{1}{T^2}). \tag{4.11b}$$

Proof. The explicit expressions of the right-hand sides of (4.11a) and (4.11b), and the proof are given in Appendix 4.6.4. It should be highlighted that the omitted constants in the first term in the right-hand side of (4.11b) do not depend on any parameters related to the communication network. □

Remark 4.3. *It has been shown in [143] that $\mathcal{O}(1/T)$ convergence rate is optimal for centralized strongly convex optimization. This rate has been established by various distributed SGD algorithms when each local cost function is strongly convex, e.g., [129, 252, 260–264]. In contrast, the linear speedup convergence rate $\mathcal{O}(1/(nT))$ established in Theorem 4.3 only requires that the global cost function satisfies the P–Ł condition, but no convexity assumption is required neither on the global cost function nor on the local cost functions. The SGD algorithms in [136, 140, 141, 257–259] also achieved this linear speedup convergence rate. However, the algorithms in [257–259] are restricted to a star graph, while our algorithm is applicable to an arbitrarily connected graph. Moreover, [257, 258] assumed that the mean of each local stochastic gradient is the gradient of the global cost function, and T has to be known to choose the algorithm parameters. The algorithm in [258] furthermore required an exponentially increasing batch size, which is not favorable in practice. In [259], it was assumed that the global cost function is strongly convex. In [136, 259], it was assumed that each local stochastic gradient has bounded second moment. In [136, 140, 141], it was assumed that each local cost function is strongly convex. It is one of our future research directions to achieve linear speedup with reduced communication rounds and communication efficiency for an arbitrarily connected graph.*

Theorem 4.3 show that the convergence rate to a global optimum is sublinear when we allow the algorithm parameters to be time-varying. The following theorem establishes that the output of Algorithm 4.1 with constant algorithm parameters linearly converges to a neighborhood of a global optimum.

Theorem 4.4. *Suppose Assumptions 4.1–4.7 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 4.1 with*

$$\alpha_k = \alpha = \kappa_1\beta,\ \beta_k = \beta,\ \eta_k = \eta = \frac{\kappa_2}{\beta},\ \forall k \in \mathbb{N}_0, \tag{4.12}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, and $\beta \geq c_0(\kappa_1, \kappa_2)$ with $c_0(\kappa_1, \kappa_2)$, c_1, $c_2(\kappa_1) > 0$ defined in Appendix 4.6.2. Then,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 + f(\bar{x}_k) - f^*\Big] \leq (1-\eta\varepsilon)^k c_4 + c_5\eta\sigma^2,\ \forall k \in \mathbb{N}_+, \tag{4.13}$$

where $\varepsilon \in (0, 1/\eta)$, c_4, $c_5 > 0$ are constants defined in Appendix 4.6.5.

Proof. The proof is given in Appendix 4.6.5. □

Remark 4.4. *It should be highlighted that we do not need to know the P–Ł constant ν in advance. Similar convergence result as stated in* (4.13) *was achieved by the distributed SGD algorithms proposed in [129, 264–267] when each local cost function is strongly convex, which obviously is stronger than the P–Ł condition assumed in Theorem 4.4. In addition to the strong convexity condition, in [129], it was also assumed that each local cost function is Lipschitz-continuous. Some information related to the Lyapunov function and global parameters, which may be difficult to get, were furthermore needed to design the stepsize. Moreover, in [264–267], the strong convexity constant was needed to design the stepsize and in [266, 267], a p-dimensional auxiliary variable, which is used to track the global gradient, was communicated between agents. The potential drawbacks of the results stated in Theorem 4.4 are that (i) we use undirected graphs rather than directed graphs as considered in [267]; and (ii) we do not analyze the robustness level to gradient noise as [264] did. We leave the extension to the (time-varying) directed graphs and the robustness level analysis as future research directions.*

The unbiased assumption, i.e., Assumption 4.6, can be removed, as shown in the following.

Theorem 4.5 (Biased SGD)**.** *Suppose Assumptions 4.1–4.5 and 4.7 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 4.1 with*

$$\alpha_k = \alpha = \kappa_1\beta,\ \beta_k = \beta,\ \eta_k = \eta = \frac{\kappa_2}{\beta},\ \forall k \in \mathbb{N}_0, \tag{4.14}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, and $\beta \geq \check{c}_0(\kappa_1, \kappa_2)$ with $\check{c}_0(\kappa_1, \kappa_2) > 0$ and c_1, $c_2(\kappa_1) > 0$ defined in Appendices 4.6.6 and 4.6.2, respectively. Then,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 + f(\bar{x}_k) - f^*\Big] \leq (1-\eta\varepsilon)^k c_4 + \check{c}_5\sigma^2,\ \forall k \in \mathbb{N}_+, \tag{4.15}$$

where $\varepsilon \in (0, 1/\eta)$, $c_4 > 0$ and $\check{c}_5 > 0$ are constants defined in Appendices 4.6.5 and 4.6.6, respectively.

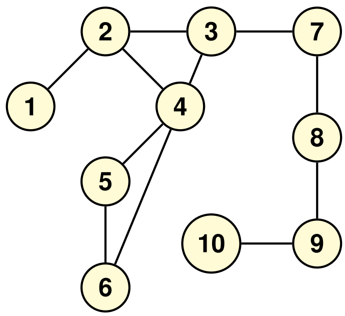


Figure 4.1: Communication network in NN experiment.

Proof. The proof is given in Appendix 4.6.6. □

Remark 4.5. *By comparing* (4.13) *with* (4.15)*, we can see that no matter the unbiased assumption holds or not, the output of Algorithm 4.1 with constant algorithm parameters linearly converges to a neighborhood of a global optimum, but the size of neighborhood is different. Specifically, in* (4.13) *the size of neighborhood is in an order of* $\tilde{\mathcal{O}}(\eta)$*, while it is* $\tilde{\mathcal{O}}(1)$ *in* (4.15).

4.4 Simulations

In this section, we evaluate the performance of the proposed distributed primal–dual SGD algorithm through numerical simulations.

4.4.1 Training of neural networks

We consider the training of neural networks (NN) for image classification tasks of the database MNIST [268]. The same NN is adopted as in [130] for each agent and the communication graph is generated randomly. The communication network is shown in Figure 4.1 and the corresponding Laplacian matrix L is given in (4.16). The corresponding mixing matrix W is constructed by metropolis weight, which is given in (4.17).

We compare our proposed distributed primal–dual SGD algorithm with time-varying and fixed parameters (DPD-SGD-T and DPD-SGD-F) with state-of-the-art algorithms: distributed momentum SGD algorithm (DM-SGD) [33], distributed SGD algorithm (D-SGD-1) [31, 129], distributed SGD algorithm (D-SGD-2) [130], D^2 [32], distributed stochastic gradient tracking algorithm (D-SGT-1) [138,267], distributed stochastic gradient tracking algorithm (D-SGT-2) [139, 266], and the baseline centralized SGD algorithm (C-SGD). We list all the parameters[1] we choose in the NN experiment for each algorithm in Table 4.2.

[1]Note: the parameter names are different in each reference.

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}. \tag{4.16}$$

$$W = \begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 3/10 & 1/4 & 1/5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 3/10 & 1/5 & 0 & 0 & 1/4 & 0 & 0 & 0 \\ 0 & 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/5 & 7/15 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/5 & 1/3 & 7/15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & 0 & 0 & 5/12 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 2/3 \end{bmatrix}. \tag{4.17}$$

Table 4.2: Parameters in each algorithm in the NN experiment.

Algorithm	η_k	α_k	β_k
DPD-SGD-T	$0.08/k^{10^{-5}}$	$4k^{10^{-5}}$	$3k^{10^{-5}}$
DPD-SGD-F	0.03	5	20
DM-SGD [33]	0.1	✗	0.8
D-SGD-1 [31, 129]	0.1	✗	✗
D-SGD-2 [130]	✗	$0.1/(10^{-5}k+1)$	$0.2/(10^{-5}k+1)^{0.3}$
D^2 [32]	0.01	✗	✗
D-SGT-1 [138, 267]	0.01	✗	✗
D-SGT-2 [139, 266]	0.01	✗	✗
C-SGD	0.1	✗	✗

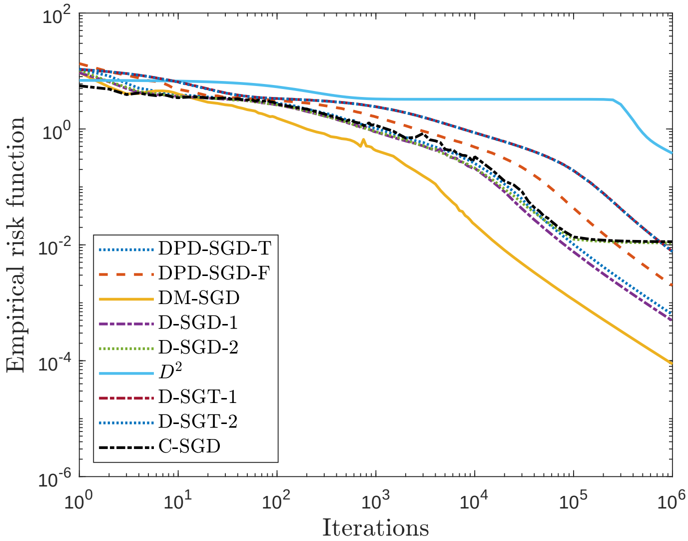


Figure 4.2: Performance of SGD optimization algorithms in the NN experiment: Evolutions of empirical risk.

We demonstrate the result in terms of the empirical risk loss [269], which is given as

$$R(\boldsymbol{z}) = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_n}\sum_{j=1}^{m_n}\sum_{k=0}^{9}(t_k \ln y_k(\boldsymbol{x},\boldsymbol{z}) + (1-t_k)\ln(1-y_k(\boldsymbol{x},\boldsymbol{z})))$$

where m_n indicates the size of data set for each agent, t_k denotes the target (ground truth) of digit k corresponding to a single image, $\boldsymbol{x}$ is a single image input, $\boldsymbol{z} = (z^{(1)}, z^{(2)})$ with $z^{(1)}$ and $z^{(2)}$ being the weights in the 2 layers separately, and $y_k \in [0,1]$ is the output which expresses the probability of digit $k = 0,\dots,9$. The mapping from input to output is given as:

$$y_k(\boldsymbol{x},\boldsymbol{z}) = \sigma\left(\sum_{j=0}^{50} z_{k,j}^{(2)}\sigma\left(\sum_{i=0}^{28\times 28} z_{j,i}^{(1)}x_i\right)\right),$$

where $\sigma(s) = \frac{1}{1+\exp(-s)}$ is the sigmoid function.

Figure 4.2 shows that the proposed distributed primal–dual SGD algorithm with time-varying parameters converges almost as fast as the distributed SGD algorithm in [31, 129] and faster than the distributed SGD algorithms in [32, 130, 138, 139, 266, 267] and the centralized SGD algorithm. Note that our algorithm converges slower than the distributed momentum SGD algorithm [33]. This is reasonable since that algorithm is an accelerated algorithm with extra requirement on the cost functions, i.e., the deviations between the gradients of local cost functions is bounded, and it requires each agent to communicate two p-dimensional variables with its neighbors at each iteration. The slope of the curves are however almost the same. The accuracy of each algorithm is given in Table 4.3. We can see that the proposed distributed primal–dual SGD algorithm with time-varying parameters has almost the same accuracy as the distributed momentum SGD algorithm [33], which is better than other algorithms.

Table 4.3: Accuracy of each algorithm in the NN experiment.

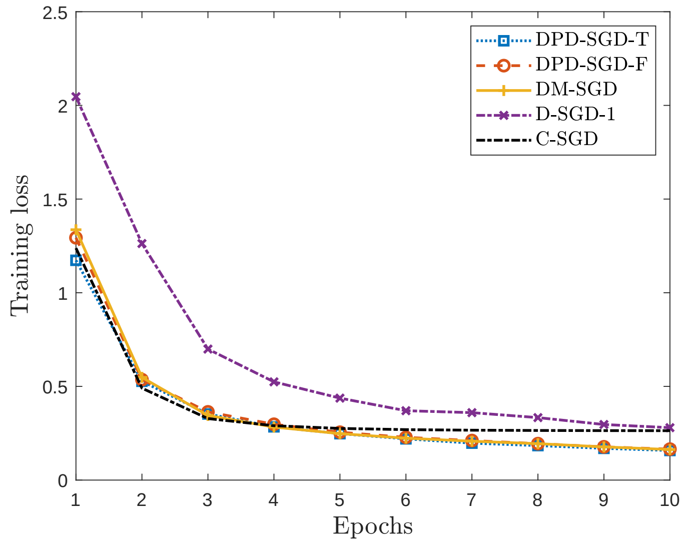
Algorithm	Accuracy
DPD-SGD-T	93.04%
DPD-SGD-F	92.76%
DM-SGD [33]	93.44%
D-SGD-1 [31, 129]	92.96%
D-SGD-2 [130]	92.88%
D^2 [32]	90.44%
D-SGT-1 [138, 267]	92.88%
D-SGT-2 [139, 266]	92.96%
C-SGD	93%

4.4.2 Training of convolutional neural networks

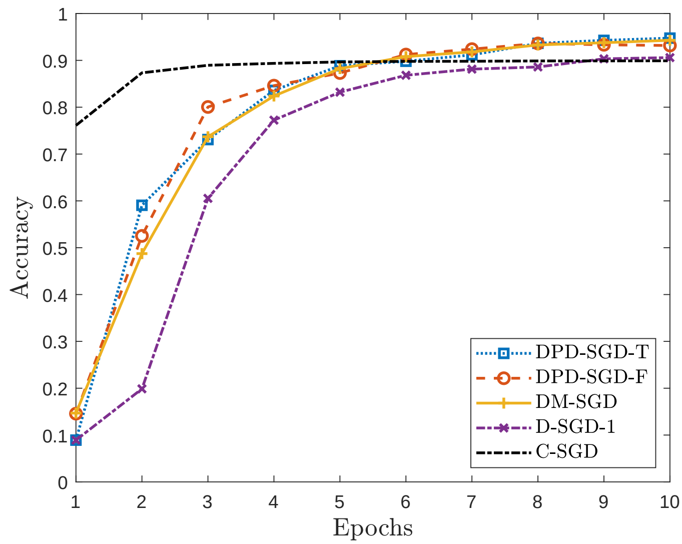
Let us consider the training of convolutional neural networks (CNNs). We build a CNN model for each agent with five 3×3 convolutional layers using ReLU as activation function, one average pooling layer with filters of size 2×2, one sigmoid layer with dimension 360, another sigmoid layer with dimension 60, one softmax layer with dimension 10. In this experiment, we use the whole MNIST data set. We use the same communication graph as in above NN experiment. Each agent is assigned 6000 data points randomly. We set the batch size as 20, which means at each iteration, 20 data points are chosen by the agent to update the gradient, which is also following a uniform distribution. For each algorithm, we do 10 epochs to train the CNN model.

We compare our algorithms DPD-SGD-T and DPD-SGD-F with the fastest one above: DM-SGD, D-SGD-1, and C-SGD. We list all the parameters we choose in the CNN experiment for each algorithm in Table 4.4.

We demonstrate the training loss and the test accuracy of each algorithm in Figures 4.3 (a) and (b), respective. Here we use Categorical Cross-Entropy loss, which is a softmax activation plus a Cross-Entropy loss. We can see that our algorithms perform almost the same as the DM-SGD algorithm and better than the D-SGD-1 and the centralized C-SGD algorithms. The accuracy of each algorithm is given in Table 4.5. We can see that the proposed distributed primal–dual SGD algorithm with time-varying parameters has the best accuracy than other algorithms.



(a) Evolutions of training loss.



(b) Evolutions of accuracy.

Figure 4.3: Performance of SGD optimization algorithms in the CNN experiment.

Table 4.4: Parameters in each algorithm in the CNN experiment.

Algorithm	η_k	α_k	β_k
DPD-SGD-T	$0.5/k^{10^{-5}}$	$0.5k^{10^{-5}}$	$0.1k^{10^{-5}}$
DPD-SGD-F	0.5	0.5	0.1
DM-SGD [33]	0.1	✗	0.8
D-SGD [31, 129]	0.1	✗	✗
C-SGD	0.1	✗	✗

Table 4.5: Accuracy of each algorithm in the CNN experiment.

Algorithm	Accuracy
DPD-SGD-T	94.75%
DPD-SGD-F	93.17%
DM-SGD [33]	94.29%
D-SGD [31, 129]	92.96%
C-SGD	89.91%

4.5 Summary

In this chapter, we studied the distributed nonconvex optimization problem with stochastic gradient information feedback. We proposed a distributed primal–dual SGD algorithm and derived its convergence rate. More specifically, the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ was established for smooth nonconvex cost functions under arbitrarily connected communication networks. The convergence rate was improved to the linear speedup convergence rate $\mathcal{O}(1/(nT))$ when the global cost function additionally satisfies the P–Ł condition. It was also shown that the output of the proposed algorithm with constant parameters linearly converges to a neighborhood of a global optimum. Interesting directions for future work include achieving linear speedup under the P–Ł condition while considering communication reduction.

4.6 Proofs

4.6.1 Notations and useful lemmas

Denote $K_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, $\boldsymbol{K} = K_n \otimes \boldsymbol{I}_p$, $\boldsymbol{H} = \frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)$, $\bar{x}_k = \frac{1}{n}(\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)\boldsymbol{x}_k$, $\bar{\boldsymbol{x}}_k = \mathbf{1}_n \otimes \bar{x}_k$, $\boldsymbol{g}_k = \nabla\tilde{f}(\boldsymbol{x}_k)$, $\bar{\boldsymbol{g}}_k = \boldsymbol{H}\boldsymbol{g}_k$, $\boldsymbol{g}_k^0 = \nabla\tilde{f}(\bar{\boldsymbol{x}}_k)$, $\bar{\boldsymbol{g}}_k^0 = \boldsymbol{H}\boldsymbol{g}_k^0 = \frac{1}{n}(\mathbf{1}_n \otimes \nabla f(\bar{x}_k))$, $\boldsymbol{g}_k^u = \mathrm{col}(g_{1,k}^u, \dots, g_{n,k}^u)$, and $\bar{\boldsymbol{g}}_k^u = \boldsymbol{H}\boldsymbol{g}_k^u$.

The distributed SGD algorithm can be rewritten as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta_k(\alpha_k \boldsymbol{L}\boldsymbol{x}_k + \beta_k \boldsymbol{v}_k + \boldsymbol{g}_k^u), \tag{4.18a}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \eta_k\beta_k \boldsymbol{L}\boldsymbol{x}_k, \ \forall \boldsymbol{x}_0 \in \mathbb{R}^{np}, \ \sum_{j=1}^{n} v_{j,0} = \boldsymbol{0}_p. \tag{4.18b}$$

Lemma 4.1. *Suppose Assumptions 4.1, 4.3, and 4.5–4.7 hold. Then the following holds for Algorithm 4.1*

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{1,k+1}] \le W_{1,k} &- \|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L}-\frac{1}{2}\eta_k\boldsymbol{K}-\frac{3}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2-\frac{1}{2}\eta_k(1+5\eta_k)L_f^2\boldsymbol{K}} \\ &- \eta_k\beta_k\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\frac{3}{2}\eta_k^2\beta_k^2\boldsymbol{K}} + 2n\sigma^2\eta_k^2,\end{aligned} \tag{4.19}$$

where $W_{1,k} = \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}$.

Proof. Noting that $\nabla\tilde{f}$ is Lipschitz-continuous with constant $L_f > 0$ since Assumption 4.3 is satisfied, we have that

$$\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k\|^2 \le L_f^2\|\bar{\boldsymbol{x}}_k - \boldsymbol{x}_k\|^2 = L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}. \tag{4.20}$$

From Assumptions 4.5–4.7, we know that

$$\mathbf{E}_{\mathfrak{F}_k}[\boldsymbol{g}_k^u] = \boldsymbol{g}_k, \tag{4.21a}$$

$$\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^u - \boldsymbol{g}_k\|^2] \le n\sigma^2. \tag{4.21b}$$

From (4.20), (4.21b), and the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^u\|^2] &= \mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k + \boldsymbol{g}_k - \boldsymbol{g}_k^u\|^2] \\ &\le 2\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k\|^2 + 2\mathbf{E}_{\mathfrak{U}_k}[\|\boldsymbol{g}_k - \boldsymbol{g}_k^u\|^2] \le 2L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + 2n\sigma^2.\end{aligned} \tag{4.22}$$

We have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{1,k+1}] &= \mathbf{E}_{\mathfrak{F}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}}\Big] = \mathbf{E}_{\mathfrak{F}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_k - \eta_k(\alpha_k\boldsymbol{L}\boldsymbol{x}_k + \beta_k\boldsymbol{v}_k + \boldsymbol{g}_k^u)\|^2_{\boldsymbol{K}}\Big] \\ &= \mathbf{E}_{\mathfrak{F}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \eta_k\alpha_k\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}} + \frac{1}{2}\eta_k^2\alpha_k^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2} + \frac{1}{2}\eta_k^2\beta_k^2\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^u\Big\|^2_{\boldsymbol{K}} \\ &\quad - \eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{I}_{np} - \eta_k\alpha_k\boldsymbol{L})\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^u\Big)\Big] \\ &= \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L}-\frac{1}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2} + \frac{1}{2}\eta_k^2\beta_k^2\mathbf{E}_{\mathfrak{F}_k}\Big[\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0 + \frac{1}{\beta_k}\boldsymbol{g}_k^u - \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\Big] \\ &\quad - \eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{I}_{np} - \eta_k\alpha_k\boldsymbol{L})\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0 + \frac{1}{\beta_k}\boldsymbol{g}_k - \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) \\ &\le W_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L}-\frac{1}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2} + \eta_k^2\beta_k^2\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \eta_k^2\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^u - \boldsymbol{g}_k^0\|^2] \\ &\quad - \eta_k\beta_k\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) + \frac{\eta_k}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{\eta_k}{2}\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2\end{aligned}$$

$$+ \frac{1}{2}\eta_k^2\alpha_k^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2} + \frac{1}{2}\eta_k^2\beta_k^2\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \frac{1}{2}\eta_k^2\alpha_k^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2} + \frac{1}{2}\eta_k^2\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2$$
$$= W_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L} - \frac{1}{2}\eta_k\boldsymbol{K} - \frac{3}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\frac{3}{2}\eta_k^2\beta_k^2\boldsymbol{K}} + \eta_k^2\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^u - \boldsymbol{g}_k^0\|^2]$$
$$+ \frac{\eta_k}{2}(1+\eta_k)\|\boldsymbol{g}_k - \boldsymbol{g}_k^0\|^2 - \eta_k\beta_k\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big), \tag{4.23}$$

where the second equality holds due to (4.18a); the third equality holds due to (2.5) in Lemma 2.5; the fourth equality holds since $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ are independent of $\mathfrak{F}_k$ and (4.21a); and the inequality holds due to the Cauchy-Schwarz inequality and $\rho(\boldsymbol{K}) = 1$.

Then, from (4.20), (4.22), and (4.23), we have (4.19). □

Lemma 4.2. *Suppose Assumptions 4.1 and 4.3 hold, and $\{\beta_k\}$ is nondecreasing. Then the following holds for Algorithm 4.1*

$$W_{2,k+1} \le W_{2,k} + (1+\omega_k)\eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{K} + \kappa_1\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) + \frac{1}{2}\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)(\omega_k + \omega_k^2)\|\boldsymbol{g}_{k+1}^0\|^2$$
$$+ \frac{1}{2}(\eta_k + \omega_k + \eta_k\omega_k)\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \|\boldsymbol{x}_k\|^2_{(1+\omega_k)\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)}$$
$$+ \frac{\eta_k}{\beta_k^2}\Big(\eta_k + \frac{1}{2}\Big)(1+\omega_k)\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)L_f^2\|\bar{\boldsymbol{g}}_k^u\|^2, \tag{4.24}$$

where $W_{2,k} = \frac{1}{2}\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}$, $\boldsymbol{Q} = R\Lambda_1^{-1}R^\top \otimes \boldsymbol{I}_p$ with matrices R and Λ_1^{-1} given in Lemma 2.5, $\omega_k = \frac{1}{\beta_k} - \frac{1}{\beta_{k+1}}$, and $\kappa_1 > 0$ is a constant.

Proof. Denote $\bar{v}_k = \frac{1}{n}(\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)\boldsymbol{v}_k$. Then, from (4.18b), we know that

$$\bar{v}_{k+1} = \bar{v}_k. \tag{4.25}$$

Then, from (4.25) and $\sum_{i=1}^n v_{i,0} = \mathbf{0}_p$, we know that

$$\bar{v}_k = \mathbf{0}_p. \tag{4.26}$$

Then, from (4.26) and (4.18a), we know that

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_k - \eta_k\bar{\boldsymbol{g}}_k^u. \tag{4.27}$$

Since $\nabla\tilde{f}$ is Lipschitz-continuous and (4.27), we have

$$\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 \le L_f^2\|\bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k\|^2 = \eta_k^2L_f^2\|\bar{\boldsymbol{g}}_k^u\|^2. \tag{4.28}$$

We know that $\omega_k \ge 0$ since $\{\beta_k\}$ is nondecreasing. We have

$$W_{2,k+1} = \frac{1}{2}\Big\|\boldsymbol{v}_{k+1} + \frac{1}{\beta_{k+1}}\boldsymbol{g}_{k+1}^0\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}} = \frac{1}{2}\Big\|\boldsymbol{v}_{k+1} + \frac{1}{\beta_k}\boldsymbol{g}_{k+1}^0 + \Big(\frac{1}{\beta_{k+1}} - \frac{1}{\beta_k}\Big)\boldsymbol{g}_{k+1}^0\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}$$

$$\leq \frac{1}{2}(1+\omega_k)\Big\|\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}^0_{k+1}\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}+\frac{1}{2}(\omega_k+\omega_k^2)\|\boldsymbol{g}^0_{k+1}\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}, \tag{4.29}$$

where the inequality holds due to the Cauchy-Schwarz inequality.

For the first term in the right-hand side of (4.29), we have

$$\begin{aligned}
&\frac{1}{2}\Big\|\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}^0_{k+1}\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}=\frac{1}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k+\eta_k\beta_k\boldsymbol{L}\boldsymbol{x}_k+\frac{1}{\beta_k}(\boldsymbol{g}^0_{k+1}-\boldsymbol{g}^0_k)\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}\\
&=\frac{1}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}+\eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{K}+\kappa_1\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big)+\|\boldsymbol{x}_k\|^2_{\frac{1}{2}\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)}\\
&\quad+\frac{1}{2\beta_k^2}\|\boldsymbol{g}^0_{k+1}-\boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}+\frac{1}{\beta_k}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k+\eta_k\beta_k\boldsymbol{L}\boldsymbol{x}_k\Big)^\top(\boldsymbol{Q}+\kappa_1\boldsymbol{K})(\boldsymbol{g}^0_{k+1}-\boldsymbol{g}^0_k)\\
&\leq W_{2,k}+\eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{K}+\kappa_1\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big)+\|\boldsymbol{x}_k\|^2_{\frac{1}{2}\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)}+\frac{1}{2\beta_k^2}\|\boldsymbol{g}^0_{k+1}-\boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}\\
&\quad+\frac{\eta_k}{2}\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}+\frac{1}{2}\eta_k^2\beta_k^2\|\boldsymbol{L}\boldsymbol{x}_k\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}+\Big(\frac{1}{2\eta_k\beta_k^2}+\frac{1}{2\beta_k^2}\Big)\|\boldsymbol{g}^0_{k+1}-\boldsymbol{g}^0_k\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}\\
&\leq W_{2,k}+\eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{K}+\kappa_1\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big)+\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\frac{1}{2}\eta_k(\boldsymbol{Q}+\kappa_1\boldsymbol{K})}\\
&\quad+\|\boldsymbol{x}_k\|^2_{\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)}+\frac{1}{\beta_k^2}\Big(1+\frac{1}{2\eta_k}\Big)\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)\|\boldsymbol{g}^0_{k+1}-\boldsymbol{g}^0_k\|^2\\
&\leq W_{2,k}+\eta_k\beta_k\boldsymbol{x}_k^\top(\boldsymbol{K}+\kappa_1\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big)+\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\frac{1}{2}\eta_k(\boldsymbol{Q}+\kappa_1\boldsymbol{K})}\\
&\quad+\|\boldsymbol{x}_k\|^2_{\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)}+\frac{\eta_k}{\beta_k^2}\Big(\eta_k+\frac{1}{2}\Big)\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)L_f^2\|\bar{\boldsymbol{g}}^u_k\|^2,
\end{aligned}\tag{4.30}$$

where the first equality holds due to (4.18b); the second equality holds due to (2.5) and (2.7) in Lemma 2.5; the first inequality holds due to the Cauchy-Schwarz inequality; the last equality holds due to (2.5) and (2.7) in Lemma 2.5; the second inequality holds due to $\rho(\boldsymbol{Q}+\kappa_1\boldsymbol{K})\leq\rho(\boldsymbol{Q})+\kappa_1\rho(\boldsymbol{K})$, (2.8), $\rho(\boldsymbol{K})=1$; and the last inequality holds due to (4.28).

For the second term in the right-hand side of (4.29), we have

$$\|\boldsymbol{g}^0_{k+1}\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}\leq\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)\|\boldsymbol{g}^0_{k+1}\|^2. \tag{4.31}$$

Also note that

$$\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}}\leq\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{K}}. \tag{4.32}$$

Then, from (4.29)–(4.32), we have (4.24). □

Lemma 4.3. *Suppose Assumptions 4.1, 4.3, and 4.5–4.7 hold, and $\{\beta_k\}$ in nondecreasing. Then the following holds for Algorithm 4.1*

$$\mathbf{E}_{\mathfrak{F}_k}[W_{3,k+1}]\leq W_{3,k}-(1+\omega_k)\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}^0_k\Big)+\frac{\eta_k}{2\beta_k^2}(1+3\eta_k)L_f^2\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}^u_k\|^2]$$

$$+\|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta_k^2(\frac{1}{2}\alpha_k^2-\alpha_k\beta_k+\beta_k^2)\boldsymbol{L}^2+\frac{1}{2}\omega_k\eta_k\alpha_k\boldsymbol{L}^2+\frac{1}{2}\eta_k(1+3\eta_k)L_f^2\boldsymbol{K}}+n\sigma^2\eta_k^2$$
$$-\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k-\frac{1}{2}-\eta_k\beta_k^2-\frac{1}{2}\omega_k\alpha_k)\boldsymbol{K}}+\frac{1}{2}\omega_k\mathbf{E}_{\mathfrak{F}_k}[2W_{1,k+1}+\|\boldsymbol{g}_{k+1}^0\|^2], \quad (4.33)$$

where $W_{3,k}=\boldsymbol{x}_k^\top\boldsymbol{K}(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0)$.

Proof. We have

$$\begin{aligned}W_{3,k+1}&=\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1}+\frac{1}{\beta_{k+1}}\boldsymbol{g}_{k+1}^0\Big)=\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}_{k+1}^0\Big)-\omega_k\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\boldsymbol{g}_{k+1}^0\\&\le\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}_{k+1}^0\Big)+\frac{1}{2}\omega_k(\|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}}+\|\boldsymbol{g}_{k+1}^0\|^2).\end{aligned} \quad (4.34)$$

For the first term in the right-hand side of (4.34), we have

$$\begin{aligned}&\mathbf{E}_{\mathfrak{F}_k}\Big[\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}_{k+1}^0\Big)\Big]\\
&=\mathbf{E}_{\mathfrak{F}_k}\Big[(\boldsymbol{x}_k-\eta_k(\alpha_k\boldsymbol{L}\boldsymbol{x}_k+\beta_k\boldsymbol{v}_k+\boldsymbol{g}_k^0+\boldsymbol{g}_k^u-\boldsymbol{g}_k^0))^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0+\eta_k\beta_k\boldsymbol{L}\boldsymbol{x}_k+\frac{\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0}{\beta_k}\Big)\Big]\\
&=\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta_k(\alpha_k+\eta_k\beta_k^2)\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)+\|\boldsymbol{x}_k\|^2_{\eta_k\beta_k(\boldsymbol{L}-\eta_k\alpha_k\boldsymbol{L}^2)}-\eta_k\beta_k\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\\
&\quad+\frac{1}{\beta_k}\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta_k\alpha_k\boldsymbol{L})\mathbf{E}_{\mathfrak{F}_k}[\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0]-\eta_k\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)^\top\boldsymbol{K}\mathbf{E}_{\mathfrak{F}_k}[\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0]\\
&\quad-\eta_k(\boldsymbol{g}_k-\boldsymbol{g}_k^0)^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0+\eta_k\beta_k\boldsymbol{L}\boldsymbol{x}_k\Big)-\frac{1}{\beta_k}\mathbf{E}_{\mathfrak{F}_k}[\eta_k(\boldsymbol{g}_k^u-\boldsymbol{g}_k^0)^\top\boldsymbol{K}(\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0)]\\
&\le\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta_k\alpha_k\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)+\frac{1}{2}\eta_k^2\beta_k^2\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}+\frac{1}{2}\eta_k^2\beta_k^2\|\boldsymbol{L}\boldsymbol{x}_k\|^2+\|\boldsymbol{x}_k\|^2_{\eta_k\beta_k(\boldsymbol{L}-\eta_k\alpha_k\boldsymbol{L}^2)}\\
&\quad-\eta_k\beta_k\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}+\frac{1}{2}\eta_k\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+\Big(\frac{1}{2\eta_k\beta_k^2}+\frac{1}{2\beta_k^2}\Big)\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2+\frac{1}{2}\eta_k^2\alpha_k^2\|\boldsymbol{L}\boldsymbol{x}_k\|^2\\
&\quad+\frac{1}{2}\eta_k^2\beta_k^2\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}+\frac{1}{2\beta_k^2}\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2]+\Big(\frac{1}{2}\eta_k+\frac{1}{2}\eta_k^2\Big)\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2\\
&\quad+\frac{1}{2}\eta_k\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}+\frac{1}{2}\eta_k^2\beta_k^2\|\boldsymbol{L}\boldsymbol{x}_k\|^2+\frac{1}{2}\eta_k^2\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^u-\boldsymbol{g}_k^0\|^2]+\frac{1}{2\beta_k^2}\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2]\\
&=\boldsymbol{x}_k^\top(\boldsymbol{K}-\eta_k\alpha_k\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)-\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k-\frac{1}{2}-\eta_k\beta_k^2)\boldsymbol{K}}+\|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta_k^2(\frac{1}{2}\alpha_k^2-\alpha_k\beta_k+\beta_k^2)\boldsymbol{L}^2}\\
&\quad+\frac{1}{2}(\eta_k+\eta_k^2)\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2+\frac{1}{2}\eta_k^2\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_k^u-\boldsymbol{g}_k^0\|^2]+\Big(\frac{1}{2\eta_k\beta_k^2}+\frac{3}{2\beta_k^2}\Big)\mathbf{E}_{\mathfrak{F}_k}[\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2]\end{aligned} \quad (4.35)$$

$$\begin{aligned}&\le\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)-(1+\omega_k)\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)+\omega_k\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)\\
&\quad-\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k-\frac{1}{2}-\eta_k\beta_k^2)\boldsymbol{K}}+\|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta_k^2(\frac{1}{2}\alpha_k^2-\alpha_k\beta_k+\beta_k^2)\boldsymbol{L}^2+\frac{1}{2}\eta_k(1+3\eta_k)L_f^2\boldsymbol{K}}\end{aligned}$$

$$+\frac{\eta_k}{2\beta_k^2}(1+3\eta_k)L_f^2\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]+n\sigma^2\eta_k^2, \tag{4.36}$$

where the first equality holds due to (4.18); the second equality holds since (2.5) in Lemma 2.5, $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ are independent of $\mathfrak{F}_k$, and (4.21a); the first inequality holds due to the Cauchy-Schwarz inequality, (2.5), $\rho(\boldsymbol{K})=1$, and the Jensen's inequality; and the last inequality holds due to (4.20), (4.22), and (4.28). For the third term in the right-hand side of (4.36), we have

$$\begin{aligned}\omega_k\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)&=\omega_k\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big)\\&\le\|\boldsymbol{x}_k\|^2_{\frac{1}{2}\omega_k\eta_k\alpha_k\boldsymbol{L}^2}+\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\frac{1}{2}\omega_k\eta_k\alpha_k\boldsymbol{K}}.\end{aligned} \tag{4.37}$$

Then, from (4.34)–(4.37), we have (4.33). □

Lemma 4.4. *Suppose Assumptions 4.2, 4.3, 4.5, and 4.6 hold. Then the following holds for Algorithm 4.1*

$$\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}]\le W_{4,k}-\frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k\|^2+\|\boldsymbol{x}_k\|^2_{\frac{\eta_k}{2}L_f^2\boldsymbol{K}}-\frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k^0\|^2+\frac{1}{2}\eta_k^2L_f\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2], \tag{4.38}$$

where $W_{4,k}=n(f(\bar{x}_k)-f^*)=\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*$.

Proof. We first note that $W_{4,k}$ is well defined due to $f^*>-\infty$ as assumed in Assumption 4.2.

From (4.20) and $\rho(\boldsymbol{H})=1$, we have that

$$\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k\|^2=\|\boldsymbol{H}(\boldsymbol{g}_k^0-\boldsymbol{g}_k)\|^2\le\|\boldsymbol{g}_k^0-\boldsymbol{g}_k\|^2\le L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}. \tag{4.39}$$

From (4.21a), we have

$$\mathbf{E}_{\mathfrak{F}_k}[\bar{\boldsymbol{g}}_k^u]=\mathbf{E}_{\mathfrak{F}_k}[\boldsymbol{H}\boldsymbol{g}_k^u]=\boldsymbol{H}\mathbf{E}_{\mathfrak{F}_k}[\boldsymbol{g}_k^u]=\bar{\boldsymbol{g}}_k. \tag{4.40}$$

We have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}]&=\mathbf{E}_{\mathfrak{F}_k}[\tilde{f}(\bar{\boldsymbol{x}}_{k+1})-\tilde{f}^*]=\mathbf{E}_{\mathfrak{F}_k}[\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*+\tilde{f}(\bar{\boldsymbol{x}}_{k+1})-\tilde{f}(\bar{\boldsymbol{x}}_k)]\\&\le\mathbf{E}_{\mathfrak{F}_k}[\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*-\eta_k(\bar{\boldsymbol{g}}_k^u)^\top\boldsymbol{g}_k^0+\frac{1}{2}\eta_k^2L_f\|\bar{\boldsymbol{g}}_k^u\|^2]\\&=\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*-\eta_k\bar{\boldsymbol{g}}_k^\top\boldsymbol{g}_k^0+\frac{1}{2}\eta_k^2L_f\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]\\&=\tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*-\eta_k\bar{\boldsymbol{g}}_k^\top\bar{\boldsymbol{g}}_k^0+\frac{1}{2}\eta_k^2L_f\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]\\&=W_{4,k}-\frac{\eta_k}{2}\bar{\boldsymbol{g}}_k^\top(\bar{\boldsymbol{g}}_k+\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k)-\frac{\eta_k}{2}(\bar{\boldsymbol{g}}_k-\bar{\boldsymbol{g}}_k^0+\bar{\boldsymbol{g}}_k^0)^\top\bar{\boldsymbol{g}}_k^0+\frac{1}{2}\eta_k^2L_f\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]\\&\le W_{4,k}-\frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k\|^2+\frac{\eta_k}{2}\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k\|^2-\frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k^0\|^2+\frac{1}{2}\eta_k^2L_f\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2],\end{aligned} \tag{4.41}$$

where the first inequality holds since that $\tilde{f}$ is smooth, (2.14) and (4.27); the third equality holds since $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ are independent of $\mathfrak{F}_k$ and (4.40); the fourth equality holds due to $\bar{\boldsymbol{g}}_k^\top \boldsymbol{g}_k^0 = \boldsymbol{g}_k^\top \boldsymbol{H}\boldsymbol{g}_k^0 = \boldsymbol{g}_k^\top \boldsymbol{H}\boldsymbol{H}\boldsymbol{g}_k^0 = \bar{\boldsymbol{g}}_k^\top \bar{\boldsymbol{g}}_k^0$; and the last inequality holds due to the Cauchy-Schwarz inequality.

Then, from (4.39) and (4.41), we have (4.38). □

4.6.2 Proof of Theorem 4.1

We denote the following notations

$$c_0(\kappa_1,\kappa_2) = \max\{4\kappa_2\varepsilon_5,\ \varepsilon_6\},\ c_1 = \frac{1}{\rho_2(L)} + 1,\ c_2(\kappa_1) = \min\Big\{\frac{\varepsilon_1}{\varepsilon_2},\ \frac{1}{5}\Big\},\ \kappa_3 = \frac{1}{\rho_2(L)} + \kappa_1 + 1,$$
$$\kappa_4 = \frac{1}{\rho_2(L)} + \kappa_1 + \frac{3}{2},\ \kappa_5 = \frac{\kappa_1+1}{2} + \frac{1}{2\rho_2(L)},\ \kappa_6 = \min\Big\{\frac{1}{2\rho(L)},\ \frac{\kappa_1-1}{2\kappa_1}\Big\},$$
$$\varepsilon_1 = (\kappa_1-1)\rho_2(L) - 1,\ \varepsilon_2 = \rho(L) + (2\kappa_1^2+1)\rho(L^2) + 1,\ \varepsilon_3 = \varepsilon_1\kappa_2 - \varepsilon_2\kappa_2^2,$$
$$\varepsilon_4 = \frac{1}{2}(\kappa_2 - 5\kappa_2^2),\ \varepsilon_5 = L_f + \frac{1}{\kappa_2\varepsilon_6}\kappa_3 L_f^2 + \frac{2}{\varepsilon_6^2}\kappa_4 L_f^2,\ \varepsilon_6 = \max\Big\{\frac{1}{2}(2+3L_f^2),\ \kappa_3\Big\}.$$

To prove Theorem 4.1, we need the following lemma.

Lemma 4.5. *Suppose Assumptions 4.1–4.3 and 4.5–4.7 hold. Suppose $\alpha_k = \alpha = \kappa_1\beta$, $\beta_k = \beta \ge c_0(\kappa_1,\kappa_2)$, and $\eta_k = \eta = \kappa_2/\beta$, where $\kappa_1 > c_1$ and $\kappa_2 \in (0, c_2(\kappa_1))$. Then, for any $k \in \mathbb{N}_0$ the following holds for Algorithm 4.1*

$$\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \le W_k - \|\boldsymbol{x}_k\|^2_{\varepsilon_3\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\varepsilon_4\boldsymbol{K}} - \frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k^0\|^2 + (\varepsilon_5+3n)\sigma^2\eta^2, \tag{4.42a}$$

$$\mathbf{E}_{\mathfrak{F}_k}[\breve{W}_{k+1}] \le \breve{W}_k - \|\boldsymbol{x}_k\|^2_{\varepsilon_3\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\varepsilon_4\boldsymbol{K}} + 2\varepsilon_5\eta^2\|\bar{\boldsymbol{g}}_k^0\|^2 + 2L_f^2\varepsilon_5\eta^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + (\varepsilon_5+3n)\sigma^2\eta^2, \tag{4.42b}$$

$$\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}] \le W_{4,k} - \frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k^0\|^2 + \|\boldsymbol{x}_k\|^2_{\frac{1}{2}\eta L_f^2\boldsymbol{K}} + L_f\sigma^2\eta^2, \tag{4.42c}$$

where $W_k = \sum_{i=1}^4 W_{i,k}$ and $\breve{W}_k = \sum_{i=1}^3 W_{i,k}$.

Proof. **(i)** Noting that $\alpha_k = \alpha = \kappa_1\beta$, $\beta_k = \beta$, $\eta_k = \eta$, and $\omega_k = \frac{1}{\beta_k} - \frac{1}{\beta_{k+1}} = 0$, from (4.19), (4.24), (4.33), and (4.38), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \le{}& W_k + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3}{2}\eta^2\beta^2\boldsymbol{K}} + 2n\sigma^2\eta^2 - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{1}{2}\eta\boldsymbol{K}-\frac{3}{2}\eta^2\alpha^2\boldsymbol{L}^2-\frac{1}{2}\eta(1+5\eta)L_f^2\boldsymbol{K}}\\
&+ \|\boldsymbol{x}_k\|^2_{\eta^2\beta^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)} + \frac{1}{2}\eta\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\\
&+ \frac{\eta}{\beta^2}\Big(\eta + \frac{1}{2}\Big)\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)L_f^2\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]\\
&+ \|\boldsymbol{x}_k\|^2_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{1}{2}\alpha^2-\alpha\beta+\beta^2)\boldsymbol{L}^2+\frac{1}{2}\eta(1+3\eta)L_f^2\boldsymbol{K}}\end{aligned}$$

$$+\frac{\eta}{2\beta^2}(1+3\eta)L_f^2\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]+n\sigma^2\eta^2-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta(\beta-\frac{1}{2}-\eta\beta^2)\boldsymbol{K}}$$
$$-\frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k\|^2+\|\boldsymbol{x}_k\|^2_{\frac{1}{2}\eta L_f^2\boldsymbol{K}}-\frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k^0\|^2+\frac{1}{2}\eta^2L_f\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]. \quad (4.43)$$

Note that

$$\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2]=\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u-\bar{\boldsymbol{g}}_k+\bar{\boldsymbol{g}}_k\|^2]\le 2\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u-\bar{\boldsymbol{g}}_k\|^2]+2\|\bar{\boldsymbol{g}}_k\|^2$$
$$=2n\mathbf{E}_{\mathfrak{F}_k}[\|\frac{1}{n}\sum_{i=1}^n(g_{i,k}^u-g_{i,k})\|^2]+2\|\bar{\boldsymbol{g}}_k\|^2=\frac{2}{n}\mathbf{E}_{\mathfrak{F}_k}[\|\sum_{i=1}^n(g_{i,k}^u-g_{i,k})\|^2]+2\|\bar{\boldsymbol{g}}_k\|^2$$
$$=\frac{2}{n}\sum_{i=1}^n\mathbf{E}_{\mathfrak{F}_k}[\|g_{i,k}^u-g_{i,k}\|^2]+2\|\bar{\boldsymbol{g}}_k\|^2\le 2\sigma^2+2\|\bar{\boldsymbol{g}}_k\|^2, \quad (4.44)$$

where the first inequality holds due to the Cauchy-Schwarz inequality; the last equality holds since $\{g_{i,k}^u,\ i\in[n]\}$ are independent of each other as assumed in Assumption 4.5, $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ are independent of $\mathfrak{F}_k$, and $\mathbf{E}_{\mathfrak{F}_k}[g_{i,k}^u]=g_{i,k}$ as assumed in Assumption 4.6; and the last inequality holds due to (4.21b).

From (4.43), (4.44), and $\alpha=\kappa_1\beta$, we have

$$\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}]\le W_k-\|\boldsymbol{x}_k\|^2_{\eta\boldsymbol{M}_1-\eta^2\boldsymbol{M}_2}-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{b_{1,k}\boldsymbol{K}}$$
$$-b_{2,k}\eta\|\bar{\boldsymbol{g}}_k\|^2-\frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k^0\|^2+b_{3,k}\sigma^2\eta^2+3n\sigma^2\eta^2, \quad (4.45)$$

where

$$\boldsymbol{M}_1=(\alpha-\beta)\boldsymbol{L}-\frac{1}{2}(2+3L_f^2)\boldsymbol{K},\ \boldsymbol{M}_2=\beta^2\boldsymbol{L}+(2\alpha^2+\beta^2)\boldsymbol{L}^2+4L_f^2\boldsymbol{K},$$
$$b_{1,k}=\frac{1}{2}(2\beta-\kappa_3)\eta-\frac{5}{2}\beta^2\eta^2,\ b_{2,k}=\frac{1}{4}-b_{3,k}\eta,\ b_{3,k}=L_f+\frac{1}{\beta^2\eta}\kappa_3L_f^2+\frac{2}{\beta^2}\kappa_4L_f^2.$$

From (2.6), $\alpha=\kappa_1\beta$, $\kappa_1>c_1>1$, $\eta=\kappa_2/\beta$, and $\beta\ge c_0(\kappa_1,\kappa_2)\ge\varepsilon_6\ge(2+3L_f^2)/2$, we have

$$\eta\boldsymbol{M}_1\ge\varepsilon_1\kappa_2\boldsymbol{K}. \quad (4.46)$$

From (2.6), $\alpha=\kappa_1\beta$, and $\beta\ge\frac{1}{2}(2+3L_f^2)>2L_f$, we have

$$\eta^2\boldsymbol{M}_2\le\varepsilon_2\kappa_2^2\boldsymbol{K}. \quad (4.47)$$

From $\beta\ge\kappa_3$, we have

$$b_{1,k}\ge\varepsilon_4. \quad (4.48)$$

From $\kappa_1>c_1=1/\rho_2(L)+1$, we have

$$\varepsilon_1>0. \quad (4.49)$$

From (4.49) and $\kappa_2 \in (0, \min\{\frac{\varepsilon_1}{\varepsilon_2}, \frac{1}{5}\})$, we have

$$\varepsilon_3 > 0, \tag{4.50a}$$
$$\varepsilon_4 > 0. \tag{4.50b}$$

From (4.50a), (4.50b), and $\beta \geq 4\kappa_2\varepsilon_5$, we have

$$b_{3,k} = L_f + \frac{1}{\beta^2\eta_k}\kappa_3 L_f^2 + \frac{2}{\beta^2}\kappa_4 L_f^2 \leq \varepsilon_5, \tag{4.51a}$$
$$b_{2,k} = \frac{1}{4} - b_{3,k}\eta \geq \frac{1}{4} - \frac{\kappa_2}{\beta}\varepsilon_5 \geq 0. \tag{4.51b}$$

From (4.45)–(4.48), (4.51a), and (4.51b), we have (4.42a).

(ii) Similar to the way to get (4.42a), we have

$$\mathbf{E}_{\mathfrak{F}_k}[\breve{W}_{k+1}] \leq \breve{W}_k - \|\boldsymbol{x}_k\|^2_{\varepsilon_3\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\varepsilon_4\boldsymbol{K}} + \varepsilon_5\eta^2\|\bar{\boldsymbol{g}}_k\|^2 + (\varepsilon_5 + 3n)\sigma^2\eta^2, \tag{4.52}$$

We have

$$\|\bar{\boldsymbol{g}}_k\|^2 = \|\bar{\boldsymbol{g}}_k - \bar{\boldsymbol{g}}_k^0 + \bar{\boldsymbol{g}}_k^0\|^2 \leq 2\|\bar{\boldsymbol{g}}_k - \bar{\boldsymbol{g}}_k^0\|^2 + 2\|\bar{\boldsymbol{g}}_k^0\|^2 \leq 2L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + 2\|\bar{\boldsymbol{g}}_k^0\|^2, \tag{4.53}$$

where the last inequality holds due to (4.39).

From (4.52) and (4.53), we have (4.42b).

(iii) From (4.38) and (4.44), we have

$$\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}] \leq W_{4,k} - \frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k\|^2 + \|\boldsymbol{x}_k\|^2_{\frac{1}{2}\eta L_f^2\boldsymbol{K}} - \frac{1}{4}\eta\|\bar{\boldsymbol{g}}_k^0\|^2 + \eta^2 L_f(\sigma^2 + \|\bar{\boldsymbol{g}}_k\|^2), \tag{4.54}$$

From $\eta = \kappa_2/\beta$ and $\beta \geq 4\kappa_2\varepsilon_5 > 4\kappa_2 L_f$, we have

$$\eta L_f < \frac{1}{4}. \tag{4.55}$$

From (4.54) and (4.55), we have (4.42c). □

Now we are ready to prove Theorem 4.1.

Denote

$$\hat{V}_k = \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + n(f(\bar{x}_k) - f^*).$$

Similar to the way to get (3.46)–(3.48),we have

$$W_k \geq \kappa_6\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\Big) + n(f(\bar{x}_k) - f^*) \tag{4.56}$$
$$\geq \kappa_6\hat{V}_k \geq 0, \tag{4.57}$$

and

$$W_k \le \kappa_5 \hat{V}_k. \tag{4.58}$$

From (4.42a) and (4.50b), we have

$$\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \le W_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{\kappa_2}{4\beta}\|\bar{\boldsymbol{g}}^0_k\|^2 + \frac{(\varepsilon_5 + 3n)\kappa_2^2\sigma^2}{\beta^2}. \tag{4.59}$$

Then, taking expectation in $\mathcal{F}_T$ and summing (4.59) over $k \in [0, T]$ yield

$$\mathbf{E}[W_{T+1}] + \sum_{k=0}^{T}\mathbf{E}\Big[\varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{\kappa_2}{4\beta}\|\bar{\boldsymbol{g}}^0_k\|^2\Big] \le W_0 + \frac{(T+1)(\varepsilon_5 + 3n)\kappa_2^2\sigma^2}{\beta^2}. \tag{4.60}$$

From (4.60), (4.57), and (4.50a), we have

$$\frac{1}{T+1}\sum_{k=0}^{T}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\Big] = \frac{1}{n(T+1)}\sum_{k=0}^{T}\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] \le \frac{W_0}{n\varepsilon_3(T+1)} + \frac{(\varepsilon_5 + 3n)\kappa_2^2\sigma^2}{n\varepsilon_3\beta^2}. \tag{4.61}$$

Noting that $W_0 = \mathcal{O}(n)$, from (4.61), we have (4.6a).

Taking expectation in $\mathcal{F}_T$ and summing (4.42c) over $k \in [0, T]$ yield

$$\frac{1}{4}n\sum_{k=0}^{T}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \frac{1}{4}\sum_{k=0}^{T}\mathbf{E}[\|\bar{\boldsymbol{g}}^0_k\|^2] \le \frac{W_{4,0}}{\eta} + \frac{1}{2}L_f^2\sum_{k=0}^{T}\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] + (T+1)L_f\sigma^2\eta. \tag{4.62}$$

From (4.62), $\eta = \kappa_2/\beta = \sqrt{n}/\sqrt{T}$, and (4.61), we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] \le \frac{4\beta}{\kappa_2 T}(f(\bar{x}_0) - f^*) + \frac{4L_f\sigma^2\kappa_2}{n\beta} + \mathcal{O}(\frac{1}{T}) + \mathcal{O}(\frac{1}{\beta^2}),$$

which gives (4.6b).

Taking expectation in $\mathcal{F}_T$ and summing (4.42c) over $k \in [0, T]$ yield

$$n(\mathbf{E}[f(\bar{x}_{T+1})] - f^*) = \mathbf{E}[W_{4,T+1}] \le W_{4,0} + \frac{1}{2}\eta L_f^2\sum_{k=0}^{T}\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] + L_f\sigma^2\eta^2(T+1). \tag{4.63}$$

Noting that $W_{4,0} = \mathcal{O}(n)$ and $\eta = \kappa_2/\beta$, from (4.60) and (4.63), we have (4.6c).

4.6.3 Proof of Theorem 4.2

In addition to the notations defined in Appendix 4.6.2, we also denote

$$\varepsilon_7 = \frac{1}{\kappa_5}\min\Big\{\varepsilon_3,\ \varepsilon_4,\ \frac{\nu}{2(T+1)^\theta}\Big\}.$$

From the conditions in Theorem 4.2, we know that all conditions needed in Lemma 4.5 are satisfied, so (4.42a)–(4.42c) still hold.

From (4.57), we have

$$\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + W_{4,k} \le \hat{V}_k \le \frac{W_k}{\kappa_6}. \tag{4.64}$$

From (2.16) and Assumption 4.4, we have that

$$\|\bar{\boldsymbol{g}}^0_k\|^2 = n\|\nabla f(\bar{x}_k)\|^2 \ge 2\nu n(f(\bar{x}_k) - f^*) = 2\nu W_{4,k}. \tag{4.65}$$

From (4.42a), (4.65), (4.57), (4.58), and (4.8), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] &\le W_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \varepsilon_4\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{K}} - \frac{1}{2}\eta\nu W_{4,k} + (\varepsilon_5 + 3n)\sigma^2\eta^2\\ &\le W_k - \frac{1}{\kappa_5}\min\Big\{\varepsilon_3,\ \varepsilon_4,\ \frac{\nu\eta}{2}\Big\}W_k + (\varepsilon_5 + 3n)\sigma^2\eta^2.\end{aligned} \tag{4.66}$$

From (4.66) and (4.8), we have

$$\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \le W_k - \varepsilon_7 W_k + \frac{(\varepsilon_5 + 3n)\sigma^2}{(T+1)^{2\theta}},\ \forall k \le T. \tag{4.67}$$

From $\kappa_1 > 1$, we have $\kappa_5 > 1$. From $0 < \kappa_2 < 1/5$, we have $\varepsilon_4 = (\kappa_2 - 5\kappa_2^2)/2 \le 1/40$. Thus,

$$0 < \varepsilon_7 \le \frac{\varepsilon_4}{\kappa_5} \le \frac{1}{40}. \tag{4.68}$$

Then, from (4.67), (4.57), and (4.68), we have

$$\begin{aligned}\mathbf{E}[W_{k+1}] &\le (1-\varepsilon_7)^{k+1}W_0 + \frac{(\varepsilon_5 + 3n)\sigma^2}{(T+1)^{2\theta}}\sum_{l=0}^{k}(1-\varepsilon_7)^l\\ &\le (1-\varepsilon_7)^{k+1}W_0 + \frac{(\varepsilon_5 + 3n)\sigma^2}{\varepsilon_7(T+1)^{2\theta}},\ \forall k \le T.\end{aligned} \tag{4.69}$$

Then, noting that $\varepsilon_7 = \mathcal{O}(1/(T+1)^{\theta})$ and $\theta \in (0,1)$, from (4.69), (2.63), and (4.64), we have

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + W_{4,k}] = \mathcal{O}(\frac{n}{T^{\theta}}),\ \forall k \le T. \tag{4.70}$$

Thus, there exists a constant $c_f > 0$ such that

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + W_{4,k}] \le nc_f,\ \forall k \le T. \tag{4.71}$$

From (4.56) and (4.58), we have

$$0 \le 2\kappa_6(W_{1,k} + W_{2,k}) \le \breve{W}_k \le 2\kappa_5(W_{1,k} + W_{2,k}). \tag{4.72}$$

From (2.15), we have

$$\|\bar{\boldsymbol{g}}_k^0\|^2 = n\|\nabla f(\bar{x}_k)\|^2 \leq 2L_f n(f(\bar{x}_k) - f^*) = 2L_f W_{4,k}. \tag{4.73}$$

Denote $\breve{z}_k = \mathbf{E}[\breve{W}_k]$. From (4.42b) and (4.71)–(4.73), we have

$$\breve{z}_{k+1} \leq (1 - a_1)\breve{z}_k + a_2\eta^2, \ \forall k \leq T, \tag{4.74}$$

where

$$a_1 = \frac{1}{\kappa_5}\min\{\varepsilon_3, \ \varepsilon_4\}, \ a_2 = n(4\varepsilon_5 L_f c_f + 2\varepsilon_5 L_f^2 c_f + (\varepsilon_5 + 3)\sigma^2). \tag{4.75}$$

From (4.68), we have

$$a_1 \leq \frac{\varepsilon_4}{\kappa_5} \leq \frac{1}{40}. \tag{4.76}$$

From (4.74) and (4.76), we have

$$\breve{z}_{k+1} \leq (1 - a_1)^{k+1}\breve{z}_0 + \frac{a_2\eta^2}{a_1}, \ \forall k \leq T,$$

which yields (4.9a).

From (4.42c) and (4.65), we have

$$\begin{aligned}
\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}] &\leq \Big(1 - \frac{\nu\eta}{2}\Big)W_{4,k} + \frac{1}{2}\eta L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + L_f\sigma^2\eta^2 \\
&\leq \Big(1 - \frac{\nu\eta}{2}\Big)^{k+1} W_{4,0} + \frac{1}{\nu}(L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + 2L_f\sigma^2\eta).
\end{aligned} \tag{4.77}$$

Noting $\eta = 1/(T+1)^\theta$, from (4.77), (2.63), and (4.9a), we have (4.9b).

4.6.4 Proof of Theorem 4.3

In addition to the notations defined in Appendix 4.6.2, we also denote

$$\begin{aligned}
&\tilde{c}_0(\kappa_1, \kappa_2) = \max\Big\{4\varepsilon_{11}, \ \varepsilon_6, \ \frac{\varepsilon_{10}}{\varepsilon_4}\Big\}, \ \hat{c}_2(\kappa_1) = \min\Big\{\frac{\varepsilon_1}{\varepsilon_2}, \ \frac{\varepsilon_8}{\varepsilon_9}, \ \frac{1}{5}\Big\}, \\
&\hat{c}_3(\kappa_0, \kappa_1, \kappa_2) = \max\Big\{\frac{\tilde{c}_0(\kappa_1, \kappa_2)}{\kappa_0}, \ \frac{8L_f\kappa_3}{\nu\kappa_2}, \ \frac{16L_f(\kappa_3 - 1)}{\nu\kappa_0\kappa_2}\Big\}, \\
&\tilde{\sigma}^2 = 2L_f f^* - 2L_f\frac{1}{n}\sum_{i=1}^n f_i^*, \ \varepsilon_8 = \kappa_1\rho_2(L) - 1, \ \varepsilon_9 = \frac{1}{2}(3\kappa_1 + 2)\kappa_1\rho(L^2) + \rho(L) + 1, \\
&\varepsilon_{10} = \kappa_2(\kappa_3 - 1) + \kappa_1\kappa_2 + \kappa_3 - 1 + 3\kappa_2^2, \ \varepsilon_{11} = \kappa_2 L_f + (2\kappa_3 - 1 + \kappa_2(10\kappa_3 - 4))L_f^2, \\
&\varepsilon_{12} = 3 + L_f + \frac{\kappa_3 L_f^2}{\kappa_0\kappa_2 t_1} + \frac{2\kappa_4 L_f^2}{\kappa_0^2 t_1^2} + \frac{2 + 2\kappa_3 L_f^2}{\kappa_0 t_1^2} + \frac{(\kappa_3 - 1)L_f^2}{\kappa_0^2\kappa_2 t_1^3} + \frac{(\kappa_3 - 1)L_f^2}{\kappa_0^2 t_1^4}\Big(\frac{2}{\kappa_0} + 2\Big),
\end{aligned}$$

$$\varepsilon_{13} = \frac{\kappa_0\kappa_3}{\kappa_2^2} + \frac{\kappa_3 - 1}{\kappa_2^2 t_1^2},\ \varepsilon_{14} = \varepsilon_{12}\sigma^2 + \varepsilon_{13}\tilde{\sigma}^2,\ \varepsilon_{15} = \frac{1}{\kappa_5}\min\Big\{\frac{\varepsilon_3\kappa_0 t_1}{\kappa_2},\ \frac{\varepsilon_4\kappa_0 t_1}{2\kappa_2},\ \frac{\nu}{8}\Big\},$$
$$\varepsilon_{16} = \frac{4L_f\sigma^2\kappa_2^2}{\kappa_0^2(\frac{\nu\kappa_2}{2\kappa_0} - 1)}.$$

To prove Theorem 4.3, we need the following lemma.

Lemma 4.6. *Suppose Assumptions 4.1–4.3 and 4.5–4.7 hold. Suppose $\alpha_k = \kappa_1\beta_k$, $\beta_k = \kappa_0(k + t_1)$, and $\eta_k = \kappa_2/\beta_k$, where $\kappa_0 \geq \tilde{c}_0(\kappa_1, \kappa_2)/t_1$, $\kappa_1 > c_1$, $\kappa_2 \in (0, \hat{c}_2(\kappa_1))$, and $t_1 \geq 1$. Then, for any $k \in \mathbb{N}_0$ the following holds for Algorithm 4.1*

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \leq{}& W_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{2}\varepsilon_4\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\\ &- \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + 2L_f b_{8,k}\eta_k^2 W_{4,k} + n\varepsilon_{14}\eta_k^2, \end{aligned} \tag{4.78a}$$

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[\breve{W}_{k+1}] \leq{}& \breve{W}_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{2}\varepsilon_4\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + n\varepsilon_{14}\eta_k^2\\ &+ 2\varepsilon_{12}L_f^2\eta_k^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + 2(2\varepsilon_{12} + \varepsilon_{13})L_f\eta_k^2 W_{4,k}, \end{aligned} \tag{4.78b}$$

$$\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}] \leq W_{4,k} - \frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 + \|\boldsymbol{x}_k\|^2_{\frac{1}{2}L_f^2\eta_k\boldsymbol{K}} + \eta_k^2 L_f\sigma^2, \tag{4.78c}$$

where $b_{8,k} = \kappa_3\frac{\omega_k}{\eta_k^2} + (\kappa_3 - 1)\frac{\omega_k^2}{\eta_k^2}$.

Proof. **(i)** We have

$$\|\boldsymbol{g}_k^0\|^2 = \sum_{i=1}^n \|\nabla f_i(\bar{x}_k)\|^2 \leq \sum_{i=1}^n 2L_f(f_i(\bar{x}_k) - f_i^*) = 2L_f n(f(\bar{x}_k) - f^*) + n\tilde{\sigma}^2, \tag{4.79}$$

where the inequality holds due to (2.15).

From the Cauchy-Schwarz inequality, (4.28), and (4.79), we have

$$\begin{aligned}\|\boldsymbol{g}_{k+1}^0\|^2 &= \|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0 + \boldsymbol{g}_k^0\|^2 \leq 2(\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 + \|\boldsymbol{g}_k^0\|^2)\\ &\leq 2(\eta_k^2 L_f^2\|\bar{\boldsymbol{g}}_k^u\|^2 + 2L_f W_{4,k} + n\tilde{\sigma}^2). \end{aligned} \tag{4.80}$$

From (4.19), (4.24), (4.33), (4.38), (4.44), (4.80), $\alpha_k = \kappa_1\beta_k$, and $\eta_k = \kappa_2/\beta_k$, we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \leq{}& W_k - \|\boldsymbol{x}_k\|^2_{\eta_k\boldsymbol{M}_{3,k} - \eta_k^2\boldsymbol{M}_{4,k} - \frac{1}{2}\kappa_1\kappa_2\omega_k + \eta_k\omega_k\boldsymbol{M}_{5,k} - \eta_k^2\omega_k\boldsymbol{M}_{6,k}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{b_{4,k}^0\boldsymbol{K}} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2\\ &- \eta_k b_{5,k}\|\bar{\boldsymbol{g}}_k\|^2 + \eta_k^2(b_{6,k}n + b_{7,k})\sigma^2 + \eta_k^2 b_{8,k}(2L_f W_{4,k} + n\tilde{\sigma}^2), \end{aligned} \tag{4.81}$$

where

$$\boldsymbol{M}_{3,k} = (\alpha_k - \beta_k)\boldsymbol{L} - \frac{1}{2}(2 + 3L_f^2)\boldsymbol{K},\ \boldsymbol{M}_{4,k} = \beta_k^2\boldsymbol{L} + (2\alpha_k^2 + \beta_k^2)\boldsymbol{L}^2 + 4L_f^2\boldsymbol{K},$$

$$\begin{aligned}
&\boldsymbol{M}_{5,k} = \alpha_k \boldsymbol{L} - \frac{1}{2}(1 + L_f^2)\boldsymbol{K},\ \boldsymbol{M}_{6,k} = \frac{3}{2}\alpha_k^2 \boldsymbol{L}^2 + \beta_k^2(\boldsymbol{L} + \kappa_1 \boldsymbol{L}^2) + \frac{5}{2}L_f^2 \boldsymbol{K},\\
&b_{4,k}^0 = \frac{1}{2}\eta_k(2\beta_k - \kappa_3) - \frac{5}{2}\eta_k^2\beta_k^2 - \frac{1}{2}\omega_k\eta_k(\kappa_3 - 1) - \frac{1}{2}\omega_k(\eta_k\alpha_k + \kappa_3 - 1 + 3\eta_k^2\beta_k^2),\\
&b_{5,k} = \frac{1}{4} - \eta_k b_{6,k},\ b_{6,k} = 3 + 2\omega_k,\\
&b_{7,k} = L_f + \frac{1}{\beta_k^2\eta_k}\kappa_3 L_f^2 + \frac{2}{\beta_k^2}\kappa_4 L_f^2 + 2\kappa_3 L_f^2\omega_k + \omega_k\Big(\frac{1}{\beta_k^2\eta_k} + \frac{2}{\beta_k^2} + 2\omega_k\Big)(\kappa_3 - 1)L_f^2.
\end{aligned}$$

From (2.6), $\alpha_k = \kappa_1\beta_k, \kappa_1 > 1, \beta_k \geq \kappa_0 t_1 \geq \tilde{c}_0(\kappa_1, \kappa_2) \geq \varepsilon_6 \geq (2+3L_f^2)/2$, and $\eta_k = \kappa_2/\beta_k$, we have

$$\eta_k \boldsymbol{M}_{3,k} \geq \varepsilon_1\kappa_2 \boldsymbol{K}. \tag{4.82}$$

From (2.6), $\alpha_k = \kappa_1\beta_k, \beta_k \geq (2 + 3L_f^2)/2 > 2L_f$, and $\eta_k = \kappa_2/\beta_k$, we have

$$\eta_k^2 \boldsymbol{M}_{4,k} \leq \varepsilon_2\kappa_2^2 \boldsymbol{K}. \tag{4.83}$$

From (2.6), $\alpha_k = \kappa_1\beta_k, \beta_k \geq (2 + 3L_f^2)/2 > (1 + L_f^2)/2$, and $\eta_k = \kappa_2/\beta_k$, we have

$$\eta_k \boldsymbol{M}_{5,k} \geq \varepsilon_8\kappa_2 \boldsymbol{K}. \tag{4.84}$$

From (2.6), $\alpha_k = \kappa_1\beta_k, \beta_k > 2L_f > \sqrt{10}L_f/2$, and $\eta_k = \kappa_2/\beta_k$, we have

$$\eta_k^2 \boldsymbol{M}_{6,k} \leq \varepsilon_9\kappa_2^2 \boldsymbol{K}. \tag{4.85}$$

From $\alpha_k = \kappa_1\beta_k, \beta_k \geq \kappa_3$, and $\eta_k = \kappa_2/\beta_k$, we have

$$b_{4,k}^0 \geq b_{4,k}, \tag{4.86}$$

where $b_{4,k} = \varepsilon_4 - \frac{1}{2}\omega_k\eta_k(\kappa_3 - 1) - \frac{1}{2}\omega_k(\kappa_1\kappa_2 + \kappa_3 - 1 + 3\kappa_2^2)$.

From $\kappa_1 > c_1 = 1/\rho_2(L) + 1$, we have

$$\varepsilon_1 > 0,\ \varepsilon_8 > 0. \tag{4.87}$$

From (4.87) and $\kappa_2 \in (0, \min\{\frac{\varepsilon_1}{\varepsilon_2},\ \frac{\varepsilon_8}{\varepsilon_9},\ \frac{1}{5}\})$, we have

$$\varepsilon_3 > 0, \tag{4.88a}$$

$$\varepsilon_8\kappa_2 - \varepsilon_9\kappa_2^2 > 0, \tag{4.88b}$$

$$\varepsilon_4 > 0. \tag{4.88c}$$

From $\beta_k = \kappa_0(k + t_1)$, we have

$$\omega_k = \frac{1}{\beta_k} - \frac{1}{\beta_{k+1}} = \frac{1}{\kappa_0}\Big(\frac{1}{k + t_1} - \frac{1}{k + t_1 + 1}\Big) = \frac{1}{\kappa_0(k + t_1)(k + t_1 + 1)} \leq \frac{\kappa_0}{\beta_k^2}. \tag{4.89}$$

From (4.88a)–(4.89), and $\kappa_0 \ge \max\{\frac{4\varepsilon_{11}}{t_1}, \frac{\varepsilon_{10}}{\varepsilon_4 t_1}\}$, we have

$$b_{4,k} \ge \varepsilon_4 - \frac{\varepsilon_{10}}{2\kappa_0 t_1^2} \ge \varepsilon_4 - \frac{\varepsilon_{10}}{2\kappa_0 t_1} \ge \frac{1}{2}\varepsilon_4 > 0, \tag{4.90a}$$

$$b_{5,k} \ge \frac{1}{4} - \frac{\varepsilon_{11}}{\kappa_0 t_1} \ge 0. \tag{4.90b}$$

From (4.89) and $\beta_k = \kappa_0(k + t_1) \ge \kappa_0 t_1$, we have

$$b_{6,k} + b_{7,k} \le \varepsilon_{12}, \tag{4.91a}$$

$$b_{8,k} \le \varepsilon_{13}. \tag{4.91b}$$

From (4.81)–(4.86), (4.88a)–(4.88c), and (4.90a)–(4.91b), we have (4.78a).
(ii) Similarly to the way to get (4.78a), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[\breve{W}_{k+1}] \le \breve{W}_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{2}\varepsilon_4\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{K}} + \varepsilon_{12}\eta_k^2\|\bar{\boldsymbol{g}}_k\|^2 \\ + 2L_f\varepsilon_{13}\eta_k^2 W_{4,k} + n\varepsilon_{14}\eta_k^2, \ \forall k \in \mathbb{N}_0,\end{aligned} \tag{4.92}$$

From (4.92), (4.53), and (4.73), we have (4.78b).
(iii) From (4.38) and (4.44), we have

$$\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}] \le W_{4,k} - \frac{\eta_k}{4}\|\bar{\boldsymbol{g}}_k\|^2 + \|\boldsymbol{x}_k\|^2_{\frac{1}{2}L_f^2\eta_k\boldsymbol{K}} - \frac{\eta_k}{4}\|\bar{\boldsymbol{g}}^0_k\|^2 + \eta_k^2 L_f(\sigma^2 + \|\bar{\boldsymbol{g}}_k\|^2). \tag{4.93}$$

From $0 < \eta_k \le \kappa_2/(\kappa_0 t_1)$ and $\kappa_0 t_1 \ge \tilde{c}_0(\kappa_1, \kappa_2) \ge 4\varepsilon_{11} > 4\kappa_2 L_f$, we have

$$\eta_k^2 L_f < \frac{1}{4}\eta_k. \tag{4.94}$$

From (4.93) and (4.94), we have (4.78c). □

Now we are ready to prove Theorem 4.3.
From $t_1 > \hat{c}_3(\kappa_0, \kappa_1, \kappa_2) \ge \tilde{c}_0(\kappa_1, \kappa_2)/\kappa_0$, we have

$$\kappa_0 > \frac{\tilde{c}_0(\kappa_1, \kappa_2)}{t_1}.$$

Thus, all conditions needed in Lemma 4.6 are satisfied, so (4.78a)–(4.78c) hold.
From (4.78a) and (4.65), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] &\le W_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{2}\varepsilon_4\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{K}} - \frac{\eta_k\nu}{2}W_{4,k} + 2L_f b_{8,k}\eta_k^2 W_{4,k} + n\varepsilon_{14}\eta_k^2 \\ &= W_k - \varepsilon_3\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{2}\varepsilon_4\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{K}} + n\varepsilon_{14}\eta_k^2 \\ &\quad - 2\Big(\frac{1}{4} - \frac{1}{\nu}L_f b_{8,k}\eta_k\Big)\nu\eta_k W_{4,k}, \ \forall k \in \mathbb{N}_0.\end{aligned} \tag{4.95}$$

From $t_1 > \hat{c}_3(\kappa_0, \kappa_1, \kappa_2) \ge 8L_f\kappa_3/(\nu\kappa_2)$, we have

$$\frac{1}{4} - \frac{L_f\kappa_3}{\nu\kappa_2 t_1} \ge \frac{1}{8}. \tag{4.96}$$

From (4.89), (4.96), and $\kappa_0 > \tilde{c}_0(\kappa_1, \kappa_2)/t_1 \ge 16L_f(\kappa_3 - 1)/(\nu\kappa_2 t_1)$, we have

$$\begin{aligned}\frac{1}{4} - \frac{1}{\nu}L_f b_{8,k}\eta_k &\ge \frac{1}{4} - \frac{L_f\kappa_0\kappa_3}{\nu\kappa_2\beta_k} - \frac{L_f\kappa_0^2(\kappa_3 - 1)}{\nu\kappa_2\beta_k^3}\\ &\ge \frac{1}{4} - \frac{L_f\kappa_3}{\nu\kappa_2 t_1} - \frac{L_f(\kappa_3 - 1)}{\nu\kappa_2\kappa_0 t_1^3} \ge \frac{1}{8} - \frac{L_f(\kappa_3 - 1)}{\nu\kappa_2\kappa_0 t_1} \ge \frac{1}{16}.\end{aligned} \tag{4.97}$$

From (4.95), (4.57), and (4.58), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] &\le W_k - \frac{\eta_k}{\kappa_5}\min\Big\{\frac{\varepsilon_3}{\eta_k}, \frac{\varepsilon_4}{2\eta_k}, \frac{\nu}{8}\Big\}W_k + n\varepsilon_{14}\eta_k^2\\ &\le W_k - \varepsilon_{15}\eta_k W_k + n\varepsilon_{14}\eta_k^2, \ \forall k \in \mathbb{N}_0.\end{aligned} \tag{4.98}$$

Denote $z_k = \mathbf{E}[W_k]$, $r_{1,k} = \varepsilon_{15}\eta_k$, and $r_{2,k} = n\varepsilon_{14}\eta_k^2$, then from (4.98), we have

$$z_{k+1} \le (1 - r_{1,k})z_k + r_{2,k}, \ \forall k \in \mathbb{N}_0. \tag{4.99}$$

From (4.10), we have

$$r_{1,k} = \eta_k\varepsilon_{15} = \frac{a_3}{k + t_1}, \tag{4.100a}$$

$$r_{2,k} = \eta_k^2\varepsilon_{14}n\sigma^2 = \frac{a_4}{(k + t_1)^2}, \tag{4.100b}$$

where

$$a_3 = \frac{\kappa_2\varepsilon_{15}}{\kappa_0}, \ a_4 = \frac{n\kappa_2^2\varepsilon_{14}}{\kappa_0^2}.$$

From (4.68), we have

$$r_{1,k} \le \frac{\varepsilon_4}{2\kappa_5} \le \frac{1}{80}. \tag{4.101}$$

Then, from (4.99)–(4.101) and (2.43), we have

$$z_k \le \phi_2(k, t_1, a_3, a_4, 2, z_0), \ \forall k \in \mathbb{N}_+, \tag{4.102}$$

where the function ϕ_1 is defined in (2.44).

From $\kappa_0 \ge \hat{c}_0\nu\kappa_2/4$, we have

$$\phi_1(k, t_1, a_3, a_4, 2, z_0) = \begin{cases} \mathcal{O}(\frac{n}{k}), & \text{if } a_3 > 1,\\ \mathcal{O}(\frac{n\ln(k-1)}{k}), & \text{if } a_3 = 1,\\ \mathcal{O}(\frac{n}{k^{a_3}}), & \text{if } a_3 < 1,\end{cases} \tag{4.103}$$

From (4.102), (4.103), and (4.64), we know that there exists a constant $c_f > 0$ such that

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + W_{4,k}] \le nc_f. \tag{4.104}$$

From (4.78b), (4.104), (4.72), and (4.10), we have

$$\breve{z}_{k+1} \le (1 - a_5)\breve{z}_k + \frac{a_6}{(t + t_1)^2}, \tag{4.105}$$

where

$$a_5 = \frac{1}{\kappa_5} \min\Big\{\varepsilon_3, \ \frac{\varepsilon_4}{2}\Big\}, \ a_6 = \frac{\kappa_2^2}{\kappa_0^2} n(2\varepsilon_{12}L_f^2 c_f + 2(2\varepsilon_{12} + \varepsilon_{13})L_f c_f + \varepsilon_{14}).$$

From (4.68), we have

$$a_5 \le \frac{\varepsilon_4}{2\kappa_5} \le \frac{1}{80}. \tag{4.106}$$

From (4.88a) and (4.88c), we know that

$$a_5 > 0 \text{ and } a_6 > 0. \tag{4.107}$$

From (4.105)–(4.107) and (2.45), we have

$$\breve{z}_k \le \phi_3(k, t_1, a_5, a_6, 2, \breve{z}_0) = \mathcal{O}(\frac{n}{k^2}), \ \forall k \in \mathbb{N}_+, \tag{4.108}$$

where the function ϕ_3 is defined in (2.46).

From (2.46), (4.72), and (4.108), we have

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] \le \frac{1}{\kappa_6}\breve{z}_k \le \frac{1}{\kappa_6}\phi_3(k, t_1, a_5, a_6, 2, \breve{z}_0) = \mathcal{O}(\frac{n}{k^2}). \tag{4.109}$$

From (4.109), we have (4.11a).

From (4.78c) and (4.65), we have

$$\mathbf{E}[W_{4,k+1}] \le \Big(1 - \frac{\nu}{2}\eta_k\Big)\mathbf{E}[W_{4,k}] + \|\boldsymbol{x}_k\|^2_{\frac{1}{2}L_f^2\eta_k\boldsymbol{K}} + L_f\sigma^2\eta_k^2. \tag{4.110}$$

From $\kappa_0 < \nu\kappa_2/4$, we have

$$\frac{\nu\kappa_2}{2\kappa_0} > 2. \tag{4.111}$$

Similar to the way to prove (2.43), from (4.109)–(4.111), we have

$$\mathbf{E}[f(\bar{x}_T) - f^*] \le \frac{\varepsilon_{16}}{nT} + \mathcal{O}(\frac{1}{T^2}), \tag{4.112}$$

where ε_{16} is determined by the last terms in (2.44) and (4.110).

From $\kappa_0 \ge \hat{c}_0\nu\kappa_2/4$, we have

$$\varepsilon_{16} = \frac{4L_f\sigma^2\kappa_2^2}{\kappa_0^2(\frac{\nu\kappa_2}{2\kappa_0} - 1)} \le \frac{4L_f\sigma^2\kappa_2^2}{\kappa_0^2(\frac{\nu\kappa_2}{2\kappa_0} - \frac{\nu\kappa_2}{4\kappa_0})} = \frac{16L_f\sigma^2\kappa_2}{\nu\kappa_0} \le \frac{64L_f\sigma^2}{\hat{c}_0\nu^2}. \tag{4.113}$$

From (4.112) and (4.113), we have (4.11b).

4.6.5 Proof of Theorem 4.4

In addition to the notations defined in Appendices 4.6.2 and 4.6.3, we also denote

$$\varepsilon = \frac{1}{\kappa_5}\min\Big\{\frac{\varepsilon_3}{\eta},\ \frac{\varepsilon_4}{\eta},\ \frac{\nu}{2}\Big\},\ c_4 = \frac{W_0}{n\kappa_6},\ c_5 = \frac{\varepsilon_5 + 3n}{n\varepsilon\kappa_6}.$$

From the conditions in Theorem 4.4, we know that (4.66) holds. Thus,

$$\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \le W_k - \eta\varepsilon W_k + (\varepsilon_5 + 3n)\sigma^2\eta^2. \tag{4.114}$$

Similar to the way to get (4.68), we have

$$0 < \eta\varepsilon < 1. \tag{4.115}$$

From (4.114) and (4.115), we have

$$\begin{aligned}\mathbf{E}[W_{k+1}] &\le (1-\eta\varepsilon)\mathbf{E}[W_k] + (\varepsilon_5 + 3n)\sigma^2\eta^2\\ &\le (1-\eta\varepsilon)^{k+1}W_0 + (\varepsilon_5 + 3n)\sigma^2\eta^2\sum_{\tau=0}^{k}(1-\eta\varepsilon)^{\tau}\\ &\le (1-\eta\varepsilon)^{k+1}W_0 + \frac{\eta(\varepsilon_5 + 3n)\sigma^2}{\varepsilon}.\end{aligned} \tag{4.116}$$

Hence, (4.116) and (4.64) give (4.13).

4.6.6 Proof of Theorem 4.5

In addition to the notations defined in Appendices 4.6.3, 4.6.2, and (4.6.5), we also denote

$$\breve{c}_0(\kappa_1,\kappa_2) = \max\{4\kappa_2\varepsilon_5,\ \breve{\varepsilon}_6\},\ \breve{\varepsilon}_6 = \max\{1 + 3L_f^2,\ \kappa_3\},\ \breve{c}_5 = \frac{3+5\eta}{\varepsilon\kappa_6}$$

Without unbiased assumption, we know that (4.30) still holds. Similar to the way to get (4.19), (4.36), and (4.38), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{1,k+1}] \le W_{1,k} &- \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2-\eta(1+3\eta)L_f^2\boldsymbol{K}} - \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big)\\ &+ \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}} + \eta(1+3\eta)n\sigma^2,\end{aligned} \tag{4.117a}$$

$$\begin{aligned}\mathbf{E}_{\mathfrak{F}_k}[W_{3,k+1}] \le W_{3,k} &- \eta\alpha\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big) + \|\boldsymbol{x}_k\|^2_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{\alpha^2}{2}-\alpha\beta+\beta^2)\boldsymbol{L}^2+\eta(1+2\eta)L_f^2\boldsymbol{K}}\\ &+ \frac{\eta}{2\beta^2}(1+3\eta)L_f^2\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2] + \eta(1+2\eta)n\sigma^2 - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\eta(\beta-\frac{1}{2}-\eta\beta^2)\boldsymbol{K}},\end{aligned} \tag{4.117b}$$

$$\mathbf{E}_{\mathfrak{F}_k}[W_{4,k+1}] \le W_{4,k} - \frac{\eta}{4}(1-2\eta L_f)\mathbf{E}_{\mathfrak{F}_k}[\|\bar{\boldsymbol{g}}_k^u\|^2] + \|\boldsymbol{x}_k\|^2_{\eta L_f^2\boldsymbol{K}} - \frac{\eta}{4}\|\bar{\boldsymbol{g}}_k^0\|^2 + n\sigma^2\eta. \tag{4.117c}$$

Then, similar to the way to get (4.42a), from (4.30) and (4.117a)–(4.117c), we have

$$\mathbf{E}_{\mathfrak{F}_k}[W_{k+1}] \le W_k - \|\boldsymbol{x}_k\|^2_{\varepsilon_3 \boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}^0_k\Big\|^2_{\varepsilon_4 \boldsymbol{K}} - \frac{1}{4}\eta\|\bar{\boldsymbol{g}}^0_k\|^2 + \eta(3+5\eta)n\sigma^2. \tag{4.118}$$

Then, similar to the way to get (4.13), from (4.118) and (4.65), we have (4.15).

Chapter 5

Distributed zeroth-order optimization algorithms

In this chapter, we consider the distributed nonconvex optimization problem with zeroth-order (ZO) oracle feedback, i.e., each agent is only able to sample ZO oracles (the values of its local cost function). We first consider the situation that deterministic zeroth-order (DZO) oracle feedback is available. We propose a distributed primal–dual DZO algorithm. In this algorithm, at each iteration each agent samples its local DZO oracles at $p + 1$ different points with an adaptive smoothing parameter, where p is the dimension of the decision variable. We show that the proposed algorithm converges to a stationary point with an $\mathcal{O}(1/T)$ convergence rate for smooth nonconvex cost functions, where T is the total number of iterations, and to a global optimum with a linear convergence rate when the global cost function satisfies the P–Ł condition in addition. In other words, our proposed distributed DZO algorithm has the same convergence properties as its FO counterpart in Chapter 3 under the same conditions. We then consider the situation that stochastic zeroth-order (SZO) oracle feedback is available. We propose two distributed SZO algorithms: distributed primal–dual and dual SZO algorithms. In both algorithms, at each iteration each agent samples its local SZO oracles at two different points with an adaptive smoothing parameter. We show that the proposed algorithms converge to a stationary point with the linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ for smooth nonconvex cost functions, and to a global optimum with the linear speedup convergence rate $\mathcal{O}(p/(nT))$ when the global cost function satisfies the P–Ł condition in addition, where n is the number of agents. To the best of our knowledge, this is the first linear speedup result for distributed SZO algorithms. We also show that the proposed algorithms converge linearly when considering centralized optimization problems with DZO oracle feedback under the P–Ł condition. We finally demonstrate through numerical simulations the efficiency of our algorithms in comparison with the baseline and recently proposed centralized and distributed ZO algorithms.

This chapter is organized as follows. Section 5.1 gives the background. Section 5.2 introduces problem formulation and assumptions. Section 5.3–5.5 provide the distributed primal–dual DZO algorithm, the distributed primal–dual SZO algorithm, and the distributed primal SZO algorithm, and analyze their convergence properties. Simulations are given in Section 5.6. Concluding remarks are offered in Section 5.7. To improve the readability, all the proofs can be found in Section 5.8.

5.1 Introduction

Many existing optimization algorithms and the algorithms proposed in the previous two chapters use at least (stochastic) gradient information of the cost functions, and sometimes even the second or higher order information. However, in many applications, the (stochastic) gradients are unavailable [144–146]. For example, many cost functions of big data problems that deal with complex data-generating processes cannot be explicitly defined [46]. Thus, gradient-free (derivative-free) optimization algorithms are needed. The study of gradient-free optimization problems has a long history, which can be traced back at least to the 1960's [270–272], and has recently gained renewed attention in machine learning community. Generally speaking, traditional gradient-free optimization methods can be classified into direct-search and model-based methods. For example, stochastic direct-search and model-based trust-region algorithms have been proposed in [273–276] and [277–279], respectively. In recent years, the more popular gradient-free optimization methods are ZO optimization methods, which are gradient-free counterparts of first-order optimization methods and thus are easy to implement. In ZO optimization methods, the full or stochastic gradients are approximated by directional derivatives which can be calculated through the sampled function values. The commonly used method to calculate directional derivatives is using the function difference at two different points [213, 214, 280].

Via modification of gradient-based optimization algorithms, various ZO optimization methods have been proposed, e.g., ZO (stochastic) gradient descent algorithms [142, 213, 281–290], ZO stochastic coordinate descent algorithms [248], ZO (stochastic) variance reduction algorithms [215, 286, 287, 289–302], ZO (stochastic) proximal algorithms [291, 298, 303, 304], ZO Frank–Wolfe algorithms [289, 300, 302, 305], ZO mirror descent algorithms [214, 296, 306], ZO adaptive momentum methods [304, 307], ZO methods of multipliers [215, 292, 308, 309], ZO stochastic path-integrated differential estimator [294, 299, 309]. Convergence properties of these algorithms have been analyzed in detail. For instance, the typical convergence results for two-point sampling based centralized DZO algorithms is that first-order stationary points can be found at an $\mathcal{O}(p/T)$ convergence rate [213, 285], while for two-point sampling based centralized SZO algorithms the convergence rate is reduced to $\mathcal{O}(\sqrt{p/T})$ [142, 248], where p and T are the dimension of the decision variable and the total number of iterations, respectively.

Aforementioned ZO optimization algorithms are all centralized and thus are not suitable to solve distributed optimization problems. Some recent works have started to modify distributed gradient-based optimization algorithms to ZO, e.g., distributed ZO gradient descent algorithms [147–151], distributed ZO push-sum algorithm [152], distributed ZO mirror descent algorithm [153], distributed ZO gradient tracking algorithm [151], distributed ZO primal–dual algorithms [155], distributed ZO sliding algorithm [154]. Convergence properties of these algorithms have been analyzed in detail. For example, in [151] it was established that the output of the $2p$-point sampling based distributed DZO algorithm achieves an $\mathcal{O}(1/T)$ convergence rate for smooth cost functions and a linear convergence rate when the global cost function satisfies the P–Ł condition in addition; and in [155] it was established that the output of the two-point sampling based distributed SZO algorithm achieves an $\mathcal{O}(p^2n/T)$ convergence rate for smooth nonconvex cost functions, where n is the number of agents. However, the algorithm in [151] requires each agent to communicate three p-dimensional variables with its neighbors, which results in a heavy communication burden when p is large, and uses the P–Ł constant, which is normally difficult to determine, to design algorithm parameters. The algorithm in [155] requires each agent to sample $\mathcal{O}(T)$ times per iteration, which is not favorable in practice.

Noting above, four core theoretical questions with important practical relevance arise when considering distributed ZO optimization problems.

(Q5.1) Are there any distributed DZO algorithms that have the same convergence properties as the $2p$-point sampling based distributed DZO algorithm in [151] under the same conditions, but require less communication and do not use the P–Ł constant?

(Q5.2) Can distributed SZO algorithms have similar convergence properties as centralized such algorithms? For instance, can two-point sampling based distributed SZO algorithms also have an $\mathcal{O}(\sqrt{p/T})$ convergence rate as their centralized counterparts did in [142, 248]?

(Q5.3) As shown in Chapter 4, distributed SGD algorithms can achieve linear speedup in the number of agents n, compared with centralized SGD algorithms. Can distributed SZO algorithms also achieve linear speedup? In particular, can two-point sampling based distributed SZO algorithms achieve the linear speedup convergence rate $\mathcal{O}(\sqrt{p/nT})$?

(Q5.4) Centralized and distributed DZO algorithms can achieve faster convergence rates under more stringent conditions such as the strong convexity and P–Ł conditions, as shown in [213, 283, 285, 299, 301, 303] and [151], respectively. Can SZO algorithms also achieve faster convergence rates under such conditions?

This chapter provides positive answers to the above four questions. We first consider the situation that DZO oracle feedback is available and have the following contributions.

(C5.1) We propose a distributed DZO primal–dual algorithm (Algorithm 5.1), by integrating the distributed FO primal–dual algorithm (3.7) with the deterministic gradient estimator (2.33). In this algorithm, at each iteration each agent samples its local DZO oracles at $p + 1$ different points.

(C5.2) We show in Theorems 5.1 and 5.2 that the proposed algorithm achieves an $\mathcal{O}(1/T)$ convergence rate for smooth nonconvex cost functions and a linear convergence rate when the global cost function satisfies the P–Ł condition in addition. In other words, our proposed distributed DZO algorithm has the same convergence properties as its FO counterpart and the $2p$-point sampling based distributed DZO algorithm in [151] under the same conditions. Two potential advantages of our distributed DZO algorithm are that it only requires each agent to communicate one p-dimensional variable with its neighbors at each iteration and does not use the P–Ł constant, thus (Q5.1) is answered.

We then consider the situation that SZO oracle feedback is available and have the following contributions.

(C5.3) We propose two distributed SZO algorithms (Algorithms 5.2 and 5.3). In both algorithms, at each iteration each agent samples its local SZO oracles at two different points with an adaptive smoothing parameter. This is different from many existing ZO algorithms and is favorable in practice.

(C5.4) We show in Theorems 5.4 and 5.10 that our SZO algorithms find a stationary point with the linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ for smooth nonconvex cost functions, and thus are faster than the centralized ZO algorithms in [142, 248, 286–289, 307] and the distributed DZO primal algorithm in [151]. To the best of our knowledge, this is the first linear speedup result for distributed SZO algorithms, thus (Q5.2) and (Q5.3) are answered.

(C5.5) We show in Theorems 5.6, 5.7, 5.12, and 5.13 that our SZO algorithms find a global optimum with an $\mathcal{O}(p/(nT))$ convergence rate when the global cost function satisfies the P–Ł condition, which is faster than the centralized ZO algorithms in [281, 282] and the distributed ZO primal algorithms in [148, 151], even though [148, 281, 282] assumed strongly convex cost functions and only considered additive sampling noise, and [151] considered the DZO oracle feedback setting. To the best of our knowledge, this is the first analysis for the performance of SZO algorithms under the P–Ł condition or the strong convexity assumption, thus (Q5.4) is answered.

(C5.6) When considering centralized optimization problems with DZO oracle feedback, we show in Theorems 5.8 and 5.14 that above two SZO algorithms linearly find a global optimum under the P–Ł condition. Compared with [213, 283, 285, 299, 301, 303] which also achieved linear convergence, we use weaker assumptions on the cost function and/or less samplings per iteration.

The detailed comparison between this chapter and other ZO optimization algorithms is summarized in Table 5.1. In this table, NoSPPI denotes the number of sampled points per iteration, and the sampling complexity is the total number of function samplings needed to attain an ϵ-accuracy, i.e., $\mathbf{E}[\|\nabla f(x_T)\|^2] \le \epsilon$ for nonconvex problems or $\mathbf{E}[f(x_T) - f^*] \le \epsilon$ for (strongly) convex problems or problems satisfying the P–Ł condition, where $\epsilon > 0$ is a constant.

Table 5.1: Comparison of Chapter 5 to some related ZO optimization algorithms.

Reference	Problem settings	NoSPPI	Convergence rate	Sampling complexity
[213]	DZO, centralized, nonconvex, smooth	Two	$O(p/T)$	$O(p/\epsilon)$
	Strongly convex in addition		Linear	$O(p\log(1/\epsilon))$
[283]	DZO, centralized, strongly convex, smooth, Lipschitz Hessian	p	Linear	$O(p\log(1/\epsilon))$
[285]	DZO, centralized, nonconvex, smooth	Two	$O(p/T)$	$O(p/\epsilon)$
	P–Ł condition in addition		Linear	$O(p\log(1/\epsilon))$
[303]	DZO, centralized, restricted strongly convex, smooth, s-sparse gradient	$4s\log(p/s)$	Linear	$O(s\log(p/s)\log(1/\epsilon))$
[281]	DZO, centralized, quadratic, additive sampling noise	One	$O(p^2/T)$	$O(p^2/\epsilon)$
[282]	DZO, centralized, strongly convex, smooth, additive sampling noise	Two	$O(p/\sqrt{T})$	$O(p^2/\epsilon^2)$
[288]	DZO, centralized, nonconvex, Lipschitz, smooth	One	$O(p^2/T^{2/3})$	$O(p^3/\epsilon^{3/2})$
	SZO, centralized, nonconvex, Lipschitz, smooth		$O(p^{4/3}/T^{1/3})$	$O(p^4/\epsilon^3)$
[142, 248]	SZO, centralized, nonconvex, smooth	Two	$O(\sqrt{p/T})$	$O(p/\epsilon^2)$
[289]	SZO, centralized, nonconvex, smooth, s-sparse gradient	Two	$O(s\log(p)/\sqrt{T})$	$O((s\log(p))^2/\epsilon^2)$
[291]	SZO, centralized, constrained, nonconvex, Lipschitz, smooth	$O(pT)$	$O(1/T)$	$O(p/\epsilon^2)$
[307]	SZO, centralized, constrained, nonconvex, Lipschitz, smooth	Two	$O(p/\sqrt{T})$	$O(p^2/\epsilon^2)$
[286]	DZO, finite-sum, nonconvex, constrained, Lipschitz, smooth	$O(\sqrt{T})$	$O(p/\sqrt{T})$	$O(p^3/\epsilon^3)$
[287]	DZO, finite-sum, nonconvex, Lipschitz, smooth	$O(pT)$	$O(\sqrt{p/T})$	$O(p^3/\epsilon^4)$
[293]	DZO, finite-sum, nonconvex, smooth, the original and mixture gradients are proportional	$2b$	$O(pn^\theta/(bT))$	$O(pn^\theta/\epsilon)$, $\forall\theta\in(0,1)$
[294]	DZO, finite-sum, nonconvex, smooth	$O(pn^{1/2})$	$O(1/T)$	$O(pn^{1/2}/\epsilon)$
[295]	DZO, finite-sum, nonconvex, smooth, similar f_i	$2n$	$O(p/T)$	$O(pn/\epsilon)$
[298]	DZO, finite-sum, nonconvex, Lipschitz, smooth	$O(pn^{2/3})$	$O(p/T)$	$O(p^2n^{2/3}/\epsilon)$
[299]	DZO, finite-sum, nonconvex, smooth, similar f_i	$O(pn^{1/2})$	$O(1/T)$	$O(pn^{1/2}/\epsilon)$
	P–Ł condition in addition		Linear	$O(pn^{1/2}\log(1/\epsilon))$
[301]	DZO, finite-sum, strongly convex, smooth	Four	Linear	$O(pn\log(p/\epsilon))$
[292]	SZO, finite-sum, nonconvex, constrained, Lipschitz, smooth	$O(nT)$	$O(p/T)$	$O(p^2n/\epsilon^2)$
[297]	SZO, finite-sum, nonconvex, smooth	Four	$O(p^{1/3}n^{2/3}/T)$	$O(p^{1/3}n^{2/3}/\epsilon)$
[147]	DZO, distributed, convex, constrained, Lipschitz	$2n$	Asymptotic	—-
[152]	DZO, distributed, convex, Lipschitz	$2n$	$O(p^3n^2/\sqrt{T})$	$O(p^6n^5/\epsilon^2)$
[153]	DZO, distributed, convex, compact constrained, Lipschitz	$2n$	$O(p\sqrt{n/T})$	$O(p^2n^2/\epsilon^2)$
	DZO, distributed, strongly convex, constrained, Lipschitz		$O(p^2n^2/T)$	$O(p^2n^3/\epsilon)$
[148]	DZO, distributed, strongly convex, smooth, additive sampling noise	$2n$	$\mathbb{O}(pn^2/\sqrt{T})$	$O(p^2n^5/\epsilon^2)$
[149]	DZO, distributed, convex, compact constrained, Lipschitz, additive sampling noise	$2n$	$O(1/\sqrt{T})$	$O(n/\epsilon^2)$
[154]	SZO, distributed, convex, compact constrained, Lipschitz	$O(pnT)$	$O(1/T)$	$O(pn/\epsilon^2)$
[151]	DZO, distributed, nonconvex, Lipschitz, smooth	$2n$	$O(\sqrt{p/T})$	$O(pn/\epsilon^2)$
	DZO, distributed, nonconvex, smooth, P–Ł condition		$O(p/T)$	$O(pn/\epsilon)$
	DZO, distributed, nonconvex, smooth	$2pn$	$O(1/T)$	$O(pn/\epsilon)$
	P–Ł condition in addition		Linear	$O(pn\log(1/\epsilon))$
[155]	SZO, distributed, nonconvex, Lipschitz, smooth	$O(nT)$	$O(p^2n/T)$	$O(p^4n^3/\epsilon^2)$
This chapter	DZO, distributed, nonconvex, smooth	$(p+1)n$	$O(1/T)$	$O(pn/\epsilon)$
	P–Ł condition in addition (without using the P–Ł constant)		Linear	$O(pn\log(1/\epsilon))$
	SZO, distributed, nonconvex, smooth, similar f_i	$2n$	$O(\sqrt{p/(nT)})$	$O(p/\epsilon^2)$
	SZO, distributed, nonconvex, smooth, P–Ł condition		$O(p/(nT))$	$O(p/\epsilon)$
	DZO, centralized, nonconvex, smooth,	Two	$O(p/T)$	$O(p/\epsilon)$
	P–Ł condition in addition (without using the P–Ł constant)		Linear	$O(p\log(1/\epsilon))$

5.2 Distributed nonconvex optimization with ZO oracle feedback

Consider a network of n agents, each of which has a local cost function $f_i : \mathbb{R}^p \to \mathbb{R}$. All agents collaborate to solve the optimization problem

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{5.1}$$

This is the same as the distributed nonconvex optimization problem (3.1). However, in this chapter, we consider the case where each agent is only able to collect ZO oracles, i.e., the values of its local cost function, rather than FO oracles, i.e., the true or stochastic gradients of its local cost function.

Both DZO and SZO oracle feedback settings are considered. Under the DZO oracle feedback setting, each agent can sample the true values of its local cost function. In this setting, based on the definitions introduced in Chapter 2, the following assumptions are made.

Assumption 5.1. *The communication among agents is described by a weighted undirected connected graph $\mathcal{G}$.*

Assumption 5.2. *The optimal set $\mathbb{X}^*$ is nonempty and $f^* > -\infty$, where $\mathbb{X}^*$ and f^* are the optimal set and the minimum function value of the optimization problem* (5.1)*, respectively.*

Assumption 5.3. *Each local cost function $f_i(x)$ is smooth with constant $L_f > 0$.*

Assumption 5.4. *The global cost function $f(x)$ satisfies the P–Ł condition with constant $\nu > 0$.*

Under the SZO oracle feedback setting, each agent can sample stochastic approximations of the true local cost function values. Let $F_i(x, \xi_i)$ be a stochastic approximation of the true local cost function value $f_i(x)$ that can be sampled by agent i, where ξ_i is a random variable. In addition to Assumptions 5.1–5.4, we also make the following assumptions.

Assumption 5.5. *The SZO oracle $F_i(x, \xi_i)$ is unbiased, i.e., $\mathbf{E}_{\xi_i}[F_i(x, \xi_i)] = f_i(x)$, $\forall i \in [n]$, $\forall x \in \mathbb{R}^p$.*

Assumption 5.6. *For almost all ξ_i, the SZO oracle $F_i(\cdot, \xi_i)$ is smooth with constant $L_f > 0$.*

Assumption 5.7. *The stochastic gradient $\nabla_x F_i(x, \xi_i)$ has bounded variance, i.e., there exists $\sigma_1 \in \mathbb{R}$ such that $\mathbf{E}_{\xi_i}[\|\nabla_x F_i(x, \xi_i) - \nabla f_i(x)\|^2] \le \sigma_1^2$, $\forall i \in [n]$, $\forall x \in \mathbb{R}^p$.*

Assumption 5.8. *Local cost functions are similar, i.e., there exists $\sigma_2 \in \mathbb{R}$ such that $\|\nabla f_i(x) - \nabla f(x)\|^2 \le \sigma_2^2$, $\forall i \in [n]$, $\forall x \in \mathbb{R}^p$.*

Remark 5.1. *It should be highlighted that no convexity assumptions are made. Assumptions 5.5–5.7 are standard when considering the SZO oracle feedback setting, e.g., [142, 155, 215, 248, 289, 291, 292, 296, 297, 304, 305]. Assumption 5.8 is slightly weaker than the assumption that each ∇f_i is bounded, which is normally used in the literature* *studying finite-sum ZO optimization, e.g., [147, 151, 152, 154, 155, 214, 215, 286, 287, 292, 293, 298, 305, 308, 309]. Bounded gradient is not the case for many unconstrained optimization problems, e.g., quadratic optimization problems. Assumption 5.8 is not needed when Assumption 5.4 holds.*

Our goal in this chapter is to answer (Q5.1)–(Q5.4), i.e., solve the following problem.

Problem 5.1. *Propose distributed DZO and SZO algorithms for the nonconvex optimization problem* (5.1) *such that stationary points or global optima can be found.*

5.3 Distributed primal–dual DZO algorithm

In this section, we consider the situation that DZO oracle feedback is available. We propose a distributed primal–dual DZO algorithm based on the deterministic gradient estimator introduced in Section 2.9 and analyze its convergence rate.

5.3.1 Algorithm description

Inspired by the deterministic gradient estimator (2.33), based on the distributed primal–dual FO algorithm (3.7), we propose the distributed primal–dual DZO algorithm

$$x_{i,k+1} = x_{i,k} - \eta\Big(\alpha \sum_{j=1}^{n} L_{ij} x_{j,k} + \beta v_{i,k} + \hat{\nabla}_p f_i(x_{i,k}, \delta_{i,k})\Big), \tag{5.2a}$$

$$v_{i,k+1} = v_{i,k} + \eta\beta \sum_{j=1}^{n} L_{ij} x_{j,k}, \forall x_{i,0} \in \mathbb{R}^p, \ \sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p, \tag{5.2b}$$

where $\hat{\nabla}_p f_i(x_{i,k}, \delta_{i,k})$ is the deterministic estimator of $\nabla f_i(x_{i,k})$ as defined in (2.33) and $\delta_{i,k}$ is the exploration parameter. Recall that

$$\hat{\nabla}_p f_i(x_{i,k}, \delta_{i,k}) = \frac{1}{\delta_{i,k}} \sum_{l=1}^{p} (f_i(x + \delta_{i,k} \boldsymbol{e}_l) - f_i(x)) \boldsymbol{e}_l.$$

Note that the gradient estimator $\hat{\nabla}_p f_i(x_{i,k}, \delta_{i,k})$ can be calculated by querying the true function values of f_i at $p + 1$ points.

We present the distributed primal–dual DZO algorithm (5.2) in pseudo-code as Algorithm 5.1.

Remark 5.2. *In [243], the authors proposed the distributed DZO gradient tracking algorithm. However, in that algorithm, at each iteration each agent i needs to communicate two additional p-dimensional variables besides the communication of $x_{i,k}$ with its neighbors, which results in a heavy burden on the communication channel when p is large. Moreover, the deterministic gradient estimator used in [243] requires that at each iteration each agent queries its local cost function values at* $2p$ *points compared with* $p + 1$ *points used in our algorithm.*

Algorithm 5.1 Distributed Primal–Dual DZO Algorithm

Input: parameters $\alpha > 0, \beta > 0, \eta > 0$, and $\{\delta_{i,k} > 0\}$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p,\ \forall i \in [n]$.
for $k = 0, 1, \dots$ **do**
  for $i = 1, \dots, n$ in parallel **do**
    Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
    Sample $f_i(x_{i,k})$ and $\{f_i(x_{i,k} + \delta_{i,k}\boldsymbol{e}_l)\}_{l=1}^p$;
    Update $x_{i,k+1}$ by (5.2a);
    Update $v_{i,k+1}$ by (5.2b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

5.3.2 Convergence analysis

In this section, we analyze convergence rate of Algorithm 5.1.

Find stationary points

Let us consider the case when Algorithm 5.1 is able to find stationary points. We have the following convergence results.

Theorem 5.1. *Suppose that Assumptions 5.1–5.3 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.1 with $\alpha \in (\beta + \kappa_1, \kappa_2\beta]$, $\beta > c_\beta$, $\eta \in (0, c_\eta)$, and $\delta_{i,k} > 0$ such that*

$$\sum_{k=0}^{+\infty} \delta_{i,k}^2 < +\infty, \tag{5.3}$$

where κ_1, κ_2, c_β, and c_η are constants given in Section 5.8.1. Then, for any $T \in \mathbb{N}_+$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 = \mathcal{O}(\frac{1}{T}), \tag{5.4}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\nabla f(\bar{x}_k)\|^2 = \mathcal{O}(\frac{1}{T}), \tag{5.5}$$

$$f(\bar{x}_T) - f^* = \mathcal{O}(1), \tag{5.6}$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}$.

Proof. The explicit expressions of the right-hand sides of (5.4)–(5.6) and the proof are given in Section 5.8.1. □

Find global optima

Let us next consider the case when Algorithm 5.1 finds global optima. We have the following convergence results.

Theorem 5.2. *Suppose that Assumptions 5.1–5.4 hold. Let* $\{\boldsymbol{x}_k\}$ *be the sequence generated by Algorithm 5.1 with the same* α, β, *and* η *given in Theorem 5.1, and* $\delta_{i,k} \in (0, \kappa_\delta^{k/2})$, *where* $\kappa_\delta \in (0,1)$ *is a constant, then*

$$\frac{1}{n}\sum_{i=1}^{n} \|x_{i,k} - \bar{x}_k\|^2 + f(\bar{x}_k) - f^* \leq \zeta_0 \zeta^k, \ \forall k \in \mathbb{N}_0, \tag{5.7}$$

where $\zeta_0 > 0$ *and* $\zeta \in (0,1)$ *given in Section 5.8.2.*

Proof. The proof is given in Section 5.8.2. □

Remark 5.3. *By comparing Theorems 3.1 and 3.2 with Theorems 5.1 and 5.2, respectively, we see that the proposed distributed FO and ZO algorithms have the same convergence properties under the same assumptions. Similar convergence results as stated in Theorems 5.1 and 5.2 were also achieved by the distributed DZO gradient tracking algorithm proposed in [243] under the same assumptions. Compared with [243], in addition to the advantages discussed in Remark 5.2, one more advantage of Theorem 5.2 is that the P–Ł constant* ν *is not needed to be known in advance.*

5.4 Distributed primal–dual SZO algorithm

In this and the next sections, we consider the situation that SZO oracle feedback is available. In this section, we propose a distributed primal–dual SZO algorithm based on the two-point sampling random gradient estimator introduced in Section 2.8 and analyze its convergence rate.

5.4.1 Algorithm description

Inspired by the two-point sampling random gradient estimator (2.26), based on the distributed primal–dual FO algorithm (3.7), we propose the distributed primal–dual SZO algorithm

$$x_{i,k+1} = x_{i,k} - \eta_k\Big(\alpha_k \sum_{j=1}^{n} L_{ij} x_{j,k} + \beta_k v_{i,k} + g^e_{i,k}\Big), \tag{5.8a}$$

$$v_{i,k+1} = v_{i,k} + \eta_k \beta_k \sum_{j=1}^{n} L_{ij} x_{j,k}, \ \forall x_{i,0} \in \mathbb{R}^p, \ \sum_{j=1}^{n} v_{j,0} = \mathbf{0}_p, \ \forall i \in [n], \tag{5.8b}$$

where

$$g^e_{i,k} = \frac{p}{\delta_{i,k}}(F_i(x_{i,k} + \delta_{i,k} u_{i,k}, \xi_{i,k}) - F_i(x_{i,k}, \xi_{i,k}))u_{i,k} \tag{5.9}$$

Algorithm 5.2 Distributed Primal–Dual SZO Algorithm

Input: positive sequences $\{\alpha_k\}$, $\{\beta_k\}$, $\{\eta_k\}$, and $\{\delta_{i,k}\}$.
Initialize: $x_{i,0} \in \mathbb{R}^p$ and $v_{i,0} = \mathbf{0}_p,\ \forall i \in [n]$.
for $k = 0, 1, \dots$ **do**
  for $i = 1, \dots, n$ in parallel **do**
    Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
    Select vector $u_{i,k} \in \mathbb{S}^p$ independently and uniformly at random;
    Select $\xi_{i,k}$ independently;
    Sample $F_i(x_{i,k}, \xi_{i,k})$ and $F_i(x_{i,k} + \delta_{i,k}u_{i,k}, \xi_{i,k})$;
    Update $x_{i,k+1}$ by (5.8a);
    Update $v_{i,k+1}$ by (5.8b).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

with $\delta_{i,k} > 0$ being an adaptive smoothing parameter and $u_{i,k} \in \mathbb{S}^p$ being a uniformly distributed random vector chosen by agent i at iteration k, $\xi_{i,k}$ being a random variable chosen by agent i according to the distribution of ξ_i, and $F_i(x_{i,k}+\delta_{i,k}u_{i,k}, \xi_{i,k})$ and $F_i(x_{i,k}, \xi_{i,k})$ being the values sampled by agent i.

Here, we assume that $u_{i,k}$ and $\xi_{i,k},\ \forall i \in [n], k \geq 1$ are mutually independent, which is commonly used when considering the SZO oracle feedback setting, e.g., [142, 148, 149, 155, 214, 215, 248, 289, 291, 292, 296, 297, 304, 305, 307]. Let $\mathfrak{L}_k$ denote the σ-algebra generated by the independent random variables $u_{1,k}, \dots, u_{n,k}, \xi_{1,k}, \dots, \xi_{n,k}$ and let $\mathcal{L}_k = \bigcup_{t=0}^{k} \mathfrak{L}_t$. It is straightforward to see that $x_{i,k}$ and $v_{i,k+1},\ i \in [p]$ depend on $\mathcal{L}_{k-1}$ and are independent of $\mathfrak{L}_t$ for all $t \geq k$.

We write the distributed primal–dual SZO algorithm (5.8) in pseudo-code as Algorithm 5.2.

Remark 5.4. *In Algorithm 5.2, at each iteration each agent samples its local SZO oracles at two different points to estimate the gradient of its local cost function. It should be highlighted that the agent-wise smoothing parameter is adaptive, which is normally larger than the fixed smoothing parameter used in many of existing ZO algorithms, and thus is favorable in practice. For example, in the following we use $\mathcal{O}(1/k^{1/4})$ smoothing parameter, which is larger than the $\mathcal{O}(1/T^{1/2})$ smoothing parameter used in [142].*

5.4.2 Convergence analysis

Find stationary points

Let us consider the case when Algorithm 5.2 is able to find stationary points. We have the following convergence result for Algorithm 5.2 with time-varying parameters.

Theorem 5.3. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Let* $\{\boldsymbol{x}_k\}$ *be the sequence generated by Algorithm 5.2 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \kappa_0(k+t_1)^{\theta},\ \eta_k = \frac{\kappa_2}{\beta_k},\ \delta_{i,k} \le \kappa_\delta\sqrt{\eta_k},\ \forall k \in \mathbb{N}_0, \tag{5.10}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\theta \in (0.5, 1)$, $t_1 \ge (\sqrt{p}c_3(\kappa_1,\kappa_2))^{1/\theta}$, $\kappa_0 \ge c_0(\kappa_1,\kappa_2)/t_1^{\theta}$, *and* $\kappa_\delta > 0$ *with* $c_0(\kappa_1,\kappa_2)$, c_1, $c_2(\kappa_1)$, *and* $c_3(\kappa_1,\kappa_2)$ *defined in Appendix 5.8.3. Then, for any* $T \in \mathbb{N}_+$,

$$\frac{\sum_{k=0}^{T-1}\eta_k\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2]}{\sum_{k=0}^{T-1}\eta_k} = \mathcal{O}(\frac{\sqrt{p}}{T^{1-\theta}}), \tag{5.11a}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1), \tag{5.11b}$$

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T}-\bar{x}_T\|^2\Big] = \mathcal{O}(\frac{1}{T^{2\theta}}), \tag{5.11c}$$

$$\lim_{T\to+\infty}\mathbf{E}[\|\nabla f(\bar{x}_T)\|^2] = 0, \tag{5.11d}$$

where $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n}x_{i,k}$.

Proof. The explicit expressions of the right-hand sides of (5.11a)–(5.11c) and the proof are given in Appendix 5.8.3. □

If the total number of iterations T and the number of agents n are known in advance, then, as shown in the following, Algorithm 5.2 can solve (5.1) with an $\mathcal{O}(\sqrt{p}/\sqrt{nT})$ convergence rate, and thus achieves linear speedup in the number of agents compared to the $\mathcal{O}(\sqrt{p}/\sqrt{T})$ convergence rate achieved by the stochastic gradient-free algorithms for solving centralized stochastic nonconvex optimization in [142, 248].

Theorem 5.4 (Linear speedup)**.** *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. For any given* $T > \max\{n(\tilde{c}_0(\kappa_1,\kappa_2)/\kappa_2)^2,\ n^3\}/p$, *let* $\{\boldsymbol{x}_k,\ k \in [T]\}$ *be the output generated by Algorithm 5.2 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \beta = \frac{\kappa_2\sqrt{pT}}{\sqrt{n}},\ \eta_k = \frac{\kappa_2}{\beta_k},\ \delta_{i,k} \le \frac{\kappa_\delta}{p^{\frac{1}{4}}n^{\frac{1}{4}}(k+1)^{\frac{1}{4}}},\ \forall k \le T, \tag{5.12}$$

where $\tilde{c}_0(\kappa_1,\kappa_2)$ *is defined in Appendix 5.8.4,* $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, *and* $\kappa_\delta > 0$ *with* c_1 *and* $c_2(\kappa_1)$ *defined in Appendix 5.8.3. Then,*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \mathcal{O}(\frac{\sqrt{p}}{\sqrt{nT}}) + \mathcal{O}(\frac{n}{T}), \tag{5.13a}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1), \tag{5.13b}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big] = \mathcal{O}(\frac{n}{T}), \tag{5.13c}$$

$$\lim_{T\to+\infty}\mathbf{E}[\|\nabla f(\bar{x}_T)\|^2] = 0, \tag{5.13d}$$

Proof. The explicit expressions of the right-hand sides of (5.13a)–(5.13b) and the proof are given in Appendix 5.8.4. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.13a) do not depend on any parameters related to the communication network. □

Remark 5.5. *To the best of our knowledge, Theorem 5.4 is the first to establish linear speedup result for distributed SZO algorithms. This rate is faster than the rates achieved by centralized ZO algorithms in [142, 248, 286–289, 307] and the distributed primal ZO algorithm in [151]. This rate is slower than rates achieved by centralized ZO algorithms in [291–295, 297–299], which is reasonable since these algorithms not only are centralized but also use variance reduction techniques. However, in [293–295, 298, 299], the considered problems are deterministic; and in [155, 291, 292], the sampling size of each agent at each iteration is $\mathcal{O}(T)$, which is difficult to execute in practice. It is one of our future research directions to establish faster convergence with reduced sampling complexity by using variance reduction techniques.*

Find global optima

Let us next consider cases when Algorithm 5.2 finds global optima.

Theorem 5.5. *Suppose Assumptions 5.1–5.8 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.2 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \kappa_0(k+t_1)^\theta,\ \eta_k = \frac{\kappa_2}{\beta_k},\ \delta_{i,k} \le \kappa_\delta\eta_k,\ \forall k \in \mathbb{N}_0, \tag{5.14}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\theta \in (0,1)$, $t_1 \in [(pc_3(\kappa_1,\kappa_2))^{1/\theta}, (pc_4c_3(\kappa_1,\kappa_2))^{1/\theta}]$, $\kappa_0 \ge c_0(\kappa_1,\kappa_2)/t_1^\theta$, and $\kappa_\delta > 0$ with $c_4 \ge 1$ being a constant, $c_0(\kappa_1,\kappa_2)$, c_1, $c_2(\kappa_1)$, and $c_3(\kappa_1,\kappa_2)$ defined in Appendix 5.8.3. Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{p}{T^{2\theta}}), \tag{5.15a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{p}{nT^\theta}) + \mathcal{O}(\frac{p}{T^{2\theta}}). \tag{5.15b}$$

Proof. The explicit expressions of the right-hand sides of (5.15a) and (5.15b), and the proof are given in Appendix 5.8.5. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.15b) do not depend on any parameters related to the communication network. □

From Theorem 5.5, we see that the convergence rate is strictly greater than $\mathcal{O}(p/(nT))$. In the following we show that the linear speedup convergence rate $\mathcal{O}(p/(nT))$ can be achieved if the P–Ł constant ν is known in advance. The total number of iterations T is not needed.

Theorem 5.6 (Linear speedup)**.** *Suppose Assumptions 5.1–5.8 hold and the P–Ł constant ν is known in advance. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.2 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \kappa_0(k+t_1),\ \eta_k = \frac{\kappa_2}{\beta_k},\ \delta_{i,k} \le \kappa_\delta\eta_k,\ \forall k \in \mathbb{N}_0, \tag{5.16}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\kappa_0 \in [3\hat{c}_0\nu\kappa_2/16, 3\nu\kappa_2/16)$, $t_1 > \hat{c}_3(\kappa_0,\kappa_1,\kappa_2)$, and $\kappa_\delta > 0$ with $\hat{c}_0 \in (0,1)$ being a constant, c_1 and $c_2(\kappa_1)$ defined in Appendix 5.8.3, and $\hat{c}_3(\kappa_0,\kappa_1,\kappa_2)$ defined in Appendix 5.8.6. Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{p}{T^2}), \tag{5.17a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{p}{nT}) + \mathcal{O}(\frac{p}{T^2}). \tag{5.17b}$$

Proof. The explicit expressions of the right-hand sides of (5.17a) and (5.17b), and the proof are given in Appendix 5.8.6. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.17b) do not depend on any parameters related to the communication network. □

Although Assumption 5.8 is weaker than the bounded gradient assumption, it can be further relaxed by a mild assumption. Specifically, if each $f_i^* > -\infty$, where $f_i^* = \min_{x\in\mathbb{R}^p} f_i(x)$, then without Assumption 5.8, the convergence results stated in (5.17a) and (5.17b) still hold, as shown in the following.

Theorem 5.7 (Linear speedup)**.** *Suppose Assumptions 5.1–5.7 hold, and the P–Ł constant ν is known in advance, and each $f_i^* > -\infty$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.2 with*

$$\alpha_k = \kappa_1\beta_k,\ \beta_k = \kappa_0(k+t_1),\ \eta_k = \frac{\kappa_2}{\beta_k},\ \delta_{i,k} \le \kappa_\delta\eta_k,\ \forall k \in \mathbb{N}_0, \tag{5.18}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\kappa_0 \in [3\hat{c}_0\nu\kappa_2/16, 3\nu\kappa_2/16)$, $t_1 > \check{c}_3(\kappa_0,\kappa_1,\kappa_2)$, and $\kappa_\delta > 0$ with $\hat{c}_0 \in (0,1)$ being a constant, c_1 and $c_2(\kappa_1)$ defined in Appendix 5.8.3, and $\check{c}_3(\kappa_0,\kappa_1,\kappa_2)$ defined in Appendix 5.8.7. Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{p}{T^2}), \tag{5.19a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{p}{nT}) + \mathcal{O}(\frac{p}{T^2}). \tag{5.19b}$$

Proof. The explicit expressions of the right-hand sides of (5.19a) and (5.19b), and the proof are given in Appendix 5.8.7. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.19b) do not depend on any parameters related to the communication network. □

Remark 5.6. *To the best of our knowledge, Theorems 5.5–5.7 are the first analysis of ZO algorithms to solve stochastic optimization problems under the P–Ł condition or the strong convexity assumption. In [281], a centralized ZO algorithm based on one-point sampling with additive sampling noise was proposed and an $\mathcal{O}(p^2/T)$ convergence rate was achieved for deterministic optimization problems strongly convex quadratic cost functions. In [282], a centralized ZO algorithm based on two-point sampling with additive noise was proposed and an $\mathcal{O}(p/\sqrt{T})$ convergence rate was achieved for deterministic strongly convex and smooth optimization problems. In [148], a distributed primal ZO algorithm based on $2p$-point sampling with additive noise was proposed and an $\mathcal{O}(pn^2/\sqrt{T})$ convergence rate was achieved for deterministic strongly convex and smooth optimization problems. In [151], a distributed primal DZO algorithm based on two-point sampling was proposed and an $\mathcal{O}(p/T)$ convergence rate was achieved for deterministic smooth optimization problems under the P–Ł condition. It is straightforward to see that aforementioned convergence rates achieved in [148, 151, 281, 282] are slower than the convergence rate achieved by our distributed primal–dual SZO algorithm as stated in Theorem 5.7, although we consider the SZO oracle feedback setting which is more general than these studies, and use the P–Ł condition which is weaker than the strong convexity condition.*

As shown in Theorems 5.5–5.7, in expectation, the convergence rate of Algorithm 5.2 with diminishing stepsizes is sublinear. The following theorem establishes that, in expectation, the output of Algorithm 5.2 with constant algorithm parameters linearly converges to a neighborhood of a global optimum.

Theorem 5.8. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.2 with*

$$\alpha_k = \kappa_1\beta,\ \beta_k = \beta,\ \eta_k = \frac{\kappa_2}{\beta},\ \delta_{i,k} \le \kappa_\delta\hat{\varepsilon}^{\frac{k}{2}},\ \forall k \in \mathbb{N}_0, \tag{5.20}$$

where $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, $\beta \ge \tilde{c}_0(\kappa_1, \kappa_2)$, $\hat{\varepsilon} \in (0,1)$, and $\kappa_\delta > 0$ with $\tilde{c}_0(\kappa_1,\kappa_2)$ defined in Appendix 5.8.4, and c_1 and $c_2(\kappa_1)$ defined in Appendix 5.8.3. Then, for any $T \in \mathbb{N}_+$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\Big] \le \frac{c_5}{T} + \eta^2(\sigma_1^2 + 3\sigma_2^2)c_6, \tag{5.21a}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] \le \frac{pc_7}{\eta T} + \eta(\sigma_1^2 + 3\eta\sigma_2^2)c_8, \tag{5.21b}$$

where c_5, c_6, c_7, and c_8 are positive constants defined in Appendix 5.8.8. Moreover, if Assumption 5.4 also holds, then

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 + f(\bar{x}_k) - f^*\Big] \le \varepsilon^k c_9 + \eta(\sigma_1^2 + 3\sigma_2^2)c_{10},\ \forall k \in \mathbb{N}_+, \tag{5.22}$$

where $\varepsilon \in (0,1)$, c_9, and c_{10} are positive constants defined in Appendix 5.8.8.

Proof. The proof is given in Appendix 5.8.8. □

Remark 5.7. *When considering centralized nonconvex smooth optimization with DZO oracle feedback, i.e., $\sigma_1 = \sigma_2 = 0$, the result stated in* (5.21b) *shows that a stationary can be found with a rate $\mathcal{O}(p/T)$. This rate is the same as that achieved by the ZO algorithms in [213,285,293,295,298]. Although the ZO variance reduced algorithms in [294,299] and the stochastic direct-search algorithms in [273–275] achieved a faster rate $\mathcal{O}(1/T)$, these algorithms require three or more samplings at each iteration, while our proposed algorithm requires only two samplings at each iteration. Moreover, the result stated in* (5.22) *shows that a global optimum can be found linearly. The ZO algorithms in [213, 283, 285, 299, 301, 303] and the stochastic direct-search algorithms in [273–276] also achieved linear convergence. However, the algorithms in [273–276, 283, 299, 301] require three or more samplings at each iteration; the P–Ł constant needs to be known in advance in [285, 299], which is not needed in Theorem 5.8; and the cost functions in [213, 273–276, 283, 301, 303] are (restricted) strongly convex, which is stronger than the P–Ł condition used in Theorem 5.8.*

5.5 Distributed primal SZO algorithm

Same as Section 5.4, in this section, we also consider the situation that SZO oracle feedback is available. We propose a distributed primal SZO algorithm based on the two-point sampling random gradient estimator introduced in Section 2.8 and analyze its convergence rate.

5.5.1 Algorithm description

Inspired by distributed first-order (sub)gradient descent algorithm proposed in [310], we propose the distributed primal SZO algorithm

$$x_{i,k+1} = x_{i,k} - \gamma \sum_{j=1}^{n} L_{ij} x_{j,k} - \eta_k g^e_{i,k}, \tag{5.23}$$

where γ is a positive constant and $\{\eta_k\}$ is a positive sequence to be specified later and $g^e_{i,k}$ is the stochastic gradient estimator defined in (5.9).

We write the distributed primal SZO algorithm (5.23) in pseudo-code as Algorithm 5.3.

5.5.2 Convergence analysis

Find stationary points

Theorem 5.9. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.3 with*

$$\gamma \in (0, d_1),\ \eta_k = \frac{\kappa_\eta}{(k+t_1)^\theta},\ \delta_{i,k} \le \kappa_\delta \sqrt{\eta_k},\ \forall k \in \mathbb{N}_0, \tag{5.24}$$

Algorithm 5.3 Distributed Primal SZO Algorithm

Input: positive constant γ and positive sequences $\{\eta_k\}$ and $\{\delta_{i,k}\}$.
Initialize: $x_{i,0} \in \mathbb{R}^p,\ \forall i \in [n]$.
for $k = 0, 1, \dots$ **do**
  for $i = 1, \dots, n$ in parallel **do**
    Broadcast $x_{i,k}$ to $\mathcal{N}_i$ and receive $x_{j,k}$ from $j \in \mathcal{N}_i$;
    Select vector $u_{i,k} \in \mathbb{S}^p$ independently and uniformly at random;
    Select $\xi_{i,k}$ independently;
    Sample $F_i(x_{i,k}, \xi_{i,k})$ and $F_i(x_{i,k} + \delta_{i,k}u_{i,k}, \xi_{i,k})$;
    Update $x_{i,k+1}$ by (5.23).
  end for
end for
Output: $\{\boldsymbol{x}_k\}$.

where $\kappa_\delta > 0$, $\kappa_\eta \in (0, d_2(\gamma)t_1^\theta]$, $\theta \in (0.5, 1)$, *and* $t_1 \geq p^{1/(2\theta)}$ *with* d_1 *and* $d_2(\gamma)$ *defined in Appendix 5.8.9. Then, for any* $T \in \mathbb{N}_+$,

$$\frac{\sum_{k=0}^{T-1} \eta_k \mathbf{E}[\|\nabla f(\bar{x}_k)\|^2]}{\sum_{k=0}^{T-1} \eta_k} = \mathcal{O}(\frac{\sqrt{p}}{T^{1-\theta}}), \tag{5.25a}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1), \tag{5.25b}$$

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n} \|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{1}{T^{2\theta}}), \tag{5.25c}$$

$$\lim_{T\to+\infty} \mathbf{E}[\|\nabla f(\bar{x}_T)\|^2] = 0. \tag{5.25d}$$

Proof. The explicit expressions of the right-hand sides of (5.25a)–(5.25c) and the proof are given in Appendix 5.8.9. □

If the total number of iterations T and the number of agents n are known in advance, then, as shown in the following, Algorithm 5.3 can solve (5.1) with $\mathcal{O}(\sqrt{p}/\sqrt{nT})$ convergence rate, and thus achieves the linear speedup with respect to the number of agents.

Theorem 5.10 (Linear speedup)**.** *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. For any given* $T \geq \max\{n/d_2^2(\gamma),\ n^3\}/p$, *let* $\{\boldsymbol{x}_k\ k \in [T]\}$ *be the output generated by Algorithm 5.3 with*

$$\gamma \in (0, d_1),\ \eta_k = \frac{\sqrt{n}}{\sqrt{pT}},\ \delta_{i,k} \leq \frac{\kappa_\delta}{p^{\frac{1}{4}} n^{\frac{1}{4}} (k+1)^{\frac{1}{4}}},\ \forall k \leq T, \tag{5.26}$$

where $\kappa_\delta > 0$ *and* d_1 *and* $d_2(\gamma)$ *are defined in Appendix 5.8.9, then*

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \mathcal{O}(\frac{\sqrt{p}}{\sqrt{nT}}) + \mathcal{O}(\frac{n}{T}), \tag{5.27a}$$

$$\mathbf{E}[f(\bar{x}_T)] - f^* = \mathcal{O}(1), \tag{5.27b}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\Big] = \mathcal{O}(\frac{n}{T}), \tag{5.27c}$$

$$\lim_{T\to+\infty}\mathbf{E}[\|\nabla f(\bar{x}_T)\|^2] = 0, \tag{5.27d}$$

Proof. The explicit expressions of the right-hand sides of (5.27a)–(5.27c) and the proof are given in Appendix 5.8.10. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.27a) do not depend on any parameters related to the communication network. □

Find global optima

Theorem 5.11. *Suppose Assumptions 5.1–5.8 hold. Let* $\{\boldsymbol{x}_k\}$ *be the sequence generated by Algorithm 5.3 with*

$$\gamma \in (0, d_1),\ \eta_k = \frac{\kappa_\eta}{(k+t_1)^\theta},\ \delta_{i,k} \le \kappa_\delta\eta_k,\ \forall k \in \mathbb{N}_0, \tag{5.28}$$

where $\kappa_\delta > 0$, $\kappa_\eta \in (0, d_2(\gamma)t_1^\theta]$, $\theta \in (0,1)$, *and* $t_1 \ge p^{1/\theta}$ *with* d_1 *and* $d_2(\gamma)$ *defined in Appendix 5.8.9. Then, for any* $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{p}{T^{2\theta}}), \tag{5.29a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{p}{nT^\theta}) + \mathcal{O}(\frac{p}{T^{2\theta}}). \tag{5.29b}$$

Proof. The explicit expressions of the right-hand sides of (5.29a) and (5.29b), and the proof are given in Appendix 5.8.11. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.29b) do not depend on any parameters related to the communication network. □

From Theorem 5.11, we see that the convergence rate is strictly greater than $\mathcal{O}(p/(nT))$. In the following we show that the linear speedup convergence rate $\mathcal{O}(p/(nT))$ can be achieved if the P–Ł constant ν is known in advance. The total number of iterations T is not needed.

Theorem 5.12 (Linear speedup)**.** *Suppose Assumptions 5.1–5.8 hold and the P–Ł constant* ν *is known in advance. Let* $\{\boldsymbol{x}_k\}$ *be the sequence generated by Algorithm 5.3 with*

$$\gamma \in (0, d_1),\ \eta_k = \frac{\kappa_\eta}{k+t_1},\ \delta_{i,k} \le \kappa_\delta\eta_k,\ \forall k \in \mathbb{N}_0, \tag{5.30}$$

where $\kappa_\delta > 0$, $\kappa_\eta > 4/\nu$, *and* $t_1 > \hat{d}_2(\gamma)$ *with* d_1 *and* $\hat{d}_2(\gamma)$ *defined in Appendices 5.8.9 and 5.8.12, respectively. Then, for any* $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{p}{T^2}), \tag{5.31a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{p}{nT}) + \mathcal{O}(\frac{p}{T^2}). \tag{5.31b}$$

Proof. The explicit expressions of the right-hand sides of (5.31a) and (5.31b), and the proof are given in Appendix 5.8.12. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.31b) do not depend on any parameters related to the communication network. □

Although Assumption 5.8 is weaker than the bounded gradient assumption, it can be further relaxed by a mild assumption. Specifically, if each $f_i^* > -\infty$, where $f_i^* = \min_{x\in\mathbb{R}^p} f_i(x)$, then without Assumption 5.8, the convergence results stated in (5.31a) and (5.31b) still hold, as shown in the following.

Theorem 5.13 (Linear speedup)**.** *Suppose Assumptions 5.1–5.7 hold, and the P–Ł constant ν is known in advance, and each $f_i^* > -\infty$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.3 with*

$$\gamma \in (0, d_1),\ \eta_k = \frac{\kappa_\eta}{k + t_1},\ \delta_{i,k} \le \kappa_\delta \eta_k,\ \forall k \in \mathbb{N}_0, \tag{5.32}$$

where $\kappa_\delta > 0$, $\kappa_\eta > 4/\nu$, and $t_1 > \check{d}_2(\gamma)$ with d_1 and $\check{d}_2(\gamma)$ defined in Appendices 5.8.9 and 5.8.13, respectively. Then, for any $T \in \mathbb{N}_+$,

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n} \|x_{i,T} - \bar{x}_T\|^2\Big] = \mathcal{O}(\frac{p}{T^2}), \tag{5.33a}$$

$$\mathbf{E}[f(\bar{x}_T) - f^*] = \mathcal{O}(\frac{p}{nT}) + \mathcal{O}(\frac{p}{T^2}). \tag{5.33b}$$

Proof. The explicit expressions of the right-hand sides of (5.33a) and (5.33b), and the proof are given in Appendix 5.8.13. It should be highlighted that the omitted constants in the first term in the right-hand side of (5.33b) do not depend on any parameters related to the communication network. □

As shown in Theorems 5.11–5.13, in expectation, the convergence rate of Algorithm 5.3 with diminishing stepsizes is sublinear. The following theorem establishes that, in expectation, the output of Algorithm 5.3 with constant algorithm parameters linearly converges to a neighborhood of a global optimum.

Theorem 5.14. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.3 with*

$$\gamma \in (0, d_1),\ \eta_k = \eta,\ \delta_{i,k} \le \hat{\epsilon}^{\frac{k}{2}},\ \forall k \in \mathbb{N}_0, \tag{5.34}$$

where $\eta \in (0, d_2(\gamma)$ and $\hat{\epsilon} \in (0,1)$ with d_1 and $d_2(\gamma)$ defined in Appendix 5.8.9. Then, for any $T \in \mathbb{N}_+$,

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n} \|x_{i,k} - \bar{x}_k\|^2\Big] \le \frac{d_3}{T} + \eta^2(\sigma_1^2 + 3\sigma_2^2)d_4, \tag{5.35a}$$

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] \leq \frac{pd_5}{\eta T} + \eta(\sigma_1^2 + 3\sigma_2^2)d_6, \tag{5.35b}$$

where d_3, d_4, d_5, and d_6 are positive constants defined in Appendix 5.8.14. Moreover, if Assumption 5.4 also holds, then

$$\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2 + f(\bar{x}_k) - f^*\Big] \leq \epsilon^k d_7 + \eta(\sigma_1^2 + 3\sigma_2^2)d_8,\ \forall k \in \mathbb{N}_+, \tag{5.36}$$

where $\epsilon \in (0,1)$, d_7, and d_8 are positive constants defined in Appendix 5.8.14.

Proof. The proof is given in Appendix 5.8.14. □

5.6 Simulations

In this section, we verify the theoretical results through numerical examples.

5.6.1 Distributed regularized logistic regression

This section evaluates the performance of Algorithm 5.1 in solving the nonconvex distributed regularized logistic regression problem considered in Section 3.6.1. In this simulation, all settings for cost functions and the communication graph are the same as those described in Section 3.6.1.

We compare Algorithm 5.1 with its FO counterpart (Algorithm 3.1) and state-of-the-art algorithms: distributed gradient descent (DGD) with diminishing stepsizes [110, 116], distributed gradient tracking algorithm (DGTA) [80, 116], distributed ZO gradient tracking algorithm (ZO-GTA) [243], xFILTER [114], Prox-GPDA [112], and D-GPDA [113].

Figure 5.1 illustrates the evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of communication rounds for these algorithms with the same initial condition. It can be seen that both zeroth-order algorithms (Algorithm 5.1 and ZO-GTA [243]) exhibit almost identical behavior as their first-order counterparts (Algorithm 3.1 and DGTA [80, 116]) during the early stage, but then slow down and converge at a sublinear rate.

In order to compare the performance of the two DZO algorithms (Algorithm 5.1 and ZO-GTA [243]), we plot the evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of samplings in Figure 5.2. It can be seen that Algorithm 5.1 gives better performance.

5.6.2 Generating adversarial examples from black-box deep neural networks

This section evaluates the performance of Algorithms 5.2 and 5.3 in generating adversarial examples from black-box deep neural networks (DNNs).

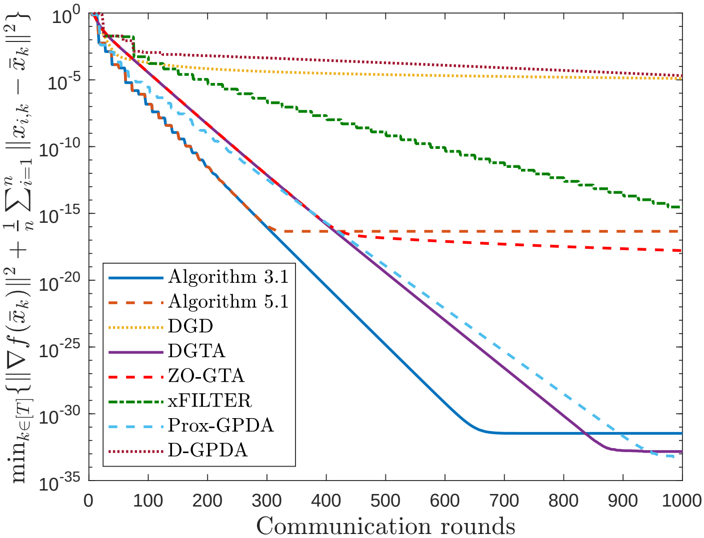


Figure 5.1: Performance of distributed FO and DZO optimization algorithms in the nonconvex distributed regularized logistic regression problem: Evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|x_{i,k} - \bar{x}_k\|^2\}$ with respect to the number of communication rounds.

In image classification tasks, CNNs are vunlberable to adversarial examples [38] even under small perturbations, which leads misclassifications. Considering the setting of zeroth-order attacks [40,295], the model is hidden and no gradient information is available. We treat this task of generating adversarial examples as an zeroth-order optimization problem. The the black-box attack loss function [40, 295] is given as

$$f_i(x) = \max\Big\{F_{y_i}\Big(\frac{1}{2}\tanh(\tanh^{-1}2a_i + x)\Big) - \max_{j\neq y_i}\Big\{F_j\Big(\frac{1}{2}\tanh(\tanh^{-1}2a_i + x)\Big)\Big\},\ 0\Big\}$$
$$+ c\Big\|\frac{1}{2}\tanh(\tanh^{-1}2a_i + x) - a_i\Big\|_2^2,$$

where c is a constant, (a_i, y_i) denotes the pair of the ith natural image a_i and its original class label y_i. The output of function $F(z) = \mathrm{col}(F_1(z), \ldots, F_m(z))$ is the well-trained model prediction of the input z in all m image classes.

The well-trained DNN model[1] on MNIST handwritten has 99.4% test accuracy on natural examples [295]. We compare the proposed distributed primal-dual SZO algorithm (Algorithm 5.2) and distributed primal SZO algorithm (Algorithm 5.3) with state-of-the-art centralized and distributed SZO algorithms: RSGF [142], SZVR-G [297], ZO-SVRG [295], distributed ZO gradient descent algorithm (ZO-GDA) [151], and ZONE-M [155].

[1] https://github.com/carlini/nn_robust_attacks

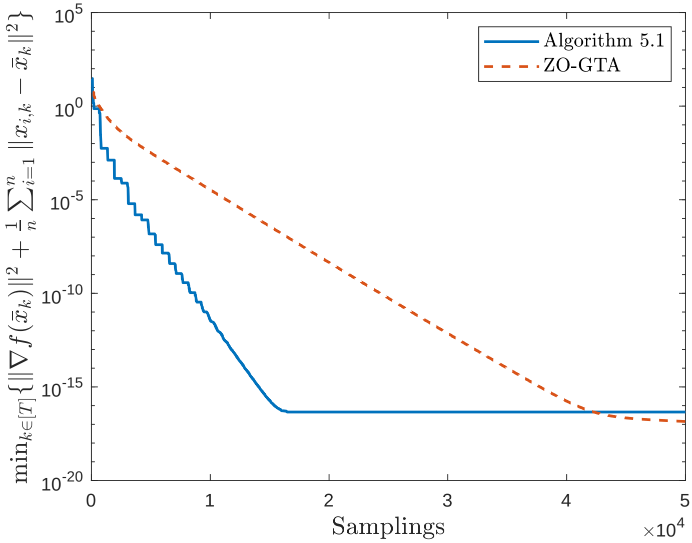


Figure 5.2: Performance of distributed DZO optimization algorithms in the nonconvex distributed regularized logistic regression problem: Evolutions of $\min_{k\in[T]}\{\|\nabla f(\bar{x}_k)\|^2+\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\}$ with respect to the number of samplings.

The communication network of 10 agents is generated randomely following the Erdős - Rényi model with probability of 0.4. All the hyper-parameters that used in the experiments are given in Table 5.2. We set the iteration number as 2500.

Table 5.2: Parameters in each algorithm.

Algorithm	Decentralized	Parameters
Algorithm 5.2	✔	$\eta = 0.5/k^{10^{-5}}$, $\alpha = 0.5k^{10^{-5}}$, $\beta = 0.1k^{10^{-5}}$
Algorithm 5.3	✔	$\gamma = 0.01$, $\eta = 0.08/k^{10^{-5}}$
ZO-GDA	✔	$\eta = 0.08/k^{10^{-5}}$
ZONE-M	✔	$\rho = 0.1\sqrt{k}$
RSGF	✗	$\mu = 0.01$
SZVR-G	✗	$\mu = 0.01$
ZO-SVRG	✗	$\mu = 0.01$

We show the black-box attack loss of each SZO algorithms in Figure 5.3 and list the least ℓ_2 distortion of the successful adversarial perturbations in Table 5.3. We can see that our proposed distributed SZO algorithms converge almost as fast as the ZO-GDA [151], and they all are faster than the other algorithms. However, the adversarial examples generated by these distributed algorithms have slightly larger ℓ_2 distortions than those

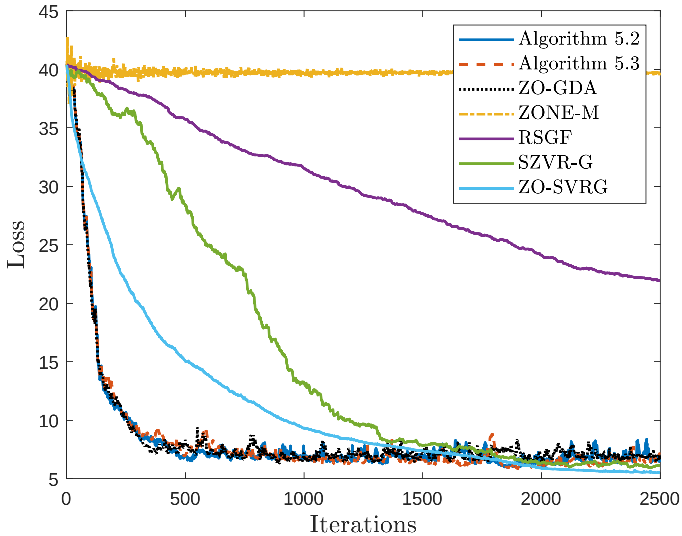


Figure 5.3: Performance of SZO optimization algorithms in generating adversarial examples: Evolutions of the black-box attack loss.

Table 5.3: Distortion

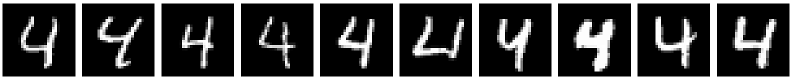
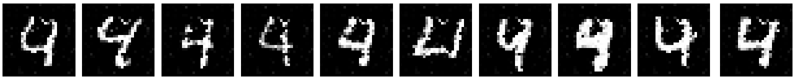
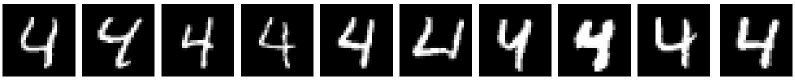
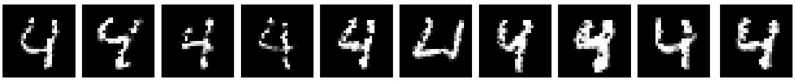
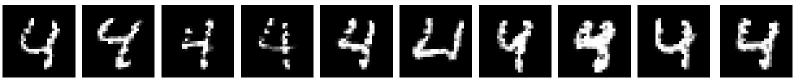
Algorithm	ℓ_2 distortion
Algorithm 5.2	6.44
Algorithm 5.3	5.77
ZO-GDA	7.23
RSGF	5.69
SZVR-G	5.16
ZO-SVRG	4.76

generated by the centralized algorithms. Table 5.4 provides the comparison of generated adversarial examples from a black-box DNN on MNIST: digit class "4".

5.7 Summary

In this chapter, we studied distributed nonconvex optimization with ZO information feedback. We first considered the case that DZO is available and proposed a distributed primal–dual DZO algorithm. We derived its convergence properties, which are the same as its FO counterpart. We then considered the case that SZO is available and proposed two distributed SZO algorithms: distributed primal–dual and primal SZO algorithms. We also

Table 5.4: Comparison of generated adversarial examples from a black-box DNN on MNIST: digit class "4".

Image ID	4	6	19	24	27	33	42	48	49	56
Original										
Algorithm 5.2										
Classified as	9	8	2	7	2	2	9	9	9	9
Algorithm 5.2										
Classified as	9	9	7	9	9	2	9	9	9	9
ZO-GDA										
Classified as	9	9	2	2	2	2	9	9	9	3
ZONE-M										
Classified as	4	4	4	4	4	4	4	4	4	4
RSGF										
Classified as	9	9	2	9	9	2	9	9	9	9
SZVR-G										
Classified as	9	8	2	2	2	2	9	9	9	9
ZO-SVRG										
Classified as	9	8	2	9	9	2	9	9	9	9

analyzed their convergence properties. More specifically, the linear speedup convergence rate $\mathcal{O}(\sqrt{p/(nT)})$ was established for smooth nonconvex cost functions under arbitrarily connected communication networks. The convergence rate was improved to $\mathcal{O}(p/(nT))$ when the global cost function satisfies the P–Ł condition in addition. It was also shown that the output of the proposed algorithms linearly converges to a neighborhood of a global optimum. Interesting directions for future work include considering asynchronous, periodic, or compressed communication, investigating an adaptive choice of the number of samplings at each iteration by different agents, and studying the trade-off between sampling complexity and convergence rate.

5.8 Proofs

5.8.1 Proof of Theorem 5.1

Denote $K_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, $\boldsymbol{K} = K_n \otimes \boldsymbol{I}_p$, $\boldsymbol{H} = \frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)$, $\bar{x}_k = \frac{1}{n}(\mathbf{1}_n^\top \otimes \boldsymbol{I}_p)\boldsymbol{x}_k$, $\bar{\boldsymbol{x}}_k = \mathbf{1}_n \otimes \bar{x}_k$, $\boldsymbol{g}_k = \nabla\tilde{f}(\boldsymbol{x}_k)$, $\bar{\boldsymbol{g}}_k = \boldsymbol{H}\boldsymbol{g}_k$, $\boldsymbol{g}_k^0 = \nabla\tilde{f}(\bar{\boldsymbol{x}}_k)$, $\bar{\boldsymbol{g}}_k^0 = \boldsymbol{H}\boldsymbol{g}_k^0 = \frac{1}{n}(\mathbf{1}_n \otimes \nabla f(\bar{x}_k))$, $h_{i,k} = \hat{\nabla}_p f_i(x_{i,k}, \delta_{i,k})$, $\boldsymbol{h}_k = \mathrm{col}(h_{1,k}, \dots, h_{n,k})$, $\bar{\boldsymbol{h}}_k = \boldsymbol{H}\boldsymbol{h}_k$, $\delta_k = \max_{i\in[n]}\{\delta_{i,k}\}$, $\delta_i^a = \sum_{k=0}^{+\infty}\delta_{i,k}^2$, $h_{i,k}^0 = \hat{\nabla}_p f_i(\bar{x}_k, \delta_k)$, $\boldsymbol{h}_k^0 = \mathrm{col}(h_{1,k}^0, \dots, h_{n,k}^0)$, and $\bar{\boldsymbol{h}}_k^0 = \boldsymbol{H}\boldsymbol{h}_k^0$.

We also denote the following notations.

$$
\begin{aligned}
& c_\beta = \max\Big\{\frac{\kappa_1}{\kappa_2 - 1},\ \kappa_3,\ \kappa_4\Big\},\ c_\eta = \min\Big\{\frac{\zeta_1}{\zeta_2},\ \frac{\zeta_3}{\zeta_4},\ \frac{\zeta_5}{\zeta_6}\Big\},\ \kappa_1 = \frac{1}{2\rho_2(L)}(2 + 9L_f^2),\ \kappa_2 > 1,\\
& \kappa_3 = \frac{1}{4}\Big(1 + \Big(1 + 8\kappa_2 + \frac{8}{\rho_2(L)}\Big)^{\frac{1}{2}}\Big),\ \kappa_4 = 6\Big(\kappa_2 + \frac{1}{\rho_2(L)}\Big)L_f^2 + 2\Big(9\Big(\kappa_2 + \frac{1}{\rho_2(L)}\Big)^2 L_f^4 + 3L_f^2\Big)^{\frac{1}{2}},\\
& \zeta_1 = (\alpha - \beta)\rho_2(L) - \frac{1}{2}(2 + 9L_f^2),\ \zeta_2 = \beta^2\rho(L) + (2\alpha^2 + \beta^2)\rho^2(L) + \frac{15}{2}L_f^2,\\
& \zeta_3 = \beta - \frac{1}{2} - \frac{\alpha}{2\beta^2} - \frac{1}{2\beta\rho_2(L)},\ \zeta_4 = 2\beta^2 + \frac{1}{2},\ \zeta_5 = \frac{1}{8} - \frac{3}{2\beta}\Big(\frac{1}{\beta} + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2,\\
& \zeta_6 = \frac{3}{\beta^2}\Big(1 + \frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2 + \frac{L_f(1 + 3L_f)}{2},\ \zeta_7 = \eta\min\Big\{\zeta_1 - \eta\zeta_2,\ \frac{1}{8}\Big\},\\
& \zeta_8 = \Big(\frac{3\eta}{4} + \eta^2\Big)\frac{15npL_f^2}{4} + \zeta_9,\ \zeta_9 = \Big(\Big(\frac{1}{\beta^2} + \frac{1}{2\eta\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big) + \frac{1}{2\eta\beta^2} + \frac{1}{\beta^2} + \frac{1}{2}\Big)\frac{3npL_f^2}{4},\\
& \zeta_{10} = \frac{\alpha + \beta}{2\beta} + \frac{1}{2\rho_2(L)},\ \zeta_{11} = \min\Big\{\frac{1}{2\rho(L)},\ \frac{\alpha - \beta}{2\alpha}\Big\}.
\end{aligned}
$$

The proof of Theorem 5.1 is similar to the proof of Theorem 3.1 with some modifications. Lemma 3.1 is replaced by the following lemma

Lemma 5.1. *Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.1. If Assumptions 5.1–5.3 hold with $\alpha > \beta$. Then,*

$$
\begin{aligned}
U_{k+1} \le\ & U_k - \|\boldsymbol{x}_k\|^2_{\eta(\zeta_1 - \eta\zeta_2)\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\eta(\zeta_3 - \eta\zeta_4)\boldsymbol{K}} - \eta(\zeta_5 - \eta\zeta_6)\|\bar{\boldsymbol{h}}_k\|^2\\
& - \frac{\eta}{8}\|\bar{\boldsymbol{g}}_k^0\|^2 + \zeta_8\delta_k^2 + \zeta_9\delta_{k+1}^2,
\end{aligned} \tag{5.37}
$$

where

$$
U_k = \sum_{i=1}^{4} U_{i,k},\ U_{1,k} = \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}},\ U_{2,k} = \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\boldsymbol{Q} + \frac{\alpha}{\beta}\boldsymbol{K}},
$$

$$U_{3,k} = \boldsymbol{x}_k^\top \boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big),\ U_{4,k} = n(f(\bar{x}_k) - f^*) = \tilde{f}(\bar{\boldsymbol{x}}_k) - \tilde{f}^*,$$

and $\boldsymbol{Q} = R\Lambda_1^{-1}R^\top \otimes \boldsymbol{I}_p$ *with matrices* R *and* Λ_1^{-1} *given in Lemma 2.5.*

Proof. The distributed deterministic zeroth-order algorithm (5.2) can be rewritten as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta(\alpha\boldsymbol{L}\boldsymbol{x}_k + \beta\boldsymbol{v}_k + \boldsymbol{h}_k), \tag{5.38a}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \eta\beta\boldsymbol{L}\boldsymbol{x}_k,\ \forall\boldsymbol{x}_0 \in \mathbb{R}^{np},\ \boldsymbol{v}_0 = \boldsymbol{0}_{np}. \tag{5.38b}$$

We know that (3.30) still holds. Similar to the way to get (3.32), we have

$$\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 = \|\nabla\tilde{f}(\bar{\boldsymbol{x}}_{k+1}) - \nabla\tilde{f}(\bar{\boldsymbol{x}}_k)\|^2 \le L_f^2\|\bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k\|^2 = \eta^2 L_f^2\|\bar{\boldsymbol{h}}_k\|^2. \tag{5.39}$$

From (2.35a), we have

$$\|\boldsymbol{h}_k - \boldsymbol{g}_k\|^2 \le \frac{npL_f^2\delta_k^2}{4},\ \|\boldsymbol{h}_k^0 - \boldsymbol{g}_k^0\|^2 \le \frac{npL_f^2\delta_k^2}{4}. \tag{5.40}$$

We have

$$\begin{aligned}
\|\boldsymbol{h}_{k+1}^0 - \boldsymbol{h}_k^0\|^2 &= \|\boldsymbol{h}_{k+1}^0 - \boldsymbol{g}_{k+1}^0 + \boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0 + \boldsymbol{g}_k^0 - \boldsymbol{h}_k^0\|^2 \\
&\le 3\|\boldsymbol{h}_{k+1}^0 - \boldsymbol{g}_{k+1}^0\|^2 + 3\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 + 3\|\boldsymbol{g}_k^0 - \boldsymbol{h}_k^0\|^2 \\
&\le \frac{3npL_f^2\delta_{k+1}^2}{4} + 3\eta^2L_f^2\|\bar{\boldsymbol{h}}_k\|^2 + \frac{3npL_f^2\delta_k^2}{4} \\
&= \frac{3npL_f^2(\delta_{k+1}^2 + \delta_k^2)}{4} + 3\eta^2L_f^2\|\bar{\boldsymbol{h}}_k\|^2,
\end{aligned} \tag{5.41}$$

where the first inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds due to (5.40) and (5.39). Similarly, from the Cauchy-Schwarz inequality, (5.40), and (3.30), we have

$$\|\boldsymbol{h}_k^0 - \boldsymbol{h}_k\|^2 \le \frac{3npL_f^2\delta_k^2}{2} + 3L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2. \tag{5.42}$$

Then, from (5.42) and $\rho(\boldsymbol{H}) = 1$, we have

$$\|\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{h}}_k\|^2 = \|\boldsymbol{H}(\boldsymbol{h}_k^0 - \boldsymbol{h}_k)\|^2 \le \|\boldsymbol{h}_k^0 - \boldsymbol{h}_k\|^2 \le \frac{3npL_f^2\delta_k^2}{2} + 3L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2. \tag{5.43}$$

We have

$$\begin{aligned}
\|\bar{\boldsymbol{h}}_k^0\|^2 &= \|\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{g}}_k^0 + \bar{\boldsymbol{g}}_k^0\|^2 = \|\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{g}}_k^0\|^2 + 2(\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{g}}_k^0)^\top\bar{\boldsymbol{g}}_k^0 + \|\bar{\boldsymbol{g}}_k^0\|^2 \\
&\ge \|\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{g}}_k^0\|^2 - 2\|\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{g}}_k^0\|^2 - \frac{1}{2}\|\bar{\boldsymbol{g}}_k^0\|^2 + \|\bar{\boldsymbol{g}}_k^0\|^2 = -\|\bar{\boldsymbol{h}}_k^0 - \bar{\boldsymbol{g}}_k^0\|^2 + \frac{1}{2}\|\bar{\boldsymbol{g}}_k^0\|^2 \\
&\ge -\|\boldsymbol{h}_k^0 - \boldsymbol{g}_k^0\|^2 + \frac{1}{2}\|\bar{\boldsymbol{g}}_k^0\|^2 \ge -\frac{npL_f^2\delta_k^2}{4} + \frac{1}{2}\|\bar{\boldsymbol{g}}_k^0\|^2,
\end{aligned} \tag{5.44}$$

where the first inequality holds due to the Cauchy-Schwarz inequality; the second inequality holds due to $\rho(\boldsymbol{H}) = 1$; and the last inequality holds due to (5.40).

Similar to the way to get (3.33), from (5.38a) and (2.5), we have

$$\begin{aligned} U_{1,k+1} \le\, & U_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2} + \frac{\eta}{2}(1+3\eta)\|\boldsymbol{h}_k - \boldsymbol{h}_k^0\|^2 \\ & - \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}}. \end{aligned} \tag{5.45}$$

Then, from (5.45) and (5.42), we have

$$\begin{aligned} U_{1,k+1} \le\, & U_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta\alpha\boldsymbol{L}-\frac{\eta}{2}\boldsymbol{K}-\frac{3\eta^2\alpha^2}{2}\boldsymbol{L}^2-\frac{3\eta}{2}(1+3\eta)L_f^2\boldsymbol{K}} - \eta\beta\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big) \\ & + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\frac{3\eta^2\beta^2}{2}\boldsymbol{K}} + \frac{3npL_f^2\delta_k^2\eta(1+3\eta)}{4}. \end{aligned} \tag{5.46}$$

Similar to the way to get (3.35), from (5.38b), (2.5), and (2.7), we have

$$\begin{aligned} U_{2,k+1} \le\, & U_{2,k} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big) + \|\boldsymbol{x}_k\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\ & + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + \Big(\frac{1}{\beta^2} + \frac{1}{2\eta\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)\|\boldsymbol{h}_{k+1}^0 - \boldsymbol{h}_k^0\|^2. \end{aligned} \tag{5.47}$$

Then, from (5.47) and (5.41), we have

$$\begin{aligned} U_{2,k+1} \le\, & U_{2,k} + \eta\boldsymbol{x}_k^\top(\beta\boldsymbol{K} + \alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big) + \|\boldsymbol{x}_k\|^2_{\eta^2\beta(\beta\boldsymbol{L}+\alpha\boldsymbol{L}^2)} \\ & + \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\frac{\eta}{2\beta}(\boldsymbol{Q}+\frac{\alpha}{\beta}\boldsymbol{K})} + +3\eta\Big(\frac{\eta}{\beta^2} + \frac{1}{2\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)L_f^2\|\bar{\boldsymbol{h}}_k\|^2 \\ & + \Big(\frac{1}{\beta^2} + \frac{1}{2\eta\beta}\Big)\Big(\frac{1}{\rho_2(L)} + \frac{\alpha}{\beta}\Big)\frac{3npL_f^2(\delta_{k+1}^2 + \delta_k^2)}{4}. \end{aligned} \tag{5.48}$$

Similar to the way to get (3.37), from (5.38) and (2.5), we have

$$\begin{aligned} U_{3,k+1} \le\, & \boldsymbol{x}_k^\top(\boldsymbol{K} - \eta\alpha\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big) + \frac{\eta}{2}(1+2\eta)\|\boldsymbol{h}_k - \boldsymbol{h}_k^0\|^2 + \|\boldsymbol{x}_k\|^2_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{\alpha^2}{2}-\alpha\beta+\beta^2)\boldsymbol{L}^2} \\ & + \Big(\frac{1}{2\eta\beta^2} + \frac{1}{\beta^2} + \frac{1}{2}\Big)\|\boldsymbol{h}_{k+1}^0 - \boldsymbol{h}_k^0\|^2 - \Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\eta(\beta-\frac{1}{2}-\frac{\eta}{2}-\frac{\eta\beta^2}{2})\boldsymbol{K}}. \end{aligned} \tag{5.49}$$

Then, from (5.49), (5.41), and (5.42), we have

$$\begin{aligned} U_{3,k+1} \le\, & U_{3,k} - \eta\alpha\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}_k^0\Big) + \|\boldsymbol{x}_k\|^2_{\eta(\beta\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta^2(\frac{\alpha^2}{2}-\alpha\beta+\beta^2)\boldsymbol{L}^2+\frac{3\eta}{2}(1+2\eta)L_f^2\boldsymbol{K}} \\ & + \frac{3npL_f^2\delta_k^2\eta(1+2\eta)}{4} + 3\eta\Big(\frac{1}{2\beta^2} + \frac{\eta}{\beta^2} + \frac{\eta}{2}\Big)L_f^2\|\bar{\boldsymbol{h}}_k\|^2 \end{aligned}$$

$$+\Big(\frac{1}{2\eta\beta^2}+\frac{1}{\beta^2}+\frac{1}{2}\Big)\frac{3npL_f^2(\delta_{k+1}^2+\delta_k^2)}{4}-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\eta(\beta-\frac{1}{2}-\frac{\eta}{2}-\frac{\eta\beta^2}{2})\boldsymbol{K}}. \tag{5.50}$$

Similar to the way to get (3.39), we have

$$\begin{aligned}
U_{4,k+1} &= \tilde{f}(\bar{\boldsymbol{x}}_{k+1})-\tilde{f}^* = \tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*+\tilde{f}(\bar{\boldsymbol{x}}_{k+1})-\tilde{f}(\bar{\boldsymbol{x}}_k)\\
&\le \tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*-\eta\bar{\boldsymbol{h}}_k^\top\boldsymbol{g}_k^0+\frac{\eta^2L_f}{2}\|\bar{\boldsymbol{h}}_k\|^2\\
&= \tilde{f}(\bar{\boldsymbol{x}}_k)-\tilde{f}^*-\eta\bar{\boldsymbol{h}}_k^\top\boldsymbol{h}_k^0+\frac{\eta^2L_f}{2}\|\bar{\boldsymbol{h}}_k\|^2-\eta\bar{\boldsymbol{h}}_k^\top(\boldsymbol{g}_k^0-\boldsymbol{h}_k^0)\\
&\le U_{4,k}-\frac{\eta}{4}(1-2\eta L_f)\|\bar{\boldsymbol{h}}_k\|^2+\frac{\eta}{2}\|\bar{\boldsymbol{h}}_k^0-\bar{\boldsymbol{h}}_k\|^2-\frac{\eta}{4}\|\bar{\boldsymbol{h}}_k^0\|^2-\eta\bar{\boldsymbol{h}}_k^\top(\boldsymbol{g}_k^0-\boldsymbol{h}_k^0).
\end{aligned} \tag{5.51}$$

Then, from (5.51), the Cauchy-Schwarz inequality, (5.40), (5.43), and (5.44), we have

$$\begin{aligned}
U_{4,k+1} &\le U_{4,k}-\frac{\eta}{4}(1-2\eta L_f)\|\bar{\boldsymbol{h}}_k\|^2+\frac{\eta}{2}\|\bar{\boldsymbol{h}}_k^0-\bar{\boldsymbol{h}}_k\|^2-\frac{\eta}{4}\|\bar{\boldsymbol{h}}_k^0\|^2+\frac{\eta}{8}\|\bar{\boldsymbol{h}}_k\|^2+2\eta\|\boldsymbol{g}_k^0-\boldsymbol{h}_k^0\|^2\\
&\le U_{4,k}-\frac{\eta}{8}(1-4\eta L_f)\|\bar{\boldsymbol{h}}_k\|^2+\frac{3npL_f^2\delta_k^2\eta}{4}+\|\boldsymbol{x}_k\|^2_{\frac{3\eta}{2}L_f^2\boldsymbol{K}}\\
&\quad+\frac{npL_f^2\delta_k^2\eta}{16}-\frac{\eta}{8}\|\bar{\boldsymbol{g}}_k^0\|^2+\frac{npL_f^2\delta_k^2\eta}{2}\\
&= U_{4,k}-\frac{\eta}{8}(1-4\eta L_f)\|\bar{\boldsymbol{h}}_k\|^2+\frac{21npL_f^2\delta_k^2\eta}{16}+\|\boldsymbol{x}_k\|^2_{\frac{3\eta}{2}L_f^2\boldsymbol{K}}-\frac{\eta}{8}\|\bar{\boldsymbol{g}}_k^0\|^2.
\end{aligned} \tag{5.52}$$

Hence, from (5.46), (5.48), (5.50), and(5.52), we have

$$\begin{aligned}
U_{k+1} &\le U_k-\|\boldsymbol{x}_k\|^2_{\eta\boldsymbol{M}_1-\eta^2\boldsymbol{M}_2}-\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\eta\boldsymbol{M}_3-\eta^2\boldsymbol{M}_4}\\
&\quad-(\eta\zeta_5-\eta^2\zeta_6)\|\bar{\boldsymbol{h}}_k\|^2-\frac{\eta}{8}\|\bar{\boldsymbol{g}}_k^0\|^2+\zeta_8\delta_k^2+\zeta_9\delta_{k+1}^2,
\end{aligned} \tag{5.53}$$

where

$$\begin{aligned}
&\boldsymbol{M}_1=(\alpha-\beta)\boldsymbol{L}-\frac{1}{2}(2+9L_f^2)\boldsymbol{K},\ \boldsymbol{M}_2=\beta^2\boldsymbol{L}+(2\alpha^2+\beta^2)\boldsymbol{L}^2+\frac{15}{2}L_f^2\boldsymbol{K},\\
&\boldsymbol{M}_3=\Big(\beta-\frac{1}{2}-\frac{\alpha}{2\beta^2}\Big)\boldsymbol{K}-\frac{1}{2\beta}\boldsymbol{Q},\ \boldsymbol{M}_4=(2\beta^2+\frac{1}{2})\boldsymbol{K}.
\end{aligned}$$

Similar to the way to get (3.26), we have (5.37). □

We are now ready to prove Theorem 5.1.
Denote

$$\hat{U}_k=\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+\Big\|\boldsymbol{v}_k+\frac{1}{\beta}\boldsymbol{h}_k^0\Big\|^2_{\boldsymbol{K}}+n(f(\bar{x}_k)-f^*).$$

Similar to the way to get (3.46)–(3.48), we have

$$U_k \geq \zeta_{11}(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \left\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{h}^0_k\right\|^2_{\boldsymbol{K}}) + n(f(\bar{x}_k) - f^*) \tag{5.54}$$

$$\geq \zeta_{11}\hat{U}_k \geq 0, \tag{5.55}$$

and

$$U_k \leq \zeta_{10}\hat{U}_k. \tag{5.56}$$

Similar to the way to get (3.49)–(3.51), we have

$$\zeta_1 > 0,\ \zeta_3 > 0,\ \text{and}\ \zeta_5 > 0. \tag{5.57}$$

From (5.57) and $0 < \eta < \min\{\frac{\zeta_1}{\zeta_2},\ \frac{\zeta_3}{\zeta_4},\ \frac{\zeta_5}{\zeta_6}\}$, we have

$$\eta(\zeta_1 - \eta\zeta_2) > 0, \tag{5.58a}$$

$$\eta(\zeta_3 - \eta\zeta_4) > 0, \tag{5.58b}$$

$$\eta(\zeta_5 - \eta\zeta_6) > 0, \tag{5.58c}$$

$$\zeta_7 > 0. \tag{5.58d}$$

From (5.37), (5.58a)–(5.58d), and $\boldsymbol{K} \geq 0$, we have

$$U_{k+1} \leq U_k - \zeta_7(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}^0_k\|^2) + \zeta_8\delta_k^2 + \zeta_9\delta_{k+1}^2. \tag{5.59}$$

Hence, summing (5.59) over $k = 0, \dots, T$ yields

$$U_{T+1} + \zeta_7\sum_{k=0}^{T}(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \|\bar{\boldsymbol{g}}^0_k\|^2) \leq U_0 + (\zeta_8 + \zeta_9)\sum_{k=0}^{T+1}\delta_k^2. \tag{5.60}$$

We know

$$\delta_k^2 = \Big(\max_{i\in[n]}\{\delta_{i,k}\}\Big)^2 \leq \sum_{i=1}^{n}\delta_{i,k}^2. \tag{5.61}$$

From (5.3) and (5.61), we have

$$\sum_{k=0}^{T+1}\delta_k^2 \leq \sum_{i=1}^{n}\delta_i^a,\ \forall T \in \mathbb{N}_0. \tag{5.62}$$

From (5.60), (5.62), (5.55), and (5.58d), we have

$$\frac{\sum_{k=0}^{T}(\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + \|\bar{\boldsymbol{g}}^0_k\|^2)}{T+1} \leq \frac{U_0 + (\zeta_8 + \zeta_9)\sum_{i=1}^{n}\delta_i^a}{\zeta_7(T+1)},\ \forall T \in \mathbb{N}_0, \tag{5.63}$$

which yields (5.4) and (5.5).

From (5.60), (5.62), and (5.58d), we have

$$f(\bar{x}_{T+1}) - f^* \leq \frac{U_0 + (\zeta_8 + \zeta_9)\sum_{i=1}^{n}\delta_i^a}{n},\ \forall T \in \mathbb{N}_0, \tag{5.64}$$

which gives (5.6).

5.8.2 Proof of Theorem 5.2

In addition to the notations defined in Section 5.8.1, we also denote the following notations.

$$\zeta_{12} = \eta \min\Big\{\zeta_1 - \eta\zeta_2,\ \zeta_3 - \eta\zeta_4,\ \frac{\nu}{4}\Big\},\ \zeta_{13} = \frac{\zeta_{12}}{\zeta_{10}},\ \zeta = \frac{1}{2} + \frac{1}{2}\max\{1 - \zeta_{13},\ \kappa_\delta\},$$
$$\zeta_{14} = \Big(\frac{\zeta_8}{1 - \zeta_{13}} + \zeta_9\Big)\frac{1}{\zeta - \kappa_\delta},\ \zeta_0 = \frac{\zeta}{n\zeta_{11}}(U_0 + \zeta_{14}).$$

From (5.58a) and (5.58b), we have

$$\zeta_{12} > 0,\ \zeta_{13} = \frac{\zeta_{12}}{\zeta_{10}} > 0. \tag{5.65}$$

Similar to the way to get (3.63), we have

$$0 < \zeta_{13} < \frac{1}{8}. \tag{5.66}$$

From (2.16) and Assumption 4.4, we have that

$$\|\bar{\boldsymbol{g}}_k^0\|^2 = n\|\nabla f(\bar{x}_k)\|^2 \ge 2\nu n(f(\bar{x}_k) - f^*) = 2\nu U_{4,k}. \tag{5.67}$$

From (5.37), (5.58c), and (5.67), we have

$$U_{k+1} \le U_k - \zeta_{12}\hat{U}_k + \zeta_{11}\delta_k^2 + \zeta_{12}\delta_{k+1}^2. \tag{5.68}$$

From (5.68), (5.56), and (5.65), we have

$$U_{k+1} \le U_k - \frac{\zeta_{12}}{\zeta_7}U_k + \zeta_8\delta_k^2 + \zeta_9\delta_{k+1}^2 = (1 - \zeta_{13})U_k + \zeta_8\delta_k^2 + \zeta_9\delta_{k+1}^2. \tag{5.69}$$

From (5.69), (5.66), and (5.55), we have

$$U_{k+1} \le (1 - \zeta_{13})^{k+1}U_0 + \zeta_8\sum_{\tau=0}^{k}(1 - \zeta_{13})^\tau\delta_{k-\tau}^2 + \zeta_9\sum_{\tau=0}^{k}(1 - \zeta_{13})^\tau\delta_{k+1-\tau}^2. \tag{5.70}$$

From $\delta_{i,k} \in (0, \kappa_\delta^{k/2})$ and (5.70), we have

$$\begin{aligned} U_{k+1} &\le (1 - \zeta_{13})^{k+1}U_0 + \Big(\frac{\zeta_8}{1 - \zeta_{13}} + \zeta_9\Big)\sum_{\tau=0}^{k+1}(1 - \zeta_{13})^\tau\delta_{k+1-\tau}^2 \\ &\le (1 - \zeta_{13})^{k+1}U_0 + \Big(\frac{\zeta_8}{1 - \zeta_{13}} + \zeta_9\Big)\sum_{\tau=0}^{k+1}(1 - \zeta_{13})^\tau\kappa_\delta^{k+1-\tau}. \end{aligned} \tag{5.71}$$

From $\kappa_\delta \in (0, 1)$, (5.65), (5.66), and (2.36), we have

$$U_{k+1} \le (1 - \zeta_{13})^{k+1}U_0 + \zeta^{k+1}\zeta_{14}. \tag{5.72}$$

From (5.54), we have

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) = \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + n(f(\bar{x}_k) - f^*) \leq \hat{U}_k \leq \frac{U_k}{\zeta_{11}}. \tag{5.73}$$

Hence, (5.72) and (5.73) give

$$\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|^2 + n(f(\bar{x}_k) - f^*) \leq \frac{1}{\zeta_{11}}((1-\zeta_{13})^{k+1}U_0 + \zeta^{k+1}\zeta_{14}),\ \forall k \in \mathbb{N}_0, \tag{5.74}$$

which yields (5.7).

5.8.3 Proof of Theorem 5.3

Denote $\boldsymbol{g}^e_k = \mathrm{col}(g^e_{1,k}, \ldots, g^e_{n,k})$, $\bar{g}^e_k = \frac{1}{n}(\boldsymbol{1}_n^\top \otimes \boldsymbol{I}_p)\boldsymbol{g}^e_k$, $\bar{\boldsymbol{g}}^e_k = \boldsymbol{1}_n \otimes \bar{g}^e_k = \boldsymbol{H}\boldsymbol{g}^e_k$, $f^s_i(x, \delta_{i,k}) = \mathbf{E}_{u\in\mathbb{B}^p}[f_i(x + \delta_{i,k}u)]$, $g^s_{i,k} = \nabla f^s_i(x_{i,k}, \delta_{i,k})$, $\boldsymbol{g}^s_k = \mathrm{col}(g^s_{1,k}, \ldots, g^s_{n,k})$, and $\bar{\boldsymbol{g}}^s_k = \boldsymbol{H}\boldsymbol{g}^s_k$.

We also denote the following notations.

$$\begin{aligned}
&c_0(\kappa_1,\kappa_2) = \max\Big\{\varepsilon_1,\ \frac{2\varepsilon_5}{\varepsilon_4},\ \Big(\frac{2p\varepsilon_7}{\varepsilon_4}\Big)^{\frac{1}{2}},\ \frac{\varepsilon_8}{2\varepsilon_6},\ \frac{24\kappa_4}{\kappa_2},\ 96p\kappa_2\varepsilon_{10}\Big\},\ c_1 = \frac{1}{\rho_2(L)} + 1,\\
&c_2(\kappa_1) = \min\Big\{\frac{\varepsilon_2}{\varepsilon_3},\ \frac{1}{5}\Big\},\ c_3(\kappa_1,\kappa_2) = \frac{24\kappa_3}{\kappa_2},\ \kappa_3 = \frac{1}{\rho_2(L)} + \kappa_1 + 1,\ \kappa_4 = \frac{1}{\rho_2(L)} + \kappa_1,\\
&\kappa_5 = \frac{1}{\rho_2(L)} + \kappa_1 + \frac{3}{2},\ \kappa_6 = \frac{\kappa_1+1}{2} + \frac{1}{2\rho_2(L)},\ \kappa_7 = \min\Big\{\frac{1}{2\rho(L)},\ \frac{\kappa_1 - 1}{2\kappa_1}\Big\},\\
&\varepsilon_1 = \max\Big\{1 + 3L_f^2,\ (8 + 12p(3 + 0.5L_f))^{\frac{1}{2}}L_f,\ p\kappa_3\Big\},\ \varepsilon_2 = (\kappa_1 - 1)\rho_2(L) - 1,\\
&\varepsilon_3 = \rho(L) + (2\kappa_1^2 + 1)\rho(L^2) + 1,\ \varepsilon_4 = 0.5(\varepsilon_2\kappa_2 - \varepsilon_3\kappa_2^2),\\
&\varepsilon_5 = 0.5 - \kappa_1\kappa_2\rho_2(L) + \kappa_2^2\rho(L) + 0.5(1 + 3\kappa_1\kappa_2 + 2\kappa_2)\kappa_1\kappa_2\rho(L^2),\ \varepsilon_6 = 0.25(\kappa_2 - 5\kappa_2^2),\\
&\varepsilon_7 = 6(1 + 6\kappa_2 + 2\kappa_4 + 10\kappa_2\kappa_4)\kappa_2L_f^4 + \frac{1}{2p}(1 + 2L_f^2)\kappa_2 + \Big(\frac{5}{p} + 24\Big)L_f^2\kappa_2^2,\\
&\varepsilon_8 = \kappa_4 + \kappa_1\kappa_2 + 3\kappa_2^2 + \kappa_2\kappa_4,\ \varepsilon_9 = \frac{3\kappa_0}{2\kappa_2^2}(2\kappa_4 + 1),\\
&\varepsilon_{10} = 10 + L_f + \frac{1}{\kappa_2}(2\kappa_4 + 1)L_f^2 + (10\kappa_4 + 6)L_f^2,\ \varepsilon_{11} = L_f^2\Big(\frac{1}{384} + \frac{1}{p}(13\kappa_2 + 4)\Big),\\
&\varepsilon_{12} = 2\varepsilon_{10}\sigma_1^2 + \frac{1}{p}\varepsilon_9\sigma_2^2 + 6\varepsilon_{10}\sigma_2^2,\ \varepsilon_{13} = \frac{1}{p}\varepsilon_9 + 6\varepsilon_{10},\ \varepsilon_{14} = \frac{W_0}{n} + \frac{2\theta p(\varepsilon_{11}\kappa_\delta^2 + \varepsilon_{12})\kappa_2^2}{(2\theta - 1)\kappa_0^2},\\
&a_1 = \frac{1}{\kappa_6}\min\{\varepsilon_4,\ \varepsilon_6\},\ a_2 = pn(\varepsilon_{11}\kappa_\delta^2 + \varepsilon_{12} + 2L_f\varepsilon_{13}\varepsilon_{14})\frac{\kappa_2^2}{\kappa_0^2}.
\end{aligned}$$

To prove Theorem 5.3, the following three lemmas are used.

Lemma 5.2. *Suppose Assumption 5.6 holds. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.2, then*

$$\boldsymbol{g}^s_k = \mathbf{E}_{\mathfrak{L}_k}[\boldsymbol{g}^e_k], \tag{5.75a}$$

$$\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^s\|^2 \le 2L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + 2nL_f^2\delta_k^2, \tag{5.75b}$$

$$\|\bar{\boldsymbol{g}}_k^0 - \bar{\boldsymbol{g}}_k^s\|^2 \le 2L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + 2nL_f^2\delta_k^2, \tag{5.75c}$$

$$\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] \le \frac{1}{n}\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] + \|\bar{\boldsymbol{g}}_k^s\|^2, \tag{5.75d}$$

$$\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^e\|^2] \le 4L_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + 4nL_f^2\delta_k^2 + 2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2], \tag{5.75e}$$

$$\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 \le \eta_k^2L_f^2\|\bar{\boldsymbol{g}}_k^e\|^2 \le \eta_k^2L_f^2\|\boldsymbol{g}_k^e\|^2, \tag{5.75f}$$

$$\|\bar{\boldsymbol{g}}_k^0\|^2 \le 2nL_f(f(\bar{x}_k) - f^*). \tag{5.75g}$$

If Assumptions 5.7 and 5.8 also hold, then

$$\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] \le 12p\|\bar{\boldsymbol{g}}_k^0\|^2 + 12pL_f^2\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + 4np\sigma_1^2 + 12np\sigma_2^2 + 0.5np^2L_f^2\delta_k^2, \tag{5.76a}$$

$$\|\boldsymbol{g}_{k+1}^0\|^2 \le 3(\eta_k^2L_f^2\|\boldsymbol{g}_k^e\|^2 + n\sigma_2^2 + \|\bar{\boldsymbol{g}}_k^0\|^2). \tag{5.76b}$$

Proof. **(i)** From $u_{i,k}$ and $\xi_{i,k}$ are mutually independent, $x_{i,k}$ is independent of $u_{i,k}$ and $\xi_{i,k}$, and (2.27), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[g_{i,k}^e] &= \mathbf{E}_{u_{i,k}}\Big[\mathbf{E}_{\xi_{i,k}}\Big[\frac{p}{\delta_{i,k}}(F_i(x_{i,k} + \delta_{i,k}u_{i,k}, \xi_{i,k} - F_i(x_{i,k}, \xi_{i,k}))u_{i,k}\Big]\Big]\\ &= \mathbf{E}_{u_{i,k}}\Big[\frac{p}{\delta_{i,k}}(f_i(x_{i,k} + \delta_{i,k}u_{i,k}) - f_i(x_{i,k}))u_{i,k}\Big]\\ &= \mathbf{E}_{u_{i,k}}[\hat{\nabla}_2 f_i(x_{i,k}, \delta_{i,k}, u_{i,k})] = \nabla f_i^s(x_{i,k}, \delta_{i,k}) = g_{i,k}^s,\end{aligned}$$

which gives (5.75a).
(ii) From Assumption 5.3, we know that (3.30) still holds.
From Assumption 5.3 and (2.32a), we have

$$\|g_{i,k}^s - g_{i,k}\| \le L_f\delta_{i,k}.$$

Thus,

$$\|\boldsymbol{g}_k^s - \boldsymbol{g}_k\|^2 = \sum_{i=1}^n \|g_{i,k}^s - g_{i,k}\|^2 \le nL_f^2\delta_k^2. \tag{5.77}$$

Noting that

$$\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^s\|^2 \le 2\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k\|^2 + 2\|\boldsymbol{g}_k - \boldsymbol{g}_k^s\|^2,$$

from (3.30) and (5.77), we know (5.75b) holds.
(iii) Noting $\|\bar{\boldsymbol{g}}_k^0 - \bar{\boldsymbol{g}}_k^s\|^2 = \|\boldsymbol{H}(\boldsymbol{g}_k^0 - \boldsymbol{g}_k^s)\|^2$, from $\rho(\boldsymbol{H}) = 1$ and (5.75b), we have (5.75c).
(iv) We have

$$\mathbf{E}_{\mathfrak{L}_k}[\|\bar{g}_k^e\|^2] = \mathbf{E}_{\mathfrak{L}_k}\Big[\Big\|\sum_{i=1}^n \frac{1}{n}g_{i,k}^e\Big\|^2\Big] = \frac{1}{n^2}\mathbf{E}_{\mathfrak{L}_k}\Big[\sum_{i=1}^n \|g_{i,k}^e\|^2 + \sum_{i=1}^n \sum_{j=1, j\neq i}^n \langle g_{i,k}^e, g_{j,k}^e\rangle\Big]$$

$$= \frac{1}{n^2}\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\langle\mathbf{E}_{\mathfrak{L}_k}[g_{i,k}^e],\mathbf{E}_{\mathfrak{L}_k}[g_{j,k}^e]\rangle$$
$$= \frac{1}{n^2}\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\langle g_{i,k}^s, g_{j,k}^s\rangle$$
$$= \frac{1}{n^2}\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] + \|\bar{g}_k^s\|^2 - \frac{1}{n^2}\|\boldsymbol{g}_k^s\|^2, \tag{5.78}$$

where the third equality holds since $u_{i,k}$ and $\xi_{i,k}$, $\forall i \in [n], k \geq 1$ are mutually independent; and the fourth equality holds due to (5.75a).

From (5.78), $\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] = n\mathbf{E}_{\mathfrak{L}_k}[\|\bar{g}_k^e\|^2]$ and $\|\bar{\boldsymbol{g}}_k^s\|^2 = n\|\bar{g}_k^s\|^2$, we know that (5.75d) holds.

(v) We have

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^e\|^2] &\leq 2\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^s\|^2 + 2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^s - \boldsymbol{g}_k^e\|^2]\\ &= 2\|\boldsymbol{g}_k^0 - \boldsymbol{g}_k^s\|^2 + 2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] - 2\|\boldsymbol{g}_k^s\|^2,\end{aligned} \tag{5.79}$$

where the inequality holds due to the Cauchy-Schwarz inequality; and the equality holds since (5.75a) and $\boldsymbol{x}_k$ is independent of $\mathfrak{L}_k$.

From (5.79) and (5.75b), we know (5.75e) holds.

(vi) The distributed ZO algorithm (5.8) can be rewritten as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta_k(\alpha_k\boldsymbol{L}\boldsymbol{x}_k + \beta_k\boldsymbol{v}_k + \boldsymbol{g}_k^e), \tag{5.80a}$$
$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \eta_k\beta_k\boldsymbol{L}\boldsymbol{x}_k,\ \forall\boldsymbol{x}_0 \in \mathbb{R}^{np},\ \sum_{i=1}^{n} v_{i,0} = \mathbf{0}_p. \tag{5.80b}$$

From (5.80b), we know that

$$\bar{v}_{k+1} = \bar{v}_k. \tag{5.81}$$

Then, from (5.81), $\sum_{i=1}^{n} v_{i,0} = \mathbf{0}_p$, and (5.80a), we know that $\bar{v}_k = \mathbf{0}_p$ and

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_k - \eta_k\bar{\boldsymbol{g}}_k^e. \tag{5.82}$$

Then, similar to the way to get (3.32), we have

$$\|\boldsymbol{g}_{k+1}^0 - \boldsymbol{g}_k^0\|^2 = \|\nabla\tilde{f}(\bar{\boldsymbol{x}}_{k+1}) - \nabla\tilde{f}(\bar{\boldsymbol{x}}_k)\|^2 \leq L_f^2\|\bar{\boldsymbol{x}}_{k+1} - \bar{\boldsymbol{x}}_k\|^2 = \eta_k^2L_f^2\|\bar{\boldsymbol{g}}_k^e\|^2 \leq \eta_k^2L_f^2\|\boldsymbol{g}_k^e\|^2,$$

which yields (5.75f).

(vii) From (2.15), we have

$$\|\bar{\boldsymbol{g}}_k^0\|^2 = n\|\nabla f(\bar{x}_k)\|^2 \leq 2nL_f(f(\bar{x}_k) - f^*), \tag{5.83}$$

which yields (5.75g).

(viii) From Assumption 5.6, $x_{i,k}$ and $\xi_{i,k}$ are independent of $u_{i,k}$, and (2.32b), we know that for almost every $\xi_{i,k}$ it holds that

$$\mathbf{E}_{u_{i,k}}[\|g^e_{i,k}\|^2] \le 2p\|\nabla_x F_i(x_{i,k},\xi_{i,k})\|^2 + 0.5p^2L_f^2\delta_{i,k}^2. \tag{5.84}$$

Then,

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[\|g^e_{i,k}\|^2] &\le 2p\mathbf{E}_{\xi_{i,k}}[\|\nabla_x F_i(x_{i,k},\xi_{i,k})\|^2] + 0.5p^2L_f^2\delta_{i,k}^2\\ &= 2p\mathbf{E}_{\xi_{i,k}}[\|\nabla_x F_i(x_{i,k},\xi_{i,k}) - \nabla f_i(x_{i,k}) + \nabla f_i(x_{i,k})\|^2] + 0.5p^2L_f^2\delta_{i,k}^2\\ &\le 4p\mathbf{E}_{\xi_{i,k}}[\|\nabla_x F_i(x_{i,k},\xi_{i,k}) - \nabla f_i(x_{i,k})\|^2 + \|\nabla f_i(x_{i,k})\|^2] + 0.5p^2L_f^2\delta_{i,k}^2\\ &\le 4p\|\nabla f_i(x_{i,k})\|^2 + 4p\sigma_1^2 + 0.5p^2L_f^2\delta_{i,k}^2,\end{aligned} \tag{5.85}$$

where the first inequality holds due to (5.84); the second inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds since Assumption 5.7 and $x_{i,k}$ is independent of $\xi_{i,k}$.

From Assumption 5.3, we have

$$\begin{aligned}\|\nabla f(x) - \nabla f(y)\|^2 &= \Big\|\frac{1}{n}\sum_{i=1}^n(\nabla f_i(x) - \nabla f_i(y))\Big\|^2\\ &\le \frac{1}{n}\sum_{i=1}^n\|\nabla f_i(x) - \nabla f_i(y)\|^2 \le L_f^2\|x-y\|^2,\ \forall x,y\in\mathbb{R}^p.\end{aligned} \tag{5.86}$$

Then, we have

$$\begin{aligned}\|\nabla f_i(x_{i,k})\|^2 &= \|\nabla f_i(x_{i,k}) - \nabla f(x_{i,k}) + \nabla f(x_{i,k}) - \nabla f(\bar{x}_k) + \nabla f(\bar{x}_k)\|^2\\ &\le 3(\|\nabla f_i(x_{i,k}) - \nabla f(x_{i,k})\|^2 + \|\nabla f(x_{i,k}) - \nabla f(\bar{x}_k)\|^2 + \|\nabla f(\bar{x}_k)\|^2)\\ &\le 3(\sigma_2^2 + L_f^2\|x_{i,k} - \bar{x}_k\|^2 + \|\nabla f(\bar{x}_k)\|^2),\end{aligned} \tag{5.87}$$

where the first inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds due to Assumption 5.8 and (5.86).

From (5.85) and (5.87), we know (5.76a) holds.

(ix) From the Cauchy-Schwarz inequality, we have

$$\|\boldsymbol{g}^0_{k+1}\|^2 = \|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k + \boldsymbol{g}^0_k - \bar{\boldsymbol{g}}^0_k + \bar{\boldsymbol{g}}^0_k\|^2 \le 3(\|\boldsymbol{g}^0_{k+1} - \boldsymbol{g}^0_k\|^2 + \|\boldsymbol{g}^0_k - \bar{\boldsymbol{g}}^0_k\|^2 + \|\bar{\boldsymbol{g}}^0_k\|^2). \tag{5.88}$$

From Assumption 5.8, we have

$$\|\boldsymbol{g}^0_k - \bar{\boldsymbol{g}}^0_k\|^2 = \sum_{i=1}^n\|f_i(\bar{x}_k) - f(\bar{x}_k)\|^2 \le n\sigma_2^2. \tag{5.89}$$

From (5.88), (5.89), and (5.75f), we know (5.76b) holds. □

Lemma 5.3. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Suppose $\{\beta_k\}$ is nondecreasing, $\alpha_k = \kappa_1\beta_k$, and $\eta_k = \frac{\kappa_2}{\beta_k}$, where $\kappa_1 > 1$ and $\kappa_2 > 0$ are constants. Moreover, suppose $\beta_k \geq \varepsilon_1$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.2, then*

$$\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \leq W_k - \|\boldsymbol{x}_k\|^2_{(2\varepsilon_4-\varepsilon_5\omega_k-b_{1,k})\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{b_{2,k}\boldsymbol{K}} + n\sigma_2^2(b_{3,k} + 6pb_{4,k})\eta_k^2 + 2pn\sigma_1^2 b_{4,k}\eta_k^2 - \eta_k(0.25 - (b_{3,k} + 6pb_{4,k})\eta_k)\|\bar{\boldsymbol{g}}^0_k\|^2 + b_{5,k}\eta_k\delta_k^2, \tag{5.90a}$$

$$\mathbf{E}_{\mathfrak{L}_k}[\breve{W}_{k+1}] \leq \breve{W}_k - \|\boldsymbol{x}_k\|^2_{(2\varepsilon_4-\varepsilon_5\omega_k-b_{1,k})\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{b_{2,k}\boldsymbol{K}} + (b_{3,k} + 6pb_{4,k})\eta_k^2\|\bar{\boldsymbol{g}}^0_k\|^2 + 2pn\sigma_1^2 b_{4,k}\eta_k^2 + n\sigma_2^2(b_{3,k} + 6pb_{4,k})\eta_k^2 + b_{5,k}\eta_k\delta_k^2, \tag{5.90b}$$

where

$$W_k = \sum_{i=1}^{4} W_{i,k},\ \breve{W}_k = \sum_{i=1}^{3} W_{i,k},\ W_{1,k} = \frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}},\ W_{2,k} = \frac{1}{2}\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{Q}+\kappa_1\boldsymbol{K}},$$

$$W_{3,k} = \boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big),\ W_{4,k} = n(f(\bar{x}_k) - f^*) = \tilde{f}(\bar{\boldsymbol{x}}_k) - f^*,$$

$$b_{1,k} = 6p\kappa_3L_f^4\frac{\eta_k}{\beta_k^2} + 12p\kappa_5L_f^4\frac{\eta_k^2}{\beta_k^2} + (0.5 + L_f^2)\eta_k\omega_k + 6p\kappa_4L_f^4\frac{\eta_k\omega_k}{\beta_k^2} + (5 + 24p + 18p\kappa_3L_f^2)L_f^2\eta_k^2\omega_k + 12p\kappa_4L_f^4\frac{\eta_k^2\omega_k}{\beta_k^2} + 18p\kappa_4L_f^4\eta_k^2\omega_k^2,$$

$$b_{2,k} = 2\varepsilon_6 - 0.5\omega_k(\kappa_1 + \kappa_4 + \kappa_1\kappa_2 + 3\kappa_2^2) - 0.5\omega_k\eta_k\kappa_4,\ b_{3,k} = \frac{3}{2}\kappa_3\frac{\omega_k}{\eta_k^2} + \frac{3}{2}\kappa_4\frac{\omega_k^2}{\eta_k^2},$$

$$b_{4,k} = 6 + L_f + \frac{\kappa_3}{\kappa_2}L_f^2\frac{1}{\beta_k} + (4 + 3\kappa_3L_f^2)\omega_k + 3\kappa_4L_f^2\omega_k^2 + 2\kappa_5L_f^2\frac{1}{\beta_k^2} + \frac{\kappa_4}{\kappa_2}L_f^2\frac{\omega_k}{\beta_k} + 2\kappa_4L_f^2\frac{\omega_k}{\beta_k^2},$$

$$b_{5,k} = nL_f^2(0.25p^2b_{4,k}\eta_k + 3 + \omega_k + 8\eta_k + 5\eta_k\omega_k).$$

Proof. Note that $W_{4,k}$ is well defined due to $f^* > -\infty$ as assumed in Assumption 5.2. Thus, W_k is well defined.

(i) Similar to the way to get (4.23), from (5.80a), (5.75a), and that $x_{i,k}$ and $v_{i,k}$ are independent of $\mathfrak{L}_k$, we have

$$\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1}] \leq W_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L}-\frac{1}{2}\eta_k\boldsymbol{K}-\frac{3}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2} + \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\frac{3}{2}\eta_k^2\beta_k^2\boldsymbol{K}} + \eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}^e_k - \boldsymbol{g}^0_k\|^2] + \frac{1}{2}\eta_k(1 + \eta_k)\|\boldsymbol{g}^s_k - \boldsymbol{g}^0_k\|^2 - \eta_k\beta_k\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big). \tag{5.91}$$

Then, from (5.91), (5.75b), and (5.75e), we have

$$\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1}] \leq W_{1,k} - \|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L}-\frac{1}{2}\eta_k\boldsymbol{K}-\frac{3}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2-\eta_k(1+5\eta_k)L_f^2\boldsymbol{K}} - \eta_k\beta_k\boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big) + \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\frac{3}{2}\eta_k^2\beta_k^2\boldsymbol{K}} + nL_f^2\eta_k(1 + 5\eta_k)\delta_k^2 + 2\eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}^e_k\|^2]. \tag{5.92}$$

(ii) Similar to the way to get (4.24), from (5.80b) and (5.75f), we have

$$
\begin{aligned}
W_{2,k+1} \le W_{2,k} &+ (1+\omega_k)\eta_k\beta_k \boldsymbol{x}_k^\top(\boldsymbol{K}+\kappa_1\boldsymbol{L})\Big(\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) + \frac{1}{2}\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)(\omega_k+\omega_k^2)\|\boldsymbol{g}_{k+1}^0\|^2 \\
&+ \frac{1}{2}(\eta_k+\omega_k+\eta_k\omega_k)\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|_{\boldsymbol{K}}^2 + \|\boldsymbol{x}_k\|^2_{(1+\omega_k)\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)} \\
&+ \frac{\eta_k}{\beta_k^2}\Big(\eta_k+\frac{1}{2}\Big)(1+\omega_k)\Big(\frac{1}{\rho_2(L)}+\kappa_1\Big)L_f^2\|\bar{\boldsymbol{g}}_k^e\|^2. \qquad (5.93)
\end{aligned}
$$

(iii) Similar to the way to get (4.35), from (5.80), (5.75a), and that $x_{i,k}$ and $v_{i,k}$ are independent of $\mathfrak{L}_k$, we have

$$
\begin{aligned}
\mathbf{E}_{\mathfrak{L}_k}\Big[\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}_{k+1}^0\Big)\Big] \le{}& \boldsymbol{x}_k^\top(\boldsymbol{K}-\eta_k\alpha_k\boldsymbol{L})\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) - \Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k-\frac{1}{2}-\eta_k\beta_k^2)\boldsymbol{K}} \\
&+ \|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta_k^2(\frac{1}{2}\alpha_k^2-\alpha_k\beta_k+\beta_k^2)\boldsymbol{L}^2} \\
&+ \frac{1}{2}(\eta_k+\eta_k^2)\|\boldsymbol{g}_k^s-\boldsymbol{g}_k^0\|^2 + \frac{1}{2}\eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e-\boldsymbol{g}_k^0\|^2] \\
&+ \Big(\frac{1}{2\eta_k\beta_k^2}+\frac{3}{2\beta_k^2}\Big)\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2]. \qquad (5.94)
\end{aligned}
$$

Then, from (5.94), (5.75b), (5.75e), and (5.75f), we have

$$
\begin{aligned}
\mathbf{E}_{\mathfrak{L}_k}\Big[\boldsymbol{x}_{k+1}^\top\boldsymbol{K}\Big(\boldsymbol{v}_{k+1}+\frac{1}{\beta_k}\boldsymbol{g}_{k+1}^0\Big)\Big] \le{}& \boldsymbol{x}_k^\top\boldsymbol{K}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) - (1+\omega_k)\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) \\
&+ \omega_k\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) - \Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k-\frac{1}{2}-\eta_k\beta_k^2)\boldsymbol{K}} \\
&+ \|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta_k^2(\frac{1}{2}\alpha_k^2-\alpha_k\beta_k+\beta_k^2)\boldsymbol{L}^2+\eta_k(1+3\eta_k)L_f^2\boldsymbol{K}} \\
&+ \frac{\eta_k}{2\beta_k^2}(1+3\eta_k)L_f^2\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] + nL_f^2\eta_k(1+3\eta_k)\delta_k^2 \\
&+ \eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2]. \qquad (5.95)
\end{aligned}
$$

Then, from (5.95), (4.34), and (4.37), we have

$$
\begin{aligned}
\mathbf{E}_{\mathfrak{L}_k}[W_{3,k+1}] \le W_{3,k} &- (1+\omega_k)\eta_k\alpha_k\boldsymbol{x}_k^\top\boldsymbol{L}\Big(\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big) + \frac{\eta_k}{2\beta_k^2}(1+3\eta_k)L_f^2\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] \\
&+ \|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L}+\frac{1}{2}\boldsymbol{K})+\eta_k^2(\frac{1}{2}\alpha_k^2-\alpha_k\beta_k+\beta_k^2)\boldsymbol{L}^2+\frac{1}{2}\omega_k\eta_k\alpha_k\boldsymbol{L}^2+\eta_k(1+3\eta_k)L_f^2\boldsymbol{K}} \\
&+ nL_f^2\eta_k(1+3\eta_k)\delta_k^2 + \eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] \\
&- \Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k-\frac{1}{2}-\eta_k\beta_k^2-\frac{1}{2}\omega_k\alpha_k)\boldsymbol{K}} + \frac{1}{2}\omega_k\mathbf{E}_{\mathfrak{L}_k}[2W_{1,k+1}+\|\boldsymbol{g}_{k+1}^0\|^2]. \qquad (5.96)
\end{aligned}
$$

(iv) Similar to the way to get (4.41), from (5.82), (5.75a), and that $x_{i,k}$ and $v_{i,k}$ are independent of $\mathfrak{L}_k$, we have

$$
\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \le W_{4,k} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^s\|^2 + \frac{1}{2}\eta_k\|\bar{\boldsymbol{g}}_k^0-\bar{\boldsymbol{g}}_k^s\|^2 - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + \frac{1}{2}\eta_k^2L_f\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2]. \quad (5.97)
$$

Then, from (5.97) and (5.75c), we have

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \le W_{4,k} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^s\|^2 + \|\boldsymbol{x}_k\|^2_{\eta_k L_f^2 \boldsymbol{K}} + nL_f^2\eta_k\delta_k^2 - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + \frac{1}{2}\eta_k^2 L_f \mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2]. \tag{5.98}$$

(v) We have

$$\begin{aligned}
\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \le{}& W_k + \frac{1}{2}\omega_k\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - (1+\omega_k)\|\boldsymbol{x}_k\|^2_{\eta_k\alpha_k\boldsymbol{L} - \frac{1}{2}\eta_k\boldsymbol{K} - \frac{3}{2}\eta_k^2\alpha_k^2\boldsymbol{L}^2 - \eta_k(1+5\eta_k)L_f^2\boldsymbol{K}} \\
&+ (1+\omega_k)\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\frac{3}{2}\eta_k^2\beta_k^2\boldsymbol{K}} + (1+\omega_k)nL_f^2\eta_k(1+5\eta_k)\delta_k^2 \\
&+ 2(1+\omega_k)\eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] + \frac{1}{2}\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)(\omega_k + \omega_k^2)\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_{k+1}^0\|^2] \\
&+ \frac{1}{2}(\eta_k + \omega_k + \eta_k\omega_k)\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + \|\boldsymbol{x}_k\|^2_{(1+\omega_k)\eta_k^2\beta_k^2(\boldsymbol{L}+\kappa_1\boldsymbol{L}^2)} \\
&+ \frac{\eta_k}{\beta_k^2}\Big(\eta_k + \frac{1}{2}\Big)(1+\omega_k)\Big(\frac{1}{\rho_2(L)} + \kappa_1\Big)L_f^2\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] + \frac{\eta_k}{2\beta_k^2}(1+3\eta_k)L_f^2\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] \\
&+ \|\boldsymbol{x}_k\|^2_{\eta_k(\beta_k\boldsymbol{L} + \frac{1}{2}\boldsymbol{K}) + \eta_k^2(\frac{1}{2}\alpha_k^2 - \alpha_k\beta_k + \beta_k^2)\boldsymbol{L}^2 + \frac{1}{2}\omega_k\eta_k\alpha_k\boldsymbol{L}^2 + \eta_k(1+3\eta_k)L_f^2\boldsymbol{K}} + nL_f^2\eta_k(1+3\eta_k)\delta_k^2 \\
&+ \eta_k^2\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2] - \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\eta_k(\beta_k - \frac{1}{2} - \eta_k\beta_k^2 - \frac{1}{2}\omega_k\alpha_k)\boldsymbol{K}} + \frac{1}{2}\omega_k\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_{k+1}^0\|^2] \\
&- \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^s\|^2 + \|\boldsymbol{x}_k\|^2_{\eta_k L_f^2\boldsymbol{K}} + nL_f^2\eta_k\delta_k^2 - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + \frac{1}{2}\eta_k^2 L_f\mathbf{E}_{\mathfrak{L}_k}[\|\bar{\boldsymbol{g}}_k^e\|^2] \\
\le{}& W_k - \|\boldsymbol{x}_k\|^2_{\eta_k\boldsymbol{M}_{1,k} - \eta_k^2\boldsymbol{M}_{2,k} - \omega_k\boldsymbol{M}_3 - b_{1,k}\boldsymbol{K}} - \eta_k\Big(\frac{1}{4} - (b_{3,k} + 6pb_{4,k})\eta_k\Big)\|\bar{\boldsymbol{g}}_k^0\|^2 \\
&- \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{b_{2,k}^0\boldsymbol{K}} + 2pn\sigma_1^2 b_{4,k}\eta_k^2 + n\sigma_2^2(b_{3,k} + 6pb_{4,k})\eta_k^2 + b_{5,k}\eta_k\delta_k^2,
\end{aligned} \tag{5.99}$$

where the first inequality holds due to (5.92), (5.93), (5.96), (5.98), and $\alpha_k = \kappa_1\beta_k$; the last inequality holds due to (5.76a), (5.76b), $\alpha_k = \kappa_1\beta_k$, $\eta_k = \frac{\kappa_2}{\beta_k}$, and

$$\begin{aligned}
\boldsymbol{M}_{1,k} &= (\alpha_k - \beta_k)\boldsymbol{L} - (1 + 3L_f^2)\boldsymbol{K}, \\
\boldsymbol{M}_{2,k} &= \beta_k^2\boldsymbol{L} + (2\alpha_k^2 + \beta_k^2)\boldsymbol{L}^2 + 8L_f^2\boldsymbol{K} + 12p(3 + 0.5L_f)L_f^2\boldsymbol{K}, \\
\boldsymbol{M}_3 &= 0.5\boldsymbol{K} - \kappa_1\kappa_2\boldsymbol{L} + 0.5\kappa_1\kappa_2\boldsymbol{L}^2 + 1.5\kappa_1^2\kappa_2^2\boldsymbol{L}^2 + \kappa_2^2(\boldsymbol{L} + \kappa_1\boldsymbol{L}^2), \\
b_{2,k}^0 &= 0.5\eta_k(2\beta_k - \kappa_3) - 2.5\kappa_2^2 - 0.5\omega_k(\kappa_1\kappa_2 + 3\kappa_2^2 + \kappa_4) - 0.5\omega_k\eta_k\kappa_4.
\end{aligned}$$

From (2.6), $\alpha_k = \kappa_1\beta_k$, $\kappa_1 > 1$, $\beta_k \ge \varepsilon_1 \ge 1 + 3L_f^2$, and $\eta_k = \frac{\kappa_2}{\beta_k}$, we have

$$\eta_k\boldsymbol{M}_{1,k} \ge \varepsilon_2\kappa_2\boldsymbol{K}. \tag{5.100}$$

From (2.6), $\alpha_k = \kappa_1\beta_k$, $\beta_k \ge \varepsilon_1 \ge (8 + 12p(3 + 0.5L_f))^{1/2}L_f$, and $\eta_k = \frac{\kappa_2}{\beta_k}$, we have

$$\eta_k^2\boldsymbol{M}_{2,k} \le \varepsilon_3\kappa_2^2\boldsymbol{K}. \tag{5.101}$$

From (2.6), $\alpha_k = \kappa_1\beta_k$, and $\eta_k = \frac{\kappa_2}{\beta_k}$, we have

$$M_3 \leq \varepsilon_5 K. \tag{5.102}$$

From $\beta_k \geq \varepsilon_1 \geq p\kappa_3 \geq \kappa_3$ and $\eta_k = \frac{\kappa_2}{\beta_k}$, we have

$$b^0_{2,k} \geq b_{2,k}. \tag{5.103}$$

From (5.99)–(5.103), we know that (5.90a) holds.

Similar to the way to get (5.90a), we have (5.90b). □

Lemma 5.4. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Suppose $\alpha_k = \kappa_1\beta_k$, $\beta_k = \kappa_0(k+t_1)^\theta$, and $\eta_k = \frac{\kappa_2}{\beta_k}$, where $\theta \in [0,1]$, $\kappa_0 \geq c_0(\kappa_1,\kappa_2)/t_1^\theta$, $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, and $t_1 \geq (c_3(\kappa_1,\kappa_2))^{1/\theta}$. Let $\{x_k\}$ be the sequence generated by Algorithm 5.2, then*

$$\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \leq W_k - \varepsilon_4\|x_k\|^2_K - \varepsilon_6\Big\|v_k + \frac{1}{\beta_k}g^0_k\Big\|^2_K - \frac{1}{16}\eta_k\|\bar{g}^0_k\|^2 + pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2, \tag{5.104a}$$

$$\mathbf{E}_{\mathfrak{L}_k}[\breve{W}_{k+1}] \leq \breve{W}_k - \varepsilon_4\|x_k\|^2_K - \varepsilon_6\Big\|v_k + \frac{1}{\beta_k}g^0_k\Big\|^2_K + p\varepsilon_{13}\eta_k^2\|\bar{g}^0_k\|^2 + pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2, \tag{5.104b}$$

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \leq W_{4,k} + \|x_k\|^2_{2\eta_k L_f^2 K} - \frac{3}{16}\eta_k\|\bar{g}^0_k\|^2 + 2p\eta_k^2 L_f(\sigma_1^2 + 3\sigma_2^2) + (n+p)L_f^2\eta_k\delta_k^2. \tag{5.104c}$$

Proof. **(i)** Noting that $\kappa_1 > c_1 > 1$ and $\beta_k = \kappa_0(k+t_1)^\theta \geq \kappa_0 t_1^\theta \geq c_0(\kappa_1,\kappa_2,t_1,\theta) \geq \varepsilon_1 \geq 1$, we know that all conditions needed in Lemma 5.3 are satisfied, so (5.90a) and (5.90b) hold.

From $\kappa_1 > c_1 = \frac{1}{\rho_2(L)} + 1$, we have

$$\varepsilon_2 > 0. \tag{5.105}$$

From (5.105) and $\kappa_2 \in (0, \min\{\frac{\varepsilon_2}{\varepsilon_3}, \frac{1}{5}\})$, we have

$$\varepsilon_4 > 0,\ \varepsilon_6 > 0. \tag{5.106}$$

From $t_1 \geq (c_3(\kappa_1,\kappa_2))^{1/\theta}$ and $c_3(\kappa_1,\kappa_2) = \frac{24\kappa_3}{\kappa_2}$, we have

$$\frac{3\kappa_3}{2\kappa_2 t_1^\theta} \leq \frac{1}{16}. \tag{5.107}$$

From $\kappa_0 \geq \frac{24\kappa_4}{\kappa_2 t_1^\theta} \geq \frac{24\kappa_4}{\kappa_2 t_1^{3\theta}}$, we have

$$\frac{3\kappa_4}{2\kappa_2\kappa_0 t_1^{3\theta}} \leq \frac{1}{16}. \tag{5.108}$$

From $\beta_k = \kappa_0(k+t_1)^\theta$, we have

$$\omega_k = \frac{1}{\beta_k} - \frac{1}{\beta_{k+1}} = \frac{1}{\kappa_0}\Big(\frac{1}{(k+t_1)^\theta} - \frac{1}{(k+t_1+1)^\theta}\Big) \le \frac{1}{\kappa_0(k+t_1)^\theta(k+t_1+1)^\theta} \le \frac{\kappa_0}{\beta_k^2} \le 1. \tag{5.109}$$

From (5.109), $\eta_k = \frac{\kappa_2}{\beta_k}$, $\beta_k \ge 1$, $\omega_k \le 1$, and $\kappa_0 \ge (\frac{2p\varepsilon_7}{\varepsilon_4 t_1^{2\theta}})^{\frac{1}{2}}$, we have

$$b_{1,k} \le \frac{p\varepsilon_7}{\kappa_0^2 t_1^{2\theta}} \le \frac{\varepsilon_4}{2}. \tag{5.110}$$

From (5.109), (5.110), $\kappa_0 \ge \frac{2\varepsilon_5}{\varepsilon_4 t_1^\theta}$, and (5.106), we have

$$2\varepsilon_4 - \varepsilon_5\omega_k - b_{1,k} \ge 2\varepsilon_4 - \frac{\varepsilon_5}{\kappa_0 t_1^\theta} - \frac{\varepsilon_4}{2} \ge \varepsilon_4 > 0. \tag{5.111}$$

From (5.109), $\eta_k = \frac{\kappa_2}{\beta_k}$, $\kappa_0 \ge \frac{\varepsilon_8}{2\varepsilon_6 t_1^\theta} \ge \frac{\varepsilon_8}{2\varepsilon_6 t_1^{2\theta}}$, and (5.106), we have

$$b_{2,k} \ge 2\varepsilon_6 - \frac{\varepsilon_8}{2\kappa_0 t_1^{2\theta}} \ge \varepsilon_6 > 0. \tag{5.112}$$

From (5.107)–(5.109) and $\eta_k = \frac{\kappa_2}{\beta_k}$, we have

$$b_{3,k}\eta_k \le \frac{3\kappa_3}{2\kappa_2\kappa_0 t_1^{3\theta}} + \frac{3\kappa_4}{2\kappa_2\kappa_0 t_1^{3\theta}} \le \frac{1}{8}. \tag{5.113}$$

From $\beta_k \ge 1$ and $\omega_k \le 1$, we have

$$b_{3,k} \le \varepsilon_9, \tag{5.114a}$$
$$b_{4,k} \le \varepsilon_{10}. \tag{5.114b}$$

From (5.113), (5.114b), and $\kappa_0 \ge \frac{96p\kappa_2\varepsilon_{10}}{t_1^\theta}$, we have

$$\frac{1}{4} - (b_{3,k} + 6pb_{4,k})\eta_k \ge \frac{1}{8} - 6pb_{4,k}\eta_k \ge \frac{1}{8} - \frac{6p\kappa_2\varepsilon_{10}}{\kappa_0 t_1^\theta} \ge \frac{1}{16}. \tag{5.115}$$

From (5.115), $\eta_k = \frac{\kappa_2}{\beta_k}$, $\beta_k \ge 1$, and $\omega_k \le 1$, we have

$$b_{5,k} \le pn\varepsilon_{11}. \tag{5.116}$$

From (5.90a), (5.111), (5.112), and (5.114a)–(5.116), we know that (5.104a) holds.
(ii) From (5.90b), (5.111), (5.112), (5.114a), (5.114b), and (5.116), we have (5.104b).
(iii) From (5.98), (5.75d), and (5.76a), we have

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \le W_{4,k} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^s\|^2 + \|\boldsymbol{x}_k\|^2_{\eta_k L_f^2 \boldsymbol{K}} + nL_f^2\eta_k\delta_k^2 - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + \frac{1}{2}\eta_k^2 L_f\Big(\frac{12p}{n}\|\bar{\boldsymbol{g}}_k^0\|^2$$

$$+\frac{12p}{n}L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+4p\sigma_1^2+12p\sigma_2^2+\frac{1}{2}p^2L_f^2\delta_k^2+\|\bar{\boldsymbol{g}}_k^s\|^2\Big). \tag{5.117}$$

From $\kappa_0 t_1^\theta \ge c_0(\kappa_1,\kappa_2)\ge 96p\kappa_2\varepsilon_{10} > 96p\kappa_2 L_f$, we have

$$\frac{6p}{n}\eta_k^2L_f \le \frac{6}{\kappa_0 t_1^\theta}p\eta_k L_f\kappa_2 < \frac{1}{16}\eta_k, \tag{5.118}$$

$$\frac{6p}{n}\eta_k^2L_f^3 < \frac{1}{16}\eta_kL_f^2,\ \frac{1}{2}\eta_k^2L_f < \frac{1}{16}\eta_k,\ \frac{1}{4}p^2\eta_k^2L_f^3 < pL_f^2\eta_k. \tag{5.119}$$

From (5.117)–(5.118), we have (5.104c). □

Now it is ready to prove Theorem 5.3.
Denote

$$\hat{V}_k = \|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} + n(f(\bar{x}_k)-f^*).$$

Similar to the way to get (3.46)–(3.48), we have

$$W_k \ge \kappa_7\Big(\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}\Big) + n(f(\bar{x}_k)-f^*) \tag{5.120}$$

$$\ge \kappa_7\hat{V}_k \ge 0, \tag{5.121}$$

and

$$W_k \le \kappa_6\hat{V}_k. \tag{5.122}$$

From (5.104a) and (5.106), we have

$$\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \le W_k - \varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \frac{1}{16}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2. \tag{5.123}$$

Then, taking expectation in $\mathcal{L}_T$, summing (5.123) over $k\in[0,T]$, and using (2.37) and $\eta_k=\frac{\kappa_2}{\kappa_0(k+t_1)^\theta}$ and $\delta_k\le\kappa_\delta\sqrt{\eta_k}$ as stated in (5.10), yield

$$\mathbf{E}[W_{T+1}] + \sum_{k=0}^{T}\mathbf{E}\Big[\varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{16}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2\Big] \le W_0 + \frac{pn(\varepsilon_{11}\kappa_\delta^2+\varepsilon_{12})\kappa_2^2}{\kappa_0^2}\sum_{k=0}^{T}\frac{1}{(k+t_1)^{2\theta}} \le n\varepsilon_{14}. \tag{5.124}$$

Noting that $t_1^\theta = \mathcal{O}(\sqrt{p})$, we have

$$\kappa_0 = \mathcal{O}(\frac{p}{t_1^\theta}) = \mathcal{O}(\sqrt{p}). \tag{5.125}$$

From $W_0=\mathcal{O}(n)$ and (5.125), we have

$$\varepsilon_{14} = \frac{W_0}{n} + \frac{2\theta p(\varepsilon_{11}\kappa_\delta^2+\varepsilon_{12})\kappa_2^2}{(2\theta-1)\kappa_0^2} = \mathcal{O}(1). \tag{5.126}$$

From (5.124), (5.121), (5.106), and $\sum_{k=0}^{T}\eta_k=\sum_{k=0}^{T}\frac{\kappa_2}{\kappa_0(k+t_1)^\theta}\ge\frac{\kappa_2(T+t_1)^{1-\theta}}{\kappa_0(1-\theta)}$, we have

$$\frac{\sum_{k=0}^{T}\eta_k\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2]}{\sum_{k=0}^{T}\eta_k}=\frac{\sum_{k=0}^{T}\eta_k\mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2]}{n\sum_{k=0}^{T}\eta_k}\le\frac{16\kappa_0(1-\theta)\varepsilon_{14}}{\kappa_2(T+t_1)^{1-\theta}}. \tag{5.127}$$

From (5.127), (5.126), and (5.125), we have (5.11a).

From (5.124), (5.120), and (5.106), we have

$$\mathbf{E}[f(\bar{x}_{T+1})]-f^*=\frac{1}{n}W_{T+1}\le\varepsilon_{14},\ \forall T\in\mathbb{N}_0, \tag{5.128}$$

which gives (5.11b).

From (5.124), (5.121), and (5.106), we have

$$\sum_{k=0}^{T}\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}]\le\frac{n\varepsilon_{14}}{\varepsilon_4},\ \forall T\in\mathbb{N}_0. \tag{5.129}$$

From (5.75g) and (5.128), we have

$$\|\bar{\boldsymbol{g}}_k^0\|^2\le 2nL_f(f(\bar{x}_k)-f^*)\le 2nL_f\varepsilon_{14}. \tag{5.130}$$

From (5.76a), (5.129), and (5.130), we know that $\mathbf{E}[\|\boldsymbol{g}_k^e\|^2]$ is bounded. Then, same as the proof of the first part of Theorem 1 in [151], we have (5.11d).

From (5.120) and (5.122), we have

$$0\le 2\kappa_7(W_{1,k}+W_{2,k})\le\breve{W}_k\le 2\kappa_6(W_{1,k}+W_{2,k}). \tag{5.131}$$

Denote $\breve{z}_k=\mathbf{E}[\breve{W}_k]$. From (5.104b), (5.130), (5.131), and (5.10), we have

$$\breve{z}_{k+1}\le(1-a_1)\breve{z}_k+\frac{a_2}{(t+t_1)^{2\theta}}. \tag{5.132}$$

From $\kappa_1>1$, we have $\kappa_6>1$. From $0<\kappa_2<\frac{1}{5}$, we have $\varepsilon_6=\frac{1}{4}(\kappa_2-5\kappa_2^2)\le\frac{1}{80}$. Thus,

$$a_1\le\frac{\varepsilon_6}{\kappa_6}\le\frac{1}{80}. \tag{5.133}$$

From (5.106), we know that

$$a_1>0,\ a_2>0. \tag{5.134}$$

From (5.132)–(5.134) and (2.45), we have

$$\breve{z}_k\le\phi_3(k,t_1,a_1,a_2,2\theta,\breve{z}_0),\ \forall k\in\mathbb{N}_+, \tag{5.135}$$

where the function ϕ_3 is defined in (2.46).

Noting that $\phi_3(k,t_1,a_1,a_2,2\theta,\breve{z}_0)=O(n/k^{2\theta})$, from (5.135) and (5.131), we have (5.11c).

5.8.4 Proof of Theorem 5.4

In addition to the notations defined in Appendix 5.8.3, we also denote the following notations.

$$\tilde{c}_0(\kappa_1,\kappa_2) = \max\Big\{\varepsilon_1,\ \Big(\frac{p\tilde{\varepsilon}_7}{\varepsilon_4}\Big)^{\frac{1}{3}},\ 48p\kappa_2\tilde{\varepsilon}_{10}\Big\},\ \tilde{\varepsilon}_7 = 6(1+3\kappa_2+\kappa_4+2\kappa_2\kappa_4)\kappa_2 L_f^4,$$
$$\tilde{\varepsilon}_{10} = 6 + L_f + \frac{1}{\kappa_2}(\kappa_4+1)L_f^2 + (3\kappa_4+3)L_f^2,\ \tilde{\varepsilon}_{11} = L_f^2\Big(\frac{1}{192}+\frac{1}{p}(8\kappa_2+3)\Big),$$
$$\tilde{\varepsilon}_{12} = 2(\sigma_1^2+3\sigma_2^2)\tilde{\varepsilon}_{10},\ \varepsilon_{15} = 2(\sigma_1^2+3\sigma_2^2)L_f,\ \varepsilon_{16} = 2L_f^2\kappa_\delta^2.$$

To prove Theorem 5.4, the following lemma is used.

Lemma 5.5. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Suppose* $\alpha_k = \alpha = \kappa_1\beta$, $\beta_k = \beta$, *and* $\eta_k = \eta = \frac{\kappa_2}{\beta}$, *where* $\beta \ge \tilde{c}_0(\kappa_1,\kappa_2)$, $\kappa_1 > c_1$, *and* $\kappa_2 \in (0, c_2(\kappa_2))$ *are constants. Let* $\{\boldsymbol{x}_k\}$ *be the sequence generated by Algorithm 5.2, then*

$$\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \le W_k - \varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - 2\varepsilon_6\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} - \frac{1}{8}\eta\|\bar{\boldsymbol{g}}_k^0\|^2 + pn\tilde{\varepsilon}_{12}\eta^2 + pn\tilde{\varepsilon}_{11}\eta\delta_k^2, \tag{5.136a}$$

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \le W_{4,k} + \|\boldsymbol{x}_k\|^2_{2\eta L_f^2\boldsymbol{K}} - \frac{1}{8}\eta\|\bar{\boldsymbol{g}}_k^0\|^2 + 2p\eta^2 L_f(\sigma_1^2+3\sigma_2^2) + (n+p)L_f^2\eta\delta_k^2. \tag{5.136b}$$

Proof. **(i)** Substituting $\alpha_k = \alpha = \kappa_1\beta, \beta_k = \beta, \eta_k = \eta = \frac{\kappa_2}{\beta}$, and $\omega_k = 0$ into (5.92), (5.93), (5.96), and (5.98), similar to the way to get (5.99), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \le{}& W_k - \|\boldsymbol{x}_k\|^2_{\eta\tilde{\boldsymbol{M}}_1-\eta^2\tilde{\boldsymbol{M}}_2-\tilde{b}_1\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\tilde{b}_2^0\boldsymbol{K}} - \eta\Big(\frac{1}{4}-6p\tilde{b}_4\eta\Big)\|\bar{\boldsymbol{g}}_k^0\|^2\\ &+ 2pn(\sigma_1^2+3\sigma_2^2)\tilde{b}_4\eta^2 + \tilde{b}_5\eta\delta_k^2,\end{aligned} \tag{5.137}$$

where

$$\tilde{\boldsymbol{M}}_1 = (\alpha-\beta)\boldsymbol{L} - (1+3L_f^2)\boldsymbol{K},\ \tilde{\boldsymbol{M}}_2 = \beta^2\boldsymbol{L} + (2\alpha^2+\beta^2)\boldsymbol{L}^2 + 8L_f^2\boldsymbol{K} + (3+0.5L_f)\frac{12p}{n}L_f^2\boldsymbol{K},$$
$$\tilde{b}_1 = \frac{6p}{n}\kappa_3 L_f^4\frac{\eta}{\beta^2} + \frac{12p}{n}\kappa_5 L_f^4\frac{\eta^2}{\beta^2},\ \tilde{b}_2^0 = \frac{1}{2}\eta(2\beta-\kappa_3) - \frac{5}{2}\kappa_2^2,$$
$$\tilde{b}_4 = 6 + L_f + \frac{\kappa_3}{\kappa_2}L_f^2\frac{1}{\beta} + 2\kappa_5 L_f^2\frac{1}{\beta^2},\ \tilde{b}_5 = nL_f^2\Big(\frac{1}{4}p^2\tilde{b}_4\eta + 3 + 8\eta\Big).$$

From (5.137), similar to the way to get (5.90a), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \le{}& W_k - \|\boldsymbol{x}_k\|^2_{(2\varepsilon_4-\tilde{b}_1)\boldsymbol{K}} - \Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{2\varepsilon_6\boldsymbol{K}} - \eta\Big(\frac{1}{4}-6p\tilde{b}_4\eta\Big)\|\bar{\boldsymbol{g}}_k^0\|^2\\ &+ 2pn(\sigma_1^2+3\sigma_2^2)\tilde{b}_4\eta^2 + \tilde{b}_5\eta\delta_k^2.\end{aligned} \tag{5.138}$$

From (5.138), similar to the way to get (5.104a), we have (5.136a).
(ii) Noting $\eta_k = \eta$, $\beta \ge 48p\kappa_2\tilde{\varepsilon}_{10} \ge 48p\kappa_2 L_f$, and $\eta = \kappa_2/\beta$, similar to the way to get (5.104c), we have (5.136b). □

We are now ready to prove Theorem 5.4.

From $\beta_k = \beta = \kappa_2 \sqrt{pT}/\sqrt{n}$ and $T > n(\tilde{c}_0(\kappa_1,\kappa_2)/\kappa_2)^2/p$, we have $\beta \geq \tilde{c}_0(\kappa_1,\kappa_2)$. Thus, all conditions needed in Lemma 5.5 are satisfied. So (5.136a) and (5.136b) hold.

From (5.136a) and (5.12), similar to the way to get (5.129), we have

$$\frac{1}{T+1}\sum_{k=0}^{T}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big] \leq \frac{1}{\varepsilon_4}\Big(\frac{W_0}{n(T+1)} + \frac{n\tilde{\varepsilon}_{12}}{T} + \frac{2n\tilde{\varepsilon}_{11}\kappa_\delta^2}{\sqrt{T(T+1)}}\Big), \tag{5.139}$$

which gives (5.13c).

From (5.136b) and (5.12), similar to the way to get (5.127), we have

$$\begin{aligned}
&\frac{1}{T+1}\sum_{k=0}^{T}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \frac{1}{n(T+1)}\sum_{k=0}^{T}\mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2] \\
&\leq 8\Big(\frac{W_{4,0}}{n(T+1)\eta} + \frac{2L_f^2}{n(T+1)}\sum_{k=0}^{T}\mathbf{E}[\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2] + \frac{p\varepsilon_{15}\eta}{n} + \frac{\sqrt{p}\varepsilon_{16}}{\sqrt{n(T+1)}}\Big).
\end{aligned} \tag{5.140}$$

Noting that $\eta = \kappa_2/\beta_k = \sqrt{n}/\sqrt{pT}$, and $n/T < \sqrt{p}/\sqrt{nT}$ due to $T > n^3/p$, from (5.140) and (5.139), we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = 8(f(\bar{x}_0) - f^* + 2(\sigma_1^2 + 3\sigma_2^2)L_f + 2L_f^2\kappa_\delta^2)\frac{\sqrt{p}}{\sqrt{nT}} + O(\frac{n}{T}),$$

which gives (5.13a).

Taking expectation in $\mathcal{L}_T$, summing (5.136b) over $k \in [0,T]$, and using (5.12) yield

$$\begin{aligned}
&n(\mathbf{E}[f(\bar{x}_{T+1})] - f^*) = \mathbf{E}[W_{4,T+1}] \\
&\leq W_{4,0} + \frac{2\sqrt{n}}{\sqrt{pT}}L_f^2\sum_{k=0}^{T}\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2 + n\varepsilon_{15}\frac{T+1}{T} + n\varepsilon_{16}\sqrt{\frac{T+1}{T}}.
\end{aligned} \tag{5.141}$$

Noting that $W_{4,0} = O(n)$ and $\sqrt{n}n/\sqrt{pT} < 1$ due to $T > n^3/p$, from (5.139) and (5.141), we have (5.13b).

Similar to the proof of (5.11d), we have (5.13d).

5.8.5 Proof of Theorem 5.5

In addition to the notations defined in Appendix 5.8.3, we also denote the following notations.

$$\begin{aligned}
&\varepsilon_{17} = \frac{1}{\kappa_6}\min\Big\{\frac{\varepsilon_4\kappa_0 t_1^\theta}{\kappa_2}, \frac{\varepsilon_6\kappa_0 t_1^\theta}{\kappa_2}, \frac{\nu}{8}\Big\}, \ \varepsilon_{18} = \frac{32\theta 4^\theta L_f(\sigma_1^2 + 3\sigma_2^2)\kappa_2}{3\nu\kappa_0}, \\
&\breve{a}_2 = pn(\varepsilon_{11}\kappa_\delta^2 + \varepsilon_{12} + \varepsilon_{13}c_g)\frac{\kappa_2^2}{\kappa_0^2}, \ a_3 = \frac{\kappa_2\varepsilon_{17}}{\kappa_0}, \ a_4 = pn(\varepsilon_{11}\kappa_\delta^2 + \varepsilon_{12})\frac{\kappa_2^2}{\kappa_0^2}.
\end{aligned}$$

All conditions needed in Lemma 5.4 are satisfied, so (5.104a)–(5.104c) hold.

From (5.104a), (4.65), (5.121), and (5.122), we have

$$\begin{aligned}
\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] &\leq W_k - \varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \varepsilon_6\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}^0_k\Big\|^2_{\boldsymbol{K}} - \frac{\eta_k \nu n}{8}W_{4,k} + pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2 \\
&\leq W_k - \frac{\eta_k}{\kappa_6}\min\Big\{\frac{\varepsilon_4}{\eta_k},\ \frac{\varepsilon_6}{\eta_k},\ \frac{\nu}{8}\Big\}W_k + pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2 \\
&\leq W_k - \eta_k\varepsilon_{17}W_k + pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2,\ \forall k \in \mathbb{N}_0.
\end{aligned} \tag{5.142}$$

Denote $z_k = \mathbf{E}[W_k]$, $r_{1,k} = \eta_k\varepsilon_{17}$, and $r_{2,k} = pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2$. From (5.142), we have

$$z_{k+1} \leq (1 - r_{1,k})z_k + r_{2,k},\ \forall k \in \mathbb{N}_0. \tag{5.143}$$

From (5.14), we have

$$r_{1,k} = \eta_k\varepsilon_{17} = \frac{a_3}{(k+t_1)^\theta}, \tag{5.144}$$

$$r_{2,k} = pn\varepsilon_{12}\eta_k^2 + pn\varepsilon_{11}\eta_k\delta_k^2 \leq \frac{a_4}{(k+t_1)^{2\theta}}. \tag{5.145}$$

From $\kappa_1 > 1$, we have $\kappa_6 > 1$. From $0 < \kappa_2 < \frac{1}{5}$, we have $\varepsilon_6 = \frac{1}{4}(\kappa_2 - 5\kappa_2^2) \leq \frac{1}{80}$. Thus,

$$r_{1,k} \leq \frac{\varepsilon_6}{\kappa_6} \leq \frac{1}{80}. \tag{5.146}$$

From (5.106), we know that

$$a_3 > 0,\ a_4 > 0. \tag{5.147}$$

Then, from $\theta \in (0,1)$, (5.143)–(5.147), and (2.41), we have

$$z_k \leq \phi_1(k, t_1, a_3, a_4, \theta, 2\theta, z_0),\ \forall k \in \mathbb{N}_+, \tag{5.148}$$

where the function ϕ_1 is defined in (2.42).

From $t_1 \geq (pc_3(\kappa_1,\kappa_2))^{1/\theta}$, we have

$$t_1^\theta = \mathcal{O}(p). \tag{5.149}$$

From $\kappa_0 \geq c_0(\kappa_1,\kappa_2)/t_1^\theta$, $t_1 \leq (pc_4c_3(\kappa_1,\kappa_2))^{1/\theta}$, $c_0(\kappa_1,\kappa_2) \geq \varepsilon_1 \geq p\kappa_3$, and $c_3(\kappa_1,\kappa_2) = 24\kappa_3/\kappa_2$, we have

$$\frac{\kappa_2}{\kappa_0} \leq \frac{\kappa_2 t_1^\theta}{c_0(\kappa_1,\kappa_2)} \leq \frac{\kappa_2 pc_4c_3(\kappa_1,\kappa_2)}{p\kappa_3} \leq 24c_4. \tag{5.150}$$

Thus,

$$\phi_1(k, t_1, a_3, a_4, \theta, 2\theta, z_0) = \mathcal{O}(\frac{pn}{(k+t_1)^\theta}). \tag{5.151}$$

From (5.121), we have

$$\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + W_{4,k} \le \hat{V}_k \le \frac{W_k}{\kappa_7}. \tag{5.152}$$

From (5.75g), (5.148), (5.151), and (5.152), we get

$$\mathbf{E}[\|\bar{\boldsymbol{g}}^0_k\|^2] = \mathcal{O}(\frac{pn}{(k+t_1)^\theta}),\ \forall k \in \mathbb{N}_+. \tag{5.153}$$

From (5.149) and (5.153), we know that there exists a constant $c_g > 0$, such that

$$\mathbf{E}[\|\bar{\boldsymbol{g}}^0_k\|^2] \le nc_g,\ \forall k \in \mathbb{N}_0. \tag{5.154}$$

From (5.104b), (5.154), (5.131), and (5.14), we have

$$\breve{z}_{k+1} \le (1-a_1)\breve{z}_k + \frac{\breve{a}_2}{(t+t_1)^{2\theta}}. \tag{5.155}$$

Using (2.45), from (5.133) and (5.155), we have

$$\breve{z}_k \le \phi_3(k, t_1, a_1, \breve{a}_2, 2\theta, \breve{z}_0),\ \forall k \in \mathbb{N}_+, \tag{5.156}$$

where the function ϕ_3 is defined in (2.46). From (5.156), (5.131), (2.46), and (5.150), we have

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] \le \frac{1}{\kappa_7}\breve{z}_k \le \frac{1}{\kappa_7}\phi_3(k, t_1, a_1, \breve{a}_2, 2\theta, \breve{z}_0) = \mathcal{O}(\frac{pn}{(k+t_1)^{2\theta}}), \tag{5.157}$$

which yields (5.15a).

From (5.104c), (4.65), and $\delta_k \le \kappa_\delta \eta_k$ we have

$$\mathbf{E}[W_{4,k+1}] \le \mathbf{E}[W_{4,k}] - \frac{3\nu}{8}\eta_k \mathbf{E}[W_{4,k}] + \|\boldsymbol{x}_k\|^2_{2\eta_k L_f^2 \boldsymbol{K}} + 2pL_f(\sigma_1^2 + 3\sigma_2^2)\eta_k^2 + (n+p)L_f^2\kappa_\delta^2\eta_k^3. \tag{5.158}$$

Similar to the way to prove (2.41), from (5.157) and (5.158), we have

$$\mathbf{E}[f(\bar{x}_T) - f^*] \le \frac{\varepsilon_{18} p}{n(T+t_1)^\theta} + \mathcal{O}(\frac{p}{(T+t_1)^{2\theta}}). \tag{5.159}$$

From (5.150), we have

$$\varepsilon_{18} = \frac{32\theta 4^\theta L_f(\sigma_1^2 + 3\sigma_2^2)\kappa_2}{3\nu\kappa_0} \le \frac{256\theta 4^\theta L_f(\sigma_1^2 + 3\sigma_2^2)c_4}{\nu}. \tag{5.160}$$

Thus, from (5.159) and (5.160), we have (5.15b).

5.8.6 Proof of Theorem 5.6

In addition to the notations defined in Appendices 5.8.3 and 5.8.5, we also denote the following notations.

$$\hat{c}_0(\kappa_1,\kappa_2)=\frac{\kappa_2}{8\kappa_6},\ \hat{c}_3(\kappa_0,\kappa_1,\kappa_2)=\max\Big\{\frac{c_0(\kappa_1,\kappa_2)}{\kappa_0},\ \frac{\kappa_6}{\varepsilon_4},\ \frac{\kappa_6}{\varepsilon_6},\ \frac{24\kappa_3}{\kappa_2},\ p^{\frac{1}{\hat{a}_3}}\Big\},$$
$$\hat{a}_3=\min\Big\{1,\ \frac{2}{3\kappa_6}\Big\},\ \breve{a}_3=pn(\varepsilon_{11}\kappa_\delta^2+\varepsilon_{12}+\varepsilon_{13}\breve{c}_g)\frac{\kappa_2^2}{\kappa_0^2}.$$

From $t_1>\hat{c}_3(\kappa_0,\kappa_1,\kappa_2)\geq\frac{c_0(\kappa_1,\kappa_2)}{\kappa_0}$, we have $\kappa_0>\frac{c_0(\kappa_1,\kappa_2)}{t_1}$. Thus, all conditions needed in Lemma 5.4 are satisfied, so (5.143)–(5.147) still hold when $\theta=1$.

From rom $t_1>\hat{c}_3(\kappa_0,\kappa_1,\kappa_2)\geq\max\{\kappa_6/\varepsilon_4,\ \kappa_6/\varepsilon_6\}$, we have

$$\frac{\varepsilon_4 t_1}{\kappa_6}>1,\ \frac{\varepsilon_6 t_1}{\kappa_6}>1. \tag{5.161}$$

From $\kappa_0\in[3\hat{c}_0\nu\kappa_2/16,3\nu\kappa_2/16)$, we have

$$\frac{16}{3\nu}<\frac{\kappa_2}{\kappa_0}\leq\frac{16}{3\hat{c}_0\nu}. \tag{5.162}$$

Thus,

$$\frac{\nu\kappa_2}{8\kappa_6\kappa_0}>\frac{2}{3\kappa_6}. \tag{5.163}$$

Hence, from (5.161) and (5.163), we have

$$a_3>\hat{a}_3. \tag{5.164}$$

Then from $\theta=1$, (5.143)–(5.147), (5.164), and (2.43), we have

$$z_k\leq\phi_2(k,t_1,a_3,a_4,2,z_0),\ \forall k\in\mathbb{N}_+, \tag{5.165}$$

where the function ϕ_2 is defined in (2.44).

From (5.164) and (5.162), we have $\phi_2(k,t_1,a_3,a_4,2,z_0)=O(pn/(k+t_1)^{\hat{a}_3})$. Hence, from (5.75g), (5.165), and (5.152), we get

$$\mathbf{E}[\|\bar{\mathbf{g}}_k^0\|^2]=O(\frac{pn}{(k+t_1)^{\hat{a}_3}}),\ \forall k\in\mathbb{N}_+. \tag{5.166}$$

Noting that $t_1>\hat{c}_3(\kappa_0,\kappa_1,\kappa_2)\geq p^{1/\hat{a}_3}$, from (5.166), we know that there exists a constant $\breve{c}_g>0$, such that

$$\mathbf{E}[\|\bar{\mathbf{g}}_k^0\|^2]\leq n\breve{c}_g,\ \forall k\in\mathbb{N}_0. \tag{5.167}$$

From (5.104b), (5.167), (5.131), and (5.14), we have

$$\breve{z}_{k+1}\leq(1-a_1)\breve{z}_k+\frac{\breve{a}_3}{(t+t_1)^2}. \tag{5.168}$$

Using (2.45), from (5.133) and (5.168), we have

$$\breve{z}_k \le \phi_3(k, t_1, a_1, \breve{a}_3, 2, \breve{z}_0),\ \forall k \in \mathbb{N}_+, \tag{5.169}$$

where the function ϕ_3 is defined in (2.46). From (5.169), (5.131), (2.46), and (5.162), we have

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] \le \frac{1}{\kappa_7}\breve{z}_k \le \frac{1}{\kappa_7}\phi_3(k, t_1, a_1, \breve{a}_3, 2, \breve{z}_0) = \mathcal{O}(\frac{pn}{(k+t_1)^2}), \tag{5.170}$$

which yields (5.17a).

From $\kappa_0 < 3\nu\kappa_2/16$, we have

$$\frac{3\nu\kappa_2}{8\kappa_0} > 2. \tag{5.171}$$

Same to the way to prove (2.43), from (5.170), (5.171), and (5.158), we have

$$\mathbf{E}[f(\bar{x}_T) - f^*] \le \frac{\hat{\varepsilon}_{18}p}{n(T+t_1)} + \mathcal{O}(\frac{p}{(T+t_1)^2}). \tag{5.172}$$

From (5.162), we have

$$\hat{\varepsilon}_{18} = \frac{8L_f(\sigma_1^2 + 3\sigma_2^2)\kappa_2^2}{\kappa_0^2(\frac{3\nu\kappa_2}{8\kappa_0} - 1)} \le \frac{128L_f(\sigma_1^2 + 3\sigma_2^2)\kappa_2}{3\nu\kappa_0} \le \frac{2048L_f(\sigma_1^2 + 3\sigma_2^2)}{9\hat{c}_0\nu^2}. \tag{5.173}$$

Thus, from (5.172) and (5.173), we have (5.17b).

5.8.7 Proof of Theorem 5.7

In addition to the notations defined in Appendices 5.8.3, 5.8.5, and 5.8.6, we also denote the following notations.

$$\breve{c}_0(\kappa_1, \kappa_2) = \max\Big\{\varepsilon_1,\ \frac{2\varepsilon_5}{\varepsilon_4},\ \Big(\frac{2p\varepsilon_7}{\varepsilon_4}\Big)^{\frac{1}{2}},\ \frac{\varepsilon_8}{2\varepsilon_6},\ 4p\kappa_2\varepsilon_{10}\Big\},$$

$$\breve{c}_3(\kappa_0, \kappa_1, \kappa_2) = \max\Big\{\frac{\breve{c}_0(\kappa_1, \kappa_2)}{\kappa_0},\ \frac{\kappa_6}{\varepsilon_4},\ \frac{\kappa_6}{\varepsilon_6},\ \Big(\frac{16L_f\kappa_3}{\nu\kappa_2}\Big)^{\frac{1}{3}},\ \Big(\frac{16L_f\kappa_4}{\nu\kappa_0\kappa_2}\Big)^{\frac{1}{3}},\ \frac{64pL_f\kappa_2\varepsilon_{10}}{\nu\kappa_0}\Big\},$$

$$\breve{\varepsilon}_{12} = 2\varepsilon_{10}\sigma_1^2 + \frac{1}{p}\varepsilon_9\tilde{\sigma}_2^2 + 6\varepsilon_{10}\tilde{\sigma}_2^2,\ \tilde{\sigma}_2^2 = 2L_f f^* - 2L_f\frac{1}{n}\sum_{i=1}^n f_i^*.$$

To prove Theorem 5.7, the following lemma is used.

Lemma 5.6. *Suppose Assumptions 5.1–5.3 and 5.5–5.7 hold and each* $f_i^* > -\infty$. *Suppose* $\alpha_k = \kappa_1\beta_k$, $\beta_k = \kappa_0(k+t_1)^\theta$, *and* $\eta_k = \frac{\kappa_2}{\beta_k}$, *where* $\theta \in [0,1]$, $\kappa_0 \ge \breve{c}_0(\kappa_1,\kappa_2)/t_1^\theta$, $\kappa_1 > c_1$, $\kappa_2 \in (0, c_2(\kappa_1))$, *and* $t_1 \ge 1$. *Let* $\{\boldsymbol{x}_k\}$ *be the sequence generated by Algorithm 5.2, then*

$$\mathbf{E}_{\mathfrak{L}_k}[W_{k+1}] \le W_k - \varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - \varepsilon_6\Big\|\boldsymbol{v}_k + \frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2$$

$$+\frac{4}{3}L_f(b_{3,k}+6pb_{4,k})\eta_k^2W_{4,k}+pn\breve{\varepsilon}_{12}\eta_k^2+pn\varepsilon_{11}\eta_k\delta_k^2, \tag{5.174a}$$

$$\mathbf{E}_{\mathfrak{L}_k}[\breve{W}_{k+1}] \le \breve{W}_k-\varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}-\varepsilon_6\Big\|\boldsymbol{v}_k+\frac{1}{\beta_k}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}}+\frac{4}{3}L_fp\varepsilon_{13}\eta_k^2W_{4,k}$$
$$+pn\breve{\varepsilon}_{12}\eta_k^2+pn\varepsilon_{11}\eta_k\delta_k^2, \tag{5.174b}$$

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \le W_{4,k}+\|\boldsymbol{x}_k\|^2_{2\eta_kL_f^2\boldsymbol{K}}-\frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2+\frac{8p}{n}L_f^2\eta_k^2W_{4,k}$$
$$+2p\eta_k^2L_f(\sigma_1^2+2\tilde{\sigma}_2^2)+(n+p)L_f^2\eta_k\delta_k^2. \tag{5.174c}$$

Proof. We know that (5.75a)–(5.75g) and (5.85) still hold since Assumptions 5.6 and 5.7 hold.

We have

$$\|\boldsymbol{g}_k^0\|^2=\sum_{i=1}^n\|\nabla f_i(\bar{x}_k)\|^2\le\sum_{i=1}^n 2L_f(f_i(\bar{x}_k)-f_i^*)=2L_fn(f(\bar{x}_k)-f^*)+n\tilde{\sigma}_2^2, \tag{5.175}$$

where the inequality holds due to (2.15).

We have

$$\|\boldsymbol{g}_k\|^2=\|\boldsymbol{g}_k-\boldsymbol{g}_k^0+\boldsymbol{g}_k^0\|^2\le 2(\|\boldsymbol{g}_k-\boldsymbol{g}_k^0\|^2+\|\boldsymbol{g}_k^0\|^2)\le 2(L_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+2L_fW_{4,k}+n\tilde{\sigma}_2^2), \tag{5.176}$$

where the first inequality holds due to the Cauchy-Schwarz inequality; and the last inequality holds due to (3.30) and (5.175).

From (5.85) and (5.176), we have

$$\mathbf{E}_{\mathfrak{L}_k}[\|\boldsymbol{g}_k^e\|^2]\le 16pL_fW_{4,k}+8pL_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+4np\sigma_1^2+8np\tilde{\sigma}_2^2+0.5np^2L_f^2\delta_k^2. \tag{5.177}$$

From the Cauchy-Schwarz inequality, (5.75e), and (5.175), we have

$$\|\boldsymbol{g}_{k+1}^0\|^2=\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0+\boldsymbol{g}_k^0\|^2\le 2(\|\boldsymbol{g}_{k+1}^0-\boldsymbol{g}_k^0\|^2+\|\boldsymbol{g}_k^0\|^2)\le 2(\eta_k^2L_f^2\|\boldsymbol{g}_k^e\|^2+2L_fW_{4,k}+n\tilde{\sigma}_2^2). \tag{5.178}$$

Then, similar to the way to get Lemma 5.4, from (5.75a)–(5.75g), (5.177), and (5.178), we get Lemma 5.6. □

Now we are ready to prove Theorem 5.7

From $t_1>\check{c}_3(\kappa_0,\kappa_1,\kappa_2)\ge\frac{\check{c}_0(\kappa_1,\kappa_2)}{\kappa_0}$, we have $\kappa_0>\frac{\check{c}_0(\kappa_1,\kappa_2)}{t_1}$. Thus, all conditions needed in Lemma 5.6 are satisfied, so (5.174a)–(5.174c) still hold when $\theta=1$.

Similar to the way to get (5.115), from $t_1\ge\max\{(\frac{16L_f\kappa_3}{\nu\kappa_2})^{\frac{1}{3}},\ (\frac{16L_f\kappa_4}{\nu\kappa_0\kappa_2})^{\frac{1}{3}},\ \frac{64pL_f\kappa_2\varepsilon_{10}}{\nu\kappa_0}\}$, we have

$$\frac{1}{2}-\frac{4}{3\nu}L_f(b_{3,k}+6pb_{4,k})\eta_k\ge\frac{1}{8}. \tag{5.179}$$

From (5.174a), (4.65), (5.179), (5.121), and (5.122), we know that (5.142) still holds when ε_{12} is replaced by $\breve{\varepsilon}_{12}$.

Then, similar to the way to get (5.17a) and (5.17b), we have (5.19a) and (5.19b).

5.8.8 Proof of Theorem 5.8

In addition to the notations defined in Appendix 5.8.4, we also denote the following notations.

$$\varepsilon = \frac{1}{2} + \frac{1}{2}\max\{1 - \tilde{\varepsilon}_{17},\ \hat{\varepsilon}\},\ \tilde{\varepsilon}_{17} = \frac{1}{4\kappa_6}\min\{4\varepsilon_4,\ 8\varepsilon_6,\ \eta\nu\},\ c_5 = \frac{1}{\varepsilon_4}\Big(\frac{W_0}{n} + \frac{p\tilde{\varepsilon}_{11}\kappa_\delta^2\eta}{1-\hat{\varepsilon}}\Big),$$
$$c_6 = \frac{2p\tilde{\varepsilon}_{10}}{\varepsilon_4},\ c_7 = 8\Big(\frac{W_0}{n} + \frac{p\tilde{\varepsilon}_{11}\kappa_\delta^2\eta}{1-\hat{\varepsilon}}\Big),\ c_8 = 16p\tilde{\varepsilon}_{10},\ c_9 = \frac{W_0}{n} + \frac{p\tilde{\varepsilon}_{11}\kappa_\delta^2\eta}{\varepsilon-\hat{\varepsilon}},\ c_{10} = \frac{2p\tilde{\varepsilon}_{10}\eta}{\tilde{\varepsilon}_{17}}.$$

All conditions needed in Lemma 5.5 are satisfied, so (5.136a) still holds.
(i) Taking expectation in $\mathcal{L}_T$, summing (5.136a) over $k \in [0, T]$, and using $\delta_{i,k} \in (0, \kappa_\delta\hat{\varepsilon}^{k/2}]$ yield

$$\mathbf{E}[W_{T+1}] + \varepsilon_4\sum_{k=0}^{T}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{8}\eta\sum_{k=0}^{T}\|\bar{\boldsymbol{g}}_k^0\|^2 \le W_0 + 2pn(\sigma_1^2 + 3\sigma_2^2)\tilde{\varepsilon}_{10}\eta^2(T+1) + \frac{pn\tilde{\varepsilon}_{11}\kappa_\delta^2\eta}{1-\hat{\varepsilon}},$$

which gives (5.21a)–(5.21b).
(ii) If Assumption 5.4 also holds, then (4.65) holds. From (5.136a), (4.65), and (5.122), for any $k \in \mathbb{N}_0$, we have

$$\begin{aligned}\mathbf{E}[W_{k+1}] &\le W_k - \varepsilon_4\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} - 2\varepsilon_6\Big\|\boldsymbol{v}_k + \frac{1}{\beta}\boldsymbol{g}_k^0\Big\|^2_{\boldsymbol{K}} - \frac{\eta\nu n}{4}(f(\bar{x}_k) - f^*)\\ &\quad + 2pn(\sigma_1^2 + 3\sigma_2^2)\tilde{\varepsilon}_{10}\eta^2 + pn\tilde{\varepsilon}_{11}\eta\delta_k^2\\ &\le W_k - \tilde{\varepsilon}_{17}W_k + 2pn(\sigma_1^2 + 3\sigma_2^2)\tilde{\varepsilon}_{10}\eta^2 + pn\tilde{\varepsilon}_{11}\eta\delta_k^2.\end{aligned} \tag{5.180}$$

From (5.146)

$$0 < \tilde{\varepsilon}_{17} \le \frac{2\varepsilon_6}{\kappa_6} \le \frac{1}{40}. \tag{5.181}$$

From (5.180), (5.121), (5.181), and $\delta_{i,k} \in (0, \kappa_\delta\hat{\varepsilon}^{\frac{k}{2}}]$, we have

$$\begin{aligned}\mathbf{E}[W_{k+1}] &\le (1 - \tilde{\varepsilon}_{17})^{k+1}W_0 + 2pn(\sigma_1^2 + 3\sigma_2^2)\tilde{\varepsilon}_{10}\eta^2\sum_{\tau=0}^{k}(1 - \tilde{\varepsilon}_{17})^\tau\\ &\quad + pn\tilde{\varepsilon}_{11}\kappa_\delta^2\eta\sum_{\tau=0}^{k}(1 - \tilde{\varepsilon}_{17})^\tau\hat{\varepsilon}^{k-\tau},\ \forall k \in \mathbb{N}_0.\end{aligned} \tag{5.182}$$

From (5.182), (2.36), and $\varepsilon > \max\{1 - \tilde{\varepsilon}_{17},\ \hat{\varepsilon}\}$, we have

$$\mathbf{E}[W_{k+1}] \le \epsilon^{k+1}c_9 + n\eta(\sigma_1^2 + 3\sigma_2^2)c_{10},\ \forall k \in \mathbb{N}_0, \tag{5.183}$$

which gives (5.22).

5.8.9 Proof of Theorem 5.9

We denote the following notations.

$$d_1=\frac{\rho_2(L)}{2\rho(L^2)},\ d_2(\gamma)=\min\Big\{\frac{4\epsilon_1}{9L_f^2},\ \frac{1}{48p(2\epsilon_2+L_f)}\Big\},\ \epsilon_1=\frac{1}{2}\gamma\rho_2(L)-\gamma^2\rho(L^2),$$
$$\epsilon_2=\frac{1+2\gamma\rho_2(L)}{2\gamma\rho_2(L)},\ \epsilon_3=2\Big(2\epsilon_2+\frac{1}{n}L_f\Big)(\sigma_1^2+3\sigma_2^2),\ \epsilon_4=\frac{1}{4}L_f^2\Big(\frac{1}{48}+\frac{4}{p}\Big),$$
$$\epsilon_5=\frac{W_{1,0}+W_{4,0}}{n}+\frac{2\theta p(\epsilon_3+\kappa_\delta^2\epsilon_4)\kappa_\eta^2}{2\theta-1},\ \epsilon_6=pn\kappa_\eta^2(24L_f\epsilon_2\epsilon_5G_f^2+4\epsilon_2(\sigma_1^2+3\sigma_2^2)+\epsilon_4\kappa_\delta^2).$$

To prove Theorem 5.9, the following lemma is used.

Lemma 5.7. *Suppose Assumptions 5.1–5.3 and 5.5–5.8 hold. Suppose $\gamma\in(0,d_1)$ and $\eta_k\in(0,d_2(\gamma)]$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.3, then*

$$\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1}+W_{4,k+1}]\le W_{1,k}+W_{4,k}-\|\boldsymbol{x}_k\|^2_{\frac{1}{2}\epsilon_1\boldsymbol{K}}-\frac{1}{8}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2+pn\epsilon_3\eta_k^2+pn\epsilon_4\eta_k\delta_k^2, \tag{5.184a}$$

$$\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1}]\le W_{1,k}-\|\boldsymbol{x}_k\|^2_{\frac{1}{2}\epsilon_1\boldsymbol{K}}+12p\epsilon_2\eta_k^2\|\bar{\boldsymbol{g}}_k^0\|^2+4pn\epsilon_2(\sigma_1^2+3\sigma_2^2)\eta_k^2+pn\epsilon_4\eta_k\delta_k^2, \tag{5.184b}$$

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}]\le W_{4,k}+\|\boldsymbol{x}_k\|^2_{2L_f^2\eta_k\boldsymbol{K}}-\frac{1}{8}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2+2pL_f(\sigma_1^2+3\sigma_2^2)\eta_k^2+(p+n)L_f^2\eta_k\delta_k^2. \tag{5.184c}$$

Proof. It is straightforward to see that for $\{\boldsymbol{x}_k\}$ generated by Algorithm 5.3, Lemma 5.2 and (5.98) still hold. Thus, (5.117) still holds.

We have

$$\begin{aligned}
\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1}]&=\mathbf{E}_{\mathfrak{L}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_{k+1}\|^2_{\boldsymbol{K}}\Big]=\mathbf{E}_{\mathfrak{L}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_k-(\gamma\boldsymbol{L}\boldsymbol{x}_k+\eta_k\boldsymbol{g}_k^e)\|^2_{\boldsymbol{K}}\Big]\\
&=\mathbf{E}_{\mathfrak{L}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}-\gamma\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}}+\frac{1}{2}\gamma^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2}-\eta_k\boldsymbol{x}_k^\top(\boldsymbol{I}_{np}-\gamma\boldsymbol{L})\boldsymbol{K}\boldsymbol{g}_k^e+\frac{1}{2}\eta_k^2\|\boldsymbol{g}_k^e\|^2_{\boldsymbol{K}}\Big]\\
&\le\mathbf{E}_{\mathfrak{L}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}-\|\boldsymbol{x}_k\|^2_{\gamma\boldsymbol{L}-\frac{1}{2}\gamma^2\boldsymbol{L}^2}+\frac{1}{2}\gamma\rho_2(L)\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+\frac{1}{2\gamma\rho_2(L)}\eta_k^2\|\boldsymbol{g}_k^e\|^2\\
&\quad+\frac{1}{2}\gamma^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{L}^2}+\frac{1}{2}\eta_k^2\|\boldsymbol{g}_k^e\|^2+\frac{1}{2}\eta_k^2\|\boldsymbol{g}_k^e\|^2\Big]\\
&\le\mathbf{E}_{\mathfrak{L}_k}\Big[\frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}-\|\boldsymbol{x}_k\|^2_{(\gamma\rho_2(L)-\gamma^2\rho(L^2)-\frac{1}{2}\gamma\rho_2(L))\boldsymbol{K}}+\frac{1+2\gamma\rho_2(L)}{2\gamma\rho_2(L)}\eta_k^2\|\boldsymbol{g}_k^e\|^2\Big]\\
&\le\frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}-\|\boldsymbol{x}_k\|^2_{\epsilon_1\boldsymbol{K}}\\
&\quad+\epsilon_2\eta_k^2\Big(12p\|\bar{\boldsymbol{g}}_k^0\|^2+12pL_f^2\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+4np\sigma_1^2+12np\sigma_2^2+\frac{1}{2}np^2L_f^2\delta_k^2\Big)\\
&=\frac{1}{2}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}-\|\boldsymbol{x}_k\|^2_{\epsilon_1\boldsymbol{K}-12pL_f^2\epsilon_2\eta_k^2\boldsymbol{K}}
\end{aligned}$$

$$+ \epsilon_2\eta_k^2\Big(12p\|\bar{\boldsymbol{g}}_k^0\|^2 + 4np\sigma_1^2 + 12np\sigma_2^2 + \frac{1}{2}np^2L_f^2\delta_k^2\Big), \tag{5.185}$$

where the second equality holds due to (5.23); the third equality holds due to (2.5); the first inequality holds due to the Cauchy-Schwarz inequality and $\rho(\boldsymbol{K}) = 1$; the second inequality holds due to (2.6); the second last equality holds since that $x_{i,k}$ is independent of $\mathfrak{L}_k$; and the last inequality holds due to (5.76a).

From (5.117) and (5.185), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1} + W_{4,k+1}] \le W_{1,k} + W_{4,k} &- \|\boldsymbol{x}_k\|^2_{\epsilon_1\boldsymbol{K}-(L_f^2\eta_k+12pL_f^2\epsilon_2\eta_k^2+\frac{6p}{n}L_f^3\eta_k^2)\boldsymbol{K}} \\ &- \frac{1}{4}\Big(1 - 48p\epsilon_2\eta_k - \frac{24p}{n}L_f\eta_k\Big)\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 \\ &- \frac{1}{4}(1 - 2L_f\eta_k)\eta_k\|\bar{\boldsymbol{g}}_k^s\|^2 + 2pn\Big(2\epsilon_2 + \frac{1}{n}L_f\Big)(\sigma_1^2 + 3\sigma_2^2)\eta_k^2 \\ &+ \frac{1}{4}pnL_f^2\Big(2p\epsilon_2\eta_k + \frac{1}{n}pL_f\eta_k + \frac{4}{p}\Big)\eta_k\delta_k^2. \end{aligned}\tag{5.186}$$

From $\gamma \in (0, d_1)$ and $\rho_2(L) \le \rho(L)$, we have

$$0 < \epsilon_1 < \frac{1}{16}. \tag{5.187}$$

From $\eta_k \le d_2(\gamma) \le 1/(48p(2\epsilon_2 + L_f))$, we have

$$48p\epsilon_2\eta_k + \frac{24p}{n}L_f\eta_k \le 24p(2\epsilon_2 + L_f)d_2(\gamma) \le \frac{1}{2}, \tag{5.188a}$$

$$2L_f\eta_k \le \frac{2L_f}{48p(2\epsilon_2 + L_f)} < \frac{1}{24p} < 1, \tag{5.188b}$$

$$\frac{1}{4}L_f^2\Big(2p\epsilon_2\eta_k + \frac{1}{n}pL_f\eta_k + \frac{4}{p}\Big) \le \epsilon_4. \tag{5.188c}$$

From $\eta_k \le d_2(\gamma) \le 4\epsilon_1/(9L_f^2)$ and (5.188a), we have

$$L_f^2\eta_k + 12pL_f^2\epsilon_2\eta_k^2 + \frac{6p}{n}L_f^3\eta_k^2 \le (1 + 6p(2\epsilon_2 + L_f)d_2(\gamma))L_f^2d_2(\gamma) \le \frac{9}{8}L_f^2d_2(\gamma) \le \frac{1}{2}\epsilon_1. \tag{5.189}$$

From (5.186)–(5.189), we have (5.184a).

Similarly, we get (5.184b) and (5.184c). □

Now it is ready to prove Theorem 5.9.

From $\kappa_\eta \in (0, d_2(\gamma)t_1^\theta]$ and $\eta_k = \kappa_\eta/(k + t_1)^\theta$, we have $\eta_k \le d_2(\gamma)$. Thus, all conditions needed in Lemma 5.7 are satisfied. So (5.184a) and (5.184b) hold.

Taking expectation in $\mathcal{L}_T$, summing (5.184a) over $k \in [0, T]$, and using (2.37) and $\eta_k = \kappa_\eta/(k + t_1)^\theta$ and $\delta_k \le \kappa_\delta\sqrt{\eta_k}$ as stated in (5.24), yield

$$\mathbf{E}[W_{1,T+1} + W_{4,T+1}] + \sum_{k=0}^{T}\mathbf{E}\Big[\frac{1}{2}\epsilon_1\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{8}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2\Big]$$

$$\leq W_{1,0} + W_{4,0} + pn(\epsilon_3 + \kappa_\delta^2\epsilon_4)\kappa_\eta^2 \sum_{k=0}^{T} \frac{1}{(k+t_1)^{2\theta}} \leq n\epsilon_5. \tag{5.190}$$

Noting that $t_1^\theta = O(\sqrt{p})$, we have

$$\kappa_\eta = O(\frac{t_1^\theta}{p}) = O(\frac{1}{\sqrt{p}}). \tag{5.191}$$

From $W_{1,0} + W_{4,0} = O(n)$ and (5.191), we have

$$\epsilon_5 = \frac{W_{1,0} + W_{4,0}}{n} + \frac{2\theta p(\epsilon_3 + \kappa_\delta^2\epsilon_4)\kappa_\eta^2}{2\theta - 1} = O(1). \tag{5.192}$$

From (5.190), (5.187), and $\sum_{k=0}^{T} \eta_k = \sum_{k=0}^{T} \frac{\kappa_\eta}{(k+t_1)^\theta} \geq \frac{\kappa_\eta(T+t_1)^{1-\theta}}{1-\theta}$, we have

$$\frac{\sum_{k=0}^{T} \eta_k \mathbf{E}[\|\nabla f(\bar{x}_k)\|^2]}{\sum_{k=0}^{T} \eta_k} = \frac{\sum_{k=0}^{T} \eta_k \mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2]}{n\sum_{k=0}^{T} \eta_k} \leq \frac{8(1-\theta)\epsilon_5}{\kappa_\eta(T+t_1)^{1-\theta}}. \tag{5.193}$$

From (5.191)–(5.193), we have (5.25a).

From (5.190) and (5.187), we have

$$\mathbf{E}[f(\bar{x}_{T+1})] - f^* \leq \frac{1}{n} W_{4,T+1} \leq \epsilon_5. \tag{5.194}$$

From (5.194) and (5.192), we have (5.25b).

From (5.190) and (5.187), we have

$$\sum_{k=0}^{T} \mathbf{E}[\|\boldsymbol{x}_k\|_{\boldsymbol{K}}^2] \leq \frac{2n\epsilon_5}{\epsilon_1}. \tag{5.195}$$

From (5.75g) and (5.194), we have

$$\|\bar{\boldsymbol{g}}_k^0\|^2 \leq 2nL_f\epsilon_5. \tag{5.196}$$

From (5.76a), (5.195), and (5.196), we know that $\mathbf{E}[\|\boldsymbol{g}_k^e\|^2]$ is bounded. Then, same as the proof of the first part of Theorem 1 in [151], we have (5.25d).

From (5.184b), (5.196), and (5.24), we have

$$\mathbf{E}[W_{1,k+1}] \leq (1-\epsilon_1)\mathbf{E}[W_{1,k}] + \frac{\epsilon_6}{(t+t_1)^{2\theta}}. \tag{5.197}$$

From (5.197), (5.187), and (2.45), we have

$$\mathbf{E}[W_{1,k}] \leq \phi_3(k, t_1, \epsilon_1, \epsilon_6, 2\theta, W_{1,0}),\ \forall k \in \mathbb{N}_+, \tag{5.198}$$

where the function ϕ_3 is defined in (2.46).

Noting that $\phi_3(k, t_1, \epsilon_1, \epsilon_6, 2\theta, W_{1,0}) = O(n/k^{2\theta})$, from (5.198), we have (5.25c).

5.8.10 Proof of Theorem 5.10

We use the notations defined in Appendix 5.8.9.

From $\eta_k = \eta = \sqrt{n}/\sqrt{pT}$ and $T \geq n/(pd_2^2(\gamma))$, we have $\eta_k \leq d_2(\gamma)$. Thus, all conditions needed in Lemma 5.7 are satisfied. So (5.184a) and (5.184c) hold.

From (5.184a), $\eta_k = \eta = \sqrt{n}/\sqrt{pT}$, and $\delta_{i,k} \leq \kappa_\delta/(pn(k+1))^{1/4}$ as stated in (5.26), similar to the way to get (5.195) and (5.194), we have

$$\frac{1}{T+1}\sum_{k=0}^{T}\mathbf{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|x_{i,k}-\bar{x}_k\|^2\Big] \leq \frac{2}{\epsilon_1}\Big(\frac{W_{1,0}+W_{4,0}}{n(T+1)}+\frac{n\epsilon_3}{T}+\frac{2n\kappa_\delta^2\epsilon_4}{\sqrt{T(T+1)}}\Big), \tag{5.199}$$

which gives (5.27c).

From (5.184c) and $\eta_k = \eta$, we have

$$\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \leq W_{4,k} + \|\boldsymbol{x}_k\|^2_{2L_f^2\eta\boldsymbol{K}} - \frac{1}{8}\eta\|\bar{\boldsymbol{g}}_k^0\|^2 + 2pL_f(\sigma_1^2+3\sigma_2^2)\eta^2 + (p+n)L_f^2\eta\delta_k^2. \tag{5.200}$$

From (5.200) and $\delta_{i,k} \leq \kappa_\delta/(pn(k+1))^{1/4}$ as stated in (5.26), similar to the way to get (5.193), we have

$$\begin{aligned}
&\frac{1}{T+1}\sum_{k=0}^{T}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = \frac{1}{n(T+1)}\sum_{k=0}^{T}\mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2]\\
&\leq 8\Big(\frac{W_{4,0}}{n(T+1)\eta}+\frac{2L_f^2}{n(T+1)}\sum_{k=0}^{T}\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}]+\frac{2pL_f(\sigma_1^2+3\sigma_2^2)\eta}{n}+\frac{2\sqrt{p}L_f^2\kappa_\delta^2}{\sqrt{n(T+1)}}\Big).
\end{aligned} \tag{5.201}$$

Noting that $\eta = \sqrt{n}/\sqrt{pT}$ and $T \geq n^3/p$, from (5.199) and (5.201), we have

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbf{E}[\|\nabla f(\bar{x}_k)\|^2] = 8(f(\bar{x}_0)-f^*+2(\sigma_1^2+3\sigma_2^2)L_f+2L_f^2\kappa_\delta^2)\frac{\sqrt{p}}{\sqrt{nT}}+O(\frac{n}{T}),$$

which gives (5.27a).

Taking expectation in $\mathcal{L}_T$, summing (5.200) over $k \in [0, T]$, and using $\delta_{i,k} \leq \kappa_\delta/(pn(k+1))^{1/4}$ yield

$$\begin{aligned}
&n(\mathbf{E}[f(\bar{x}_{T+1})]-f^*) = \mathbf{E}[W_{4,T+1}]\\
&\leq W_{4,0}+2\eta L_f^2\sum_{k=0}^{T}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}+(T+1)2p\eta^2L_f(\sigma_1^2+3\sigma_2^2)+2\sqrt{pn}L_f^2\eta\sqrt{T+1}.
\end{aligned} \tag{5.202}$$

Noting that $W_{4,0} = O(n)$, $\eta = \sqrt{n}/\sqrt{pT}$, and $T \geq n^3/p$, from (5.199) and (5.202), we have (5.27b).

Similar to the proof of (5.25d), we have (5.27d).

5.8.11 Proof of Theorem 5.11

In addition to the notations defined in Appendix 5.8.9, we also denote the following notations.

$$\tilde{\epsilon}_6 = pn\kappa_\eta^2(12L_f\epsilon_2 d_g + 4\epsilon_2(\sigma_1^2 + 3\sigma_2^2) + \epsilon_4\kappa_\delta^2),$$
$$\epsilon_7 = \min\Big\{\frac{\epsilon_1 t_1^\theta}{\kappa_\eta},\ \frac{\nu}{4}\Big\},\ b_1 = \epsilon_7\kappa_\eta,\ b_2 = pn(\epsilon_3 + \epsilon_4\kappa_\delta^2)\kappa_\eta^2.$$

All conditions needed in Lemma 5.7 are satisfied, so (5.184a)–(5.184c) hold.

Denote $\check{W}_k = W_{1,k} + W_{4,k}$. From (5.184a) and (4.65), we have

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[\check{W}_{k+1}] &\le \check{W}_k - \|\boldsymbol{x}_k\|^2_{\frac{1}{2}\epsilon_1\boldsymbol{K}} - \frac{\nu}{4}\eta_k W_{4,k} + pn\epsilon_3\eta_k^2 + pn\epsilon_4\eta_k\delta_k^2\\ &\le \Big(1 - \eta_k\min\Big\{\frac{\epsilon_1}{\eta_k},\ \frac{\nu}{4}\Big\}\Big)\check{W}_k + pn\epsilon_3\eta_k^2 + pn\epsilon_4\eta_k\delta_k^2\\ &\le (1-\eta_k\epsilon_7)\check{W}_k + pn\epsilon_3\eta_k^2 + pn\epsilon_4\eta_k\delta_k^2,\ \forall k\in\mathbb{N}_0.\end{aligned} \tag{5.203}$$

Denote $\check{z}_k = \mathbf{E}[\check{W}_k]$, $s_{1,k} = \eta_k\epsilon_7$, and $s_{2,k} = pn\epsilon_3\eta_k^2 + pn\epsilon_4\eta_k\delta_k^2$. From (5.203), we have

$$\check{z}_{k+1} \le (1 - s_{1,k})\check{z}_k + s_{2,k},\ \forall k\in\mathbb{N}_0. \tag{5.204}$$

From (5.28), we have

$$s_{1,k} = \eta_k\epsilon_7 = \frac{b_1}{(k+t_1)^\theta}, \tag{5.205}$$

$$s_{2,k} = pn\epsilon_3\eta_k^2 + pn\epsilon_4\eta_k\delta_k^2 \le \frac{b_2}{(k+t_1)^{2\theta}}. \tag{5.206}$$

From (5.187), we have

$$0 < s_{1,k} \le \epsilon_1 \le \frac{1}{16}. \tag{5.207}$$

Then, from $\theta\in(0,1)$, (5.204)–(5.207), and (2.41), we have

$$\check{z}_k \le \phi_1(k, t_1, b_1, b_2, \theta, 2\theta, \check{z}_0),\ \forall k\in\mathbb{N}_+, \tag{5.208}$$

where the function ϕ_1 is defined in (2.42).

Noting that $t_1^\theta = \mathcal{O}(p)$, we have

$$\kappa_\eta = \mathcal{O}(\frac{t_1^\theta}{p}) = \mathcal{O}(1). \tag{5.209}$$

From (5.75g), (5.208), and (5.209), we get

$$\mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2] = \mathcal{O}(\frac{pn}{(k+t_1)^\theta}),\ \forall k\in\mathbb{N}_+. \tag{5.210}$$

From (5.149) and (5.210), we know that there exists a constant $d_g > 0$, such that

$$\mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2] \leq nd_g,\ \forall k \in \mathbb{N}_0. \tag{5.211}$$

From (5.184b), (5.211), and (5.28), we have

$$\mathbf{E}[W_{1,k+1}] \leq (1-\epsilon_1)\mathbf{E}[W_{1,k}] + \frac{\tilde{\epsilon}_6}{(t+t_1)^{2\theta}}. \tag{5.212}$$

Using (2.45), from (5.187) and (5.212), we have

$$\mathbf{E}[W_{1,k}] \leq \phi_3(k, t_1, \epsilon_1, \tilde{\epsilon}_6, 2\theta, W_{0,k}),\ \forall k \in \mathbb{N}_+, \tag{5.213}$$

where the function ϕ_3 is defined in (2.46). From (5.213), (2.46), and (5.209), we have

$$\mathbf{E}[\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}}] \leq 2\mathbf{E}[W_{1,k}] \leq 2\phi_3(k, t_1, \epsilon_1, \tilde{\epsilon}_6, 2\theta, W_{0,k}) = \mathcal{O}(\frac{pn}{(k+t_1)^{2\theta}}), \tag{5.214}$$

which yields (5.29a).

From (5.184c), (4.65), and $\delta_k \leq \kappa_\delta \eta_k$ we have

$$\mathbf{E}[W_{4,k+1}] \leq \mathbf{E}[W_{4,k}] - \frac{\nu}{4}\eta_k\mathbf{E}[W_{4,k}] + \|\boldsymbol{x}_k\|^2_{2L_f^2\eta_k\boldsymbol{K}} + 2pL_f(\sigma_1^2+3\sigma_2^2)\eta_k^2 + (p+n)L_f^2\kappa_\delta^2\eta_k^3. \tag{5.215}$$

Similar to the way to prove (2.41), from (5.214) and (5.215), we have (5.29b).

5.8.12 Proof of Theorem 5.12

In addition to the notations defined in Appendices 5.8.9 and 5.8.11, we also denote $\hat{d}_2(\gamma) = \max\{\frac{1}{\epsilon_1}, \frac{\kappa_\eta}{d_2(\gamma)}\}$

From $t_1 > \hat{d}_2(\gamma) \geq \frac{\kappa_\eta}{d_2(\gamma)}$, we have $\eta_k = \frac{\kappa_\eta}{k+t_1} \leq \frac{\kappa_\eta}{t_1} < d_2(\gamma)$. Thus, all conditions needed in Lemma 5.7 are satisfied, so (5.204)–(5.207) still hold when $\theta = 1$.

From $t_1 > \hat{d}_2(\gamma) \geq \frac{1}{\epsilon_1}$ and $\kappa_\eta > 4/\nu$, we have

$$b_1 = \epsilon_6\kappa_\eta > 1. \tag{5.216}$$

Then from $\theta = 1$, (5.204)–(5.207), (5.216), and (2.43), we have

$$\check{z}_k \leq \phi_2(k, t_1, b_1, b_2, 2, \check{z}_0),\ \forall k \in \mathbb{N}_+, \tag{5.217}$$

where the function ϕ_2 is defined in (2.44).

From $\kappa_\eta > 4/\nu$, we know $\kappa_\eta = \mathcal{O}(1)$, thus $\phi_2(k, t_1, b_1, b_2, 2, \check{z}_0) = \mathcal{O}(pn/k)$. Hence, from (5.75g) and (5.217), we get

$$\mathbf{E}[\|\bar{\boldsymbol{g}}_k^0\|^2] = \mathcal{O}(\frac{pn}{k+t_1}),\ \forall k \in \mathbb{N}_+. \tag{5.218}$$

Then, similar to the way to get (5.29a) and (5.29b), we get (5.31a) and (5.31b).

5.8.13 Proof of Theorem 5.13

In addition to the notations defined in Appendices 5.8.9, 5.8.11, and 5.8.12, we also denote

$$\tilde{d}_2(\gamma) = \min\Big\{\frac{\epsilon_1}{4L_f^2},\ \frac{1}{4p(2\epsilon_2 + L_f)}\Big\},\ \check{d}_2(\gamma) = \max\Big\{\frac{1}{\epsilon_1},\ \frac{\kappa_\eta}{\tilde{d}_2(\gamma)},\ \frac{\kappa_\eta}{8\nu\epsilon_8}\Big\},$$
$$\check{\epsilon}_3 = 2\Big(2\epsilon_2 + \frac{1}{n}L_f\Big)(\sigma_1^2 + 2\tilde{\sigma}_2^2),\ \check{\epsilon}_4 = \frac{1}{4}L_f^2\Big(\frac{1}{8} + \frac{4}{p}\Big),\ \epsilon_8 = 8p(2\epsilon_2 + L_f)L_f$$

To prove Theorem 5.13, the following lemma is used.

Lemma 5.8. *Suppose Assumptions 5.1–5.3 and 5.5–5.7 hold and each $f_i^* > -\infty$. Suppose $\gamma \in (0, d_1)$ and $\eta_k \in (0, \tilde{d}_2(\gamma)]$. Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Algorithm 5.3, then*

$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1} + W_{4,k+1}] \le{}& W_{1,k} + W_{4,k} - \|\boldsymbol{x}_k\|^2_{\frac{1}{2}\epsilon_1\boldsymbol{K}} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2\\ &+ \epsilon_8\eta_k^2 W_{4,k} + pn\check{\epsilon}_3\eta_k^2 + pn\check{\epsilon}_4\eta_k\delta_k^2, \end{aligned} \tag{5.219a}$$
$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[W_{1,k+1}] \le{}& W_{1,k} - \|\boldsymbol{x}_k\|^2_{\frac{1}{2}\epsilon_1\boldsymbol{K}} + 16p\epsilon_2 L_f\eta_k^2 W_{4,k}\\ &+ 4pn\epsilon_2(\sigma_1^2 + 2\tilde{\sigma}_2^2)\eta_k^2 + pn\check{\epsilon}_4\eta_k\delta_k^2,\end{aligned} \tag{5.219b}$$
$$\begin{aligned}\mathbf{E}_{\mathfrak{L}_k}[W_{4,k+1}] \le{}& W_{4,k} + \|\boldsymbol{x}_k\|^2_{2L_f^2\eta_k\boldsymbol{K}} - \frac{1}{4}\eta_k\|\bar{\boldsymbol{g}}_k^0\|^2 + \frac{8p}{n}L_f^2 W_{4,k}\\ &+ 2pL_f(\sigma_1^2 + 2\tilde{\sigma}_2^2)\eta_k^2 + (p+n)L_f^2\eta_k\delta_k^2.\end{aligned} \tag{5.219c}$$

Proof. We know that (5.75a)–(5.75g) and (5.177) still hold since Assumptions 5.6 and 5.7 hold, and each $f_i^* > -\infty$. Then, similar to the way to get Lemma 5.7, we get Lemma 5.8. □

Now we are ready to prove Theorem 5.13.

From $t_1 > \check{d}_2(\gamma) \ge \max\{\kappa_\eta/\tilde{d}_2(\gamma),\ \kappa_\eta/(4\nu\epsilon_8)\}$, we have

$$\eta_k = \frac{\kappa_\eta}{k + t_1} \le \frac{\kappa_\eta}{t_1} < \min\Big\{\tilde{d}_2(\gamma),\ \frac{1}{4\nu\epsilon_8}\Big\}. \tag{5.220}$$

Thus, all conditions needed in Lemma 5.8 are satisfied, so (5.219a)–(5.219c) hold

From (5.219a), (4.65), and (5.220), we know that (5.203) still holds when ϵ_3 and ϵ_4 are replaced by $\check{\epsilon}_3$ and $\check{\epsilon}_4$, respectively.

Then, similar to the way to get (5.31a) and (5.31b), we have (5.33a) and (5.33b).

5.8.14 Proof of Theorem 5.14

In addition to the notations defined in Appendix 5.8.9, we also denote the following notations.

$$\epsilon = 0.5 + 0.5\max\{1 - \tilde{\epsilon}_7,\ \hat{\epsilon}\},\ \tilde{\epsilon}_7 = \min\Big\{\epsilon_1,\ \frac{1}{4}\nu\eta\Big\},\ d_3 = \frac{2}{\epsilon_1}\Big(\frac{W_{1,0} + W_{4,0}}{n} + \frac{p\epsilon_4\kappa_\delta^2\eta}{1 - \hat{\epsilon}}\Big),$$

$$d_4 = \frac{4p}{\epsilon_1}\Big(2\epsilon_2 + \frac{1}{n}L_f\Big),\ d_5 = 8\Big(\frac{W_{1,0} + W_{4,0}}{n} + \frac{p\epsilon_4\kappa_\delta^2\eta}{1-\hat{\epsilon}}\Big),\ d_6 = 16p\Big(2\epsilon_2 + \frac{1}{n}L_f\Big),$$
$$d_7 = \frac{W_{1,0} + W_{4,0}}{n} + \frac{p\epsilon_4\kappa_\delta^2\eta}{\epsilon-\hat{\epsilon}},\ d_8 = \frac{2p\eta}{\tilde{\epsilon}_7}\Big(2\epsilon_2 + \frac{1}{n}L_f\Big).$$

All conditions needed in Lemma 5.7 are satisfied, so (5.184a) still holds.
(i) Taking expectation in $\mathcal{L}_T$, summing (5.184a) over $k \in [0, T]$, and using $\eta_k = \eta$ and $\delta_{i,k} \in (0, \kappa_\delta\hat{\epsilon}^{k/2}]$ yield

$$\begin{aligned}&\mathbf{E}[W_{1,T+1} + W_{4,T+1}] + \frac{1}{2}\epsilon_1\sum_{k=0}^{T}\|\boldsymbol{x}_k\|^2_{\boldsymbol{K}} + \frac{1}{8}\eta\sum_{k=0}^{T}\|\bar{\boldsymbol{g}}_k^0\|^2\\ &\leq W_{1,0} + W_{4,0} + pn\epsilon_3\eta^2(T+1) + \frac{pn\epsilon_4\kappa_\delta^2\eta}{1-\hat{\epsilon}},\end{aligned}$$

which gives (5.35a)–(5.35b).
(ii) If Assumption 5.4 also holds, then (4.65) holds. Thus, (5.203) also holds when $\eta_k = \eta$. From (5.203) and $\eta_k = \eta$, for all $k \in \mathbb{N}_0$, we have

$$\mathbf{E}_{\mathfrak{L}_k}[\check{W}_{k+1}] \leq (1-\tilde{\epsilon}_7)\check{W}_k + pn\epsilon_3\eta^2 + pn\epsilon_4\eta\delta_k^2. \tag{5.221}$$

From (5.187)

$$0 < \tilde{\epsilon}_7 \leq \epsilon_1 < \frac{1}{16}. \tag{5.222}$$

From (5.221), (5.222), and $\delta_{i,k} \in (0, \kappa_\delta\hat{\epsilon}^{\frac{k}{2}}]$, we have

$$\mathbf{E}[\check{W}_{k+1}] \leq (1-\tilde{\epsilon}_7)^{k+1}\check{W}_0 + pn\epsilon_3\eta^2\sum_{\tau=0}^{k}(1-\tilde{\epsilon}_7)^\tau + pn\epsilon_4\kappa_\delta^2\eta\sum_{\tau=0}^{k}(1-\tilde{\epsilon}_7)^\tau\hat{\epsilon}^{k-\tau},\ \forall k \in \mathbb{N}_0. \tag{5.223}$$

From (5.223), (2.36), and $\epsilon > \max\{1-\tilde{\epsilon}_7,\ \hat{\epsilon}\}$, we have

$$\mathbf{E}[\check{W}_{k+1}] \leq \epsilon^{k+1}d_7 + n(\sigma_1^2 + 3\sigma_2^2)d_8,\ \forall k \in \mathbb{N}_0, \tag{5.224}$$

which gives (5.36).

Part II

Distributed Online Convex Optimization

Chapter 6

Distributed online primal–dual optimization algorithm

This and the next chapters consider on online convex optimization problems, which view optimization as a process or a repeated game. This chapter considers distributed online convex optimization with time-varying coupled inequality constraints. The global objective function is composed of local convex cost and regularization functions and the coupled constraint function is the sum of local convex functions. A distributed online primal–dual dynamic mirror descent algorithm is proposed to solve this problem, where the local cost, regularization, and constraint functions are held privately and revealed only after each time slot. Without assuming Slater's condition, we first derive regret and constraint violation bounds for the proposed algorithm and show how they depend on the stepsize sequences, the accumulated dynamic variation of the comparator sequence, the number of agents, and the network connectivity. As a result, under some natural decreasing stepsize sequences, we prove that the proposed algorithm achieves sublinear dynamic regret and constraint violation if the accumulated dynamic variation of the optimal sequence also grows sublinearly. In particular, we show that it achieves $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$ static regret and $\mathcal{O}(T^{1-\kappa/2})$ constraint violation bounds, where $\kappa \in (0,1)$ is a user-defined trade-off parameter. Assuming Slater's condition, we show that the dynamic regret bound is similar to the bound without assuming Slater's condition, but the constraint violation bound can be reduced to $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$. Moreover, we show that both static regret and constraint violation bounds grow as $\mathcal{O}(\sqrt{T})$. In addition, smaller bounds on the static regret are achieved when the objective function is strongly convex. Numerical simulations are provided to illustrate the effectiveness of the theoretical results.

This chapter is organized as follows. Section 6.1 gives the background. Section 6.2 introduces the problem formulation. Section 6.3 provides the distributed online primal–dual dynamic mirror descent algorithm and analyzes the bounds of the regret and constraint violation for this algorithm. Section 6.4 presents numerical simulations. Section 6.5 concludes this chapter. To improve the readability, all the proofs can be found in Section 6.6.

6.1 Introduction

Centralized online convex optimization with static set constraints was first studied by Zinkevich [163]. Specifically, he developed a projection-based online gradient descent algorithm and achieved $O(\sqrt{T})$ static regret bound for an arbitrary sequence of convex objective functions with bounded subgradients. It was later shown that this is a tight bound up to constant factors [166]. The regret bound can be reduced under more stringent strong convexity conditions on the objective functions [157, 165–167] or by allowing to query the gradient of the objective function multiple times [168]. When the static constrained sets are characterized by inequalities, the conventional projection-based online algorithms are difficult to implement and may be inefficient in practice due to high computational complexity of the projection operation. To overcome these difficulties, some researchers proposed primal–dual algorithms for centralized online convex optimization with time-invariant inequality constraints, e.g., [169–172]. The authors of [173] showed that the algorithms proposed in [169, 170] are general enough to handle time-varying inequality constraints. The authors of [174] used the modified saddle-point method to handle time-varying constraints. The authors of [175, 176] used a virtual queue, which essentially is a modified Lagrange multiplier, to handle stochastic and time-varying constraints and the authors of [311] extended the algorithm proposed in [175] with bandit feedback. The authors of [312] studied online convex optimization with time-varying constraints in the continuous-time setting and showed that the static regret in continuous-time can be bounded by a constant independent of the time horizon, as opposed to the sublinear static regret observed in the discrete-time setting.

Distributed online convex optimization has been extensively studied, so here we only list some of the most relevant work. Firstly, the authors of [180–182, 186–188] proposed distributed online algorithms to solve convex optimization problems with static set constraints and achieved sublinear regret. For instance, the authors of [181] proposed a decentralized variant of the dynamic mirror descent algorithm proposed in [313]. Mirror descent generalizes classical gradient descent to Bregman divergences and is suitable for solving high-dimensional convex optimization problems. The weighted majority algorithm in machine learning [314] can be viewed as a special case of mirror descent. Secondly, the authors of [189] extended the adaptive algorithm proposed in [170] to a distributed setting to solve an online convex optimization problem with a static inequality constraint. Finally, the authors of [190, 191] proposed distributed primal–dual algorithms to solve an online convex optimization with static coupled inequality constraints. To the best of our knowledge, no existing studies considered distributed online convex optimization with time-varying constraints in the discrete-time setting. In the continuous-time setting, the authors of [315] extended the online saddle point algorithm proposed in [312] to a distributed version.

This chapter considers distributed online optimization with time-varying coupled inequality constraints. The global objective function is composed of local convex cost and regularization functions and the coupled constraint function is the sum of local convex functions. Compared to the literature the contributions of this chapter are summarized as follows.

(C6.1) We propose a novel distributed online primal–dual dynamic mirror descent algorithm (Algorithm 6.1). In this algorithm, each agent i maintains two local sequences: the local decision (primal) and dual sequences. An agent averages its local dual variable with its in-neighbors in a consensus step, and takes into account the estimated dynamics of the optimal sequences. The proposed algorithm uses different nonincreasing stepsize sequences for the primal and dual updates, and a nonincreasing sequence to design penalty terms such that the dual variables are not growing too large. These sequences give some freedom in the regret and constraint violation bounds, as they allow the trade-off between how fast these two bounds tend to zero. The algorithm uses the subgradients of the local cost and constraint functions at the previous decision, but the total number of iterations or any other parameters related to the objective or constraint functions are not used.

(C6.2) Without assuming Slater's condition, i.e., that the feasible region has an interior point, in Lemma 6.3 we derive regret and constraint violation bounds for the algorithm and show how they depend on the stepsize sequences, the accumulated dynamic variation of the comparator sequence, the number of agents, and the network connectivity. The same regret bound was achieved by the centralized dynamic mirror descent proposed in [313] for static set constraints. Particularly, we show in Theorem 6.1 that our algorithm simultaneously achieves sublinear dynamic regret and constraint violation if the accumulated dynamic variation of the optimal sequence grows sublinearly with a known order. Moreover, we show in Corollary 6.1 that the algorithm achieves $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$ static regret and $\mathcal{O}(T^{1-\kappa/2})$ constraint violation bounds, where $\kappa \in (0, 1)$ is a user-defined trade-off parameter. Same results have been achieved in [170]. Compared with [169, 170, 172, 173, 191], which assumed the same assumption on the cost and constraint functions as this chapter, the proposed algorithm has the following advantages. The parameter κ enables the user to trade-off static regret bound for constraint violation bound, while recovering the $\mathcal{O}(\sqrt{T})$ static regret and $\mathcal{O}(T^{3/4})$ constraint violation bounds from [169, 173] as special cases. The algorithms proposed in [169, 170, 173] are centralized and the constraint functions in [169, 170] are time-invariant. Moreover, in [169, 173] the total number of iterations and in [169, 170, 173] the upper bounds of the objective and constraint functions and their subgradients need to be known in advance to design the stepsizes. The proposed algorithm achieves smaller static regret and constraint violation bounds than [191], although time-invariant coupled inequality constraints were considered. The algorithm proposed in [172] achieved a better constraint violation bound than ours, but their algorithm is centralized and the constraint function is time-invariant.

(C6.3) Assuming Slater's condition, we show in Theorem 6.2 that the dynamic regret bound is similar to the bound without assuming Slater's condition, but the constraint violation bound can be reduced to $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$. Our results are superior to [174] in the sense that the accumulated variation of constraints, $V(\{g_t\}_{t=1}^T) = \sum_{t=1}^T \max_{x\in\mathcal{X}} \|[g_{t+1}(x) - g_t(x)]_+\|$, appears in their bounds and more assumptions are

Table 6.1: Comparison of Chapter 6 to some related online convex optimization algorithms.

Reference	Problem type	Constraint type	Regret and constraint violation bounds
[169]	Centralized	$g(x) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(\sqrt{T})$, $\\|[\sum_{t=1}^T g(x_t)]_+\\| = \mathcal{O}(T^{3/4})$
[170]	Centralized	$g(x) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$, $\\|[\sum_{t=1}^T g(x_t)]_+\\| = \mathcal{O}(T^{1-\kappa/2})$, $\kappa \in (0,1)$
[172]	Centralized	$g(x) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(\sqrt{T})$, $\sum_{t=1}^T \\|[g(x_t)]_+\\|^2 = \mathcal{O}(\sqrt{T})$
[173]	Centralized	$g_t(x) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(\sqrt{T})$, $\\|[\sum_{t=1}^T g_t(x_t)]_+\\| = \mathcal{O}(T^{3/4})$
[174]	Centralized	$g_t(x) \leq \mathbf{0}_m$ and Slater's condition	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}) = \mathcal{O}(\max\{T^{1/3}\sum_{t=1}^T \\|x_t^* - x_{t-1}^*\\|, T^{1/3}V(\{g_t\}_{t=1}^T), T^{2/3}\})$, $\\|[\sum_{t=1}^T g_t(x_t)]_+\\| = \mathcal{O}(T^{2/3})$,
[175]	Centralized	$g_t(x) \leq \mathbf{0}_m$ and Slater's condition	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})/T \leq c\epsilon$, $\\|[\sum_{t=1}^T g_t(x_t)]_+\\|/T \leq c\epsilon$ for $T \geq 1/\epsilon^2$
[190]	Distributed	$g(x) = \sum_{i=1}^n g_i(\mathbb{X}_i) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(\sqrt{T})$, $\\|[\sum_{t=1}^T g(x_t)]_+\\| = \mathcal{O}(\sqrt{T})$ if dual variables generated by the proposed algorithm are bounded
[191]	Distributed	$g(x) = \sum_{i=1}^n g_i(\mathbb{X}_i) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) = \mathcal{O}(T^{1/2+2\kappa})$, $\\|[\sum_{t=1}^T g(x_t)]_+\\| = \mathcal{O}(T^{1-\kappa/2})$, $\kappa \in (0, 1/4)$
This chapter	Distributed	$g_t(x) = \sum_{i=1}^n g_{i,t}(\mathbb{X}_i) \leq \mathbf{0}_m$	$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}) = \mathcal{O}(\max\{T^{\kappa}\sum_{t=1}^{T-1} \\|x_{t+1}^* - x_t^*\\|, T^{\max\{1-\kappa,\kappa\}}\})$, $\\|[\sum_{t=1}^T g_t(x_t)]_+\\| = \mathcal{O}(T^{1-\kappa/2})$ (without Slater's condition), $\\|[\sum_{t=1}^T g_t(x_t)]_+\\| = \mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$ (with Slater's condition), $\kappa \in (0,1)$

needed. We show that our algorithm simultaneously achieves sublinear dynamic regret and constraint violation, if the accumulated variation of the optimal sequence grows sublinearly. Moreover, we show in Corollary 6.2 that both static regret and constraint violation bounds grow as $\mathcal{O}(\sqrt{T})$, which are better than the results achieved by the centralized algorithm in [175]. The authors of [190] achieved the same bounds, but they assumed that the coupled inequality constraints are time-invariant and they explicitly assumed boundedness of the dual variable sequence. The conditions to guarantee this assumption are not so obvious since the dual variable sequence is generated by the algorithm. In this chapter, we show that the dual variable sequence is indeed bounded.

(C6.4) When the local objective functions are assumed to be strongly convex, we show that in Theorem 6.3, without Slater's condition, the proposed algorithm achieves $\mathcal{O}(T^{\kappa})$ static regret and $\mathcal{O}(T^{1-\kappa/2})$ constraint violation bounds. Moreover, we show in Corollary 6.3 that the constraint violation bound can be reduced to $\mathcal{O}(T^{\max\{1-\kappa,\kappa\}})$ when Slater's condition holds.

Table 6.1 compares this chapter with other online convex optimization algorithms.

6.2 Distributed OCO with time-varying coupled inequality constraints

We consider the problem of distributed online convex optimization with time-varying coupled inequality constraints. Specifically, consider a network of n agents indexed by $i \in [n]$. For each i, let $\{f_{i,t} : \mathbb{R}^{p_i} \to \mathbb{R}\}$, $\{r_{i,t} : \mathbb{R}^{p_i} \to \mathbb{R}\}$, and $\{g_{i,t} : \mathbb{R}^{p_i} \to \mathbb{R}^m\}$ be arbitrary sequences of local convex cost, regularization, and constraint functions over time $t = 1, 2, \ldots$, respectively, where p_i and m are positive integers. At time t, each agent i selects a decision $x_{i,t} \in \mathbb{X}_i$, where $\mathbb{X}_i \subseteq \mathbb{R}^{p_i}$ is a known convex set. After the selection, the agent receives its cost function $f_{i,t}$ and regularization $r_{i,t}$ together with its constraint function $g_{i,t}$, and obtains the loss $l_{i,t}(x_{i,t}) = f_{i,t}(x_{i,t}) + r_{i,t}(x_{i,t})$. Here the regularization function is used to influence the structure of the decisions. Examples of regularization include ℓ_1-regularization $r_{i,t}(x_i) = \lambda_i \|x_i\|_1$ and ℓ_2-regularization $r_{i,t}(x_i) = \frac{\lambda_i}{2}\|x_i\|$ with $\lambda_i > 0$. At the same moment, the agents exchange data with their neighbors over a time-varying directed graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where $\mathcal{V} = [n]$ is the agent set and $\mathcal{E}_t \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. The network's objective is to choose a global decision sequence $\boldsymbol{x}_{[T]} = (x_1, \ldots, x_T)$ with $x_t = \mathrm{col}(x_{1,t}, \ldots, x_{n,t})$ so that the accumulated global loss $\sum_{t=1}^{T} l_t(x_t)$ is competitive with the loss of any comparator sequence $\boldsymbol{y}_{[T]} = (y_1, \ldots, y_T)$ with $y_t = \mathrm{col}(y_{1,t}, \ldots, y_{n,t})$ (i.e., the regret grows sublinearly in T) and at the same time the constraint violation grows sublinearly in T, where T is the total number of iterations and $l_t(x_t) = \sum_{i=1}^{n} l_{i,t}(x_{i,t})$ is the global loss function.

From (1.5), we know that the regret of a global decision sequence $\boldsymbol{x}_{[T]}$ with respect to a comparator sequence $\boldsymbol{y}_{[T]}$ is

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) = \sum_{t=1}^{T} l_t(x_t) - \sum_{t=1}^{T} l_t(y_t).$$

For the above distributed online convex optimization problem with time-varying coupled inequality constraints, there are two commonly used comparator sequences. One is the optimal dynamic decision sequence $\boldsymbol{y}_{[T]} = \boldsymbol{x}^*_{[T]} = (x^*_1, \ldots, x^*_T)$ solving the following constrained convex optimization problem when the sequences of cost, regularization, and constraint functions are known a priori:

$$\begin{aligned} \min_{x_t \in \mathbb{X}} \quad & \sum_{t=1}^{T} l_t(x_t) \\ \text{s.t.} \quad & g_t(x_t) \leq \mathbf{0}_m, \ \forall t \in [T], \end{aligned} \tag{6.1}$$

where $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n \subseteq \mathbb{R}^p$ is the global decision set, $p = \sum_{i=1}^{n} p_i$, and $g_t(x_t) = \sum_{i=1}^{n} g_{i,t}(x_{i,t})$ is the coupled constraint function. In order to guarantee that problem (6.1) is feasible, for any $T \in \mathbb{N}_+$, we assume that $\mathcal{X}_T$, the set of all feasible decision sequences, is nonempty, where

$$\mathcal{X}_T = \Big\{(x_1, \ldots, x_T) : \ x_t \in \mathbb{X}, \ g_t(x_t) \leq \mathbf{0}_m, \ \forall t \in [T]\Big\}.$$

With this standing assumption, an optimal dynamic decision sequence to (6.1) always exists. In this case $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})$ is called the dynamic regret for $\boldsymbol{x}_{[T]}$. Another comparator sequence is $\boldsymbol{y}_{[T]} = \check{\boldsymbol{x}}^*_{[T]} = (\check{x}^*_T, \ldots, \check{x}^*_T)$, where $\check{x}^*_T$ is the optimal static decision solving

$$\begin{aligned} \min_{x \in \mathbb{X}} \quad & \sum_{t=1}^{T} l_t(x) \\ \text{s.t.} \quad & g_t(x) \leq \mathbf{0}_m, \ \forall t \in [T]. \end{aligned} \tag{6.2}$$

Similar to above, in order to guarantee that problem (6.2) is feasible, for any $T \in \mathbb{N}_+$, we assume that $\check{\mathcal{X}}_T$, the set of all feasible static decision sequences, is nonempty, where

$$\check{\mathcal{X}}_T = \Big\{(x, \dots, x) : \; x \in \mathbb{X}, \; g_t(x) \le \mathbf{0}_m, \; \forall t \in [T]\Big\} \subseteq \mathcal{X}_T.$$

In this case $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})$ is called the static regret. It is straightforward to see that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) \le \mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}), \; \forall \boldsymbol{y}_{[T]} \in \mathcal{X}_T$, and that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) \le \mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})$.

From (1.7), we know that the constraint violation of a decision sequence $\boldsymbol{x}_{[T]}$ is

$$\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|.$$

This definition implicitly allows constraint violations at some times to be compensated by strictly feasible decisions at other times. This is appropriate for constraints that have a cumulative nature such as energy budgets enforced through average power constraints.

Based on the definitions introduced in Chapter 2, the following mild assumption is made on the time-varying directed graph.

Assumption 6.1. *For any $t \in \mathbb{N}_+$, the graph $\mathcal{G}_t$ satisfies the following conditions:*

(i) *The mixing matrix W_t is doubly stochastic, i.e., $\sum_{i=1}^{n}[W_t]_{ij} = \sum_{j=1}^{n}[W_t]_{ij} = 1, \; \forall i, j \in [n]$.*

(ii) *There exists a constant $w \in (0, 1)$, such that $[W_t]_{ij} \ge w$ if $[W_t]_{ij} > 0$.*

(iii) *There exists an integer $\iota > 0$ such that the graph $(\mathcal{V}, \cup_{l=0,\dots,\iota-1}\mathcal{E}_{t+l})$ is strongly connected.*

We make the following standing assumption on the cost, regularization, and constraint functions.

Assumption 6.2. **(i)** *For each $i \in [n]$, the convex set $\mathbb{X}_i$ is compact, i.e., there exists a positive constant $d(\mathbb{X})$ such that*

$$\|x - y\| \le d(\mathbb{X}), \; \forall x, y \in \mathbb{X}_i, \; \forall i \in [n]. \tag{6.3}$$

(ii) *The functions $\{f_{i,t}\}$, $\{r_{i,t}\}$, and $\{g_{i,t}\}$ are convex and uniformly bounded on $\mathbb{X}_i$, i.e., there exists a constant $F > 0$ such that*

$$|f_{i,t}(x)| \le F, \; |r_{i,t}(x)| \le F, \; \|g_{i,t}(x)\| \le F, \; \forall t \in \mathbb{N}_+, \; \forall i \in [n], \; \forall x \in \mathbb{X}_i. \tag{6.4}$$

(iii) *The subgradients $\nabla f_{i,t}$, $\nabla r_{i,t}$, and $\nabla g_{i,t}$ exist and they are uniformly bounded on $\mathbb{X}_i$, i.e., there exists a constant $G > 0$ such that*

$$\|\nabla f_{i,t}(x)\| \le G, \; \|\nabla r_{i,t}(x)\| \le G, \; \|\nabla g_{i,t}(x)\| \le G, \; \forall t \in \mathbb{N}_+, \; \forall i \in [n], \; \forall x \in \mathbb{X}_i. \tag{6.5}$$

Our goal in this chapter is to solve the following problem.

Problem 6.1. *Develop a distributed algorithm to solve the problem of distributed online optimization with time-varying coupled inequality constraints with guaranteed performance measured by regret and constraint violation.*

We are satisfied with low regret and constraint violation, by which we mean that both $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]})$ and $\|[\sum_{t=1}^{T} g_t(x_t)]_+\|$ grow sublinearly with T, i.e., there exist $\kappa_1, \ \kappa_2 \in (0, 1)$ such that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) = \mathcal{O}(T^{\kappa_1})$ and $\|[\sum_{t=1}^{T} g_t(x_t)]_+\| = \mathcal{O}(T^{\kappa_2})$. This implies that the upper bound of the time averaged difference between the accumulated cost of the decision sequence and the accumulated cost of any comparator sequences tends to zero as T goes to infinity. The same thing holds for the upper bound of the time averaged constraint violation. The novel algorithm we design explores the stepsize sequences in a way that allows the trade-off between how fast these two bounds tend to zero.

6.3 Distributed online primal–dual dynamic mirror descent algorithm

In this section, we first propose a distributed online primal–dual dynamic mirror descent algorithm. Then, we derive regret and constraint violation bounds for this algorithm.

6.3.1 Algorithm description

The regularized Lagrangian function associated with the considered problem at each time t is

$$\mathcal{A}_t(x_t, u_t) = f_t(x_t) + r_t(x_t) + u_t^\top g_t(x_t) - \frac{\beta_{t+1}}{2}\|u_t\|^2, \tag{6.6}$$

where $\{u_t \in \mathbb{R}_+^m\}$ is the dual variable or Lagrange multiplier vector sequence and $\{\beta_t > 0\}$ is the regularization sequence. Inspired by the dynamic mirror descent [313], which is a generalization of the composite objective mirror descent algorithm [316], a centralized online primal–dual dynamic mirror descent algorithm to solve the considered problem is

$$\tilde{x}_{t+1} = \operatorname*{argmin}_{x\in\mathbb{X}}\{\alpha_{t+1}(\langle x, \nabla f_t(x_t) + (\nabla g_t(x_t))^\top u_t\rangle + r_t(x_t)) + \mathcal{D}_\psi(x, x_t)\}, \tag{6.7a}$$

$$u_{t+1} = [u_t + \gamma_{t+1}(g_t(x_t) - \beta_{t+1}u_t)]_+, \tag{6.7b}$$

$$x_{t+1} = \Phi_{t+1}(\tilde{x}_{t+1}), \tag{6.7c}$$

where $\{\alpha_t > 0\}$ and $\{\gamma_t > 0\}$ are the stepsize sequences used in the primal and dual updates, respectively; $\psi_i : \mathbb{R}^{p_i} \to \mathbb{R}$ is a function to define the Bregman divergence $\mathcal{D}_\psi(\cdot, \cdot)$, which is differentiable and strongly convex with convexity parameter $\sigma_i > 0$ on $\mathbb{X}_i$; and $\Phi_t : \mathbb{X} \to \mathbb{X}$ is a dynamic model and characterizes a prior knowledge of the considered problem, akin to developing a state space model for stochastic filters [313], and if the prior knowledge is lacking then Φ_t is simply set to the identity mapping. When r_t is a constant mapping and Φ_t is the identity mapping, then the centralized online algorithm (6.7) is Algorithm 1 in [173]. The potential drawback of that algorithm is that the upper bounds of the objective

Algorithm 6.1 Distributed Online Primal–Dual Dynamic Mirror Descent Algorithm

Input: nonincreasing sequences $\{\alpha_t\}$, $\{\beta_t\}$, $\{\gamma_t\} \subseteq (0, 1]$; differentiable and strongly convex functions $\{\psi_i,\ i \in [n]\}$.
Initialize: $x_{i,1} \in \mathbb{X}_i$ and $q_{i,1} = \mathbf{0}_m,\ \forall i \in [n]$.
for $t = 2, \dots$ **do**
  for $i = 1, \dots, n$ in parallel **do**
    Observe $\nabla f_{i,t-1}(x_{i,t-1})$, $\nabla g_{i,t-1}(x_{i,t-1})$, $g_{i,t-1}(x_{i,t-1})$, and $r_{i,t-1}(\cdot)$;
    Determine $\Phi_{i,t}(\cdot)$;
    Update

$$\tilde{q}_{i,t} = \sum_{j=1}^{n} [W_{t-1}]_{ij} q_{j,t-1}, \tag{6.9a}$$

$$a_{i,t} = \nabla f_{i,t-1}(x_{i,t-1}) + (\nabla g_{i,t-1}(x_{i,t-1}))^\top \tilde{q}_{i,t}, \tag{6.9b}$$

$$\tilde{x}_{i,t} = \operatorname*{argmin}_{x \in \mathbb{X}_i} \{\alpha_t \langle x, a_{i,t} \rangle + \alpha_t r_{i,t-1}(x) + \mathcal{D}_{\psi_i}(x, x_{i,t-1})\}, \tag{6.9c}$$

$$b_{i,t} = \nabla g_{i,t-1}(x_{i,t-1})(\tilde{x}_{i,t} - x_{i,t-1}) + g_{i,t-1}(x_{i,t-1}), \tag{6.9d}$$

$$q_{i,t} = [\tilde{q}_{i,t} + \gamma_t(b_{i,t} - \beta_t \tilde{q}_{i,t})]_+, \tag{6.9e}$$

$$x_{i,t} = \Phi_{i,t}(\tilde{x}_{i,t}); \tag{6.9f}$$

    Broadcast $q_{i,t}$ to $\mathcal{N}_i^{\text{out}}(\mathcal{G}_t)$ and receive $q_{j,t}$ from $j \in \mathcal{N}_i^{\text{in}}(\mathcal{G}_t)$.
  end for
end for
Output: $\{x_t\}$.

and constraint functions and their subgradients need to be known in advance to choose the stepsize sequences. In order to avoid using these upper bounds, inspired by the algorithm proposed in [175], we slightly modify the dual update equation (6.7b) as

$$u_{t+1} = [u_t + \gamma_{t+1}(g_t(x_t) + \nabla g_t(x_t)(x_{t+1} - x_t) - \beta_{t+1} u_t)]_+. \tag{6.8}$$

Then we modify the centralized online primal–dual dynamic mirror descent algorithm (6.7a), (6.8), and (6.7c) to a distributed manner, which is given in pseudo-code as Algorithm 6.1. The key difficulty caused by the distributed setting is that each agent does not know the global dual variable. In order to overcome this, the consensus step (6.9a) is introduced such that each agent has an estimation of the global dual variable.

Remark 6.1. *In order to execute Algorithm 6.1, at each iteration t, each agent i needs to know the regularization function at the previous time $t - 1$, i.e., $r_{i,t-1}(\cdot)$. This is in many situations a mild assumption since regularization functions are normally predefined to influence the structure of the decision. Furthermore, $g_{i,t-1}(x_{i,t-1})$, $\nabla f_{i,t-1}(x_{i,t-1})$, and $\nabla g_{i,t-1}(x_{i,t-1})$ rather than the full knowledge of $f_{i,t-1}(\cdot)$ and $g_{i,t-1}(\cdot)$ are needed, similar to the assumption on most online algorithms in the literature, cf., [169,170,172,173,191].*

Note that the total number of iterations or any parameters related to the objective or constraint functions, such as upper bounds of the objective and constraint functions or their subgradients, are not used in the algorithm. Also note that no local information related to the primal is exchanged between the agents, but only local dual variables.

Remark 6.2. *In Algorithm 6.1, the sequences $\{\alpha_t\}$, $\{\beta_t\}$, and $\{\gamma_t\}$ play a key role in deriving the regret and constraint violation bounds. They allow the trade-off between how fast these two bounds tend to zero, as will be seen in the next section. With some modifications, all the results in this chapter still hold if the coordinated sequences $\{\alpha_t\}$, $\{\beta_t\}$, and $\{\gamma_t\}$ are replaced by uncoordinated ones $\{\alpha_{i,t}\}$, $\{\beta_{i,t}\}$, and $\{\gamma_{i,t}\}$, respectively.*

The minimization problem (6.9c) is the composite objective mirror descent [316] and is strongly convex, so it is solvable at a linear convergence rate and closed-form solutions are available in special cases. For example, if $r_{i,t}$ is a constant mapping and Euclidean distance is used as the Bregman distance, i.e., $\mathcal{D}_{\psi_i}(x,y) = \|x-y\|^2$, then (6.9c) can be solved by the projection $\tilde{x}_{i,t} = \mathcal{P}_{\mathbb{X}_i}(x_{i,t-1} - \frac{\alpha_t}{2}a_{i,t})$. One mild assumption on the Bregman divergence is stated as follows.

Assumption 6.3. *For all $i \in [n]$, function $\psi_i : \mathbb{R}^{p_i} \to \mathbb{R}$ is differentiable and strongly convex with convexity parameter $\sigma_i > 0$ on $\mathbb{X}_i$. Moreover, for all $y \in \mathbb{X}_i$, $\mathcal{D}_{\psi_i}(\cdot, y) : \mathbb{R}^{p_i} \to \mathbb{R}$ is Lipschitz-continuous on $\mathbb{X}_i$, i.e., there exists a constant $K > 0$ such that*

$$|\mathcal{D}_{\psi_i}(x_1,y) - \mathcal{D}_{\psi_i}(x_2,y)| \le K\|x_1 - x_2\|,\ \forall x_1, x_2 \in \mathbb{X}_i. \tag{6.10}$$

This assumption is satisfied when ψ_i is Lipschitz-continuous on $\mathbb{X}_i$. From Assumptions 6.2 and 6.3, it follows that

$$\mathcal{D}_{\psi_i}(x,y) \le d(\mathbb{X})K,\ \forall x, y \in \mathbb{X}_i,\ \forall i \in [n]. \tag{6.11}$$

The dynamic mapping $\Phi_{i,t}$ used in (6.9f) plays the role of a prediction, which is a decentralized variant of the dynamical model Φ_t introduced in [313] and a generalization of the time-invariant linear mapping A used in [181]. If the optimal sequence of agent i has the dynamics $x^*_{i,t} = \Phi^*_{i,t}(x^*_{i,t-1})$ for some true dynamic mapping $\Phi^*_{i,t} : \mathbb{X}_i \to \mathbb{X}_i$, then $\Phi_{i,t}$ can be viewed as an estimate of $\Phi^*_{i,t}$. If $\Phi_{i,t}$ is equal or close enough to $\Phi^*_{i,t}$, then $x^*_{i,t} - \Phi_{i,t}(x^*_{i,t-1}) = \Phi^*_{i,t}(x^*_{i,t-1}) - \Phi_{i,t}(x^*_{i,t-1})$ is small. $\Phi_{i,t}$ is chosen as the identity mapping if at time t agent i has no knowledge about the dynamics of the optimal sequence. The following assumption on the dynamic mapping $\Phi_{i,t}$ is needed.

Assumption 6.4. *For any $t \in \mathbb{N}_+$ and $i \in [n]$, the dynamic mapping $\Phi_{i,t}$ is nonexpansive, i.e.,*

$$\mathcal{D}_{\psi_i}(\Phi_{i,t}(x), \Phi_{i,t}(y)) \le \mathcal{D}_{\psi_i}(x,y),\ \forall x, y \in \mathbb{X}_i. \tag{6.12}$$

The assumption is used to exclude the situation that any poor prediction made at one step could be exacerbated as the algorithm moves forward. The same assumption can also be found in [181, 313]. An example of the mapping $\Phi_{i,t}$ that satisfies his assumption is the identity mapping.

6.3.2 Regret and constraint violation bounds

This section presents the main results on regret and constraint violation bounds for Algorithm 6.1, but first some preliminary results are given.

Preliminary results

Firstly, we state some results on the local dual variables.

Lemma 6.1. *Suppose that Assumptions 6.1–6.2 hold. For all $i \in [n]$ and $t \in \mathbb{N}_+$, $\tilde{q}_{i,t}$ and $q_{i,t}$ generated by Algorithm 6.1 satisfy*

$$\|q_{i,t}\| \leq \frac{F}{\beta_t},\ \|\tilde{q}_{i,t+1}\| \leq \frac{F}{\beta_t}, \tag{6.13a}$$

$$\|\tilde{q}_{i,t+1} - \bar{q}_t\| \leq n\tau B_1 \sum_{s=1}^{t-1} \gamma_{s+1}\lambda^{t-1-s}, \tag{6.13b}$$

$$\frac{\Delta_{t+1}}{2\gamma_{t+1}} \leq \frac{n(B_1)^2}{2}\gamma_{t+1} + [\bar{q}_t - q]^\top g_t(x_t) + E_1(t) + E_2(t) + n\Big(\frac{G^2\alpha_{t+1}}{\underline{\sigma}} + \frac{\beta_{t+1}}{2}\Big)\|q\|^2, \tag{6.13c}$$

where q is an arbitrary vector in $\mathbb{R}^m_+$, w and ι are constants given in Assumption 6.1, F, G, and $d(\mathbb{X})$ are constants given in Assumption 6.2, and

$$\bar{q}_t = \frac{1}{n}\sum_{i=1}^n q_{i,t},\ \tau = \Big(1 - \frac{w}{2n^2}\Big)^{-2} > 1,\ B_1 = 2F + Gd(\mathbb{X}),\ \lambda = \Big(1 - \frac{w}{2n^2}\Big)^{1/\iota},$$

$$\Delta_t = \sum_{i=1}^n \|q_{i,t} - q\|^2 - (1 - \beta_t\gamma_t)\sum_{i=1}^n \|q_{i,t-1} - q\|^2,\ E_1(t) = n^2\tau B_1 F \sum_{s=1}^t \gamma_{s+1}\lambda^{t-s},$$

$$E_2(t) = \frac{\underline{\sigma}}{4\alpha_{t+1}}\sum_{i=1}^n \|\tilde{x}_{i,t+1} - x_{i,t}\|^2 + \sum_{i=1}^n (\tilde{q}_{i,t+1})^\top \nabla g_{i,t}(x_{i,t})(\tilde{x}_{i,t+1} - x_{i,t}),\ \underline{\sigma} = \min_{i\in[n]}\{\sigma_i\}.$$

Proof. See Section 6.6.1. □

Remark 6.3. *An upper bound of the local dual variables is given in (6.13a) even without Slater's condition. (6.13b) is a standard estimate from the consensus protocol with perturbations and time-varying communication graphs [190] and presents an upper bound on the deviation of the local estimate from the average value of the local dual variables at each iteration. (6.13c) gives an upper bound on the regularized drift of the local dual variables Δ_t, which extends Lemma 3 in [313] from a centralized setting to a distributed one.*

Next, we provide an upper bound on the regret for one update step.

Lemma 6.2. *Suppose that Assumptions 6.1–6.4 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 6.1 and $\{y_t\}$ be an arbitrary sequence in X, then*

$$[\bar{q}_t]^\top g_t(x_t) + l_t(x_t) - l_t(y_t) \leq [\bar{q}_t]^\top \frac{4nG^2\alpha_{t+1}}{\underline{\sigma}} + \frac{K}{\alpha_{t+1}}\sum_{i=1}^n \|y_{i,t+1} - \Phi_{i,t+1}(y_{i,t})\|$$

$$+ g_t(y_t) + 2E_1(t) - E_2(t) + E_3(t),\ \forall t \in \mathbb{N}_+, \tag{6.14}$$

where K is a constant given in Lemma 6.3, and

$$E_3(t) = \frac{1}{\alpha_{t+1}} \sum_{i=1}^{n} (\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1})).$$

Proof. See Section 6.6.2. □

Finally, we derive regret and constraint violation bounds for Algorithm 6.1.

Lemma 6.3. *Suppose that Assumptions 6.1–6.4 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 6.1. Then, for any $T \in \mathbb{N}_+$ and any comparator sequence $\boldsymbol{y}_{[T]} \in \mathcal{X}_T$,*

$$\begin{aligned}\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) \le{}& \frac{KV_\Phi(\boldsymbol{y}_{[T]})}{\alpha_T} - \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} + \beta_{t+1}\Big)\|q_{i,t}\|^2 \\ &+ C_{1,1}\sum_{t=1}^{T}\gamma_{t+1} + C_{1,2}\sum_{t=1}^{T}\alpha_{t+1} + \sum_{t=1}^{T}E_3(t),\end{aligned} \tag{6.15a}$$

$$\begin{aligned}\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|^2 \le{}& E_4(T)\Big(2nFT + \frac{KV_\Phi^*}{\alpha_T} - \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} + \beta_{t+1}\Big)\|q_{i,t} - q_c\|^2 \\ &+ C_{1,1}\sum_{t=1}^{T}\gamma_{t+1} + C_{1,2}\sum_{t=1}^{T}\alpha_{t+1} + \sum_{t=1}^{T}E_3(t)\Big),\end{aligned} \tag{6.15b}$$

where $V_\Phi(\boldsymbol{y}_{[T]})$ and V_Φ^* are the accumulated dynamic variation of the sequence $\boldsymbol{y}_{[T]}$ with respect to $\{\Phi_{i,t}\}$ and the minimum accumulated dynamic variation of all feasible sequences, respectively, defined as

$$V_\Phi(\boldsymbol{y}_{[T]}) = \sum_{t=1}^{T-1}\sum_{i=1}^{n}\|y_{i,t+1} - \Phi_{i,t+1}(y_{i,t})\|,\ V_\Phi^* = \min_{\boldsymbol{y}_{[T]}\in\mathcal{X}_T} V_\Phi(\boldsymbol{y}_{[T]}),$$

and

$$C_{1,1} = \frac{3n^2\tau B_1 F}{1-\lambda} + \frac{n(B_1)^2}{2},\ C_{1,2} = \frac{4nG^2}{\underline{\sigma}},$$
$$q_c = \frac{2[\sum_{t=1}^{T} g_t(x_t)]_+}{E_4(T)},\ E_4(T) = 4n\Big(\frac{1}{\gamma_1} + \sum_{t=1}^{T}\Big(\frac{G^2\alpha_{t+1}}{\underline{\sigma}} + \frac{\beta_{t+1}}{2}\Big)\Big).$$

Proof. See Section 6.6.3. □

Remark 6.4. *Note that the dependence on the stepsize sequences, the accumulated dynamic variation of the comparator sequence, the number of agents, and the network connectivity is characterized in* (6.15a) *and* (6.15b). *The accumulated variation of* *constraints or the pointwise maximum variation of consecutive constraints defined in [174] do, however, not appear in* (6.15a) *and* (6.15b). *This regret bound is the same as the regret bound achieved by the centralized dynamic mirror descent in [313], while [313] only considered static set constraints. The term* V^*_Φ *in (6.15b) can be replaced by* $V_\Phi(\boldsymbol{y}_{[T]})$ *due to* $V^*_\Phi \le V_\Phi(\boldsymbol{y}_{[T]})$. *Moreover, if all* $\{\Phi_{t,i}\}$ *are the identity mapping, then* $V^*_\Phi = \min_{\boldsymbol{y}_{[T]}\in\check{\mathcal{X}}_T} V_\Phi(\boldsymbol{y}_{[T]}) = V_\Phi(\check{\boldsymbol{x}}^*_T) = 0$.

In order to obtain sublinear regret and constraint violation bounds, the sequences $\{\alpha_t\}$, $\{\beta_t\}$, and $\{\gamma_t\}$ should be properly chosen. Firstly, note that α_t appears in both the denominator and numerator of (6.15a) and (6.15b), so we should let $\alpha_t = \mathcal{O}(\frac{1}{t^c})$ with $c \in (0,1)$ because otherwise one of the terms that contained α_t will grow linearly or superlinearly. Then, noting that the upper bound of the dual sequence is unclear, we should let $\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} - \beta_{t+1}\alpha_{t+1} \le 0$. In the next section, we characterize the regret and constraint violation bounds based on such sequences.

Dynamic regret and constraint violation bounds

This section states the main results on dynamic regret and constraint violation bounds for Algorithm 6.1. The succeeding theorem characterizes the bounds based on some natural decreasing stepsize sequences.

Theorem 6.1. *Suppose that Assumptions 6.1–6.4 hold. Let* $\{x_t\}$ *be the sequence generated by Algorithm 6.1 with*

$$\alpha_t = \frac{1}{t^c},\ \beta_t = \frac{1}{t^\kappa},\ \gamma_t = \frac{1}{t^{1-\kappa}},\ \forall t \in \mathbb{N}_+, \tag{6.16}$$

where $\kappa \in (0,1)$ *and* $c \in (0,1)$ *are constants. Then, for any* $T \in \mathbb{N}_+$,

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}) \le C_1 T^{\max\{1-c,c,\kappa\}} + 2KT^c V_\Phi(\boldsymbol{x}^*_{[T]}), \tag{6.17a}$$

$$\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|^2 \le C_2 T^{\max\{2-c,2-\kappa\}} + KC_{2,1}T^{\max\{1,1+c-\kappa\}}V^*_\Phi, \tag{6.17b}$$

where

$$C_1 = \frac{C_{1,1}}{\kappa} + \frac{C_{1,2}}{1-c} + 2nd(\mathbb{X})K,\ C_2 = C_{2,1}(2nF + C_1),\ C_{2,1} = 2n\Big(\frac{2G^2}{(1-c)\underline{\sigma}} + \frac{1}{1-\kappa} + 2\Big).$$

Proof. See Section 6.6.4. □

Remark 6.5. *Sublinear dynamic regret and constraint violation is thus achieved if* $V_\Phi(\boldsymbol{x}^*_{[T]})$ *grows sublinearly. If, in this case, there exists a constant* $\nu \in [0,1)$, *such that* $V_\Phi(\boldsymbol{x}^*_{[T]}) = \mathcal{O}(T^\nu)$, *then setting* $c \in (0, 1-\nu)$ *in Theorem 6.1 gives* $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}) = \mathbf{o}(T)$ *and* $\|[\sum_{t=1}^T g_t(x_t)]_+\| = \mathbf{o}(T)$. *Note that* $V_\Phi(\boldsymbol{x}^*_{[T]})$ *depends on the dynamic mapping* $\Phi_{i,t}$. *In practice, agents may not know what is a good estimate of* $\Phi_{i,t}$ *and* $\Phi_{i,t}$ *may change stochastically. It is for future research how to estimate* $\Phi_{i,t}$ *from a finite or parametric class of candidates.*

From (6.17b), we can see that the constraint violation bound is strictly greater than $O(\sqrt{T})$ due to $\max\{2-c, 2-\kappa\} > 1$. In the following we show that an $O(\sqrt{T})$ bound on constraint violation can be achieved if all $\{\Phi_{i,t}\}$ are the identity mapping and the constraint functions $\{g_{i,t}\}$ satisfy Slater's condition, which was also assumed in [174, 175].

Assumption 6.5. *(Slater's condition) There exists a constant $\varepsilon > 0$ and a vector $x_c \in X$, such that*

$$g_t(x_c) \leq -\varepsilon \mathbf{1}_m,\ t \in \mathbb{N}_+. \tag{6.18}$$

Theorem 6.2. *Suppose that Assumptions 6.1–6.5 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 6.1 with all $\{\Phi_{i,t}\}$ being the identity mapping, and*

$$\alpha_t = \frac{1}{t^{1-\kappa}},\ \beta_t = \frac{1}{t^{\kappa}},\ \gamma_t = \frac{1}{t^{1-\kappa}},\ \forall t \in \mathbb{N}_+, \tag{6.19}$$

where $\kappa \in (0,1)$. Then, for any $T \in \mathbb{N}_+$,

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}) \leq C_1 T^{\max\{1-\kappa,\kappa\}} + 2KT^{1-\kappa} V_I(x^*_T), \tag{6.20a}$$

$$\Big\| \Big[\sum_{t=1}^{T} g_t(x_t) \Big]_+ \Big\| \leq C_3 T^{\max\{1-\kappa,\kappa\}}, \tag{6.20b}$$

where $V_I(\boldsymbol{x}^*_{[T]})$ is the accumulated variation of the optimal sequence $\boldsymbol{x}^*_{[T]}$ defined ad

$$V_I(\boldsymbol{x}^*_{[T]}) = \sum_{t=1}^{T-1} \|x^*_{t+1} - x^*_t\|,$$

and

$$C_3 = n\Big(2B_2 + \frac{B_2}{1-\kappa} + \frac{G^2(B_2+2)\sqrt{m}}{\underline{\sigma}\kappa}\Big),$$

$$B_2 = \max\Big\{2\varepsilon + 2(\varepsilon^2 + nd(\mathbb{X})K)^{\frac{1}{2}},\ \frac{2B_3}{\varepsilon}\Big\},\ B_3 = 2F + C_{1,1}.$$

Proof. See Section 6.6.5. □

Remark 6.6. *From (6.20b), we note that under Slater's condition the constraint violation bound is not affected by the optimal sequences or the pointwise maximum variation of consecutive constraints, which is different from the bounds obtained in [174]. From (6.20a), it follows that sublinear dynamic regret could be achieved if $V_I(\boldsymbol{x}^*_{[T]})$ grows sublinearly with a known upper bound. Then, there exists a constant $\nu \in [0,1)$, such that $V_I(\boldsymbol{x}^*_{[T]}) = O(T^{\nu})$, so setting $\kappa \in (\nu, 1)$ in Theorem 6.2 gives* $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]}) = \mathbf{o}(T)$ *and* $\|[\sum_{t=1}^{T} g_t(x_t)]_+\| = \mathbf{o}(T)$. *Under the additional assumption that the accumulated variation of constraints, $V(\{g_t\}_{t=1}^T) = \sum_{t=1}^T \max_{x\in\mathcal{X}} \|[g_{t+1}(x) - g_t(x)]_+\|$, grows sublinearly with a known upper bound, similar results have been achieved by the modified centralized online saddle-point method proposed in [174]. However, [174] assumed not only that the time-varying* *constraint functions satisfy Slater's condition but also that the slack constant is larger than the pointwise maximum variation of consecutive constraints. The latter assumption is not always satisfied. Moreover, in [174] the total number of iterations T needs to be known in advance.*

Static regret and constraint violation bounds

This section states the main results on static regret and constraint violation bounds for Algorithm 6.1. When considering static regret, $\{\Phi_{i,t}\}$ should be set to the identity mapping since the static optimal sequence is used as the comparator sequence. In this case, replacing $\boldsymbol{x}^*_{[T]}$ by the static sequence $\check{\boldsymbol{x}}^*_T$ in Theorem 6.1 gives the following results on the bounds of static regret and constraint violation.

Corollary 6.1. *Under the same conditions as stated in Theorem 6.1 with all $\{\Phi_{i,t}\}$ being the identity mapping and $c = \kappa$, it holds that*

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) \leq C_1 T^{\max\{1-\kappa,\kappa\}}, \tag{6.21a}$$

$$\Big\| \Big[\sum_{t=1}^{T} g_t(x_t) \Big]_+ \Big\| \leq \sqrt{C_2} T^{1-\kappa/2}. \tag{6.21b}$$

Proof. Substituting $c = \kappa$ in Theorem 6.1 gives the results. □

Remark 6.7. *From Corollary 6.1, we know that Algorithm 6.1 achieves the same static regret and constraint violation bounds as in [170]. As discussed in [170], $\kappa \in (0, 1)$ is a user-defined parameter which enables the trade-off between the static regret bound and the constraint violation bound. Corollary 6.1 recovers the $\mathcal{O}(\sqrt{T})$ static regret and $\mathcal{O}(T^{3/4})$ constraint violation bounds from [169, 173] when $\kappa = 0.5$. Moreover, the result extends the $\mathcal{O}(T^{2/3})$ bound for both static regret and constraint violation achieved in [169] for linear constraint functions. However, the algorithms proposed in [169, 170, 173] are centralized and the constraint functions considered in [169, 170] are time-invariant. Moreover, in [169, 173] the total number of iterations and in [169, 170, 173] the upper bounds of the objective and constraint functions and their subgradients need to be known in advance to choose the stepsize sequences. Furthermore, Corollary 6.1 achieves smaller static regret and constraint violation bounds than [191], although [191] considered time-invariant coupled inequality constraints. However, [191] did not require the time-varying directed graph to be balanced. Although the algorithm proposed in [172] achieved more strict constraint violation bound than our Algorithm 6.1, that algorithm assumed time-invariant constraint functions and the centralized computations.*

Similarly, replacing $\boldsymbol{x}^*_{[T]}$ by the static sequence $\check{\boldsymbol{x}}^*_T$ in Theorem 6.2 gives the following results on the bounds of static regret and constraint violation.

Corollary 6.2. *Under the same conditions as stated in Theorem 6.2, it holds that*

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) \leq C_1 T^{\max\{1-\kappa,\kappa\}}, \tag{6.22a}$$

$$\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\| \leq C_3 T^{\max\{1-\kappa,\kappa\}}. \tag{6.22b}$$

Remark 6.8. *Setting $\kappa = 0.5$ in Corollary 6.2 gives $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_T) = \mathcal{O}(\sqrt{T})$ and $\|[\sum_{t=1}^{T} g_t(x_t)]_+\| = \mathcal{O}(\sqrt{T})$. Hence, Algorithm 6.1 achieves stronger results than [175] and the same results as [176, 190]. However, the algorithms proposed in [175, 176] are centralized and in [176] it is assumed that the constraint functions are independent and identically distributed. Moreover, in [190] the coupled inequality constraints are time-invariant and the boundedness of the dual variable sequence generated by the proposed algorithm is explicitly assumed.*

The static regret bounds in Corollaries 6.1 and 6.2 can be reduced, if a generalized strong convexity of the local objective functions $f_{i,t} + r_{i,t}$ is assumed. We put the strong convexity assumption on the local cost functions $f_{i,t}$ so $r_{i,t}$ can be simply convex, such as an ℓ_1-regularization.

Assumption 6.6. *For any $i \in [n]$, there exist constants $\mu_i > 0$ such that for any $t \in \mathbb{N}_+$, $f_{i,t}$ are μ_i-strongly convex on $\mathbb{X}_i$ with respect to ψ_i.*

Theorem 6.3. *Suppose Assumptions 6.1–6.4 and 6.6 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 6.1 with*

$$\alpha_t = \frac{1}{t^{\max\{1-\kappa,\kappa\}}},\ \beta_t = \frac{1}{t^{\kappa}},\ \gamma_t = \frac{1}{t^{1-\kappa}},\ \forall t \in \mathbb{N}_+, \tag{6.23}$$

where $\kappa \in (0, 1)$. Then, for any $T \in \mathbb{N}_+$,

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) \leq \max\{C_1, C_4\} T^{\kappa}, \tag{6.24a}$$

$$\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\| \leq \sqrt{C_2}\, T^{1-\kappa/2}, \tag{6.24b}$$

where

$$C_4 = \frac{n(B_1)^2}{2\kappa} + \frac{B_1 C_{1,1}}{\kappa} + \frac{C_{1,2}}{\kappa} + 2nd(\mathbb{X})K(B_4)^{1-\kappa},\ B_4 = \Big\lceil \frac{1}{(\underline{\mu})^{\frac{1}{\kappa}}} \Big\rceil,\ \underline{\mu} = \min_{i\in[n]}\{\mu_i\}.$$

Proof. See Section 6.6.6. □

Corollary 6.3. *Under the same conditions as stated in Theorem 6.2, if Assumption 6.6 also holds. Then,*

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]}) \leq C_4 T^{\kappa}, \tag{6.25a}$$

$$\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\| \leq C_3 T^{\max\{1-\kappa,\kappa\}}. \tag{6.25b}$$

Proof. (6.25a) follows from the first step in the proof of (6.24a) and (6.25b) follows from (6.20b). □

Remark 6.9. *With some minor modifications, the results stated in Theorem 6.3 and Corollary 6.3 still hold if Assumption 6.6 is replaced by the assumption that for any $i \in [n]$ and $t \in \mathbb{N}_+$, $f_{i,t}$ or $r_{i,t}$ is μ_i-strongly convex on $\mathbb{X}_i$ with respect to ψ_i.*

6.4 Simulations

This section evaluates the performance of Algorithm 6.1 in solving the multi-target tracking problem introduced in Section 1.1. In the simulations, for each agent $i \in [n]$, $\Phi_{i,t}$ is set as the identity mapping and the strongly convex function $\psi_i(x) = \sigma\|x\|^2$ is used to define the Bregman divergence $\mathcal{D}_{\psi_i}$. Thus, $\mathcal{D}_{\psi_i}(x, y) = \sigma\|x - y\|^2, \forall i \in [n]$. The stepsize sequences given (6.23) are used. Moreover, agent i could use a regularization function $r_{i,t}(x_{i,t}) = \lambda_{i,1}\|x_{i,t}\|_1 + \lambda_{i,2}\|x_{i,t}\|^2$ to influence the structure of its action, where $\lambda_{i,1}$ and $\lambda_{i,2}$ are nonnegative constants. At each time t, an undirected graph is used as the communication graph. Specifically, connections between vertices are random and the probability of two vertices being connected is ρ. To guarantee that Assumption 6.1 holds, edges $(i, i+1)$, $i \in [n-1]$ are added and $[W_t]_{ij} = \frac{1}{n}$ if $(j, i) \in \mathcal{E}_t$ and $[W_t]_{ii} = 1 - \sum_{j\in\mathcal{N}_i^{\text{in}}(\mathcal{G}_t)}[W_t]_{ij}$.

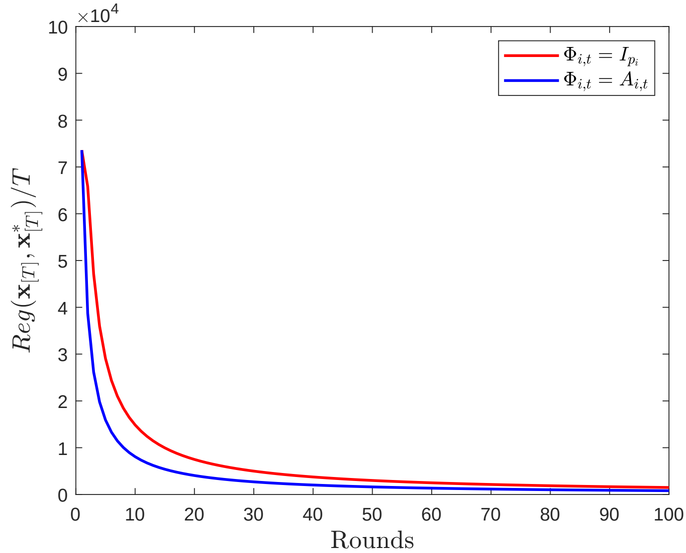
We assume $n = 50$, $m = 5$, $\sigma = 10$, $p_i = 6$, $\mathbb{X}_i = [0, 5]^{p_i}$, $\zeta_{i,1} = \lambda_{i,1} = 1$, $\zeta_{i,2} = \lambda_{i,2} = 30$, $i \in [n]$, and $\rho = 0.2$. Each component of $\pi_{i,t}$ is drawn from the discrete uniform distribution in $[0, 10]$ and each component of $D_{i,t}$ is drawn from the discrete uniform distribution in $[-5, 5]$. We let $\xi_{i,t} = (2(\zeta_{i,2} + \lambda_{i,2})x^0_{i,t} + \zeta_{i,1}\pi_{i,t} + \lambda_{i,1}\mathbf{1}_{p_i})/(2\zeta_{i,2})$, where $x^0_{i,t+1} = A_{i,t}x^0_{i,t}$ with $A_{i,t}$ being a doubly stochastic matrix and $x^0_{i,1}$ being a vector that is uniformly drawn from $\mathbb{X}_i$. In order to guarantee the constraints are feasible, we let $d_{i,t} = D_{i,t}x^0_{i,t}$.

6.4.1 Dynamics of optimal sequences

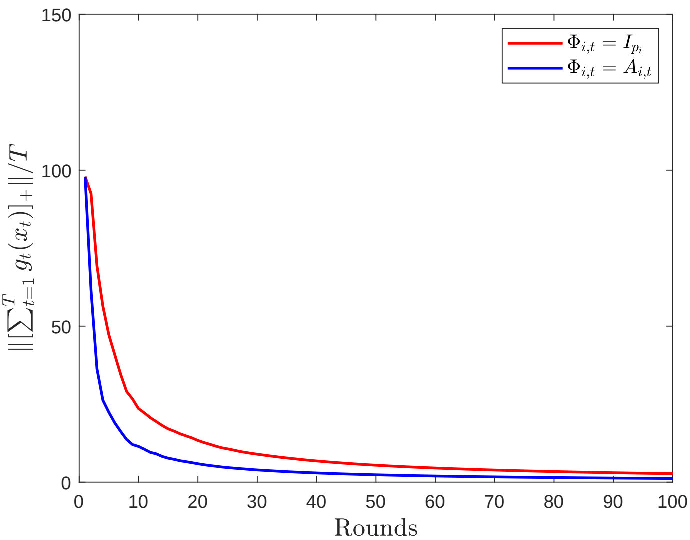
Under the above settings, we have that $x^*_{i,t} = x^0_{i,t}$. To investigate the dependence of the dynamic regret and constraint violation with $\Phi_{i,t}$, we run Algorithm 6.1 for two cases: $\Phi_{i,t}$ is the identity mapping and the linear mapping $A_{i,t}$. Figures 6.1 (a) and (b) show the evolutions of $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ and $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$, respectively, and we can see that knowing the dynamics of the optimal sequence leads to smaller dynamic regret and constraint violation.

6.4.2 Regularization function

To highlight the dependence of the dynamic regret and constraint violation with the regularization function, we run Algorithm 6.1 for two cases. Case I: $f_{i,t}(x_i) = \zeta_{i,1}\langle\pi_{i,t}, x_i\rangle + \zeta_{i,2}\|H_{i,t}x_i - y_{i,t}\|^2$, $r_{i,t}(x_i) = \lambda_{i,1}\|x_i\|_1 + \lambda_{i,2}\|x_i\|^2$ and Case II: $f_{i,t}(x_i) = \zeta_{i,1}\langle\pi_{i,t}, x_i\rangle + \zeta_{i,2}\|H_{i,t}x_i - y_{i,t}\|^2 + \lambda_{i,1}\|x_i\|_1 + \lambda_{i,2}\|x_i\|^2$, $r_{i,t}(x_i) = 0$. Figures 6.2 (a) and (b) show the evolutions of $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ and $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$, respectively, for these two cases. From these two figures, we can see that having the regularization term explicitly leads to smaller dynamic regret and constraint violation.

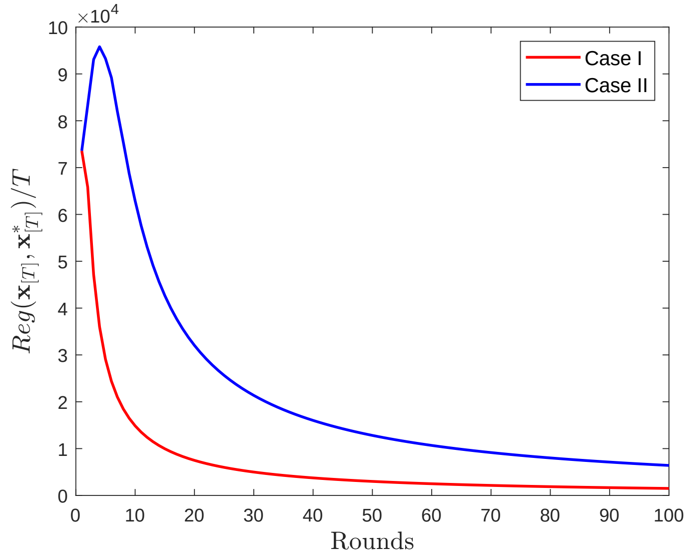


(a) Evolutions of dynamic regret $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$.

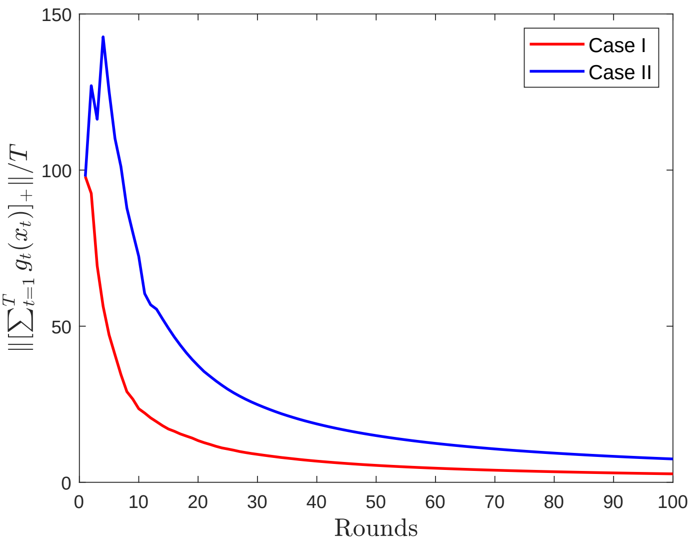


(b) Evolutions of constraint violation $\|[\sum_{t=1}^{T} g_t(x_t)]_+\|/T$.

Figure 6.1: Comparison of different $\Phi_{i,t}$ in the multi-target tracking problem.



(a) Evolutions of dynamic regret $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$.



(b) Evolutions of constraint violation $\|[\sum_{t=1}^{T} g_t(x_t)]_+\|/T$.

Figure 6.2: Effects of the regularization function in the multi-target tracking problem.

6.4.3 Effects of parameter κ

To investigate the dependence of the dynamic regret and constraint violation with the parameter κ, we run Algorithm 6.1 with $\kappa = 0.1, 0.3, 0.5, 0.7, 0.9$. Figures 6.3 (a) and (b) show effects of κ on $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ and $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$, respectively, when $T = 100, 500, 1000$. From these two figures, we can see that κ almost does not affect $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ and $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$ when T is large (e.g., $T \geq 500$). This phenomenon is not contradictory to the theoretical results shown in Theorem 6.3 since the theoretical results provide upper bounds of $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ and $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$.

6.4.4 Comparison to other algorithms

Since there are no distributed online algorithms to solve the problem of distributed online optimization with time-varying coupled inequality constraints, we compare Algorithm 6.1 with the centralized online algorithms in [173–175]. Here, Algorithm 1 in [173] with $\alpha = 10$, $\delta = 1$, and $\mu = 1/\sqrt{T}$, Algorithm 1 in [174] with $\alpha = \mu = T^{-1/3}$, and the virtual queue algorithm in [175] with $V = \sqrt{T}$ and $\alpha = V^2$ are used. Figures 6.4 (a) and (b) show the evolutions of $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ and $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$, respectively, for these algorithms. From these two figures, we can see that in this example Algorithm 6.1 achieves smaller dynamic regret and constraint violation than the algorithms in [174, 175] and almost the same values as the algorithm in [173].

6.5 Summary

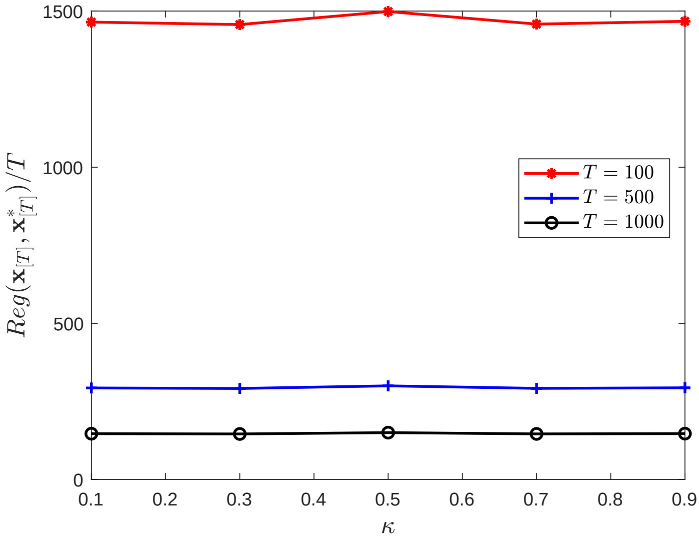
In this chapter, we considered an online convex optimization problem with time-varying coupled inequality constraints. We proposed a distributed online primal–dual dynamic mirror descent algorithm to solve this problem. We derived regret and constraint violation bounds for the algorithm and showed how they depend on the stepsize sequences, the accumulated dynamic variation of the comparator sequence, the number of agents, and the network connectivity. We proved that the algorithm achieves sublinear regret and constraint violation for both convex and strongly convex objective functions. We showed that the results in this chapter can be cast as extensions of existing literature. Future research directions include considering a strict form of the constraint violations and learning the dynamics of the optimal sequence.

6.6 Proofs

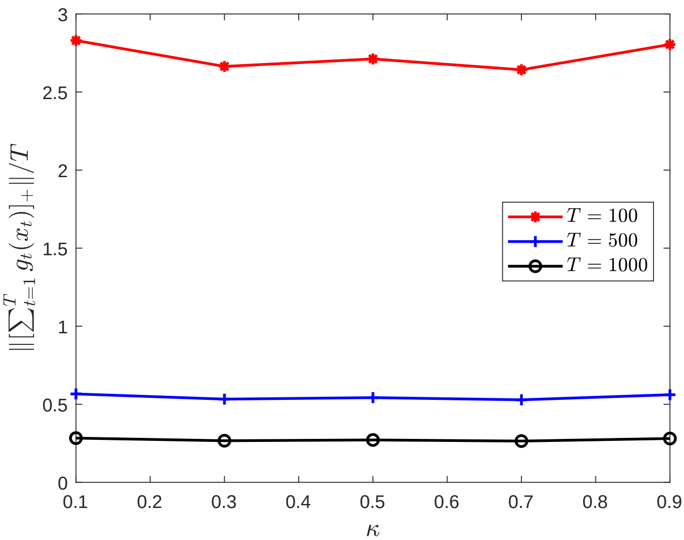
6.6.1 Proof of Lemma 6.1

(i) We prove (6.13a) by induction.

It is straightforward to see that $q_{i,1} = \tilde{q}_{i,2} = \mathbf{0}_m$, $\forall i \in [n]$, thus $\|q_{i,1}\| \leq \frac{F}{\beta_1}$, $\|\tilde{q}_{i,2}\| \leq \frac{F}{\beta_1}$, $\forall i \in [n]$. Assume that (6.13a) is true at time t for all $i \in [n]$. We show that it remains

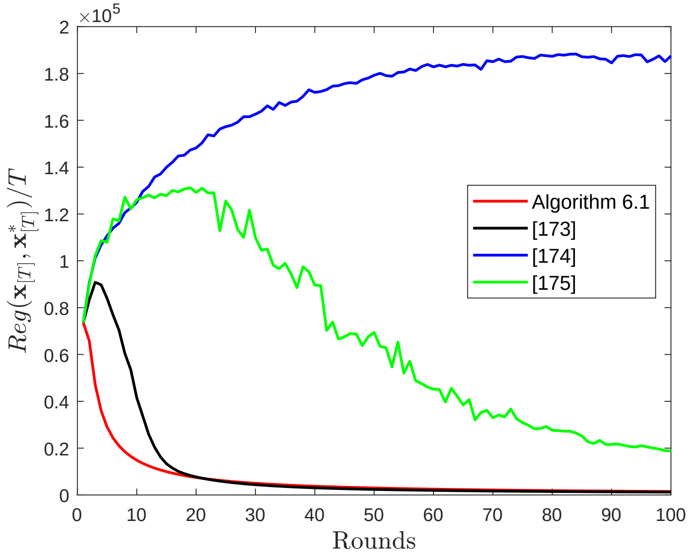


(a) Dynamic regret $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$ at $T = 100, 500, 1000$.

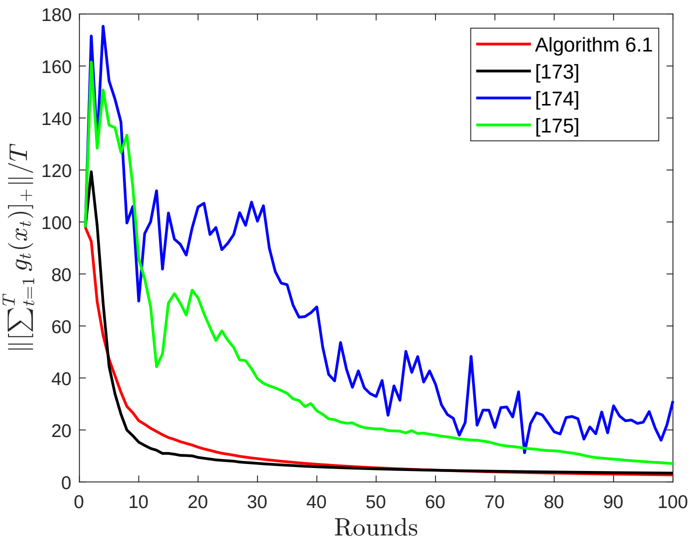


(b) Constraint violation $\|[\sum_{t=1}^T g_t(x_t)]_+\|/T$ at $T = 100, 500, 1000$.

Figure 6.3: Effects of parameter κ in the multi-target tracking problem.



(a) Evolutions of dynamic regret $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})/T$.



(b) Evolutions of constraint violation $\|[\sum_{t=1}^{T} g_t(x_t)]_+\|/T$.

Figure 6.4: Performance of online convex optimization algorithms in the multi-target tracking problem.

true at time $t+1$. (2.9) and (6.9d) imply

$$(1-\gamma_{t+1}\beta_{t+1})\tilde{q}_{i,t+1}+\gamma_{t+1}b_{i,t+1}\leq(1-\gamma_{t+1}\beta_{t+1})\tilde{q}_{i,t+1}+\gamma_{t+1}g_{i,t}(\tilde{x}_{i,t+1}). \tag{6.26}$$

Noting that $\|[x]_+\|\leq\|y\|$ for all $x\leq y$, (6.9e), (6.26), and (6.4) imply

$$\begin{aligned}\|q_{i,t+1}\|&\leq(1-\gamma_{t+1}\beta_{t+1})\|\tilde{q}_{i,t+1}\|+\gamma_{t+1}\|g_{i,t}(\tilde{x}_{i,t+1})\|\\&\leq(1-\gamma_{t+1}\beta_{t+1})\frac{F}{\beta_t}+\gamma_{t+1}F\leq(1-\gamma_{t+1}\beta_{t+1})\frac{F}{\beta_{t+1}}+\gamma_{t+1}F=\frac{F}{\beta_{t+1}},\ \forall i\in[n],\end{aligned}$$

where the last inequality holds due to the sequence $\{\beta_t\}$ is nonincreasing. The convexity of norms and $\sum_{j=1}^n[W_t]_{ij}=1$ yield

$$\|\tilde{q}_{i,t+2}\|\leq\sum_{j=1}^n[W_t]_{ij}\|q_{j,t+1}\|\leq\sum_{j=1}^n[W_t]_{ij}\frac{F}{\beta_{t+1}}=\frac{F}{\beta_{t+1}},\ \forall i\in[n].$$

Thus, (6.13a) follows.

(ii) We can rewrite (6.9e) as

$$q_{i,t+1}=\sum_{j=1}^n[W_t]_{ij}q_{j,t}+\epsilon^q_{i,t},$$

where $\epsilon^q_{i,t}=[(1-\gamma_{t+1}\beta_{t+1})\tilde{q}_{i,t+1}+\gamma_{t+1}b_{i,t+1}]_+-\tilde{q}_{i,t+1}$. From (6.4), (6.5), and (6.3), we have

$$\|b_{i,t+1}\|\leq\|g_{i,t}(x_{i,t})\|+\|\nabla g_{i,t}(x_{i,t})\|\|(\tilde{x}_{i,t+1}-x_{i,t})\|\leq F+Gd(\mathbb{X}),\ \forall i\in[n]. \tag{6.27}$$

Thus, (2.10), (6.13a), and (6.27) give

$$\|\epsilon^q_{i,t}\|\leq\|-\gamma_{t+1}\beta_{t+1}\tilde{q}_{i,t+1}+\gamma_{t+1}b_{i,t+1}\|\leq B_1\gamma_{t+1},\ \forall i\in[n]. \tag{6.28}$$

Then, Lemma 2 in [190], $q_{i,1}=\mathbf{0}_m,\ \forall i\in[n]$, and (6.28) yield

$$\|q_{i,t+1}-\bar{q}_{t+1}\|\leq n\tau B_1\sum_{s=1}^t\gamma_{s+1}\lambda^{t-s},\ \forall i\in[n].$$

So (6.13b) follows due to $\sum_{j=1}^n[W_t]_{ij}=1$ and $\|\tilde{q}_{i,t+1}-\bar{q}_t\|=\|\sum_{j=1}^n[W_t]_{ij}q_{j,t}-\bar{q}_t\|\leq\sum_{j=1}^n[W_t]_{ij}\|q_{j,t}-\bar{q}_t\|$.

(iii) Applying (2.10) to (6.9e) gives

$$\begin{aligned}\|q_{i,t}-q\|^2&\leq\|(1-\beta_t\gamma_t)\tilde{q}_{i,t}+\gamma_tb_{i,t}-q\|^2\\&=\|\tilde{q}_{i,t}-q\|^2+(\gamma_t)^2\|b_{i,t}-\beta_t\tilde{q}_{i,t}\|^2+2\gamma_t(\tilde{q}_{i,t})^\top\nabla g_{i,t-1}(x_{i,t-1})(\tilde{x}_{i,t}-x_{i,t-1})\\&\quad-2\gamma_tq^\top\nabla g_{i,t-1}(x_{i,t-1})(\tilde{x}_{i,t}-x_{i,t-1})+2\gamma_t(\tilde{q}_{i,t}-q)^\top g_{i,t-1}(x_{i,t-1})\\&\quad-2\beta_t\gamma_t(\tilde{q}_{i,t}-q)^\top\tilde{q}_{i,t}.\end{aligned} \tag{6.29}$$

For the first term of the right-hand side of (6.29), by convexity of norms and $\sum_{j=1}^{n}[W_{t-1}]_{ij} = 1$, it can be concluded that

$$\|\tilde{q}_{i,t} - q\|^2 = \Big\| \sum_{j=1}^{n} [W_{t-1}]_{ij} q_{j,t-1} - \sum_{j=1}^{n} [W_{t-1}]_{ij} q \Big\|^2 \le \sum_{j=1}^{n} [W_{t-1}]_{ij} \|q_{j,t-1} - q\|^2. \tag{6.30}$$

For the second term of the right-hand side of (6.29), (6.13a) and (6.27) yield

$$(\gamma_t)^2 \|b_{i,t} - \beta_t \tilde{q}_{i,t}\|^2 \le (B_1 \gamma_t)^2. \tag{6.31}$$

For the fourth term of the right-hand side of (6.29), (6.5) and the Cauchy-Schwarz inequality yield

$$-2\gamma_t q^\top \nabla g_{i,t-1}(x_{i,t-1})(\tilde{x}_{i,t} - x_{i,t-1}) \le 2\gamma_t \Big(\frac{G^2 \alpha_t}{\underline{\sigma}} \|q\|^2 + \frac{\underline{\sigma}}{4\alpha_t} \|\tilde{x}_{i,t} - x_{i,t-1}\|^2 \Big). \tag{6.32}$$

For the fifth term of the right-hand side of (6.29), we have

$$2\gamma_t (\tilde{q}_{i,t} - q)^\top g_{i,t-1}(x_{i,t-1}) = 2\gamma_t (\bar{q}_{t-1} - q)^\top g_{i,t-1}(x_{i,t-1}) + 2\gamma_t (\tilde{q}_{i,t} - \bar{q}_{t-1})^\top g_{i,t-1}(x_{i,t-1}). \tag{6.33}$$

Moreover, from (6.4) and (6.13b), we have

$$2\gamma_t (\tilde{q}_{i,t} - \bar{q}_{t-1})^\top g_{i,t-1}(x_{i,t-1}) \le 2\gamma_t \|\tilde{q}_{i,t} - \bar{q}_{t-1}\| \|g_{i,t-1}(x_{i,t-1})\| \le \frac{2\gamma_t E_1(t-1)}{n}. \tag{6.34}$$

For the last term of the right-hand side of (6.29), neglecting the nonnegative term $\beta_t \gamma_t \|\tilde{q}_{i,t}\|^2$ gives

$$-2\beta_t \gamma_t (\tilde{q}_{i,t} - q)^\top \tilde{q}_{i,t} \le \beta_t \gamma_t (\|q\|^2 - \|\tilde{q}_{i,t} - q\|^2). \tag{6.35}$$

Then, combining (6.29)–(6.35), summing over $i \in [n]$, and dividing by $2\gamma_t$, and using $\sum_{i=1}^{n}[W_{t-1}]_{ij} = 1,\ \forall t \in \mathbb{N}_+$ yields (6.13c).

6.6.2 Proof of Lemma 6.2

From (2.9), we have

$$\begin{aligned}
& l_{i,t}(x_{i,t}) - l_{i,t}(y_{i,t}) \\
&= f_{i,t}(x_{i,t}) - f_{i,t}(y_{i,t}) + r_{i,t}(x_{i,t}) - r_{i,t}(\tilde{x}_{i,t+1}) + r_{i,t}(\tilde{x}_{i,t+1}) - r_{i,t}(y_{i,t}) \\
&\le \langle \nabla f_{i,t}(x_{i,t}), x_{i,t} - y_{i,t} \rangle + \langle \nabla r_{i,t}(x_{i,t}), x_{i,t} - \tilde{x}_{i,t+1} \rangle + \langle \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t} \rangle \\
&= \langle \nabla f_{i,t}(x_{i,t}) + \nabla r_{i,t}(x_{i,t}), x_{i,t} - \tilde{x}_{i,t+1} \rangle + \langle \nabla f_{i,t}(x_{i,t}) + \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t} \rangle.
\end{aligned} \tag{6.36}$$

We now bound each of the two terms above. For the first term, (6.5) and the Cauchy-Schwarz inequality give

$$\langle \nabla f_{i,t}(x_{i,t}) + \nabla r_{i,t}(x_{i,t}), x_{i,t} - \tilde{x}_{i,t+1} \rangle \le 2G \|x_{i,t} - \tilde{x}_{i,t+1}\| \le \frac{\underline{\sigma}}{4\alpha_{t+1}} \|x_{i,t} - \tilde{x}_{i,t+1}\|^2 + \frac{4G^2 \alpha_{t+1}}{\underline{\sigma}}. \tag{6.37}$$

For the second term, we have

$$\begin{aligned}
&\langle \nabla f_{i,t}(x_{i,t}) + \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t}\rangle \\
&= \langle (\nabla g_{i,t}(x_{i,t}))^\top \tilde{q}_{i,t+1}, y_{i,t} - \tilde{x}_{i,t+1}\rangle + \langle a_{i,t+1} + \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t}\rangle \\
&= \langle (\nabla g_{i,t}(x_{i,t}))^\top \tilde{q}_{i,t+1}, y_{i,t} - x_{i,t}\rangle + \langle (\nabla g_{i,t}(x_{i,t}))^\top \tilde{q}_{i,t+1}, x_{i,t} - \tilde{x}_{i,t+1}\rangle \\
&\quad + \langle a_{i,t+1} + \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t}\rangle. \qquad (6.38)
\end{aligned}$$

From (2.9) and $\tilde{q}_{i,t} \geq \mathbf{0}_m,\ \forall t \in \mathbb{N}_+,\ \forall i \in [n]$, we have

$$\begin{aligned}
&\langle (\nabla g_{i,t}(x_{i,t}))^\top \tilde{q}_{i,t+1}, y_{i,t} - x_{i,t}\rangle \leq (\tilde{q}_{i,t+1})^\top g_{i,t}(y_{i,t}) - (\tilde{q}_{i,t+1})^\top g_{i,t}(x_{i,t}) \\
&= (\bar{q}_t)^\top (g_{i,t}(y_{i,t}) - g_{i,t}(x_{i,t})) + (\tilde{q}_{i,t+1} - \bar{q}_t)^\top (g_{i,t}(y_{i,t}) - g_{i,t}(x_{i,t})). \qquad (6.39)
\end{aligned}$$

Similar to (6.34), we have

$$(\tilde{q}_{i,t+1} - \bar{q}_t)^\top (g_{i,t}(y_{i,t}) - g_{i,t}(x_{i,t})) \leq \frac{2E_1(t)}{n}. \qquad (6.40)$$

Applying (2.21) to the update rule (6.9c), we get

$$\begin{aligned}
&\langle a_{i,t+1} + \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t}\rangle \\
&\leq \frac{1}{\alpha_{t+1}}(\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \mathcal{D}_{\psi_i}(y_{i,t}, \tilde{x}_{i,t+1}) - \mathcal{D}_{\psi_i}(\tilde{x}_{i,t+1}, x_{i,t})) \\
&= \frac{1}{\alpha_{t+1}}(\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1}) + \mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1}) - \mathcal{D}_{\psi_i}(\Phi_{i,t+1}(y_{i,t}), x_{i,t+1}) \\
&\quad + \mathcal{D}_{\psi_i}(\Phi_{i,t+1}(y_{i,t}), x_{i,t+1}) - \mathcal{D}_{\psi_i}(y_{i,t}, \tilde{x}_{i,t+1}) - \mathcal{D}_{\psi_i}(\tilde{x}_{i,t+1}, x_{i,t})) \\
&\leq \frac{1}{\alpha_{t+1}}\Big(\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1}) + K\|y_{i,t+1} - \Phi_{i,t+1}(y_{i,t})\| - \frac{\sigma}{2}\|\tilde{x}_{i,t+1} - x_{i,t}\|^2\Big), \\
&\qquad (6.41)
\end{aligned}$$

where the last inequality holds due to (6.9f), (6.12), (6.10), and (2.18).

Combining (6.36)–(6.41) and summing over $i \in [n]$ yields (6.14).

6.6.3 Proof of Lemma 6.3

(i) The definition of Δ_t given in Lemma 6.1 yields

$$\begin{aligned}
-\frac{\Delta_t}{2\gamma_t} &= \frac{1}{2\gamma_t}\sum_{i=1}^{n}((1 - \beta_t\gamma_t)\|q_{i,t-1} - q\|^2 - \|q_{i,t} - q\|^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_{t-1}}\|q_{i,t-1} - q\|^2 - \frac{1}{\gamma_t}\|q_{i,t} - q\|^2\Big) \\
&\quad + \frac{1}{2}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_t} - \frac{1}{\gamma_{t-1}} - \beta_t\Big)\|q_{i,t-1} - q\|^2. \qquad (6.42)
\end{aligned}$$

For any nonnegative sequence $\zeta_1, \zeta_2, \dots$, it holds that

$$\sum_{t=1}^{T}\sum_{s=1}^{t}\zeta_{s+1}\lambda^{t-s} = \sum_{t=1}^{T}\zeta_{t+1}\sum_{s=0}^{T-t}\lambda^{s} \le \frac{1}{(1-\lambda)}\sum_{t=1}^{T}\zeta_{t+1}. \tag{6.43}$$

Let $g_c : \mathbb{R}^m_+ \to \mathbb{R}$ be a function defined as

$$g_c(q) = \Big(\sum_{t=1}^{T} g_t(x_t)\Big)^{\top} q - n\Big(\frac{1}{\gamma_1} + \sum_{t=1}^{T}\Big(\frac{G^2\alpha_{t+1}}{\underline{\sigma}} + \frac{\beta_{t+1}}{2}\Big)\Big)\|q\|^2. \tag{6.44}$$

Combining (6.13c) and (6.14), summing over $t \in [T]$, neglecting the nonnegative term $\|q_{i,T+1} - q\|^2$, and using (6.42)–(6.44), $\|q_{i,1} - q\|^2 \le 2\|q_{i,1}\|^2 + 2\|q\|^2 = 2\|q\|^2$, and $g_t(y_t) \le \mathbf{0}_m$, $\boldsymbol{y}_{[T]} \in \mathcal{X}_T$ yields

$$\begin{aligned} g_c(q) + \mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) \le{}& C_{1,1}\sum_{t=1}^{T}\gamma_{t+1} + \frac{4nG^2}{\underline{\sigma}}\sum_{t=1}^{T}\alpha_{t+1} + \sum_{t=1}^{T}E_3(t) \\ &- \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} + \beta_{t+1}\Big)\|q_{i,t} - q\|^2 \\ &+ K\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{\|y_{i,t+1} - \Phi_{i,t+1}(y_{i,t})\|}{\alpha_{t+1}}, \ \forall q \in \mathbb{R}^m_+. \end{aligned} \tag{6.45}$$

Then, substituting $q = \mathbf{0}_m$ into (6.45), setting $y_{i,T+1} = \Phi_{i,T+1}(y_{i,T})$, noting that $\{\alpha_t\}$ is nonincreasing, and rearranging the terms yields (6.15a).
(ii) Substituting $q = q_c$ into $g_c(q)$ gives

$$g_c(q_c) = \frac{\|[\sum_{t=1}^{T} g_t(x_t)]_+\|^2}{E_4(T)}. \tag{6.46}$$

Moreover, (6.4) gives

$$|\,\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]})| \le 2nFT, \ \forall \boldsymbol{y}_{[T]} \in \mathcal{X}_T. \tag{6.47}$$

Substituting $q = q_c$ into (6.45), combining (6.46)–(6.47), and rearranging the terms gives (6.15b).

6.6.4 Proof of Theorem 6.1

(i) Applying (2.37) to the third and forth terms of the right-hand side of (6.15a) gives

$$C_{1,1}\sum_{t=1}^{T}\gamma_{t+1} \le \frac{C_{1,1}}{\kappa}T^{\kappa}, \tag{6.48a}$$

$$C_{1,2}\sum_{t=1}^{T}\alpha_{t+1} \le \frac{C_{1,2}}{1-c}T^{1-c}. \tag{6.48b}$$

Noting that $\{\alpha_t\}$ is nonincreasing and (6.11), for any $s \in [T]$, we have

$$\begin{aligned}\sum_{t=s}^{T} E_3(t) &= \sum_{t=s}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_t}\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \frac{1}{\alpha_{t+1}}\mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1})\Big)\\ &\quad + \sum_{t=s}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t}\Big)\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t})\\ &\le \frac{1}{\alpha_s}\sum_{i=1}^{n}\mathcal{D}_{\psi_i}(y_{i,s}, x_{i,s}) - \frac{1}{\alpha_{T+1}}\sum_{i=1}^{n}\mathcal{D}_{\psi_i}(y_{i,T+1}, x_{i,T+1}) + n\Big(\frac{1}{\alpha_{T+1}} - \frac{1}{\alpha_s}\Big)d(\mathbb{X})K\\ &\le \frac{nd(\mathbb{X})K}{\alpha_{T+1}}. \end{aligned} \tag{6.49}$$

Combining (6.15a) and (6.48a)–(6.49), setting $y_{i,t} = x^*_{i,t}$, $\forall t \in [T]$, and noting that the second last term of the right-hand side of (6.15a) is nonpositive due to $\frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} + \beta_{t+1} > 0$ yield (6.17a).

(ii) Using (2.37) gives

$$E_4(T) \le C_{2,1}T^{\max\{1-c,1-\kappa\}}. \tag{6.50}$$

Combining (6.15b) and (6.48a)–(6.50) and noting that the last term of the right-hand side of (6.15b) is nonpositive due to $\frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} + \beta_{t+1} > 0$ give (6.17b).

6.6.5 Proof of Theorem 6.2

(i) Substituting $c = 1 - \kappa$ in (6.17a) gives (6.20a).

(ii) We first show that $\|q_t\| \le B_2$ by induction, where $q_t = \mathrm{col}(q_{1,t}, \ldots, q_{n,t})$.

It is straightforward to see that $\|q_1\| = 0 \le B_2$. Suppose that there exists $T_1 \in \mathbb{N}_+$ such that $\|q_t\| \le B_2$, $\forall t \in [T_1]$. We show that $\|q_{T_1+1}\| \le B_2$ by contradiction. Now suppose that $\|q_{T_1+1}\| > B_2$. Noting that $\|\bar{q}_{T_1+1}\|_1 = \|q_{T_1+1}\|_1 \ge \|q_{T_1+1}\| > B_2$ and $\|\bar{q}_1\|_1 = 0$, we know that there exists $t_0 \in [T_1]$ such that $\|\bar{q}_{t_0}\|_1 \le \frac{B_2}{2}$. Let $t_1 = \max\{t_0 : \|\bar{q}_{t_0}\|_1 \le \frac{B_2}{2}, \ t_0 \in [T_1]\}$. Combining (6.13c) and (6.14), substituting $q = \mathbf{0}_m$ and $y_t = x_c$, setting $\{\Phi_{t,i}\}$ as the identity mapping, and using $|l_t(x_t) - l_t(x_c)| \le 2F$ and (6.18) yields

$$\|q_{t+1}\|^2 - (1 - \beta_{t+1}\gamma_{t+1})\|q_t\|^2 \le 2B_3\gamma_{t+1} + 2\gamma_{t+1}E_3(t+1) - 2\varepsilon\|\bar{q}_t\|_1\gamma_{t+1}. \tag{6.51}$$

Summing (6.51) over $t \in \{t_1, \ldots, T_1\}$, using (6.11), $\alpha_t = \gamma_t = \frac{1}{t^{1-\kappa}}$ and $\beta_t \ge 0$, and noting that $\|q_{T_1+1}\| > B_2$, $\|q_{t_1}\| \le \|\bar{q}_{t_1}\|_1 \le \frac{B_2}{2}$, and $\|\bar{q}_t\|_1 > \frac{B_2}{2}$, $\forall t \in \{t_1 + 1, \ldots, T_1\}$ gives

$$\begin{aligned}\frac{3(B_2)^2}{4} &< \|q_{T_1+1}\|^2 - \|q_{t_1}\|^2 + \sum_{t=t_1}^{T_1}\beta_{t+1}\gamma_{t+1}\|q_t\|^2\\ &\le 2B_3\sum_{t=t_1}^{T_1}\gamma_{t+1} + 2nd(\mathbb{X})K - 2\varepsilon\sum_{t=t_1}^{T_1}\|\bar{q}_t\|_1\gamma_{t+1}\end{aligned}$$

$$
\begin{aligned}
&\le \frac{2B_3}{\kappa}((T_1+1)^\kappa - (t_1+1)^\kappa) + 2B_3 + 2nd(\mathbb{X})K \\
&\quad - \frac{\varepsilon B_2}{\kappa}((T_1+1)^\kappa - (t_1+1)^\kappa) + \varepsilon B_2 - 2\varepsilon\|\bar{q}_{t_1}\|_1 \\
&\le 2nd(\mathbb{X})K + 2\varepsilon B_2 \le \frac{(B_2)^2}{2},
\end{aligned} \tag{6.52}
$$

which is a contradiction. Thus, $\|q_{T_1+1}\| \le B_2$.

We now show (6.20b) holds. Applying (2.22) to the update (6.9c) and noting $\|\tilde{q}_{i,t+1}\| \le \|q_t\| \le B_2$ gives

$$
\|\tilde{x}_{i,t+1} - x_{i,t}\| \le \frac{\|\alpha_{t+1}a_{i,t+1}\| + \alpha_{t+1}G}{\underline{\sigma}} \le \frac{G\alpha_{t+1}}{\underline{\sigma}}(B_2+2). \tag{6.53}
$$

(6.9a) and (6.9e) give

$$
q_{i,t+1} \ge (1-\beta_{t+1}\gamma_{t+1})\sum_{j=1}^{n}[W_t]_{ij}q_{j,t} + \gamma_{t+1}b_{i,t+1}. \tag{6.54}
$$

Summing (6.54) over $i \in [n]$, dividing by $n\gamma_{t+1}$, and using $\sum_{i=1}^{n}[W_t]_{ij} = 1,\ \forall t \in \mathbb{N}_+$, (6.5), (6.9d), and (6.53) yields

$$
\begin{aligned}
\frac{\bar{q}_{t+1}}{\gamma_{t+1}} &\ge \Big(\frac{1}{\gamma_{t+1}} - \beta_{t+1}\Big)\bar{q}_t + \frac{1}{n}\sum_{i=1}^{n} b_{i,t+1} \\
&\ge \Big(\frac{1}{\gamma_{t+1}} - \beta_{t+1}\Big)\bar{q}_t + \frac{1}{n}g_t(x_t) - \frac{G^2\alpha_{t+1}}{\underline{\sigma}}(B_2+2)\mathbf{1}_m.
\end{aligned} \tag{6.55}
$$

Summing (6.55) over $t \in [T]$ gives

$$
\frac{1}{n}\sum_{t=1}^{T} g_t(x_t) \le \frac{\bar{q}_{T+1}}{\gamma_{T+1}} + \sum_{t=1}^{T}\beta_{t+1}\bar{q}_t + \sum_{t=1}^{T}\frac{G^2\alpha_{t+1}}{\underline{\sigma}}(B_2+2)\mathbf{1}_m. \tag{6.56}
$$

Noting that $\|[x]_+\| \le \|y\|$ for all $x \le y$ and using $\|\bar{q}_t\| \le \|q_t\| \le B_2$ and (2.37) yields (6.20b).

6.6.6 Proof of Theorem 6.3

(i) We first show that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}_T^*) \le C_4 T^\kappa$ when $\alpha_t = \frac{1}{t^{1-\kappa}}$.

Under Assumption 6.6, (6.36) can be replaced by

$$
\begin{aligned}
l_{i,t}(x_{i,t}) - l_{i,t}(y_{i,t}) &\le \langle \nabla f_{i,t}(x_{i,t}), x_{i,t} - y_{i,t}\rangle + \langle \nabla r_{i,t}(x_{i,t}), x_{i,t} - \tilde{x}_{i,t+1}\rangle \\
&\quad + \langle \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t}\rangle - \underline{\mu}\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) \\
&= \langle \nabla f_{i,t}(x_{i,t}) + \nabla r_{i,t}(x_{i,t}), x_{i,t} - \tilde{x}_{i,t+1}\rangle \\
&\quad + \langle \nabla f_{i,t}(x_{i,t}) + \nabla r_{i,t}(\tilde{x}_{i,t+1}), \tilde{x}_{i,t+1} - y_{i,t}\rangle - \underline{\mu}\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}).
\end{aligned} \tag{6.57}
$$

Thus, (6.14)–(6.15b) still hold if replacing $E_3(t)$ by

$$E_5(t) = \sum_{i=1}^{n}\Big(\frac{1}{\alpha_{t+1}}(\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1})) - \underline{\mu}\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t})\Big).$$

Then,

$$\begin{aligned}\sum_{t=1}^{T} E_5(t) = &\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_t}\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \frac{1}{\alpha_{t+1}}\mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1})\Big)\\ &+ \sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \underline{\mu}\Big)\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}).\end{aligned} \tag{6.58}$$

Noting that $\underline{\mu} > 0$, $\mathcal{D}_{\psi_i}(\cdot,\cdot) \geq 0$, and $\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \underline{\mu} = \frac{t+1}{(t+1)^\kappa} - \frac{t}{t^\kappa} - \underline{\mu} < \frac{1}{t^\kappa} - \underline{\mu} \leq 0,\ \forall t \geq B_4$ and using (6.49) and (6.58) yields

$$\begin{aligned}\sum_{t=1}^{T} E_5(t) &= \sum_{t=1}^{B_4-1} E_3(t) + \sum_{t=B_4}^{T} E_5(t)\\ &\leq \frac{nd(\mathbb{X})K}{\alpha_{B_4}} + \sum_{t=B_4}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_{t+1}} - \frac{1}{\alpha_t} - \underline{\mu}\Big)\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t})\\ &\quad + \sum_{t=B_4}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_t}\mathcal{D}_{\psi_i}(y_{i,t}, x_{i,t}) - \frac{1}{\alpha_{t+1}}\mathcal{D}_{\psi_i}(y_{i,t+1}, x_{i,t+1})\Big)\\ &\leq \frac{2nd(\mathbb{X})K}{\alpha_{B_4}}.\end{aligned} \tag{6.59}$$

Replacing (6.49) with (6.59) and along the same line as the proof of (6.17a) in Theorem 6.1 gives that $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}_T^*) \leq C_4 T^\kappa$ when $\alpha_t = \frac{1}{t^{1-\kappa}}$.

Next, we show that (6.24a) holds. When $\kappa \in (0, 0.5)$, we have $\alpha_t = 1/t^{(1-\kappa)}$. Thus, from the above result, we have $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}_T^*) \leq C_4 T^\kappa$. When $\kappa \in [0.5, 1)$, we have $\alpha_t = 1/t^\kappa$. Thus, (6.21a) gives $\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}_T^*) \leq C_1 T^\kappa$. In conclusion, (6.24a) holds.

(ii) Substituting $c = 1 - \kappa$ when $\kappa \in (0, 0.5)$ and $c = \kappa$ when $\kappa \in [0.5, 1)$ in (6.17b) gives (6.24b).

Chapter 7

Distributed bandit online primal–dual optimization algorithms

In this chapter, distributed bandit online convex optimization with time-varying coupled inequality constraints is considered, motivated by a repeated game between a group of learners and an adversary. The learners attempt to minimize a sequence of global loss functions and at the same time satisfy a sequence of coupled constraint functions, where the constraints are coupled across the distributed learners at each round. The global loss and the coupled constraint functions are the sum of local convex loss and constraint functions, respectively, which are adaptively generated by the adversary. The local loss and constraint functions are revealed in a bandit manner, i.e., only the values of loss and constraint functions are revealed to the learners at the sampling instance, and the revealed function values are held privately by each learner. Both one- and two-point bandit feedback are studied with the two corresponding distributed bandit online algorithms used by the learners. We show that sublinear expected dynamic regret and constraint violation are achieved by these two algorithms, if the accumulated variation of the comparator sequence also grows sublinearly. In particular, we show that $\mathcal{O}(T^{\theta_1})$ expected static regret and $\mathcal{O}(T^{7/4-\theta_1})$ constraint violation bounds are achieved in the one-point bandit feedback setting, and $\mathcal{O}(T^{\max\{\kappa,1-\kappa\}})$ expected static regret and $\mathcal{O}(T^{1-\kappa/2})$ constraint violation bounds in the two-point bandit feedback setting, where T is the total number of rounds and $\theta_1 \in (3/4, 5/6]$ and $\kappa \in (0, 1)$ are user-defined trade-off parameters. The effectiveness of the theoretical results is illustrated by numerical simulations.

This chapter is organized as follows. Section 7.1 introduces the background. Section 7.2 gives the problem formulation. Sections 7.3 and 7.4 provide the distributed bandit online algorithms for one- and two-point bandit feedback, respectively, and present their expected regret and constraint violation bounds. Section 7.5 presents numerical simulations. Section 7.6 concludes this chapter. To improve the readability, all the proofs can be found in Section 7.7.

7.1 Introduction

Bandit online convex optimization is online convex optimization with bandit feedback, i.e., at each round only the values of the loss functions are revealed, rather than the entire loss function, the gradient of the loss function, or some other information. Bandit feedback is suitable to model various applications, where the entire function or gradient information is not available, such as online source localization, online routing in data networks, and online advertisement placement in web search [165]. For such applications, existing online algorithms are inapplicable but gradient-free (zeroth-order) optimization methods are needed. Gradient-free optimization methods have a long history [271] and have an evident advantage since computing a function value is much simpler than computing its gradient. Gradient-free optimization methods have gained renewed interests in recent years, e.g., [147, 150, 213, 243]. Essentially, a bandit online convex optimization algorithm is a gradient-free method to solve online convex optimization problems. In a bandit setting, a sublinear static regret bound may not be guaranteed if the adversary still can arbitrarily choose the loss function. Under completely adaptive adversary, the authors of [164] gave an example to show that any algorithm suffer at least linear regret. Therefore, the power of the adversary should be limited to achieve a sublinear regret bound. For a so called adaptive adversary [164], the adversary chooses f_t based only on the learner's past decisions $x_1, \dots, x_{t-1}$, but not on her current decision x_t. In other words, the adversary chooses f_t at the beginning of round t, before the learner chooses her decision.

A key step in bandit online convex optimization is to estimate the gradient of the loss function by sampling the loss function. Various algorithms have been developed and can be divided into two categories depending on the number of samplings. Algorithms with one sampling at each round have been proposed in [212, 317–324]. Specifically, in [212], $O(T^{3/4})$ expected static regret bound was achieved for Lipschitz-continuous functions, where T is the total number of rounds. Better regret bounds can be guaranteed if additional assumptions are made. The authors of [317] considered linear loss functions and achieved $O(\sqrt{T})$ expected static regret bound. The authors of [318, 319] also considered linear loss functions and proposed algorithms that achieved $O(\sqrt{T\log(T)})$ expected static regret bound. The authors of [320] studied smooth loss functions and achieved $O(T^{2/3}(\log(T))^{1/3})$ expected static regret bound. The authors of [321] considered strongly convex and smooth loss functions and achieved $O(\sqrt{T\log(T)})$ expected static regret bound. One common assumption in [318–321] is that the convex domain admits a self-concordant barrier. The authors of [322] showed that $O(\sqrt{T}\log(T))$ expected static regret bound can be achieved for Lipschitz-continuous loss functions with one-dimensional domains, but they did not develop any explicit algorithm. This result was extended to arbitrary dimensions in [323], but still without any explicit algorithm. Based on the application of the ellipsoid method to online learning, the authors of [324] proposed an algorithm for Lipschitz-continuous loss functions and achieved $O(\sqrt{T}\log(T))$ expected static regret bound.

Algorithms with two or more samplings at each round have been proposed in [164, 214, 280, 325–327]. The expected static regret bounds can then be reduced compared to the one-sample case. The authors of [164] extended the one-point sampling bandit algorithm proposed in [212] to a two-point sampling bandit algorithm and obtained

$\mathcal{O}(\log(T))$ expected static regret bound for Lipschitz-continuous and strongly convex loss functions. Moreover, with $p + 1$ samplings at each round, where p is the state dimension, they proposed a deterministic algorithm and showed that the algorithm can achieve $\mathcal{O}(\sqrt{T})$ static regret bound for Lipschitz-continuous and smooth loss functions, and $\mathcal{O}(\log(T))$ static regret bound for strongly convex and smooth loss functions. The author of [280] proposed a simple algorithm with two samplings at each round and obtained $\mathcal{O}(\sqrt{T})$ expected static regret bound for Lipschitz-continuous loss functions. Without assuming that the decision set is bounded, the author of [326] proposed a class of algorithms with one and two samplings at each round and obtained $\mathcal{O}(T^{2/3})$ and $\mathcal{O}(\sqrt{T})$ expected static regret bounds, respectively, for smooth loss functions.

Aforementioned studies did not consider equality or inequality constraints. In the literature, there are few studies considering bandit online convex optimization with such constraints, although such constraints are common in applications. The authors of [169] studied online convex optimization with static inequality constraints and bandit feedback for constraints, while the authors of [328] studied online convex optimization with time-varying inequality constraints and bandit feedback for loss functions. The authors of [329] studied online convex optimization with time-varying inequality constraints and bandit feedback for both loss and constraint functions.

Most existing bandit online convex optimization studies are in a centralized setting and only few studies considered distributed bandit online convex optimization. When loss functions are strongly convex, the authors of [330] proposed a consensus-based distributed bandit online algorithm with one sampling at each round and obtained $\mathcal{O}(\sqrt{T}\log(T))$ expected static regret bound. When loss functions are quadratic, the authors of [188] proposed a consensus-based distributed bandit online algorithm with two samplings at each round and obtained $\mathcal{O}(\sqrt{T})$ expected static regret bound when there are set constraints. When there are static linear inequality constraints, they also established $\mathcal{O}(T^{\max\{\beta,1-\beta\}})$ and $\mathcal{O}(T^{1-\beta/2})$ bounds on the expected static regret and constraint violation, respectively, where $\beta \in (0, 1)$ is a user-defined trade-off parameter of the proposed algorithm.

This chapter considers the problem of distributed bandit online convex optimization with time-varying coupled inequality constraints. This problem can be interpreted as a repeated game between a group of learners and an adversary. The learners attempt to minimize a sequence of global loss functions and at the same time satisfy a sequence of coupled constraint functions. The global loss and the coupled constraint functions are the sum of local convex loss and constraint functions, respectively. They are generated adaptively by the adversary. The local loss and constraint functions are revealed in a bandit manner and the revealed information is held privately by each learner.

We first consider that the situation that at each round each learner can sample both her local loss and constraint functions at one point. We have the following contributions.

(C7.1) We propose a distributed bandit online algorithm based on the one-point sampling gradient estimator (Algorithm 7.1) to solve the considered optimization problem. To the best of our knowledge, this is the first algorithm to solve the online convex optimization problem with time-varying inequality constraints in the one-point bandit feedback setting. An advantage of our algorithm is that the total number of rounds is not used in the algorithm and thus does not need to be known a priori, which is an improvement compared to the one-point sampling bandit algorithms in [212, 318–321, 328, 330]. Moreover, note that these studies did not consider bandit feedback for time-varying inequality constraints or did not even consider time-varying inequality constraints at all.

(C7.2) We show in Theorem 7.1 that sublinear expected regret and constraint violation bounds are achieved by the proposed algorithm if $V(\boldsymbol{x}^*_{[T]})$, the path-length of the optimal dynamic decision sequence, grows sublinearly with a known order. We also show in Corollary 7.1 that $\tilde{O}(T^{\theta_1})$ expected static regret and $O(T^{7/4-\theta_1})$ constraint violation bounds are achieved, where $\theta_1 \in (3/4, 5/6]$ is a user-defined trade-off parameter. As a special case, when there are no inequality constraints, the proposed algorithm achieves $\tilde{O}(T^{3/4})$ expected static regret bound, which is the same expected static regret bound that has been achieved by the one-point sampling bandit algorithm in [212]. However, in [212] the total number of iterations, T, as well as the Lipschitz constant and upper bound of the loss functions are needed for the algorithm.

We then consider that the situation that at each round each learner can sample both her local loss and constraint functions at two points. We have the following contributions.

(C7.3) We propose a distributed bandit online algorithm based on the two-point sampling gradient estimator (Algorithm 7.2). This algorithm does not require the total number of rounds or any other parameters related to the loss or constraint functions, which is different from the two-point sampling bandit algorithms in [164, 169, 188, 214, 280, 325, 327–329].

(C7.4) In an average sense, the two-point sampling based distributed bandit online algorithm is as efficient as the algorithms proposed in [169, 170, 173] and Chapter 6, although [170, 173] and Chapter 6 are in a full-information feedback setting and [169] considers bandit setting only for the constraint functions. Specifically, we show in Theorem 7.2 that sublinear expected regret and constraint violation bounds are achieved by the proposed algorithm if the path-length of the optimal dynamic decision sequence grows sublinearly with a known order $\nu \in [0, 1)$. For example, $\tilde{O}(T^{(1+\nu)/2})$ expected dynamic regret and $\tilde{O}(T^{(3+\nu)/4})$ constraint violation bounds are achieved by our algorithm. Thus the bounds achieved by the centralized two-point sampling bandit algorithms in [325, 329] are recovered by our algorithm. We also show in Corollary 7.2 that $\tilde{O}(T^{\max\{\kappa,1-\kappa\}})$ expected static regret and $O(T^{1-\kappa/2})$ constraint violation bounds are also achieved, where $\kappa \in (0, 1)$ is a user-defined parameter. Thus the bounds achieved by the centralized two-point sampling bandit algorithm in [169, 280] are also recovered with $\kappa = 1/2$. However, in [280, 325] static set constraints rather than time-varying inequality constraints are considered; in [169] static inequality constraints and full-information feedback for the cost function are studied; and in [169, 280, 325, 329] the total number of rounds as well as the Lipschitz constant of the loss function are needed.

Table 7.1: Comparison of Chapter 7 to some related bandit online convex optimization algorithms.

Reference	Problem type	Constraint type	Information feedback	Regret and constraint violation bounds
[212]	Centralized	$g_t(x) \equiv \mathbf{0}_m$	One-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = O(T^{3/4})$
[324]	Centralized	$g_t(x) \equiv \mathbf{0}_m$	One-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = O(T^{1/2}\log(T))$
[280]	Centralized	$g_t(x) \equiv \mathbf{0}_m$	Two-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = O(T^{1/2})$
[325]	Centralized	$g_t(x) \equiv \mathbf{0}_m$	Two-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = O(\max\{(TV(\boldsymbol{x}^*_{[T]}))^{1/2},\ T^{1/2}\})$
[169]	Centralized	$g(x) \leq \mathbf{0}_m$	∇f_t and two-point sampling for g	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = O(T^{1/2})$, $\mathbf{E}[\|[\sum_{t=1}^T g(x_t)]_+\|] = O(T^{3/4})$
[328]	Centralized	$g_t(x) \leq \mathbf{0}_m$ and Slater's condition	∇g_t and one-point sampling for f_t	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = O(\max\{T^{3/4}V(\boldsymbol{x}^*_{[T]}),\ T^{3/4}\})$, $\|[\sum_{t=1}^T g(x_t)]_+\| = O(T^{3/4})$
			∇g_t and two-point sampling for f_t	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = O(\max\{T^{1/2}V(\boldsymbol{x}^*_{[T]}),\ T^{1/2}\})$, $\|[\sum_{t=1}^T g(x_t)]_+\| = O(T^{1/2})$
[329]	Centralized	$g_t(x) \leq \mathbf{0}_m$	Two-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = O((TV(\boldsymbol{x}^*_{[T]}))^{1/2})$, $\mathbf{E}[\|[\sum_{t=1}^T g(x_t)]_+\|] = O((T^3V(\boldsymbol{x}^*_{[T]}))^{1/4})$
This chapter	Distributed	$g_t(x) = \sum_{i=1}^n g_{i,t}(x_i) \leq \mathbf{0}_m$	One-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = O(\max\{T^{\theta_1}V(\boldsymbol{x}^*_{[T]}),\ T^{\theta_1}\})$, $\|[\sum_{t=1}^T g(x_t)]_+\| = O(T^{7/4-\theta_1})$, where $\theta_1 \in (3/4, 5/6]$
			Two-point sampling	$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = O(\max\{T^{\kappa}V(\boldsymbol{x}^*_{[T]}),\ T^{\max\{\kappa,1-\kappa\}}\})$, $\|[\sum_{t=1}^T g(x_t)]_+\| = O(T^{1-\kappa/2})$, where $\kappa \in (0, 1)$

Table 7.1 compares this chapter with other bandit online convex optimization algorithms.

7.2 Distributed bandit OCO with time-varying coupled inequality constraints

We consider the problem of distributed bandit online convex optimization with time-varying coupled inequality constraints. This problem can be defined as a repeated game between a group of n learners indexed by $i \in [n]$ and an adversary. At round t of the game, the adversary first arbitrarily chooses n local loss functions $\{f_{i,t} : \mathbb{R}^{p_i} \to \mathbb{R},\ i \in [n]\}$ and n local constraint functions $\{g_{i,t} : \mathbb{R}^{p_i} \to \mathbb{R}^m,\ i \in [n]\}$, where p_i and m are positive integers. Then, without knowing $\{f_{i,t},\ i \in [n]\}$ and $\{g_{i,t},\ i \in [n]\}$, all learners simultaneously choose their decisions $\{x_{i,t} \in \mathbb{X}_i,\ i \in [n]\}$, where $\mathbb{X}_i \subseteq \mathbb{R}^{p_i}$ are known convex sets. Each learner i samples the values of $f_{i,t}$ and $g_{i,t}$ at the point $x_{i,t}$ as well as at other potential points, i.e., the learners receive bandit feedback from the adversary. These values are held privately by each learner. At the same moment, the learners exchange data with their neighbors over a time-varying directed graph $\mathcal{G}_t$. The goal of the learners is to cooperatively choose a global decision sequence $\boldsymbol{x}_{[T]} = (x_1, \ldots, x_T)$, where T is the total number of rounds and $x_t = \mathrm{col}(x_{1,t}, \ldots, x_{n,t})$ is the decision vector, such that the accumulated global loss $\sum_{t=1}^T f_t(x_t)$, where $f_t(x_t) = \sum_{i=1}^n f_{i,t}(x_{i,t})$ is the global loss function, is competitive with the loss of any comparator sequence $\boldsymbol{y}_{[T]} = (y_1, \ldots, y_T)$ with $y_t = \mathrm{col}(y_{1,t}, \ldots, y_{n,t})$ (i.e., the regret is as small as possible) and at the same time the constraint violation is as small as possible.

From (1.5), we know that the regret of a global decision sequence $\boldsymbol{x}_{[T]}$ with respect to a comparator sequence $\boldsymbol{y}_{[T]}$ is

$$\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]}) = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(y_t).$$

For the above distributed bandit online convex optimization problem with time-varying coupled inequality constraints, two commonly used comparator sequences are $\boldsymbol{y}_{[T]} = \boldsymbol{x}^*_{[T]} = (x^*_1, \ldots, x^*_T)$ and $\boldsymbol{y}_{[T]} = \check{\boldsymbol{x}}^*_{[T]} = (\check{x}^*_T, \ldots, \check{x}^*_T)$, where $\check{x}^*_T$, which solve the following two constrained convex optimization problems

$$\begin{aligned} \min_{x_t \in \mathbb{X}} \quad & \sum_{t=1}^{T} f_t(x_t) \\ \text{s.t.} \quad & g_t(x_t) \leq \mathbf{0}_m, \ \forall t \in [T], \end{aligned} \tag{7.1}$$

and

$$\begin{aligned} \min_{x \in \mathbb{X}} \quad & \sum_{t=1}^{T} f_t(x) \\ \text{s.t.} \quad & g_t(x) \leq \mathbf{0}_m, \ \forall t \in [T], \end{aligned} \tag{7.2}$$

respectively, where $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n \subseteq \mathbb{R}^p$ is the global decision set, $p = \sum_{i=1}^n p_i$, and $g_t(x_t) = \sum_{i=1}^n g_{i,t}(x_{i,t})$ is the coupled constraint function.

From (1.7), we know that the constraint violation of a decision sequence $\boldsymbol{x}_{[T]}$ is

$$\Big\| \Big[\sum_{t=1}^{T} g_t(x_t) \Big]_+ \Big\|.$$

Based on the definitions introduced in Chapter 2, we make the following assumptions on the time-varying directed graph $\mathcal{G}_t$ as well as the loss and constraint functions.

Assumption 7.1. *For any $t \in \mathbb{N}_+$, the directed graph $\mathcal{G}_t$ satisfies the following conditions:*

(i) *The mixing matrix W_t is doubly stochastic, i.e., $\sum_{i=1}^n [W_t]_{ij} = \sum_{j=1}^n [W_t]_{ij} = 1, \ \forall i, j \in [n]$.*

(ii) *There exists a constant $w \in (0, 1)$, such that $[W_t]_{ij} \geq w$ if $[W_t]_{ij} > 0$.*

(iii) *There exists an integer $\iota > 0$ such that the directed graph $(\mathcal{V}, \cup_{l=0,\ldots,\iota-1} \mathcal{E}_{t+l})$ is strongly connected.*

Assumption 7.2. **(i)** *For each $i \in [n]$, the convex set $\mathbb{X}_i$ is closed. Moreover, there exist $r_i > 0$ and $R_i > 0$ such that*

$$r_i \mathbb{B}^{p_i} \subseteq \mathbb{X}_i \subseteq R_i \mathbb{B}^{p_i}, \tag{7.3}$$

and r_i is known a priori.

(ii) *For each* $i \in [n]$, $\{f_{i,t}(x)\}$ *and* $\{[g_{i,t}(x)]_j,\ j \in [m]\}$ *are convex and uniformly bounded on* $\mathbb{X}_i$, *i.e., there exist constants* $F_{f_i} > 0$ *and* $F_{g_i} > 0$ *such that for all* $t \in \mathbb{N}_+,\ j \in [m],\ x \in \mathbb{X}_i$,

$$|f_{i,t}(x)| \le F_{f_i},\ |[g_{i,t}(x)]_j| \le F_{g_i}. \tag{7.4}$$

(iii) *For each* $i \in [n]$, $f_{i,t}$ *and* $g_{i,t}$ *are differentiable on* $\mathbb{X}_i$. *Moreover,* $\{\nabla f_{i,t}\}$ *and* $\{\nabla[g_{i,t}(x)]_j,\ j \in [m]\}$ *are uniformly bounded on* $\mathbb{X}_i$, *i.e., there exist constants* $G_{f_i} > 0$ *and* $G_{g_i} > 0$ *such that for all* $t \in \mathbb{N}_+,\ j \in [m],\ x \in \mathbb{X}_i$,

$$\|\nabla f_{i,t}(x)\| \le G_{f_i},\ \|\nabla[g_{i,t}(x)]_j\| \le G_{g_i}. \tag{7.5}$$

Assumption 7.1 is a mild assumption and common in the literature on distributed optimization. Assumption 7.2 appears often in the literature of bandit online convex optimization. From Assumption 7.2 and Lemma 2.6 in [157], it follows that for all $t \in \mathbb{N}_+,\ i \in [n],\ j \in [m],\ x,\ y \in \mathbb{X}_i$,

$$|f_{i,t}(x) - f_{i,t}(y)| \le G_{f_i}\|x - y\|, \tag{7.6a}$$

$$|[g_{i,t}(x)]_j - [g_{i,t}(y)]_j| \le G_{g_i}\|x - y\|, \tag{7.6b}$$

i.e., $\{f_{i,t}(x)\}$ and $\{[g_{i,t}(x)]_j\}$ are Lipschitz-continuous on $\mathbb{X}_i$ with constants G_{f_i} and G_{g_i}, respectively.

Our goal in this chapter is to solve the following problem.

Problem 7.1. *Develop distributed algorithms to solve the problem of distributed bandit online optimization with time-varying coupled inequality constraints with guaranteed performance measured by expected regret and constraint violation.*

The considered problem can be viewed as an extension of the problem studied in Chapter 6, from full information feedback to bandit feedback. As discussed in Section 7.1, two main motivations of considering bandit feedback are that (i) gradient information is not available in many applications [165]; and (ii) computing a function value is much simpler than computing its gradient [213]. We consider two scenarios: one-point and two-point bandit feedback. More specifically, one-point bandit feedback means that at each round each learner samples her local loss and constraint function values at her decision point in the last round. Two-point bandit feedback means that each learner can do one more sampling at an any other point.

7.3 Distributed bandit online primal–dual algorithm based on one-point sampling

In this section, we consider the one-point feedback scenario. We propose a distributed bandit online primal–dual algorithm based on the one-point sampling random gradient estimator introduced in Section 2.8 to solve the considered optimization problem and derive its expected regret and constraint violation bounds.

Algorithm 7.1 Distributed Bandit Online Primal–Dual Algorithm Based on One-Point Sampling

Input: nonincreasing sequences $\{\alpha_{i,t}\}$, $\{\beta_{i,t}\}$, $\{\gamma_{i,t}\} \subseteq (0, +\infty)$, $\{\xi_{i,t}\} \subseteq (0, 1)$, and $\{\delta_{i,t}\} \subseteq (0, r_i\xi_{i,t-1}]$, $i \in [n]$, $t \in \mathbb{N}_+$.
Initialize: $u_{i,1} \in \mathbb{S}^{p_i}$, $z_{i,1} \in (1 - \xi_{i,1})\mathbb{X}_i$, $x_{i,1} = z_{i,1} + \delta_{i,1}u_{i,1}$, and $q_{i,1} = \mathbf{0}_m$, $i \in [n]$.
for $t = 2, \dots$ **do**
  for $i \in [n]$ in parallel **do**
    Select vector $u_{i,t} \in \mathbb{S}^{p_i}$ independently and uniformly at random.
    Sample $f_{i,t-1}(x_{i,t-1})$ and $g_{i,t-1}(x_{i,t-1})$.
    Update

$$\tilde{q}_{i,t} = \sum_{j=1}^{n} [W_{t-1}]_{ij} q_{j,t-1}, \tag{7.7a}$$

$$z_{i,t} = \mathcal{P}_{(1-\xi_{i,t})\mathbb{X}_i}(z_{i,t-1} - \alpha_{i,t}a_{i,t}), \tag{7.7b}$$

$$x_{i,t} = z_{i,t} + \delta_{i,t}u_{i,t}, \tag{7.7c}$$

$$q_{i,t} = [(1 - \beta_{i,t}\gamma_{i,t})\tilde{q}_{i,t} + \gamma_{i,t}g_{i,t-1}(x_{i,t-1})]_+. \tag{7.7d}$$

    Broadcast $q_{i,t}$ to $\mathcal{N}_i^{\text{out}}(\mathcal{G}_t)$ and receive $q_{j,t}$ from $j \in \mathcal{N}_i^{\text{in}}(\mathcal{G}_t)$.
  end for
end for
Output: $\{x_t\}$.

7.3.1 Algorithm description

The proposed algorithm is given in pseudo-code as Algorithm 7.1. In this algorithm, each agent i maintains four local sequences: the local primal decision variable sequence $\{x_{i,t}\} \subseteq \mathbb{X}_i$, the local intermediate decision variable sequence $\{z_{i,t}\} \subseteq (1 - \xi_{i,t})\mathbb{X}_i$, the local dual variable sequence $\{q_{i,t}\} \subseteq \mathbb{R}^m_+$, and the estimates of the average of local dual variables $\{\tilde{q}_{i,t}\} \subseteq \mathbb{R}^m_+$. They are updated recursively by the update rules (7.7a)–(7.7d). In (7.7b), $a_{i,t}$ is the updating direction information for the local intermediate decision variable defined as

$$a_{i,t} = \hat{\nabla}_1 f_{i,t-1}(z_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}) + (\hat{\nabla}_1 g_{i,t-1}(z_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}))^\top \tilde{q}_{i,t}, \tag{7.8}$$

where $\hat{\nabla}_1 f_{i,t-1}(z_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})$ and $\hat{\nabla}_1 g_{i,t-1}(z_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})$ are the one-point sampling random estimators of $\nabla f_{i,t-1}(z_{i,t-1})$ and $\nabla g_{i,t-1}(z_{i,t-1})$, respectively, as defined in (2.25), $\delta_{i,t-1} > 0$ is an adaptive smoothing parameter, and $u_{i,t-1} \in \mathbb{S}^{p_i}$ is a uniformly distributed random vector. Recall that

$$\begin{aligned}\hat{\nabla}_1 f_{i,t-1}(z_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}) &= \frac{p_i}{\delta_{i,t-1}} f_{i,t-1}(z_{i,t-1} + \delta_{i,t-1}u_{i,t-1})u_{i,t-1} \\ &= \frac{p_i}{\delta_{i,t-1}} f_{i,t-1}(x_{i,t-1})u_{i,t-1} \in \mathbb{R}^{p_i},\end{aligned}$$

and

$$\hat{\nabla}_1 g_{i,t-1}(z_{i,t-1},\delta_{i,t-1},u_{i,t-1}) = \begin{bmatrix} (\hat{\nabla}_1[g_{i,t-1}(z_{i,t-1},\delta_{i,t-1},u_{i,t-1})]_1)^\top \\ (\hat{\nabla}_1[g_{i,t-1}(z_{i,t-1},\delta_{i,t-1},u_{i,t-1})]_2)^\top \\ \vdots \\ (\hat{\nabla}_1[g_{i,t-1}(z_{i,t-1},\delta_{i,t-1},u_{i,t-1})]_m)^\top \end{bmatrix} \in \mathbb{R}^{m\times p_i},$$

where

$$\begin{aligned}\hat{\nabla}_1[g_{i,t-1}(z_{i,t-1},\delta_{i,t-1},u_{i,t-1})]_l &= \frac{p_i}{\delta_{i,t-1}}[g_{i,t-1}(z_{i,t-1}+\delta_{i,t-1}u_{i,t-1})]_l u_{i,t-1} \\ &= \frac{p_i}{\delta_{i,t-1}}[g_{i,t-1}(x_{i,t-1})]_l u_{i,t-1} \in \mathbb{R}^{p_i},\ \forall l \in [m].\end{aligned}$$

The intuition of the update rules (7.7a)–(7.7d) is as follows. The regularized Lagrangian function associated with the constrained optimization problem with cost function f and constraint function g is

$$\mathcal{A}(x,\mu) = f(x) + \mu^\top g(x) - \frac{\beta}{2}\|\mu\|^2, \tag{7.9}$$

where $\mu \in \mathbb{R}^m_+$ is the Lagrange multiplier and $\beta > 0$ is the regularization parameter. $\mathcal{A}(x,\mu)$ is a convex-concave function. A standard primal–dual algorithm to find its saddle point is

$$x_{k+1} = \mathcal{P}_{\mathbb{X}}(x_k - \alpha(\nabla f(x_k) + (\nabla g(x_k))^\top \mu_k)), \tag{7.10a}$$

$$\mu_{k+1} = [\mu_k + \gamma(g(x_k) - \beta\mu_k)]_+, \tag{7.10b}$$

where $\alpha > 0$ and $\gamma > 0$ are the stepsizes used in the primal and dual updates, respectively. The update rules (7.7a)–(7.7d) are the distributed, online, and gradient-free extensions of (7.10a) and (7.10b).

Remark 7.1. *The differences between Algorithm 7.1 and the centralized one-point sampling bandit algorithm in [328] are (i) in [328] full-information feedback for the constraint functions is used; and (ii) in the update of the dual variables in Algorithm 7.1, i.e., (7.7d), there is an additional term $-\beta_{i,t}\gamma_{i,t}\tilde{q}_{i,t}$, which comes from the regularized Lagrangian function and it plays a key role to bound the dual variables as shown later in Lemma 7.1.*

The sequences $\{\alpha_{i,t}\}$, $\{\beta_{i,t}\}$, $\{\gamma_{i,t}\}$, $\{\xi_{i,t}\}$, and $\{\delta_{i,t}\}$ used in Algorithm 7.1 are predetermined and the vector sequences $\{u_{i,t}\}$ are randomly selected. Moreover, $\{\tilde{q}_{i,t}\}$, $\{z_{i,t}\}$, $\{x_{i,t}\}$, and $\{q_{i,t}\}$ are random vector sequences generated by Algorithm 7.1. Let $\mathfrak{U}_t$ denote the σ-algebra generated by the independent and identically distributed (i.i.d.) random variables $u_{1,t},\dots,u_{n,t}$ and let $\mathcal{U}_t = \bigcup_{s=1}^t \mathfrak{U}_s$. It is straightforward to see that $\tilde{q}_{t+1}$, $z_{i,t}$, $x_{i,t-1}$, and $q_{i,t}$, $i \in [n]$ depend on $\mathcal{U}_{t-1}$ and are independent of $\mathfrak{U}_s$ for all $s \geq t$.

7.3.2 Expected regret and constraint violation bounds

This section states the main results on the expected regret and constraint violation bounds for Algorithm 7.1. The following theorem characterizes these bounds based on some specially selected stepsizes, shrinkage coefficients, and exploration parameters.

Theorem 7.1. *Suppose that Assumptions 7.1–7.2 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 7.1 with*

$$\alpha_{i,t} = \frac{r_i^2}{4mp_i^2F_{g_i}^2t^{\theta_1}},\ \beta_{i,t} = \frac{2}{t^{\theta_2}},\ \gamma_{i,t} = \frac{1}{t^{1-\theta_2}},\ \xi_{i,t} = \frac{1}{(t+1)^{\theta_3}},\ \delta_{i,t} = \frac{r_i}{(t+1)^{\theta_3}},\ i \in [n],\ t \in \mathbb{N}_+, \tag{7.11}$$

where $\theta_1 \in (0,1)$, $\theta_2 \in (0, \theta_1/3)$ and $\theta_3 \in (\theta_2, (\theta_1 - \theta_2)/2]$ are constants. Then, for any $T \in \mathbb{N}_+$ and any comparator sequence $\boldsymbol{y}_{[T]} \in \mathcal{X}_T$,

$$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]})] \le C_1 T^{\max\{\theta_1, 1-\theta_1+2\theta_3, 1-\theta_3+\theta_2\}} + C_{1,1}T^{\theta_1}V(\boldsymbol{y}_{[T]}), \tag{7.12a}$$

$$\mathbf{E}\Big[\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|\Big] \le C_2 T^{1-\theta_2/2}, \tag{7.12b}$$

where

$$C_1 = \sum_{j=1}^{4} C_{1,j} + \frac{C_0}{\theta_2},\ C_{1,1} = \sum_{i=1}^{n} \frac{8mp_i^2F_{g_i}^2R_i^2}{r_i^2},\ C_2 = \Big(C_{2,1}\Big(2\sum_{i=1}^{n} F_{f_i} + C_1\Big)\Big)^{\frac{1}{2}},$$

$$C_{1,2} = \sum_{i=1}^{n} \frac{mF_gG_{g_i}(2r_i + R_i)}{1-\theta_3+\theta_2},\ C_{1,3} = \sum_{i=1}^{n} \frac{G_{f_i}(2r_i + R_i)}{1-\theta_3},\ C_{1,4} = \sum_{i=1}^{n} \frac{F_{f_i}^2}{4mF_{g_i}^2(1-\theta_1+2\theta_3)},$$

$$F_g = \max_{i\in[n]}\{F_{g_i}\},\ C_0 = \frac{6mn^2F_g^2\tau}{1-\lambda} + 2mnF_g^2,\ \tau = \Big(1 - \frac{w}{2n^2}\Big)^{-2} > 1,$$

$$\lambda = \Big(1 - \frac{w}{2n^2}\Big)^{\frac{1}{\iota}},\ C_{2,1} = 2n\Big(1 + \max_{i\in[n]}\Big\{\frac{F_{f_i}^2}{F_{g_i}^2(1-\theta_1+2\theta_3)}\Big\} + \frac{1}{1-\theta_2}\Big),$$

w and ι are constants given in Assumption 7.1, r_i, R_i, F_{f_i}, F_{g_i}, G_{f_i}, and G_{g_i} are constants given in Assumption 7.2, and

$$V(\boldsymbol{y}_{[T]}) = \sum_{t=1}^{T-1}\sum_{i=1}^{n} \|y_{i,t+1} - y_{i,t}\|$$

is the accumulated variation (path-length) of the comparator sequence $\boldsymbol{y}_{[T]}$.

Proof. See Section 7.7.1. □

Remark 7.2. *From* (7.12b)*, we see that Algorithm 7.1 achieves sublinear expected constraint violation. From* (7.12a)*, we see that Algorithm 7.1 can achieve sublinear* *expected dynamic regret if* $V(\boldsymbol{x}^*_{[T]})$ *grows sublinearly with a known order. In this case, there exists a known constant* $\nu \in [0,1)$*, such that* $V(\boldsymbol{x}^*_{[T]}) = \mathcal{O}(T^\nu)$*, then setting* $\boldsymbol{y}_{[T]} = \boldsymbol{x}^*_{[T]}$ *and* $\theta_1 \in (0, 1-\nu)$ *in Theorem 7.1 gives* $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = \mathbf{o}(T)$*.*

Remark 7.3. *To the best of our knowledge, Algorithm 7.1 is the first algorithm to solve the online convex optimization problem with time-varying inequality constraints in the one-point bandit feedback setting. In Algorithm 7.1 the information about the total number of rounds is not used, which is an improvement compared to the one-point sampling bandit algorithms in [212,317–321,328,330,331]. Note that these studies did not consider bandit feedback for time-varying inequality constraints or did not even consider time-varying inequality constraints at all. The potential drawback of Algorithm 7.1 is that in order to use the sequences defined in* (7.11)*, each learner i needs to know* F_{g_i}*, the uniform upper bound of her time-varying constraint function. One way to overcome this is to let* $\alpha_{i,t} = \tau_i/t^{\theta_1}$ *and* $\theta_3 \in (\theta_2, (\theta_1-\theta_2)/2)$*, where* $\tau_i > 0$ *is a user-defined parameter. In this case, similar to the way we prove* (7.12a) *and* (7.12b)*, we can establish similar results as* (7.12a) *and* (7.12b) *for* $T \geq (4m\max_{i\in[n]}\{p_i^2F_{g_i}^2\tau_i/r_i^2\})^{1/(\theta_1-\theta_2-2\theta_3)}$ *rather than any* $T \in \mathbb{N}_+$*.*

Remark 7.4. *The preliminary results on the expected regret and constraint violation bounds are stated by* (7.43a) *and* (7.43b) *in Lemma 7.3 in Section 7.7. The intuition of the choices of the sequences given in* (7.11) *is to let the terms in the right-hand side of* (7.43a) *and* (7.43b) *be as small as possible. Specifically, the first four terms in the right-hand side of* (7.43a) *need to be sublinear. Moreover, the coefficient of* $\mathbf{E}[\|q_{i,t}\|^2]$ *should be nonpositive otherwise it is unclear how to show that the last terms in the right-hand side of* (7.43a) *and* (7.43b) *are sublinear.*

Setting $\boldsymbol{y}_{[T]} = \check{\boldsymbol{x}}^*_{[T]}$ in Theorem 7.1 gives following results, which characterize the expected static regret and constraint violation bounds.

Corollary 7.1. *Under the same conditions as in Theorem 7.1 with* $\theta_1 \in (3/4, 5/6]$*,* $\theta_2 = 2\theta_1 - 3/2$*, and* $\theta_3 = \theta_1 - 1/2$*, it holds that*

$$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] \leq C_1 T^{\theta_1}, \tag{7.13a}$$

$$\mathbf{E}\Big[\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|\Big] \leq C_2 T^{7/4-\theta_1}. \tag{7.13b}$$

Remark 7.5. *The parameter* θ_1 *in Corollary 7.1 is a user-defined parameter influencing the step length in* (7.11)*. It enables the trade-off between the expected static regret bound and the expected constraint violation bound. Same as [212], if there are no inequality constraints, i.e.,* $g_{i,t} \equiv \mathbf{0}_m$*,* $\forall i \in [n]$*,* $\forall t \in \mathbb{N}_+$*, then by setting* $\alpha_{i,t} = \frac{1}{t^{3/4}}$*,* $\beta_{i,t} = \gamma_{i,t} = 0$*,* $\xi_{i,t} = \frac{1}{(t+1)^{1/4}}$*,* $\delta_{i,t} = \frac{r_i}{(t+1)^{1/4}}$ *in* (7.11)*, we have that* (7.13a) *can be replaced by* $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] \leq \hat{C}_1 T^{3/4}$*, where* $\hat{C}_1 = \sum_{i=1}^n(\frac{4G_{f_i}(2r_i+R_i)}{3} + 6R_i^2 + \frac{4p_i^2F_{f_i}^2}{3r_i^2})$*. Hence, Algorithm 7.1 achieves the same expected static regret bound as the bandit algorithm in [212]. However, in [212] the total number of rounds, the Lipschitz constant, and upper bound of the loss functions need to be known in advance to run the algorithm.*

Algorithm 7.2 Distributed Bandit Online Primal–Dual Algorithm Based on Two-Point Sampling

Input: nonincreasing sequences $\{\alpha_{i,t}\}$, $\{\beta_{i,t}\}$, $\{\gamma_{i,t}\} \subseteq (0, +\infty)$, $\{\xi_{i,t}\} \subseteq (0, 1)$, and $\{\delta_{i,t}\} \subseteq (0, r_i\xi_{i,t-1}]$, $i \in [n]$, $t \in \mathbb{N}_+$.
Initialize: $x_{i,1} \in (1 - \xi_{i,1})\mathbb{X}_i$ and $q_{i,1} = \mathbf{0}_m$, $i \in [n]$.
for $t = 2, \ldots$ **do**
  for $i \in [n]$ in parallel **do**
    Select vector $u_{i,t-1} \in \mathbb{S}^{p_i}$ independently and uniformly at random.
    Sample $f_{i,t-1}(x_{i,t-1} + \delta_{i,t-1}u_{i,t-1})$, $f_{i,t-1}(x_{i,t-1})$, $g_{i,t-1}(x_{i,t-1} + \delta_{i,t-1}u_{i,t-1})$ and $g_{i,t-1}(x_{i,t-1})$.
    Update

$$\tilde{q}_{i,t} = \sum_{j=1}^{n}[W_{t-1}]_{ij}q_{j,t-1}, \tag{7.14a}$$

$$x_{i,t} = \mathcal{P}_{(1-\xi_{i,t})\mathbb{X}_i}(x_{i,t-1} - \alpha_{i,t}b_{i,t}), \tag{7.14b}$$

$$q_{i,t} = [(1 - \gamma_{i,t}\beta_{i,t})\tilde{q}_{i,t} + \gamma_{i,t}c_{i,t}]_+. \tag{7.14c}$$

    Broadcast $q_{i,t}$ to $\mathcal{N}_i^{\text{out}}(\mathcal{G}_t)$ and receive $q_{j,t}$ from $j \in \mathcal{N}_i^{\text{in}}(\mathcal{G}_t)$.
  end for
end for
Output: $\{x_t\}$.

7.4 Distributed bandit online primal–dual algorithm based on two-point sampling

In this section, we consider the two-point feedback scenario. We propose a distributed bandit online primal–dual algorithm based on the two-point random sampling gradient estimator introduced in Section 2.8 to solve the considered optimization problem and derive its expected regret and constraint violation bounds.

7.4.1 Algorithm description

With two-point bandit feedback at each round each learner samples the values of her local loss and constraint at two points. This gives the freedom to design a more efficient algorithm which at the same time avoids the potential drawback of Algorithm 7.1 stated in Remark 7.3 on knowing the upper bounds of the time-varying constraint functions. The proposed algorithm is given in pseudo-code as Algorithm 7.2. In (7.14b), $b_{i,t}$ is the updating direction information for the local primal decision variable defined as

$$b_{i,t} = \hat{\nabla}_2 f_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}) + (\hat{\nabla}_2 g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}))^\top \tilde{q}_{i,t}, \tag{7.15}$$

where $\hat{\nabla}_2 f_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})$ and $\hat{\nabla}_2 g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})$ are the two-point sampling random estimators of $\nabla f_{i,t-1}(x_{i,t-1})$ and $\nabla g_{i,t-1}(x_{i,t-1})$, respectively, as defined in (2.26),

$\delta_{i,t-1} > 0$ is an adaptive smoothing parameter, and $u_{i,t-1} \in \mathbb{S}^{p_i}$ is a uniformly distributed random vector. Recall that

$$\hat{\nabla}_2 f_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}) = \frac{p_i}{\delta_{i,t-1}}(f_{i,t-1}(x_{i,t-1} + \delta_{i,t-1}u_{i,t-1}) - f_{i,t-1}(x_{i,t-1}))u_{i,t-1} \in \mathbb{R}^{p_i},$$

and

$$\hat{\nabla}_2 g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1}) = \begin{bmatrix} (\hat{\nabla}_2[g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})]_1)^\top \\ (\hat{\nabla}_2[g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})]_2)^\top \\ \vdots \\ (\hat{\nabla}_2[g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})]_m)^\top \end{bmatrix} \in \mathbb{R}^{m\times p_i},$$

where

$$\begin{aligned}&\hat{\nabla}_2[g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})]_l \\ &= \frac{p_i}{\delta_{i,t-1}}[g_{i,t-1}(x_{i,t-1} + \delta_{i,t-1}u_{i,t-1}) - g_{i,t-1}(x_{i,t-1})]_l u_{i,t-1} \in \mathbb{R}^{p_i}, \ \forall l \in [m].\end{aligned}$$

Similarly, in (7.14c), $c_{i,t}$ is the updating direction information for the local dual variable defined as

$$c_{i,t} = \hat{\nabla}_2 g_{i,t-1}(x_{i,t-1}, \delta_{i,t-1}, u_{i,t-1})(x_{i,t} - x_{i,t-1}) + g_{i,t-1}(x_{i,t-1}). \tag{7.16}$$

In addition to that Algorithm 7.2 uses a two-point sampling gradient estimator, another difference between Algorithms 7.1 and 7.2 is that when updating the local dual variable, in Algorithm 7.2, $c_{i,t}$ is used to replace $g_{i,t-1}(x_{i,t-1})$, which is a key difference between Algorithm 7.2 and the centralized two-point sampling bandit algorithm in [329]. This modification is inspired by the algorithm proposed in [175] and Algorithm 6.1, and helps to avoid using the uniform upper bound of each learner's time-varying constraint function, i.e., to remove the potential drawback stated in Remark 7.3.

7.4.2 Expected regret and constraint violation bounds

This section states the main results on the expected regret and constraint violation bounds for Algorithm 7.2.

Theorem 7.2. *Suppose that Assumptions 7.1–7.2 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 7.2 with*

$$\alpha_t = \frac{1}{t^\kappa},\ \beta_t = \frac{1}{t^\kappa},\ \gamma_t = \frac{1}{t^{1-\kappa}},\ \xi_{i,t} = \frac{1}{t+1},\ \delta_{i,t} = \frac{r_i}{t+1},\ i \in [n],\ t \in \mathbb{N}_+, \tag{7.17}$$

where $\kappa \in (0,1)$ is a constant. Then, for any $T \in \mathbb{N}_+$ and any comparator sequence $\boldsymbol{y}_{[T]} \in \mathcal{X}_T$,

$$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]})] \le C_3 T^{\max\{\kappa,1-\kappa\}} + 2R_{\max}T^\kappa V(\boldsymbol{y}_{[T]}), \tag{7.18a}$$

$$\mathbf{E}\Big[\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|\Big] \le C_4 T^{1-\kappa/2}, \tag{7.18b}$$

where

$$C_3 = \sum_{i=1}^{n}\Big(2G_{f_i}(r_i + R_i) + 8R_i^2 + \frac{2\sqrt{m}B_1 G_{g_i} R_i}{\kappa} + \frac{p_i^2 G_{f_i}^2}{1-\kappa}\Big) + \frac{\hat{C}_0}{\kappa},$$

$$C_4 = \Big(C_{4,1}\Big(2\sum_{i=1}^{n} F_{f_i} + C_3\Big)\Big)^{\frac{1}{2}},\ C_{4,1} = \sum_{i=1}^{n} 2\Big(\frac{2mp_i^2 G_{g_i}^2 + 1}{1-\kappa} + 1\Big),$$

$$\hat{C}_0 = \frac{6n^2\sqrt{m}\tau B_1 F_g}{1-\lambda} + 2nB_1^2,\ B_1 = \sqrt{m}F_g + \sqrt{m}pG_g R_{\max},\ R_{\max} = \max_{i\in[n]}\{R_i\}.$$

Proof. See Appendix 7.7.2. □

Remark 7.6. *The bounds obtained in* (7.18a) *and* (7.18b) *are the same as the bounds shown in* (6.17a) *and* (6.17b) *achieved by Algorithm 6.1 in Chapter 6 under the same assumptions, although Chapter 6 considered a full-information feedback setting. In other words, in an average sense, Algorithm 7.2, which only uses two-point bandit feedback, is as efficient as Algorithm 6.1, which uses full-information feedback. By comparing* (7.11), (7.12a), *and* (7.12b) *with* (7.17), (7.18a), *and* (7.18b), *respectively, we see that if a two-point sampling gradient estimator is used, then not only the uses of F_{g_i}, the uniform upper bound of the time-varying constraint functions, is avoided, but also the upper bounds of the expected regret and constraint violation are both reduced. An advantage of Algorithm 7.2 is that the total number of rounds or any other parameters related to loss or constraint functions are not used, which is different from the two-point sampling bandit algorithms in [164, 169, 188, 214, 280, 325, 327–329].*

Remark 7.7. *Similar to the analysis in Remark 7.2, from* (7.18b), *we know that Algorithm 7.2 achieves sublinear expected constraint violation. Algorithm 7.2 can also achieve sublinear expected dynamic regret if $V(\boldsymbol{x}^*_{[T]})$ grows sublinearly with a known order. In this case, there exists a known constant $\nu \in [0, 1)$, such that $V(\boldsymbol{x}^*_{[T]}) = \mathcal{O}(T^{\nu})$. Then setting $\boldsymbol{y}_{[T]} = \boldsymbol{x}^*_{[T]}$ and $\kappa \in (0, 1-\nu)$ in Theorem 7.2 gives $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})] = \mathbf{o}(T)$. One special case is to set $\kappa = (1-\nu)/2$ in* (7.18a) *and* (7.18b). *It gives $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = \mathcal{O}(T^{(1+\nu)/2})$ and $\mathbf{E}[\|[\sum_{t=1}^{T} g_t(x_t)]_+\|] = \mathcal{O}(T^{(3+\nu)/4})$, which recovers the bounds achieved by the centralized two-point sampling bandit algorithms in [325, 329].*

Setting $\boldsymbol{y}_{[T]} = \check{\boldsymbol{x}}^*_{[T]}$ in Theorem 7.2 gives the following results.

Corollary 7.2. *Under the same conditions as stated in Theorem 7.2, it holds that*

$$\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] \le C_3 T^{\max\{\kappa, 1-\kappa\}}, \tag{7.19a}$$

$$\mathbf{E}\Big[\Big\|\Big[\sum_{t=1}^{T} g_t(x_t)\Big]_+\Big\|\Big] \le C_4 T^{1-\kappa/2}. \tag{7.19b}$$

Remark 7.8. *The parameter κ for the sequences $\{\alpha_{i,t}\}$, $\{\beta_{i,t}\}$, and $\{\gamma_{i,t}\}$ in Corollary 7.2 enables the user to trade-off the expected static regret bound for the expected constraint violation bound. For example, setting $\kappa = 1/2$ in Corollary 7.2 gives* $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \check{\boldsymbol{x}}^*_{[T]})] = \mathcal{O}(\sqrt{T})$ *and* $\mathbf{E}[\|[\sum_{t=1}^T g_t(x_t)]_+\|] = \mathcal{O}(T^{3/4})$. *These two bounds are the same as the bounds achieved in [169, 170, 173]. In other words, Algorithm 7.2 is as efficient as the algorithms proposed in [169, 170, 173]. However, [170, 173] use full-information feedback and [169] considers bandit setting only for the constraint functions. The algorithms proposed in [169, 170, 173] are centralized and the constraint functions considered in [169, 170] are time-invariant. Moreover, in [169, 173] the total number of rounds and in [169, 170, 173] the upper bounds of the loss and constraint functions and their subgradients need to be known in advance to execute the algorithms. Also, an $\mathcal{O}(\sqrt{T})$ expected static regret bound was achieved by the bandit algorithm in [280]. However, in [280] static set constraints (rather than time-varying inequality constraints) are considered and the proposed algorithm is centralized (rather than distributed). Moreover, in [280] the total number of rounds and the Lipschitz constant need to be known in advance.*

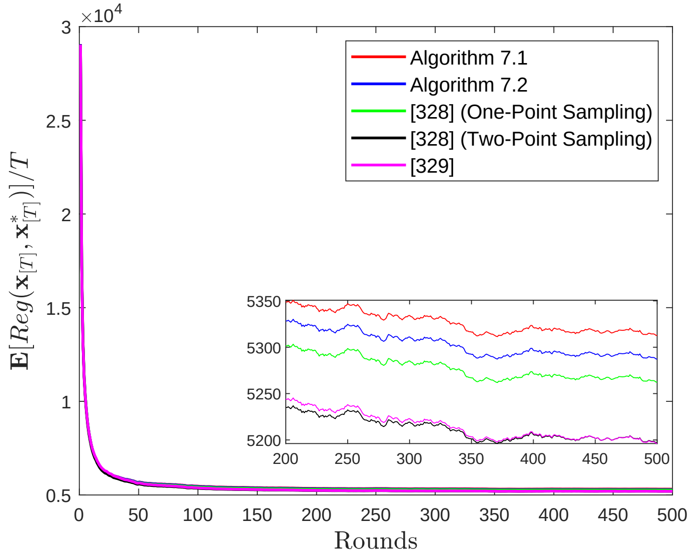
7.5 Simulations

This section evaluates the performance of Algorithms 7.1 and 7.2 in solving the DERs coordination problem introduced in Section 1.1. The local cost and constraint functions are given as

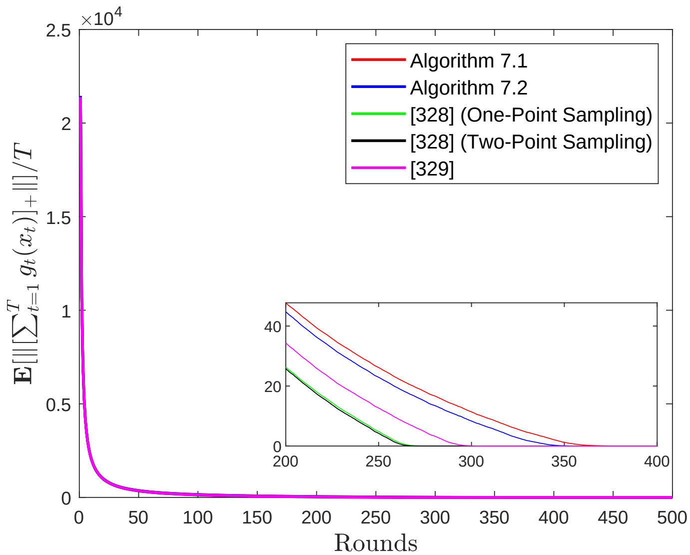
$$f_{i,t}(x_{i,t}) = x_{i,t}^\top \Pi_{i,t}^\top \Pi_{i,t} x_{i,t} + \langle \pi_{i,t}, x_{i,t} \rangle,\ g_{i,t}(x_{i,t}) = x_{i,t}^\top \Phi_{i,t}^\top \Phi_{i,t} x_{i,t} + \langle \phi_{i,t}, x_{i,t} \rangle + c_{i,t},$$

respectively, where $\Pi_{i,t} \in \mathbb{R}^{p_i \times p_i}$, $\pi_{i,t} \in \mathbb{R}_+^{p_i}$, $\Phi_{i,t} \in \mathbb{R}^{p_i \times p_i}$, $\phi_{i,t} \in \mathbb{R}^{p_i}$, and $c_{i,t} \in \mathbb{R}$. At each time t, an undirected graph is used as the communication graph. Specifically, connections between vertices are random and the probability of two vertices being connected is $\rho > 0$. Moreover, edges $(i, i+1)$, $i \in [n-1]$ are added and $[W_t]_{ij} = 1/n$ if $(j,i) \in \mathcal{E}_t$ and $[W_t]_{ii} = 1 - \sum_{j \in \mathcal{N}_i^{\mathrm{in}}(\mathcal{G}_t)} [W_t]_{ij}$. The parameters are set as: $n = 50$, $m = 1$, $p_i = 6$, $\mathbb{X}_i = [-10, 10]^{p_i}$, and $\rho = 0.2$. Each element of $\Pi_{i,t}$, $\pi_{i,t}$, $\Phi_{i,t}$, $\phi_{i,t}$, and $c_{i,t}$ are drawn from the discrete uniform distribution in $[-5, 5]$, $[0, 10]$, $[-5, 5]$, $[-5, 5]$, and $[-5, -1]$, respectively. Under above settings, Assumptions 7.1–7.2 hold.

Since there are no other distributed bandit online algorithms to solve the problem of online optimization with time-varying coupled inequality constraints, we compare our Algorithms 7.1 and 7.2 with the centralized one- and two-point sampling bandit algorithms in [328], which use full-information feedback for the constraint functions, and the centralized two-point sampling bandit algorithm in [329]. Figures 7.1 (a) and (b) show the evolutions of $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})]/T$ and $\mathbf{E}[\|[\sum_{t=1}^T g_t(x_t)]_+\|]/T$, respectively. The average is taken over 100 realizations. Note that $\mathbf{E}[\|[\sum_{t=1}^T g_t(x_t)]_+\|]/T \to 0$. This is in agreement with (7.12b), (7.18b), and the theoretical results shown in [328, 329]. From the zoomed figures, we see that the centralized algorithms in [328, 329] achieve smaller expected dynamic regret and constraint violation than our distributed algorithms, which is reasonable. We also see that Algorithm 7.2 achieves smaller expected dynamic regret and constraint violation than Algorithm 7.1, which is consistent with our theoretical results.



(a) Evolutions of expected dynamic regret $\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{x}^*_{[T]})]/T$.



(b) Evolutions of expected constraint violation $\mathbf{E}[\|[\sum_{t=1}^{T} g_t(x_t)]_+\|]/T$.

Figure 7.1: Performance of bandit online convex optimization algorithms in the DERs coordination problem.

7.6 Summary

In this chapter, we considered the distributed bandit online convex optimization problem with time-varying coupled inequality constraints. We proposed distributed bandit online algorithms with one- and two-point bandit feedback. We showed that sublinear expected regret and constraint violation can be achieved by both proposed algorithms. We showed that the results can be cast as nontrivial extensions of existing literature on online optimization and bandit feedback. Future research directions include considering an adaptive choice of the number of samplings at each round by different learners, relaxing the doubly stochastic assumption, studying sampling noise, achieving a smaller regret bound under stronger assumptions for the cost functions, and trying to establish sublinear constraint violation under a stricter constraint violation metric.

7.7 Proofs

7.7.1 Proof of Theorem 7.1

To prove Theorem 7.1, the following three lemmas are used. Lemma 7.1 presents the results on the local dual variables, while Lemma 7.2 provides an upper bound for the regret of one round. Lemma 7.3 provides the expected regret constraint violation bounds for Algorithm 7.1 for the general case.

To simplify notation, we denote $\beta_t = \beta_{i,t}$, $\gamma_t = \gamma_{i,t}$, and $\xi_t = \xi_{i,t}$.

Lemma 7.1. *Suppose that Assumptions 7.1–7.2 hold. For all $i \in [n]$ and $t \in \mathbb{N}_+$, $\tilde{q}_{i,t}$ and $q_{i,t}$ generated by Algorithm 7.1 satisfy*

$$\|\tilde{q}_{i,t+1}\| \le \frac{\sqrt{m}F_g}{\beta_t},\ \|q_{i,t}\| \le \frac{\sqrt{m}F_g}{\beta_t}, \tag{7.20a}$$

$$\|\tilde{q}_{i,t+1} - \bar{q}_t\| \le 2\sqrt{m}nF_g\tau\sum_{s=1}^{t-1}\gamma_{s+1}\lambda^{t-1-s}, \tag{7.20b}$$

$$\frac{\Delta_{t+1}}{2\gamma_{t+1}} \le (\bar{q}_t - q)^\top g_t(x_t) + 2mnF_g^2\gamma_{t+1} + \frac{n\beta_{t+1}}{2}\|q\|^2 + d_1(t), \tag{7.20c}$$

where q is an arbitrary vector in $\mathbb{R}^m_+$, and

$$\Delta_t = \sum_{i=1}^{n}\|q_{i,t} - q\|^2 - (1-\beta_t\gamma_t)\sum_{i=1}^{n}\|q_{i,t-1} - q\|^2,$$

$$\bar{q}_t = \frac{1}{n}\sum_{i=1}^{n}q_{i,t},\ d_1(t) = 2mn^2F_g^2\tau\sum_{s=1}^{t}\gamma_{s+1}\lambda^{t-s}.$$

Proof. This lemma is Lemma 6.1 under bandit setting and the ideas of the proofs of these two lemmas are similar.

(i) From (7.4), we have

$$\|g_{i,t}(x_{i,t})\| \le \sqrt{m}F_g,\ \forall i \in [n],\ \forall t \in \mathbb{N}_+. \tag{7.21}$$

We prove (7.20a) by induction.

It is straightforward to see that $q_{i,1} = \tilde{q}_{i,2} = \mathbf{0}_m$, $\forall i \in [n]$, thus $\|\tilde{q}_{i,2}\| \le \frac{\sqrt{m}F_g}{\beta_1}$, $\|q_{i,1}\| \le \frac{\sqrt{m}F_g}{\beta_1}$, $\forall i \in [n]$. Assume that (7.20a) is true at time t for all $i \in [n]$. We show that it remains true at time $t+1$. Firstly, from (2.11a), (7.7d), (7.21), $1 - \gamma_{t+1}\beta_{t+1} \ge 0$, and $\beta_t \ge \beta_{t+1}$ we know that for all $i \in [n]$,

$$\begin{aligned}\|q_{i,t+1}\| &\le (1-\gamma_{t+1}\beta_{t+1})\|\tilde{q}_{i,t+1}\| + \gamma_{t+1}\|g_{i,t}(x_{i,t})\|\\ &\le (1-\gamma_{t+1}\beta_{t+1})\frac{\sqrt{m}F_g}{\beta_t} + \gamma_{t+1}\sqrt{m}F_g\\ &\le (1-\gamma_{t+1}\beta_{t+1})\frac{\sqrt{m}F_g}{\beta_{t+1}} + \gamma_{t+1}\sqrt{m}F_g\\ &\le \frac{\sqrt{m}F_g}{\beta_{t+1}}.\end{aligned}$$

Then, the convexity of norms and $\sum_{j=1}^n [W_t]_{ij} = 1$ yield

$$\begin{aligned}\|\tilde{q}_{i,t+2}\| &\le \sum_{j=1}^n [W_{t+1}]_{ij}\|q_{j,t+1}\| \le \sum_{j=1}^n [W_t]_{ij}\frac{\sqrt{m}F_g}{\beta_{t+1}}\\ &= \frac{\sqrt{m}F_g}{\beta_{t+1}},\ \forall i \in [n].\end{aligned}$$

Thus, (7.20a) follows.
(ii) Note that (7.7d) can be rewritten as

$$q_{i,t+1} = \sum_{j=1}^n [W_t]_{ij} q_{j,t} + \epsilon^q_{i,t}, \tag{7.22}$$

where $\epsilon^q_{i,t} = [(1-\gamma_{t+1}\beta_{t+1})\tilde{q}_{i,t+1} + \gamma_{t+1}g_{i,t}(x_{i,t})]_+ - \tilde{q}_{i,t+1}$. Then, (2.10), (7.20a), and (7.21) give

$$\|\epsilon^q_{i,t}\| \le \|-\gamma_{t+1}\beta_{t+1}\tilde{q}_{i,t+1} + \gamma_{t+1}g_{i,t}(x_{i,t})\| \le 2\sqrt{m}F_g\gamma_{t+1},\ \forall i \in [n]. \tag{7.23}$$

Then, from Assumption 7.1, Lemma 2 in [190], $q_{i,1} = \mathbf{0}_m$, $\forall i \in [n]$, and (7.23), we know that for any $i \in [n]$ and $t \in \mathbb{N}_+$,

$$\|q_{i,t+1} - \bar{q}_{t+1}\| \le 2\sqrt{m}nF_g\tau \sum_{s=1}^t \gamma_{s+1}\lambda^{t-s}. \tag{7.24}$$

Thus, (7.20b) follows due to $\sum_{j=1}^n [W_t]_{ij} = 1$ and $\|\tilde{q}_{i,t+1} - \bar{q}_t\| = \|\sum_{j=1}^n [W_t]_{ij} q_{j,t} - \bar{q}_t\| \le \sum_{j=1}^n [W_t]_{ij}\|q_{j,t} - \bar{q}_t\|$.

(iii) Applying (2.10) to (7.7d) yields

$$\begin{aligned}\|q_{i,t}-q\|^2 &\le \|(1-\beta_t\gamma_t)\tilde{q}_{i,t}+\gamma_t g_{i,t-1}(x_{i,t-1})-q\|^2\\ &= \|\tilde{q}_{i,t}-q\|^2+\gamma_t^2\|g_{i,t-1}(x_{i,t-1})-\beta_t\tilde{q}_{i,t}\|^2\\ &\quad+2\gamma_t(\tilde{q}_{i,t}-q)^\top g_{i,t-1}(x_{i,t-1})-2\beta_t\gamma_t(\tilde{q}_{i,t}-q)^\top\tilde{q}_{i,t}. \end{aligned}\tag{7.25}$$

For the first term of the right-hand side of (7.25), by convexity of norms and $\sum_{j=1}^{n}[W_{t-1}]_{ij}=1$, it can be concluded that

$$\begin{aligned}\|\tilde{q}_{i,t}-q\|^2 &= \|\sum_{j=1}^{n}[W_{t-1}]_{ij}q_{j,t-1}-\sum_{j=1}^{n}[W_{t-1}]_{ij}q\|^2\\ &\le \sum_{j=1}^{n}[W_{t-1}]_{ij}\|q_{j,t-1}-q\|^2.\end{aligned}\tag{7.26}$$

For the second term of the right-hand side of (7.25), (7.20a) and (7.21) yield

$$\gamma_t^2\|g_{i,t-1}(x_{i,t-1})-\beta_t\tilde{q}_{i,t}\|^2 \le (2\sqrt{m}F_g\gamma_t)^2.\tag{7.27}$$

For the fourth term of the right-hand side of (7.25), we have

$$\begin{aligned}&2\gamma_t(\tilde{q}_{i,t}-q)^\top g_{i,t-1}(x_{i,t-1})\\ &=2\gamma_t(\bar{q}_{t-1}-q)^\top g_{i,t-1}(x_{i,t-1})+2\gamma_t(\tilde{q}_{i,t}-\bar{q}_{t-1})^\top g_{i,t-1}(x_{i,t-1}).\end{aligned}\tag{7.28}$$

Moreover, from (7.21) and (7.20b), we have

$$\begin{aligned}2\gamma_t(\tilde{q}_{i,t}-\bar{q}_{t-1})^\top g_{i,t-1}(x_{i,t-1}) &\le 2\gamma_t\|\tilde{q}_{i,t}-\bar{q}_{t-1}\|\|g_{i,t-1}(x_{i,t-1})\|\\ &\le \frac{2\gamma_t d_1(t-1)}{n}.\end{aligned}\tag{7.29}$$

For the last term of the right-hand side of (7.25), neglecting the nonnegative term $\beta_t\gamma_t\|\tilde{q}_{i,t}\|^2$ gives

$$-2\beta_t\gamma_t(\tilde{q}_{i,t}-q)^\top\tilde{q}_{i,t} \le \beta_t\gamma_t(\|q\|^2-\|\tilde{q}_{i,t}-q\|^2).\tag{7.30}$$

Combining (7.25)–(7.30), summing over $i\in[n]$, dividing by $2\gamma_t$, using $\sum_{i=1}^{n}[W_{t-1}]_{ij}=1$, $\forall t\in\mathbb{N}_+$, setting $t=t+1$, and rearranging the terms yields (7.20c). □

Lemma 7.2. *Suppose that Assumptions 7.1–7.2 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 7.1 and $\{y_t\}$ be an arbitrary sequence in $\mathbb{X}$, then*

$$\begin{aligned}f_t(x_t)-f_t(y_t) &\le (\bar{q}_t)^\top(g_t(y_t)-g_t(x_t))+2d_1(t)+d_2(t)+d_3(t)+\mathbf{E}_{\mathfrak{U}_t}[d_4(t)]\\ &\quad+\sum_{i=1}^{n}\frac{p_i^2F_{f_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}+\sum_{i=1}^{n}\frac{2R_i\|y_{i,t+1}-y_{i,t}\|}{\alpha_{i,t+1}},\ \forall t\in\mathbb{N}_+,\end{aligned}\tag{7.31}$$

where $d_1(t)$ is given in Lemma 7.1, and

$$d_2(t) = \sum_{i=1}^{n} \Big((2\delta_{i,t} + R_i\xi_t)(\sqrt{m}G_{g_i}\|q_{i,t}\| + G_{f_i}) + \frac{2R_i^2(\xi_t - \xi_{t+1})}{\alpha_{i,t+1}} \Big),$$

$$d_3(t) = 2m \max_{i\in[n]} \Big\{ \frac{p_i^2 F_{g_i}^2 \alpha_{i,t+1}}{\delta_{i,t}^2} \Big\} \Big(n\|q\|^2 + \sum_{i=1}^{n} \|q_{i,t} - q\|^2 \Big),$$

$$d_4(t) = \sum_{i=1}^{n} \frac{\|\check{y}_{i,t} - z_{i,t}\|^2 - \|\check{y}_{i,t+1} - z_{i,t+1}\|^2}{2\alpha_{i,t+1}}, \ \check{y}_{i,t} = (1-\xi_t)y_{i,t}.$$

Proof. For any $i \in [n]$, $t \in \mathbb{N}_+$ and $x \in (1-\xi_t)\mathbb{X}_i$, denote

$$f_{i,t}^s(x) = \mathbf{E}_{v\in\mathbb{B}^p}[f_{i,t}(x + \delta_{i,t}v)], \ \hat{g}_{i,t}(x) = \mathbf{E}_{v\in\mathbb{B}^p}[g_{i,t}(x + \delta_{i,t}v)].$$

From Lemma 2.16, (7.4), (7.21), (7.6a), and (7.6b), we know that $f_{i,t}^s(x)$ and $\hat{g}_{i,t}(x)$ are convex on $(1-\xi_t)\mathbb{X}_i$, and for any $i \in [n]$, $t \in \mathbb{N}_+$ and $x \in (1-\xi_t)\mathbb{X}_i$,

$$\nabla f_{i,t}^s(x) = \mathbf{E}_{\mathfrak{U}_t}[\hat{\nabla}_1 f_{i,t}(x)], \tag{7.32a}$$

$$f_{i,t}(x) \le f_{i,t}^s(x) \le f_{i,t}(x) + G_{f_i}\delta_{i,t}, \tag{7.32b}$$

$$\|\hat{\nabla}_1 f_{i,t}(x)\| \le \frac{p_i F_{f_i}}{\delta_{i,t}}, \tag{7.32c}$$

$$\nabla \hat{g}_{i,t}(x) = \mathbf{E}_{\mathfrak{U}_t}[\hat{\nabla}_1 g_{i,t}(x)], \tag{7.32d}$$

$$g_{i,t}(x) \le \hat{g}_{i,t}(x) \le g_{i,t}(x) + G_{g_i}\delta_{i,t}\mathbf{1}_m, \tag{7.32e}$$

$$\|\hat{\nabla}_1 g_{i,t}(x)\| \le \frac{\sqrt{m}p_i F_{g_i}}{\delta_{i,t}}, \tag{7.32f}$$

$$\|\hat{g}_{i,t}(x)\| \le \sqrt{m}F_{g_i}. \tag{7.32g}$$

Then, (7.6a), (7.6b), (7.3), and (7.32b) yield

$$|f_{i,t}(x_{i,t}) - f_{i,t}(z_{i,t})| \le G_{f_i}\|x_{i,t} - z_{i,t}\| \le G_{f_i}\delta_{i,t}, \tag{7.33a}$$

$$\|g_{i,t}(x_{i,t}) - g_{i,t}(z_{i,t})\| \le \sqrt{m}G_{g_i}\|x_{i,t} - z_{i,t}\| \le \sqrt{m}G_{g_i}\delta_{i,t}, \tag{7.33b}$$

$$\begin{aligned} f_{i,t}^s(\check{y}_{i,t}) - f_{i,t}(y_{i,t}) &= f_{i,t}(\check{y}_{i,t}) - f_{i,t}(y_{i,t}) + f_{i,t}^s(\check{y}_{i,t}) - f_{i,t}(\check{y}_{i,t}) \\ &\le G_{f_i}\|\check{y}_{i,t} - y_{i,t}\| + f_{i,t}^s(\check{y}_{i,t}) - f_{i,t}(\check{y}_{i,t}) \\ &\le G_{f_i}R_i\xi_t + G_{f_i}\delta_{i,t}, \end{aligned} \tag{7.33c}$$

$$f_{i,t}(z_{i,t}) - f_{i,t}^s(z_{i,t}) \le 0, \tag{7.33d}$$

$$\|g_{i,t}(\check{y}_{i,t}) - g_{i,t}(y_{i,t})\| \le \sqrt{m}G_{g_i}R_i\xi_t. \tag{7.33e}$$

From that $f_{i,t}^s(x)$ is convex on $(1-\xi_t)\mathbb{X}_i$, we have that

$$\begin{aligned} f_{i,t}^s(z_{i,t}) - f_{i,t}^s(\check{y}_{i,t}) &\le \langle \nabla f_{i,t}^s(z_{i,t}), z_{i,t} - \check{y}_{i,t} \rangle \\ &= \langle \mathbf{E}_{\mathfrak{U}_t}[\hat{\nabla}_1 f_{i,t}(z_{i,t})], z_{i,t} - \check{y}_{i,t} \rangle \end{aligned}$$

$$= \mathbf{E}_{\mathfrak{U}_t}[\langle \hat{\nabla}_1 f_{i,t}(z_{i,t}), z_{i,t} - \check{y}_{i,t} \rangle], \tag{7.34}$$

where the first equality holds from (7.32a) and the last equality holds since $z_{i,t}$ is independent of $\mathfrak{U}_t$.

Next, we rewrite the right-hand side of (7.34) into two terms and bound them individually.

$$\begin{aligned}
&\mathbf{E}_{\mathfrak{U}_t}[\langle \hat{\nabla}_1 f_{i,t}(z_{i,t}), z_{i,t} - \check{y}_{i,t} \rangle] \\
&= \mathbf{E}_{\mathfrak{U}_t}[\langle \hat{\nabla}_1 f_{i,t}(z_{i,t}), z_{i,t} - z_{i,t+1} \rangle] + \mathbf{E}_{\mathfrak{U}_t}[\langle \hat{\nabla}_1 f_{i,t}(z_{i,t}), z_{i,t+1} - \check{y}_{i,t} \rangle].
\end{aligned} \tag{7.35}$$

For the first term of the right-hand side of (7.35), the Cauchy-Schwarz inequality and (7.32c) give

$$\begin{aligned}
\langle \hat{\nabla}_1 f_{i,t}(z_{i,t}), z_{i,t} - z_{i,t+1} \rangle &\le \|\hat{\nabla}_1 f_{i,t}(z_{i,t})\| \|z_{i,t} - z_{i,t+1}\| \le \frac{p_i F_{f_i}}{\delta_{i,t}} \|z_{i,t} - z_{i,t+1}\| \\
&\le \frac{p_i^2 F_{f_i}^2 \alpha_{i,t+1}}{\delta_{i,t}^2} + \frac{1}{4\alpha_{i,t+1}} \|z_{i,t} - z_{i,t+1}\|^2.
\end{aligned} \tag{7.36}$$

For the second term of the right-hand side of (7.35), it follows from (7.8) that

$$\begin{aligned}
\mathbf{E}_{\mathfrak{U}_t}[\langle \hat{\nabla}_1 f_{i,t}(z_{i,t}), z_{i,t+1} - \check{y}_{i,t} \rangle] &= \mathbf{E}_{\mathfrak{U}_t}[\langle (\hat{\nabla}_1 g_{i,t}(z_{i,t}))^\top \tilde{q}_{i,t+1}, \check{y}_{i,t} - z_{i,t+1} \rangle] \\
&\quad + \mathbf{E}_{\mathfrak{U}_t}[\langle a_{i,t+1}, z_{i,t+1} - \check{y}_{i,t} \rangle] \\
&= \mathbf{E}_{\mathfrak{U}_t}[\langle (\hat{\nabla}_1 g_{i,t}(z_{i,t}))^\top \tilde{q}_{i,t+1}, \check{y}_{i,t} - z_{i,t} \rangle] \\
&\quad + \mathbf{E}_{\mathfrak{U}_t}[\langle (\hat{\nabla}_1 g_{i,t}(z_{i,t}))^\top \tilde{q}_{i,t+1}, z_{i,t} - z_{i,t+1} \rangle] \\
&\quad + \mathbf{E}_{\mathfrak{U}_t}[\langle a_{i,t+1}, z_{i,t+1} - \check{y}_{i,t} \rangle].
\end{aligned} \tag{7.37}$$

For the first term of the right-hand side of (7.37), noting that $x_{i,t}$ and $\tilde{q}_{i,t+1}$ are dependent of $\mathfrak{U}_t$, from (7.32d), $\tilde{q}_{i,t+1} \ge \mathbf{0}_m$, $\bar{q}_t \ge \mathbf{0}_m$, (7.32e), and that $\hat{g}_{i,t}$ is convex, we have

$$\begin{aligned}
\mathbf{E}_{\mathfrak{U}_t}[\langle (\hat{\nabla}_1 g_{i,t}(z_{i,t}))^\top \tilde{q}_{i,t+1}, \check{y}_{i,t} - z_{i,t} \rangle] &= \langle (\mathbf{E}_{\mathfrak{U}_t}[\hat{\nabla}_1 g_{i,t}(z_{i,t})])^\top \tilde{q}_{i,t+1}, \check{y}_{i,t} - z_{i,t} \rangle \\
&= \langle (\nabla \hat{g}_{i,t}(z_{i,t}))^\top \tilde{q}_{i,t+1}, \check{y}_{i,t} - z_{i,t} \rangle \\
&\le (\tilde{q}_{i,t+1})^\top \hat{g}_{i,t}(\check{y}_{i,t}) - [\tilde{q}_{i,t+1})^\top \hat{g}_{i,t}(z_{i,t}) \\
&= (\bar{q}_t)^\top (\hat{g}_{i,t}(\check{y}_{i,t}) - \hat{g}_{i,t}(z_{i,t})) \\
&\quad + (\tilde{q}_{i,t+1} - \bar{q}_t)^\top (\hat{g}_{i,t}(\check{y}_{i,t}) - \hat{g}_{i,t}(z_{i,t})) \\
&\le (\bar{q}_t)^\top (g_{i,t}(\check{y}_{i,t}) + \delta_{i,t} G_{g_i} \mathbf{1}_m - g_{i,t}(z_{i,t})) \\
&\quad + (\tilde{q}_{i,t+1} - \bar{q}_t)^\top (\hat{g}_{i,t}(\check{y}_{i,t}) - \hat{g}_{i,t}(z_{i,t})).
\end{aligned} \tag{7.38}$$

From (7.20b) and (7.32g), we have

$$(\tilde{q}_{i,t+1} - \bar{q}_t)^\top (\hat{g}_{i,t}(\check{y}_{i,t}) - \hat{g}_{i,t}(z_{i,t})) \le \frac{2d_1(t)}{n}. \tag{7.39}$$

For the second term of the right-hand side of (7.37), from the Cauchy-Schwarz inequality, (7.32f), and (7.26) we have

$$\begin{aligned}\langle(\hat{\nabla}_1 g_{i,t}(z_{i,t}))^\top \tilde{q}_{i,t+1}, z_{i,t}-z_{i,t+1}\rangle &= q^\top \hat{\nabla}_1 g_{i,t}(z_{i,t})(z_{i,t}-z_{i,t+1})\\ &\quad+(\tilde{q}_{i,t+1}-q)^\top \hat{\nabla}_1 g_{i,t}(z_{i,t})(z_{i,t}-z_{i,t+1})\\ &\le \frac{2mp_i^2F_{g_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}\|q\|^2+\frac{1}{8\alpha_{i,t+1}}\|z_{i,t+1}-z_{i,t}\|^2\\ &\quad+\frac{2mp_i^2F_{g_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}\|\tilde{q}_{i,t+1}-q\|^2+\frac{1}{8\alpha_{i,t+1}}\|z_{i,t+1}-z_{i,t}\|^2\\ &\le 2m\max_{i\in[n]}\Big\{\frac{p_i^2F_{g_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}\Big\}\Big(\|q\|^2+\sum_{j=1}^n[W_t]_{ij}\|q_{j,t}-q\|^2\Big)\\ &\quad+\frac{1}{4\alpha_{i,t+1}}\|z_{i,t+1}-z_{i,t}\|^2. \end{aligned}\tag{7.40}$$

For the last term of the right-hand side of (7.37), noting that $\check{y}_{i,t}\in(1-\xi_t)\mathbb{X}_i\subseteq(1-\xi_{t+1})\mathbb{X}_i$ due to $\xi_t\ge\xi_{t+1}$ and applying (2.12) to the update rule (7.7b) yield

$$\begin{aligned}2\alpha_{i,t+1}\langle a_{i,t+1}, z_{i,t+1}-\check{y}_{i,t}\rangle &\le \|\check{y}_{i,t}-z_{i,t}\|^2-\|\check{y}_{i,t}-z_{i,t+1}\|^2-\|z_{i,t+1}-z_{i,t}\|^2\\ &=\|\check{y}_{i,t+1}-z_{i,t+1}\|^2-\|\check{y}_{i,t}-z_{i,t+1}\|^2+\|\check{y}_{i,t}-z_{i,t}\|^2\\ &\quad-\|\check{y}_{i,t+1}-z_{i,t+1}\|^2-\|z_{i,t+1}-z_{i,t}\|^2. \end{aligned}\tag{7.41}$$

The first two terms of the right-hand side of (7.41) can be bounded by

$$\begin{aligned}\|\check{y}_{i,t+1}-z_{i,t+1}\|^2-\|\check{y}_{i,t}-z_{i,t+1}\|^2 &\le \|\check{y}_{i,t+1}-\check{y}_{i,t}\|\|\check{y}_{i,t+1}+\check{y}_{i,t}-2z_{i,t+1}\|\\ &\le 4R_i\|(1-\xi_{t+1})y_{i,t+1}-(1-\xi_t)y_{i,t}\|\\ &=4R_i\|(1-\xi_{t+1})(y_{i,t+1}-y_{i,t})+(\xi_t-\xi_{t+1})y_{i,t}\|\\ &\le 4R_i\|y_{i,t+1}-y_{i,t}\|+4R_i^2(\xi_t-\xi_{t+1}), \end{aligned}\tag{7.42}$$

where the last inequality holds since $\{\xi_t\}\subseteq(0,1)$ is nonincreasing.

Combining (7.33c)–(7.42), taking expectation in $\mathfrak{U}_t$, summing over $i\in[n]$, and rearranging the terms yields (7.31). □

Lemma 7.3. *Suppose that Assumptions 7.1–7.2 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 7.1. Then, for any $T\in\mathbb{N}_+$ and any comparator sequence $\boldsymbol{y}_{[T]}\in\mathcal{X}_T$,*

$$\begin{aligned}\mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]},\boldsymbol{y}_{[T]})] &\le \sum_{t=1}^T\mathbf{E}[d_2(t)]+C_0\sum_{t=1}^T\gamma_{t+1}+\sum_{t=1}^T\sum_{i=1}^n\frac{p_i^2F_{f_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}+\sum_{i=1}^n\frac{2R_i^2}{\alpha_{i,T+1}}\\ &\quad+\sum_{t=1}^{T-1}\sum_{i=1}^n\frac{2R_i\|y_{i,t+1}-y_{i,t}\|}{\alpha_{i,t+1}}+\frac{1}{2}\sum_{t=1}^T\tilde{\alpha}_t\mathbf{E}[\|q_{i,t}\|^2], \end{aligned}\tag{7.43a}$$

$$\mathbf{E}[\|[\sum_{t=1}^{T} g_t(x_t)]_+\|^2] \le d_5(T)\Big(\sum_{t=1}^{T}\mathbf{E}[d_2(t)] + C_0\sum_{t=1}^{T}\gamma_{t+1} + \sum_{t=1}^{T}\sum_{i=1}^{n}\frac{p_i^2F_{f_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2} + \sum_{i=1}^{n}\frac{2R_i^2}{\alpha_{i,T+1}} + 2T\sum_{i=1}^{n}F_{f_i} + \frac{1}{2}\sum_{t=1}^{T}\tilde{\alpha}_t\mathbf{E}[\|q_{i,t}-q_c\|^2]\Big), \tag{7.43b}$$

where

$$\tilde{\alpha}_t = \sum_{i=1}^{n}\Big(4m\max_{i\in[n]}\Big\{\frac{p_i^2F_{g_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}\Big\} + \frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} - \beta_{t+1}\Big),$$

$$d_5(T) = 2n\Big(\frac{1}{\gamma_1} + \sum_{t=1}^{T}\Big(4m\max_{i\in[n]}\Big\{\frac{p_i^2F_{g_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2}\Big\} + \beta_{t+1}\Big)\Big),$$

$$q_c = \frac{2[\sum_{t=1}^{T} g_t(x_t)]_+}{d_5(T)} \in \mathbb{R}_+^m.$$

Proof. **(i)** For any $\lambda \in (0,1)$ and nonnegative sequence $\zeta_1, \zeta_2, \ldots$, it holds that

$$\sum_{t=1}^{T}\sum_{s=1}^{t}\zeta_{s+1}\lambda^{t-s} = \sum_{t=1}^{T}\zeta_{t+1}\sum_{s=0}^{T-t}\lambda^s \le \frac{1}{1-\lambda}\sum_{t=1}^{T}\zeta_{t+1}. \tag{7.44}$$

Thus,

$$\sum_{t=1}^{T} d_1(t) \le \frac{2\sqrt{m}n^2\tau B_1F_g}{1-\lambda}\sum_{t=1}^{T}\gamma_{t+1}. \tag{7.45}$$

The definition of Δ_t given in Lemma 7.1 yields

$$\begin{aligned}
-\sum_{t=1}^{T}\frac{\Delta_{t+1}}{2\gamma_{t+1}} &= \sum_{t=1}^{T}\frac{1}{2\gamma_{t+1}}\sum_{i=1}^{n}((1-\beta_{t+1}\gamma_{t+1})\|q_{i,t}-q\|^2 - \|q_{i,t+1}-q\|^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{t=1}^{T}\Big(\frac{1}{\gamma_t}\|q_{i,t}-q\|^2 - \frac{1}{\gamma_{t+1}}\|q_{i,t+1}-q\|^2\Big) \\
&\quad + \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} - \beta_{t+1}\Big)\|q_{i,t}-q\|^2 \\
&= \frac{1}{2}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_1}\|q_{i,1}-q\|^2 - \frac{1}{\gamma_{T+1}}\|q_{i,T+1}-q\|^2\Big) \\
&\quad + \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} - \beta_{t+1}\Big)\|q_{i,t}-q\|^2 \\
&\le \frac{n}{2\gamma_1}\|q\|^2 + \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} - \beta_{t+1}\Big)\|q_{i,t}-q\|^2,
\end{aligned} \tag{7.46}$$

where the last inequality holds due to $q_{i,1} = \mathbf{0}_m$ and $\|q_{i,T+1} - q\|^2 \geq 0$.

From the properties of conditional expectation, we know that

$$\mathbf{E}_{\mathcal{U}_T}[\mathbf{E}_{\mathfrak{U}_t}[d_4(t)]] = \mathbf{E}[d_4(t)], \ \forall t \in [T], \tag{7.47}$$

where we recall the definition $\mathcal{U}_T = \bigcup_{s=1}^{T} \mathfrak{U}_s$.

Noting that $\{\alpha_t\}$ is nonincreasing and (7.3), for any $s \in [T]$, we have

$$\begin{aligned}
\sum_{t=s}^{T} d_4(t) &= \frac{1}{2}\sum_{t=s}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_{i,t}}\|\check{y}_{i,t} - z_{i,t}\|^2 - \frac{1}{\alpha_{i,t+1}}\|\check{y}_{i,t+1} - z_{i,t+1}\|^2\Big) \\
&\quad + \frac{1}{2}\sum_{t=s}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\alpha_{i,t+1}} - \frac{1}{\alpha_{i,t}}\Big)\|\check{y}_{i,t} - z_{i,t}\|^2 \\
&\leq \frac{1}{2\alpha_{i,s}}\sum_{i=1}^{n}\|\check{y}_{i,s} - z_{i,s}\|^2 - \frac{1}{2\alpha_{i,T+1}}\sum_{i=1}^{n}\|\check{y}_{i,T+1} - z_{i,T+1}\|^2 \\
&\quad + 2\sum_{i=1}^{n}\Big(\frac{1}{\alpha_{i,T+1}} - \frac{1}{\alpha_{i,s}}\Big)R_i^2 \\
&\leq \sum_{i=1}^{n}\frac{2R_i^2}{\alpha_{i,T+1}}.
\end{aligned} \tag{7.48}$$

Let $g_c : \mathbb{R}_+^m \to \mathbb{R}$ be a function defined as

$$g_c(q) = \Big(\sum_{t=1}^{T} g_t(x_t)\Big)^\top q - \frac{d_5(T)}{4}\|q\|^2. \tag{7.49}$$

Combining (7.20c) and (7.31), summing over $t \in [T]$, using (7.45)–(7.49) and $g_t(y_t) \leq \mathbf{0}_m$, $\boldsymbol{y}_{[T]} \in \mathcal{X}_T$, and taking expectation in $\mathcal{U}_T$ yields

$$\begin{aligned}
\mathbf{E}[g_c(q)] + \mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]}, \boldsymbol{y}_{[T]})] &\leq \sum_{t=1}^{T}\mathbf{E}[d_2(t)] + C_0\sum_{t=1}^{T}\gamma_{t+1} + \sum_{t=1}^{T}\sum_{i=1}^{n}\frac{p_i^2F_{f_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2} \\
&\quad + \sum_{i=1}^{n}\frac{2R_i^2}{\alpha_{i,T+1}} + \sum_{t=1}^{T}\sum_{i=1}^{n}\frac{2R_i\|y_{i,t+1} - y_{i,t}\|}{\alpha_{i,t+1}} \\
&\quad + \frac{1}{2}\sum_{t=1}^{T}\tilde{\alpha}_t\mathbf{E}[\|q_{i,t} - q\|^2], \ \forall q \in \mathbb{R}_+^m.
\end{aligned} \tag{7.50}$$

Then, substituting $q = \mathbf{0}_m$ into (7.50), setting $y_{i,T+1} = y_{i,T}$, and noting that $\{\alpha_t\}$ is nonincreasing yields (7.43a).

(ii) Substituting $q = q_c$ into $g_c(q)$ gives

$$g_c(q_c) = \frac{\|[\sum_{t=1}^{T} g_t(x_t)]_+\|^2}{d_5(T)}. \tag{7.51}$$

Moreover, (7.4) gives

$$|\operatorname{Reg}(\boldsymbol{x}_{[T]},\boldsymbol{y}_{[T]})| \le 2T\sum_{i=1}^{n}F_{f_i},\ \forall \boldsymbol{y}_{[T]} \in \mathcal{X}_T. \tag{7.52}$$

Substituting $q = q_c$ and $y_t = \check{x}^*_T$, $t \in [T+1]$ into (7.50), combining (7.51)–(7.52), and rearranging the terms gives (7.43b). □

We are now ready to prove Theorem 7.1.

(i) Applying (2.65), (2.37), and (7.20a) to the first three terms of the right-hand side of (7.43a) and noting $\theta_2 < \theta_3$ gives

$$\sum_{t=1}^{T}\mathbf{E}[d_2(t)] \le C_{1,2}T^{1-\theta_3+\theta_2} + C_{1,3}T^{1-\theta_3} + C_{1,1}\log(T), \tag{7.53a}$$

$$C_0\sum_{t=1}^{T}\gamma_{t+1} \le \frac{C_0}{\theta_2}T^{\theta_2}, \tag{7.53b}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{n}\frac{p_i^2F_{f_i}^2\alpha_{i,t+1}}{\delta_{i,t}^2} \le C_{1,4}T^{1-\theta_1+2\theta_3}. \tag{7.53c}$$

From (7.11) and $\theta_1 - 2\theta_3 \ge \theta_2$ we know that

$$\begin{aligned}\tilde{\alpha}_t &= \frac{1}{(t+1)^{\theta_1-2\theta_3}} + \frac{t+1}{(t+1)^{\theta_2}} - \frac{t}{t^{\theta_2}} - \frac{2}{(t+1)^{\theta_2}}\\ &\le \frac{1}{(t+1)^{\theta_2}} + \frac{t+1}{(t+1)^{\theta_2}} - \frac{t}{t^{\theta_2}} - \frac{2}{(t+1)^{\theta_2}}\\ &= \frac{t}{(t+1)^{\theta_2}} - \frac{t}{t^{\theta_2}} < 0.\end{aligned} \tag{7.54}$$

Combining (7.43a) and (7.53a)–(7.54) yields (7.12a).

(ii) Using (2.37) and noting $\theta_1 - 2\theta_3 \ge \theta_2$ gives

$$d_5(T) \le C_{2,1}T^{1-\theta_2}. \tag{7.55}$$

Combining (7.43b) and (7.53a)–(7.55) gives

$$\mathbf{E}\Big[\Big\|\Big[\sum_{t=1}^{T}g_t(x_t)\Big]_+\Big\|^2\Big] \le C_2T^{2-\theta_2}. \tag{7.56}$$

Finally, combining (7.56) and $(\mathbf{E}[\|[\sum_{t=1}^{T}g_t(x_t)]_+\|])^2 \le \mathbf{E}[\|[\sum_{t=1}^{T}g_t(x_t)]_+\|^2]$ (which follows from Jensen's inequality) gives (7.12b).

7.7.2 Proof of Theorem 7.2

The proof is similar to the proof of Theorem 7.1 with some modifications. Lemmas 7.1–7.3 are replaced by Lemmas 7.4–7.6.

To simplify notation, we denote $\alpha_t = \alpha_{i,t}$, $\beta_t = \beta_{i,t}$, $\gamma_t = \gamma_{i,t}$, and $\xi_t = \xi_{i,t}$.

Lemma 7.4. *Suppose that Assumptions 7.1–7.2 hold. For all $i \in [n]$ and $t \in \mathbb{N}_+$, $\tilde{q}_{i,t}$ and $q_{i,t}$ generated by Algorithm 7.2 satisfy*

$$\|\tilde{q}_{i,t+1}\| \le \frac{B_1}{\beta_t},\ \|q_{i,t}\| \le \frac{B_1}{\beta_t}, \tag{7.57a}$$

$$\|\tilde{q}_{i,t+1} - \bar{q}_t\| \le 2nB_1\tau \sum_{s=1}^{t-1} \gamma_{s+1}\lambda^{t-1-s}, \tag{7.57b}$$

$$\begin{aligned}\frac{\Delta_{t+1}}{2\gamma_{t+1}} &\le (\bar{q}_t - q)^\top g_t(x_t) + 2nB_1^2\gamma_{t+1} + d_6(t)\\ &\quad + \frac{1}{2}\sum_{i=1}^n (2mp_i^2G_{g_i}^2\alpha_{t+1} + \beta_{t+1})\|q\|^2 + d_7(t),\end{aligned} \tag{7.57c}$$

where q is an arbitrary vector in $\mathbb{R}^m_+$, and

$$d_6(t) = 2\sqrt{m}n^2B_1F_g\tau\sum_{s=1}^{t}\gamma_{s+1}\lambda^{t-s},$$

$$d_7(t) = \frac{1}{4\alpha_{t+1}}\sum_{i=1}^n \|x_{i,t+1} - x_{i,t}\|^2 + \sum_{i=1}^n (\tilde{q}_{i,t+1})^\top \hat{\nabla}_2 g_{i,t}(x_{i,t})(x_{i,t+1} - x_{i,t}).$$

Proof. From the fifth part in Lemma 2.16 and (7.6b), we know that for all $i \in [n]$, $x \in (1-\xi_{i,t})\mathbb{X}_i$, and $t \in \mathbb{N}_+$,

$$\|\hat{\nabla}_2 g_{i,t}(x)\| \le \sqrt{m}p_iG_{g_i}. \tag{7.58}$$

Hence, (7.16), (7.3), (7.4), and (7.58) yield

$$\begin{aligned}\|c_{i,t+1}\| &\le \|g_{i,t}(x_{i,t})\| + \|\hat{\nabla}_2 g_{i,t}(x_{i,t})\|\|(x_{i,t+1} - x_{i,t})\|\\ &\le \sqrt{m}F_{g_i} + 2\sqrt{m}p_iG_{g_i}R_i \le B_1,\ \forall i \in [n],\ \forall t \in \mathbb{N}_+.\end{aligned} \tag{7.59}$$

Replacing $z_{i,t}$ and $g_{i,t}(z_{i,t})$ by $x_{i,t}$ and $c_{i,t+1}$, respectively, and following steps similar to those used to prove (7.20a) and (7.20b) yields (7.57a) and (7.57b).

Applying (2.10) to (7.14c) yields

$$\begin{aligned}\|q_{i,t} - q\|^2 &\le \|(1-\beta_t\gamma_t)\tilde{q}_{i,t} + \gamma_t c_{i,t} - q\|^2\\ &= \|\tilde{q}_{i,t} - q\|^2 + \gamma_t^2\|c_{i,t} - \beta_t\tilde{q}_{i,t}\|^2 + 2\gamma_t(\tilde{q}_{i,t})^\top\hat{\nabla}_2 g_{i,t-1}(x_{i,t-1})(x_{i,t} - x_{i,t-1})\\ &\quad - 2\gamma_t q^\top \hat{\nabla}_2 g_{i,t-1}(x_{i,t-1})(x_{i,t} - x_{i,t-1}) + 2\gamma_t(\tilde{q}_{i,t} - q)^\top g_{i,t-1}(x_{i,t-1})\\ &\quad - 2\beta_t\gamma_t(\tilde{q}_{i,t} - q)^\top\tilde{q}_{i,t}.\end{aligned} \tag{7.60}$$

For the fourth term of the right-hand side of (7.60), (7.58) and the Cauchy-Schwarz inequality yield

$$-2\gamma_t q^\top \hat{\nabla}_2 g_{i,t-1}(x_{i,t-1})(x_{i,t}-x_{i,t-1}) \le 2\gamma_t\Big(mp_i^2 G_{g_i}^2\alpha_t\|q\|^2 + \frac{1}{4\alpha_t}\|x_{i,t}-x_{i,t-1}\|^2\Big). \quad (7.61)$$

Replacing (7.25) by (7.60), using (7.61), and following steps similar to those used to prove (7.20c) yields (7.57c). □

Lemma 7.5. *Suppose that Assumptions 7.1–7.2 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 7.2 and $\{y_t\}$ be an arbitrary sequence in $\mathbb{X}$, then*

$$\begin{aligned} f_t(x_t) - f_t(y_t) \le{}& (\bar{q}_t)^\top(g_t(y_t) - g_t(x_t)) + 2d_6(t) - E_{\mathfrak{U}_t}[d_7(t)] + d_8(t) + \mathbf{E}_{\mathfrak{U}_t}[d_9(t)] \\ &+ \sum_{i=1}^n p_i^2 G_{f_i}^2\alpha_{t+1} + \sum_{i=1}^n \frac{2R_i\|y_{i,t+1}-y_{i,t}\|}{\alpha_{t+1}},\ \forall t\in\mathbb{N}_+, \end{aligned} \quad (7.62)$$

where

$$\begin{aligned} d_8(t) &= \sum_{i=1}^n \Big((\delta_{i,t}+R_i\xi_t)(\sqrt{m}G_{g_i}\|q_{i,t}\| + G_{f_i}) + \frac{2R_i^2(\xi_t-\xi_{t+1})}{\alpha_{t+1}}\Big), \\ d_9(t) &= \frac{1}{2\alpha_{t+1}}\sum_{i=1}^n(\|\check{y}_{i,t}-x_{i,t}\|^2 - \|\check{y}_{i,t+1}-x_{i,t+1}\|^2),\ \check{y}_{i,t} = (1-\xi_t)y_{i,t}. \end{aligned}$$

Proof. Replacing $z_{i,t}$, $a_{i,t}$, and (7.32c) by $x_{i,t}$, $b_{i,t}$, and

$$\|\hat{\nabla}_2 f_{i,t}(x)\| \le p_i G_{f_i}, \quad (7.63)$$

respectively, deleting (7.40), and following steps similar to those used to prove (7.31) yields (7.62). □

Lemma 7.6. *Suppose that Assumptions 7.1–7.2 hold. Let $\{x_t\}$ be the sequence generated by Algorithm 7.2. Then, for any $T\in\mathbb{N}_+$ and any comparator sequence $\boldsymbol{y}_{[T]}\in\mathcal{X}_T$,*

$$\begin{aligned} \mathbf{E}[\mathrm{Reg}(\boldsymbol{x}_{[T]},\boldsymbol{y}_{[T]})] \le{}& \sum_{t=1}^T \mathbf{E}[d_8(t)] + \hat{C}_0\sum_{t=1}^T\gamma_{t+1} + \sum_{i=1}^n\frac{2R_i^2}{\alpha_{T+1}} + \sum_{t=1}^T\sum_{i=1}^n p_i^2G_{f_i}^2\alpha_{t+1} \\ &+ \frac{2R_{\max}V(\boldsymbol{y}_{[T]})}{\alpha_T} + \frac{1}{2}\sum_{t=1}^T\sum_{i=1}^n\Big(\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t} - \beta_{t+1}\Big)\mathbf{E}[\|q_{i,t}\|^2], \quad (7.64a) \end{aligned}$$

$$\begin{aligned} \mathbf{E}\Big[\Big\|\Big[\sum_{t=1}^T g_t(x_t)\Big]_+\Big\|^2\Big] \le{}& d_{10}(T)\Big(\sum_{t=1}^T\mathbf{E}[d_8(t)] + \hat{C}_0\sum_{t=1}^T\gamma_{t+1} + \sum_{i=1}^n\frac{2R_i^2}{\alpha_{T+1}} \\ &+ \sum_{t=1}^T\sum_{i=1}^n p_i^2G_{f_i}^2\alpha_{t+1} + 2T\sum_{i=1}^n F_{f_i} \end{aligned}$$

$$+\frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\frac{1}{\gamma_{t+1}}-\frac{1}{\gamma_t}-\beta_{t+1}\Big)\mathbf{E}[\|q_{i,t}-\hat{q}_c\|^2]\Big), \tag{7.64b}$$

where

$$d_{10}(T)=2n\Big(\frac{1}{\gamma_1}+\sum_{t=1}^{T}(2mp_i^2G_{g_i}^2\alpha_{t+1}+\beta_{t+1})\Big),\ \hat{q}_c=\frac{2[\sum_{t=1}^{T}g_t(x_t)]_+}{d_{10}(T)}\in\mathbb{R}_+^m.$$

Proof. With Lemmas 7.4 and 7.5 at hand, the proof of Lemma 7.6 follows steps similar to those used to prove Lemma 7.3. □

With Lemmas 7.4–7.6 at hand, the proofs of (7.18a) and (7.18b) in Theorem 7.2 follow steps similar to those used to prove (7.12a) and (7.12b) in Theorem 7.1.

Part III

Distributed Event-Triggered Control

Chapter 8

Distributed dynamic event-triggered control algorithms

In this and the following two chapters, in order to achieve a common control objective for a networked system, we propose distributed event-triggered algorithms to reduce the amount of information exchanged. In this chapter, we propose two novel dynamic event-triggered control laws to solve the average consensus problem for first-order continuous-time multi-agent systems over undirected graphs. Compared with most existing triggering laws, the proposed laws involve internal dynamic variables, which play an essential role in guaranteeing that the triggering time sequence does not exhibit Zeno behavior. Moreover, some existing triggering laws are special cases of ours. For the proposed self-triggered algorithm, continuous agent listening is avoided as each agent predicts its next triggering time and broadcasts it to its neighbors at the current triggering time. Thus, each agent only needs to sense and broadcast at its triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. It is proved that the proposed triggering laws ensure that the state of each agent converge exponentially to the average of all agents' initial states if and only if the underlying graph is connected. The idea behind these approaches will also play an important role in the following two chapters. Numerical simulations are provided to illustrate the effectiveness of the theoretical results.

This chapter is organized as follows. Section 8.1 gives the background. Section 8.2 introduces the problem formulation. Section 8.3 presents two distributed dynamic event-triggered control algorithms to determine triggering times such that average consensus is achieved exponentially. A self-triggered control algorithm to solve the aforementioned problem is presented in Section 8.4. Simulations are given in Section 8.5. This chapter is concluded in Section 8.6.

8.1 Introduction

The consensus problem has a long history in computer science, particular in distributed computing [332]. For multi-agent systems, consensus means that the group of agents reach an agreement upon a certain quantity of interest that may depend on the initial states of all agents. In the study of complex networks, the synchronization has sometime a similar meaning as consensus.

There is a huge amount of research work on consensus or synchronization in the past decades. Here we only recall some of them. In [333–337], the authors introduced theoretical frameworks for analysis of consensus for first-order linear multi-agent systems with an emphasis on the role of directed information flow, robustness to changes in network topology due to link/node failures, time-delays, and performance guarantees. One fundamental result is that the performance of the consensus protocol is determined by the algebraic connectivity. Consensus is achieved if and only if the underlying fixed undirected graph is connected or directed graph has a directed spanning tree [333–335]. In [338], the authors studied general linear multi-agent systems with directed communication graphs. Similar work can be found in earlier studies [198, 339], in which the authors presented a framework for analysing synchronization of linearly coupled ordinary differential equations. In [340], the authors used a high-gain methodology to construct linear decentralized consensus controllers for general linear multi-agent systems with time-invariant and time-varying topologies. In [341], the authors considered consensus for first-order multi-agent systems with stochastically switching topologies modeled as a stochastic process. In [342], the authors studied asynchronous consensus problems for continuous-time multi-agent systems with discontinuous information transmission. In [343], the authors investigated the joint effect of agent dynamics, network topologies and communication data rate on the consensus problem. In [344], the authors considered nonlinear consensus protocols.

The average consensus problem involves a group of agents in a network who seeks the average of a set of network-wide measurements or states. It has been widely investigated because its many applications in sensor networks, mobile robots, autonomous underwater vehicles, and unmanned air vehicles, e.g., [336] and the references therein. In these studies, agents have continuous-time dynamics and actuation. However, continuous communication cannot usually be implemented in multi-agent systems, since the interactions among agents are typically realized over a digital communication channel with limited capacity. Moreover, in order to simplify and reduce communication, the information exchange should be kept as small as possible. In order to realize this, in practice, typically agents communicate with their neighbors and take actions at discrete time points. There are various studies considered agents with discrete-time dynamics or continuous-time dynamics but discontinuous information transmission, e.g., [342, 343]. In these studies, time-triggered sampling was used to determine when agents should establish communication with its neighbors, which is often implemented periodically. A nice feature of such a model is that analysis and design becomes rather straightforward and the vast literature on sample-data control can be used [192]. Drawbacks are that agents need to take actions in a synchronous manner, which is often hard to implement when the number of agents is large, and it is not energy-efficient to communicate when the state has not changed much.

Event-triggered sampling has been proposed for single-agent systems [345–347]. The concept was originally extended to multi-agent systems in [193]. In event-triggered multi-agent systems actuation updates and inter-agent communications occur only when some specific events are triggered, for instance, a measure of the state error exceeds a specified threshold. The control is often constant between any two consecutive triggering times. In [348], by introducing an internal dynamic variable, a new class of event-triggered mechanisms was presented and it was extended to discrete-time setting in [349]. The idea of using internal dynamic variables in event- and self-triggered control can also be found in [350–354]. Many researchers studied event-triggered control for multi-agent systems recently [193–195, 200, 355–360]. A key challenge is how to design triggering laws to determine the corresponding triggering times, while excluding Zeno behavior, i.e., infinite number of triggers in a finite time interval [196].

To overcome the drawback of continuous monitoring of the triggering law, self-triggered control were proposed for single-agent systems [361–363]. Many researchers have investigated self-triggered control for multi-agent systems [193, 200, 358]. For self-triggered single-agent systems, the next triggering time is determined at the previous triggering instance. However, the self-triggered approaches for multi-agent systems mentioned above are not in accordance with this. Although continuous sensing of each agent's own and neighbors' states is avoided in these studies, continuous listening is still needed since the triggering times are determined during runtime and not known in advance. To overcome this drawback, some researchers introduced local clock variables in the self-triggering policy [350], others combined event-triggered control with periodic sampling [351, 355, 357], and some proposed cloud-supported algorithms [364]. By introducing an internal dynamic variable, a new class of event-triggering mechanisms was presented in [348] and later extended to a discrete-time setting in [349]. The idea of using internal dynamic variables in event- and self-triggered control can also be found in [350–352, 365]. In this chapter, we make essential modifications to the dynamic event-triggering mechanism for single-agent systems in [348] and extend it to multi-agent systems.

In this chapter, we propose two novel dynamic event-triggered control laws to solve the average consensus problem for first-order continuous-time multi-agent systems over undirected graphs. We have the following contributions.

(C8.1) The first main contribution of this chapter is in the introduction and convergence analysis of dynamic event- and self-triggered control laws for multi-agent systems. The control laws are truly distributed in the sense that they do not require any a priori knowledge of global network parameters. We prove that the proposed dynamic triggering laws yield consensus exponentially fast, and we show that they are free from Zeno behavior by verifying that the triggering time sequence of each agent is divergent. We show also that the triggering laws in [194, 195] are special cases of our event-triggered law.

(C8.2) To overcome the main disadvantage of event-triggered laws, i.e., avoid continuous sensing and listening, we present a self-triggered control law. The main idea to avoid continuous listening is that each agent predicts its next triggering time and broadcasts it to its neighbors at the current triggering time. As a result, each agent only needs to sense and broadcast at its triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. This is to say that, in terms of avoiding continuous listening, our self-triggered algorithm improves the ones in [193, 200, 358] and other studies using a similar approach. Although continuous sensing, broadcasting, listening, and receiving are also avoided in [351, 355, 357] by combining event-triggered control with periodic sampling, the additional periodic sensing and listening are still needed. Moreover, it is not clear how to show that the average inter-event time is strictly larger than the required sampling period. Our self-triggered control law is reminiscent of the event-triggered cloud access in [364]. The main difference is that we do not need the cloud to store data and we use different analysis techniques.

8.2 Average consensus for first-order multi-agent systems

We consider a set of n agents modelled as single integrators

$$\dot{x}_i(t) = u_i(t),\ i \in [n],\ t \geq 0, \tag{8.1}$$

where $x_i(t) \in \mathbb{R}$ is the state and $u_i(t) \in \mathbb{R}$ is the control input.

Remark 8.1. *For the ease of presentation, we study the case where all the agents have scalar states, i.e., $x_i \in \mathbb{R}$. However, the analysis in this chapter is also valid for the cases where the agents have vector-valued states, i.e., $x_i \in \mathbb{R}^p$.*

Definition 8.1 (Average consensus). *We say average consensus for the multi-agent system (8.1) is achieved if* $\lim_{t\to\infty} x_i(t) = \frac{1}{n}\sum_{j=1}^{n} x_j(0),\ \forall i \in [n]$.

The classic distributed consensus protocol is given by [336, 337],

$$u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t), \tag{8.2}$$

where L_{ij} is the element of the Laplacian matrix L. In this chapter, we assume that the underlying graph $\mathcal{G}$ is undirected. Figure 1.10 shows how agents communicate when the control input has the form (8.2).

To implement the consensus protocol (8.2), a continuous exchange of information among agents and a continuous update of actuators are needed. However, it is often impractical to require continuous communication and update in real applications.

Inspired by the idea of event-triggered control for multi-agent systems [193], we use the event-triggered control input

$$u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t^j_{k_j(t)}). \tag{8.3}$$

Note that the event-triggered control input (8.3) only updates at the triggering times and it remains constant between any two consecutive triggering times. Figure 1.9 shows how agents communicate when the control input has the form (8.3).

Our goal in this chapter is to solve the following problem.

Problem 8.1. *Propose methods to determine the triggering times such that average consensus is reached, while continuous exchange of information, continuous update of actuators, and Zeno behavior are avoided.*

For simplicity, let $x(t) = \mathrm{col}(x_1(t), \dots, x_n(t))$, $\hat{x}_i(t) = x_i(t^i_{k_i(t)})$, $\hat{x}(t) = \mathrm{col}(\hat{x}_1(t), \dots, \hat{x}_n(t))$, $e_i(t) = \hat{x}_i(t) - x_i(t)$, and $e(t) = \mathrm{col}(e_1(t), \cdots, e_n(t)) = \hat{x}(t) - x(t)$. Then we can rewrite the multi-agent system with agent dynamics as in (8.1) and event-triggered control input as in (8.3) in the stack vector form

$$\dot{x}(t) = -L\hat{x}(t) = -L(x(t) + e(t)).$$

8.3 Distributed dynamic event-triggered control algorithms

In this section, we will propose two distributed dynamic event-triggered control algorithms to design the triggering times such that the average consensus can be achieved.

8.3.1 Continuous approach

We first show that the average state in (8.1) is constant.

Lemma 8.1. *Consider the multi-agent system (8.1)–(8.3), and assume that the underlying graph $\mathcal{G}$ is undirected. The average of all agents' states $\bar{x}(t) = \frac{1}{n}\sum_{i=1}^n x_i(t)$ is constant, i.e., $\bar{x}(t) = \bar{x}(0),\ \forall t \geq 0$.*

Proof. It follows from (8.1)–(8.3) that the time derivative of the average value is given by

$$\dot{\bar{x}}(t) = \frac{1}{n}\sum_{i=1}^n \dot{x}_i(t) = -\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n L_{ij}x_j(t^j_{k_j(t)}) = -\frac{1}{n}\sum_{j=1}^n x_j(t^j_{k_j(t)})\sum_{i=1}^n L_{ij} = 0.$$

Thus $\bar{x}(t)$ is constant. □

Now, consider a Lyapunov candidate as follows

$$\begin{aligned} V(x(t)) &= \frac{1}{2}x^\top(t)K_n x(t) = \frac{1}{2}x^\top(t)(I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)x(t) \\ &= \frac{1}{2}\sum_{i=1}^n x_i^2(t) - \frac{n}{2}\bar{x}^2(0) = \frac{1}{2}\sum_{i=1}^n (x_i(t) - \bar{x}(0))^2. \end{aligned} \tag{8.4}$$

Then the derivative of $V(x(t))$ along the trajectories of the multi-agent system (8.1)–(8.3) satisfies

$$\begin{aligned} \dot{V}(x(t)) &= \sum_{i=1}^n [x_i(t) - \bar{x}(0)]\dot{x}_i(t) = \sum_{i=1}^n x_i(t)\dot{x}_i(t) - \bar{x}(0)\sum_{i=1}^n \dot{x}_i(t) = \sum_{i=1}^n x_i(t)\dot{x}_i(t) \\ &= \sum_{i=1}^n x_i(t)\sum_{j=1}^n(-L_{ij}x_j(t^j_{k_j(t)})) = -\sum_{i=1}^n x_i(t)\sum_{j=1}^n L_{ij}(x_j(t) + e_j(t)) \end{aligned}$$

$$
\begin{aligned}
&\overset{*}{=} -\sum_{i=1}^{n} q_i(t) - \sum_{i=1}^{n}\sum_{j=1}^{n} x_i(t)L_{ij}e_j(t) = -\sum_{i=1}^{n} q_i(t) - \sum_{i=1}^{n}\sum_{j=1}^{n} e_i(t)L_{ij}x_j(t) \\
&= -\sum_{i=1}^{n} q_i(t) - \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} e_i(t)L_{ij}(x_j(t)-x_i(t)) \\
&\leq -\sum_{i=1}^{n} q_i(t) - \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} L_{ij}e_i^2(t) - \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} L_{ij}\frac{1}{4}(x_j(t)-x_i(t))^2 \\
&= -\sum_{i=1}^{n} q_i(t) + \sum_{i=1}^{n} L_{ii}e_i^2(t) - \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{4}L_{ij}(x_j(t)-x_i(t))^2 \\
&\overset{*}{=} -\sum_{i=1}^{n} \frac{1}{2}q_i(t) + \sum_{i=1}^{n} L_{ii}e_i^2(t),
\end{aligned}
\tag{8.5}
$$

where

$$
q_i(t) = -\frac{1}{2}\sum_{j=1}^{n} L_{ij}(x_j(t)-x_i(t))^2 \geq 0, \tag{8.6}
$$

and the equalities denoted by $\overset{*}{=}$ hold due to

$$
\sum_{i=1}^{n} q_i(t) = -\sum_{i=1}^{n}\frac{1}{2}\sum_{j=1}^{n} L_{ij}(x_j(t)-x_i(t))^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i(t)L_{ij}x_j(t) = x^\top(t)Lx(t),
$$

and the inequality holds due to $ab \leq a^2 + \frac{1}{4}b^2$.

Similar to [193] and [358], the following law can be used to determine the triggering times:

$$
t_1^i = 0,\ t_{k+1}^i = \min\Big\{t:\ L_{ii}e_i^2(t) - \frac{\sigma_i}{2}q_i(t) \geq 0,\ t \geq t_k^i\Big\},\ k \in \mathbb{N}_+, \tag{8.7}
$$

where $\sigma_i \in (0,1)$ is a design parameter. From the way to determine the triggering times by (8.7), we have

$$
L_{ii}e_i^2(t) \leq \frac{\sigma_i}{2}q_i(t),\ \forall t \geq 0. \tag{8.8}
$$

Then, from (8.5) and (8.8), we have

$$
\begin{aligned}
\dot V(x(t)) &\leq -\sum_{i=1}^{n}\frac{1}{2}q_i(t) + \sum_{i=1}^{n} L_{ii}e_i^2(t) \leq -\frac{1}{2}(1-\sigma_{\max})\sum_{i=1}^{n} q_i(t) = -\frac{1}{2}(1-\sigma_{\max})x^\top(t)Lx(t) \\
&\leq -\frac{1}{2}(1-\sigma_{\max})\rho_2(L)x^\top(t)K_nx(t) = -(1-\sigma_{\max})\rho_2(L)V(x(t)),
\end{aligned}
$$

where $\sigma_{\max} = \max_{i\in[n]}\{\sigma_i\} < 1$ and the last inequality holds due to (2.6). Then

$$
V(x(t)) \leq V(x(0))e^{-(1-\sigma_{\max})\rho_2(L)t},\ \forall t \geq 0. \tag{8.9}
$$

This implies that system (8.1)–(8.3) reaches average consensus exponentially if the underlying graph $\mathcal{G}$ is connected.

Remark 8.2. *Note that (8.7) is a static triggering law since it does not involve any extra dynamic variables but the agent state variables $x_i(t), \hat{x}_i(t)$ and $x_j(t), j \in \mathcal{N}_i$. The static triggering law (8.7) is distributed since each agent's control input only depends on its own state information and its neighbors' state information, without any a prior knowledge of any global parameters, such as the eigenvalues of the Laplacian matrix.*

Remark 8.3. *If we consider the same unweighted graph as in [193], i.e., $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, then $L_{ii} = |\mathcal{N}_i|$. Noting $a(1 - a|\mathcal{N}_i|) \le \frac{1}{4|\mathcal{N}_i|}$ and $(\sum_{j=1}^n (x_j(t) - x_i(t)))^2 \le 2|\mathcal{N}_i| \sum_{j=1}^n (x_j(t) - x_i(t))^2$, we have $\frac{\sigma_i a(1-a|\mathcal{N}_i|)}{|\mathcal{N}_i|}(\sum_{j=1}^n (x_j(t) - x_i(t)))^2 \le \frac{\sigma_i}{2|\mathcal{N}_i|} q_i(t)$. In other words, the distributed triggering law (10) proposed in [193] is a special case of the static triggering law (8.7).*

The main purpose of using event-triggered control is to reduce the overall need of actuation updates and communication between agents, so it is essential to exclude Zeno behavior. However, as stated in [193], Zeno behavior may not be excluded under (8.7). In order to explicitly exclude Zeno behavior, in the following we propose a dynamic triggering law to determine the triggering times.

Inspired by [348], we propose the following internal dynamic variable η_i to agent i:

$$\dot{\eta}_i(t) = -\beta_i \eta_i(t) - \delta_i\Big(L_{ii} e_i^2(t) - \frac{\sigma_i}{2} q_i(t)\Big), \ i \in [n], \tag{8.10}$$

where $\eta_i(0) > 0$, $\beta_i > 0$, $\delta_i \in [0, 1]$, and $\sigma_i \in [0, 1)$ are design parameters and can be arbitrarily chosen in the given intervals. These dynamic variables are correlated in the triggering law, as defined in our first main result.

Theorem 8.1. *Consider the multi-agent system (8.1)–(8.3). Suppose that the underlying graph $\mathcal{G}$ is undirected. Given $\theta_i > \frac{1-\delta_i}{\beta_i}$ and the first triggering time $t_1^i = 0$, agent i determines the triggering times $\{t_k^i\}_{k=2}^\infty$ by*

$$t_{k+1}^i = \min\Big\{t : \ \theta_i\Big(L_{ii} e_i^2(t) - \frac{\sigma_i}{2} q_i(t)\Big) \ge \eta_i(t), \ t \ge t_k^i\Big\}, \tag{8.11}$$

where $q_i(t)$ and $\eta_i(t)$ are defined in (8.6) and (8.10), respectively. Then,

(i) *average consensus is achieved exponentially if and only if $\mathcal{G}$ is connected;*

(ii) *there is no Zeno behavior.*

Proof. **(i)** The necessity is straightforward and we only prove sufficiency here. From the way to determine the triggering times by (8.11), we have

$$\theta_i\Big(L_{ii} e_i^2(t) - \frac{\sigma_i}{2} q_i(t)\Big) \le \eta_i(t), \ \forall t \ge 0. \tag{8.12}$$

From (8.10) and (8.12), we have

$$\dot{\eta}_i(t) \ge -\beta_i \eta_i(t) - \frac{\delta_i}{\theta_i} \eta_i(t), \ \forall t \ge 0.$$

Thus

$$\eta_i(t) \geq \eta_i(0)e^{-(\beta_i+\frac{\delta_i}{\theta_i})t} > 0,\ \forall t \geq 0. \tag{8.13}$$

Consider a Lyapunov candidate

$$W(x(t),\eta(t)) = V(x(t)) + \sum_{i=1}^{n}\eta_i(t),$$

where $\eta(t) = \mathrm{col}(\eta_1(t),\ldots,\eta_n(t))$. Then the derivative of $W(x(t),\eta(t))$ along the trajectories of the multi-agent system (8.1)–(8.3) and system (8.10) satisfies

$$\begin{aligned}
\dot{W}(x(t),\eta(t)) &= \dot{V}(x(t)) + \sum_{i=1}^{n}\dot{\eta}_i(t)\\
&\leq -\sum_{i=1}^{n}\frac{1}{2}q_i(t) + \sum_{i=1}^{n}L_{ii}e_i^2(t) - \sum_{i=1}^{n}\beta_i\eta_i(t) + \sum_{i=1}^{n}\delta_i\Big(\frac{\sigma_i}{2}q_i(t) - L_{ii}e_i^2(t)\Big)\\
&= -\sum_{i=1}^{n}\frac{1}{2}(1-\sigma_i)q_i(t) - \sum_{i=1}^{n}\beta_i\eta_i(t) + \sum_{i=1}^{n}(\delta_i - 1)\Big(\frac{\sigma_i}{2}q_i(t) - L_{ii}e_i^2(t)\Big)\\
&\leq -\sum_{i=1}^{n}\frac{1}{2}(1-\sigma_i)q_i(t) - \sum_{i=1}^{n}\beta_i\eta_i(t) + \sum_{i=1}^{n}\frac{1-\delta_i}{\theta_i}\eta_i(t)\\
&= -\sum_{i=1}^{n}\frac{1}{2}(1-\sigma_i)q_i(t) - \sum_{i=1}^{n}\Big(\beta_i - \frac{1-\delta_i}{\theta_i}\Big)\eta_i(t)\\
&\leq -(1-\sigma_{\max})\sum_{i=1}^{n}\frac{1}{2}q_i(t) - k_d\sum_{i=1}^{n}\eta_i(t)\\
&\leq -(1-\sigma_{\max})\rho_2(L)V(x(t)) - k_d\sum_{i=1}^{n}\eta_i(t)\\
&\leq -k_W W(x(t),\eta(t)),
\end{aligned}$$

where

$$k_d = \min_i\{\beta_i - \frac{1-\delta_i}{\theta_i}\} > 0,\ k_W = \min\{(1-\sigma_{\max})\rho_2(L),\ k_d\} > 0.$$

Then

$$V(x(t)) \leq W(x(t),\eta(t)) \leq W(x(0),\eta(0))e^{-k_W t},\ \forall t \geq 0. \tag{8.14}$$

This implies that system (8.1)–(8.3) reaches average consensus exponentially.
(ii) Next, we prove that there is no Zeno behavior by contradiction. Suppose there exists Zeno behavior. Then there exists an agent i, such that $\lim_{k\to+\infty} t_k^i = T_0$, where T_0 is a positive constant.

Whether $\mathcal{G}$ is connected or not, from the proof in (i) we know that all the agents in the same connected component reach consensus and there is a result similar to (8.14). Thus, we know that there exists a positive constant $M_0 > 0$ such that $|x_i(t)| \leq M_0$ for all $t \geq 0$ and $i = 1, \ldots, n$. Then, we have

$$|u_i(t)| \leq 2M_0 L_{ii}, \ \forall t \geq 0.$$

Let $\varepsilon_0 = \frac{\sqrt{\eta_i(0)}}{4\sqrt{\theta_i L_{ii}^3} M_0} e^{-\frac{1}{2}(\beta_i + \frac{\delta_i}{\theta_i})T_0} > 0$. Then from the property of limits, there exists a positive integer $N(\varepsilon_0)$ such that

$$t_k^i \in [T_0 - \varepsilon_0, T_0], \ \forall k \geq N(\varepsilon_0). \tag{8.15}$$

Noting $q_i(t) \geq 0$ and (8.13), we can conclude that one necessary condition to guarantee that the inequality in (8.11) holds is

$$|\hat{x}_i(t) - x_i(t)| \geq \sqrt{\frac{\eta_i(0)}{\theta_i L_{ii}}} e^{-\frac{1}{2}(\beta_i + \frac{\delta_i}{\theta_i})t}.$$

Again noting $|\dot{x}_i(t)| = |u_i(t)| \leq 2M_0 L_{ii}$ and $|\hat{x}_i(t_k^i) - x_i(t_k^i)| = 0$ for any triggering time t_k^i, we can conclude that one necessary condition to guarantee that the above inequality holds is

$$(t - t_k^i) 2M_0 L_{ii} \geq \frac{\sqrt{\eta_i(0)}}{\sqrt{\theta_i L_{ii}}} e^{-\frac{1}{2}(\beta_i + \frac{\delta_i}{\theta_i})t}. \tag{8.16}$$

Now suppose that the $N(\varepsilon_0)$-th triggering time of agent i, $t_{N(\varepsilon_0)}^i$, has been determined. Let $t_{N(\varepsilon_0)+1}^i$ and $\tilde{t}_{N(\varepsilon_0)+1}^i$ denote the next triggering time determined by (8.11) and (8.16), respectively. Then

$$\begin{aligned} t_{N(\varepsilon_0)+1}^i - t_{N(\varepsilon_0)}^i &\geq \tilde{t}_{N(\varepsilon_0)+1}^i - t_{N(\varepsilon_0)}^i = \frac{\sqrt{\eta_i(0)}}{2\sqrt{\theta_i L_{ii}^3} M_0} e^{-\frac{1}{2}(\beta_i + \frac{\delta_i}{\theta_i})\tilde{t}_{N(\varepsilon_0)+1}^i} \\ &\geq \frac{\sqrt{\eta_i(0)}}{2\sqrt{\theta_i L_{ii}^3} M_0} e^{-\frac{1}{2}(\beta_i + \frac{\delta_i}{\theta_i})t_{N(\varepsilon_0)+1}^i} \geq \frac{\sqrt{\eta_i(0)}}{2\sqrt{\theta_i L_{ii}^3} M_0} e^{-\frac{1}{2}(\beta_i + \frac{\delta_i}{\theta_i})T_0} = 2\varepsilon_0, \end{aligned} \tag{8.17}$$

which contradicts to (8.15). Therefore, Zeno behavior is excluded. □

Remark 8.4. *Note that (8.11) is a dynamic triggering law since it involves the extra dynamic variables $\eta_i(t)$. Similar to the static triggering law (8.7), it is also distributed. The static triggering law (8.7) can be seen as a limit case of the dynamic triggering law (8.11) when θ_i grows large. Thus, from the analysis in Remark 8.3, we can conclude that the distributed triggering law (10) proposed in [193] is a special case of the dynamic triggering law (8.11).*

Remark 8.5. *If we choose $\delta_i = 0$ in (8.10) and $\sigma_i = 0$ in (8.11), then $\eta_i(t) = \eta_i(0)e^{-\beta_i t}$ and now the inequality in (8.11) is $|e_i(t)| \geq \sqrt{\eta_i(0)} e^{-\frac{\beta_i}{2}t} / \sqrt{\theta_i L_{ii}}$. The later is the triggering function (7) proposed in [195] with $c_0 = 0$, $c_1 = \sqrt{\eta_i(0)} / \sqrt{\theta_i L_{ii}}$, $\alpha = \beta_i/2$. However, we do not need the constraint $\alpha < \rho_2(L)$ which is necessary in [195].*

If we choose β_i large enough, then $k_W = (1-\sigma_{\max})\rho_2(L)$. Hence, in this case, from (8.9) and (8.14), we know that the trajectories of the multi-agent system (8.1) –(8.3) under static triggering law (8.7) and dynamic triggering law (8.11) have the same guaranteed decay rate given by (8.9).

Remark 8.6. *Intuitively, from (8.16), one can conclude that the larger $\eta_i(0)$ the larger the inter-event time. This is also consistent with the definition of ε_0. However, how those design parameters $\eta_i(0), \beta_i, \xi_i, \sigma_i, \theta_i$ affect the inter-event times and decay rate in theory is unclear. We leave this as a future research direction.*

8.3.2 Discontinuous approach

In the above static and dynamic triggering laws, continuous updating of the control input is avoided. However, in order to monitor the inequalities (8.7) and (8.11), each agent still needs to continuously monitor its neighbors's states, which means continuous broadcasting and receiving are still needed. In what follows, we will modify the above results to avoid these two requirements.

We estimate the upper bound of the derivative of $V(x(t))$ along the trajectories of the multi-agent system (8.1)–(8.3) in a different way. Similar to the derivation process to get (8.5), we have

$$
\begin{aligned}
\dot{V}(x(t)) &= \sum_{i=1}^{n} x_i(t) \sum_{j=1}^{n} -L_{ij}\hat{x}_j(t) = -\sum_{i=1}^{n}(\hat{x}_i(t) - e_i(t)) \sum_{j=1}^{n} L_{ij}\hat{x}_j(t) \\
&\overset{**}{=} -\sum_{i=1}^{n} \hat{q}_i(t) + \sum_{i=1}^{n}\sum_{j=1}^{n} e_i(t)L_{ij}\hat{x}_j(t) \\
&= -\sum_{i=1}^{n} \hat{q}_i(t) + \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} e_i(t)L_{ij}(\hat{x}_j(t) - \hat{x}_i(t)) \\
&\leq -\sum_{i=1}^{n} \hat{q}_i(t) - \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} L_{ij}e_i^2(t) - \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} L_{ij}\frac{1}{4}(\hat{x}_j(t) - \hat{x}_i(t))^2 \\
&= -\sum_{i=1}^{n} \hat{q}_i(t) + \sum_{i=1}^{n} L_{ii}e_i^2(t) - \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{4}L_{ij}(\hat{x}_j(t) - \hat{x}_i(t))^2 \\
&\overset{**}{=} -\sum_{i=1}^{n} \frac{1}{2}\hat{q}_i(t) + \sum_{i=1}^{n} L_{ii}e_i^2(t),
\end{aligned} \tag{8.18}
$$

where

$$
\hat{q}_i(t) = -\frac{1}{2}\sum_{j=1}^{n} L_{ij}(\hat{x}_j(t) - \hat{x}_i(t))^2 \geq 0, \tag{8.19}
$$

and the equalities denoted by $\stackrel{**}{=}$ hold due to

$$\sum_{i=1}^{n}\hat{q}_i(t) = -\sum_{i=1}^{n}\frac{1}{2}\sum_{j=1}^{n}L_{ij}(\hat{x}_j(t)-\hat{x}_i(t))^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\hat{x}_i(t)L_{ij}\hat{x}_j(t) = \hat{x}^\top(t)L\hat{x}(t),$$

and the inequality holds due to $ab \le a^2 + \frac{1}{4}b^2$.

Similar to [194] and [358], the following law can be used to determine the triggering times:

$$t_1^i = 0,\ t_{k+1}^i = \min\Big\{t:\ L_{ii}e_i^2(t) - \frac{\sigma_i}{2}\hat{q}_i(t) \ge 0,\ t \ge t_k^i\Big\},\ k \in \mathbb{N}_+, \tag{8.20}$$

where $\sigma_i \in (0,1)$ is a design parameter. From the way to determine the triggering times by (8.20), we have

$$L_{ii}e_i^2(t) \le \frac{\sigma_i}{2}\hat{q}_i(t),\ \forall t \ge 0. \tag{8.21}$$

Then, from (8.18) and (8.21), we have

$$\begin{aligned}\dot{V}(x(t)) &\le -\sum_{i=1}^{n}\frac{1}{2}\hat{q}_i(t) + \sum_{i=1}^{n}L_{ii}e_i^2(t) \le -\frac{1}{2}(1-\sigma_{\max})\sum_{i=1}^{n}\hat{q}_i(t)\\ &= -\frac{1}{2}(1-\sigma_{\max})\hat{x}^\top(t)L\hat{x}(t).\end{aligned}$$

Furthermore,

$$\begin{aligned}x^\top(t)Lx(t) &= (\hat{x}(t)+e(t))^\top L(\hat{x}(t)+e(t)) \le 2\hat{x}^\top(t)L\hat{x}(t) + 2e^\top(t)Le(t)\\ &\le 2\hat{x}^\top(t)L\hat{x}(t) + 2\|L\|\|e(t)\|^2 \le 2\hat{x}^\top(t)L\hat{x}(t) + \frac{\|L\|\sigma_{\max}}{\min_{i\in[n]}\{L_{ii}\}}\sum_{i=1}^{n}\hat{q}_i(t)\\ &= \Big(2 + \frac{\|L\|\sigma_{\max}}{\min_{i\in[n]}\{L_{ii}\}}\Big)\hat{x}^\top(t)L\hat{x}(t),\end{aligned} \tag{8.22}$$

where the first inequality holds since L is positive semi-definite as well as $2a^\top Lb \le a^\top La + b^\top Lb, \forall a,b \in \mathbb{R}^n$, and the second inequality holds due to $a^\top La \le \|L\|\|a\|^2, \forall a \in \mathbb{R}^n$, and the last inequality holds due to (8.21). We then obtain

$$\begin{aligned}\dot{V}(x(t)) &\le -\frac{(1-\sigma_{\max})\min_{i\in[n]}\{L_{ii}\}}{4\min_i L_{ii} + 2\|L\|\sigma_{\max}}x^\top(t)Lx(t)\\ &\le -\frac{(1-\sigma_{\max})\min_{i\in[n]}\{L_{ii}\}}{4\min_{i\in[n]}\{L_{ii}\} + 2\|L\|\sigma_{\max}}\rho_2(L)x^\top(t)K_nx(t)\\ &= -\frac{(1-\sigma_{\max})\min_{i\in[n]}\{L_{ii}\}}{2\min_{i\in[n]}\{L_{ii}\} + \|L\|\sigma_{\max}}\rho_2(L)V(x(t)).\end{aligned}$$

Hence,

$$V(x(t)) \le V(x(0))e^{-\frac{(1-\sigma_{\max})\min_{i\in[n]}\{L_{ii}\}}{2\min_{i\in[n]}\{L_{ii}\}+\|L\|\sigma_{\max}}\rho_2(L)t},\ \forall t \ge 0. \tag{8.23}$$

This implies that system (8.1)–(8.3) reaches average consensus exponentially if the underlying graph $\mathcal{G}$ is connected.

Remark 8.7. *Similar to the analysis in Remark 8.2, (8.20) is a static triggering law and it is also distributed. Moreover, similar to the analysis in Remark 8.3, we can conclude that the distributed triggering law (6) proposed in [194] is a special case of the static triggering law (8.20).*

In [357] it is argued that the distributed triggering law (6) in [194] "does not discard the possibility of an infinite number of events happening in a finite time period". Zeno behavior may also not be excluded under the static triggering law (8.20). In the following, in order to explicitly exclude Zeno behavior, we will replace the static triggering law (8.20) by the dynamic one.

Similar to (8.10), we propose an internal dynamic variable χ_i to agent i:

$$\dot{\chi}_i(t) = -\beta_i\chi_i(t) - \delta_i\Big(L_{ii}e_i^2(t) - \frac{\sigma_i}{2}\hat{q}_i(t)\Big),\ i \in [n], \tag{8.24}$$

where $\chi_i(0) > 0$, $\beta_i > 0$, $\delta_i \in [0, 1]$, and $\sigma_i \in [0, 1)$ are design parameters and can be arbitrarily chosen in the given intervals. Our second main result is given in the following theorem.

Theorem 8.2. *Consider the multi-agent system (8.1)–(8.3). Suppose that the underlying graph $\mathcal{G}$ is undirected. Given $\theta_i > \frac{1-\delta_i}{\beta_i}$ and the first triggering time $t_1^i = 0$, agent i determines the triggering times $\{t_k^i\}_{k=2}^{\infty}$ by*

$$t_{k+1}^i = \min\Big\{t :\ \theta_i\Big(L_{ii}e_i^2(t) - \frac{\sigma_i}{2}\hat{q}_i(t)\Big) \geq \chi_i(t),\ t \geq t_k^i\Big\}, \tag{8.25}$$

where $\hat{q}_i(t)$ and $\chi_i(t)$ are defined in (8.19) and (8.24), respectively. Then,

(i) *average consensus is achieved exponentially if and only if $\mathcal{G}$ is connected;*

(ii) *there is no Zeno behavior.*

Proof. **(i)** The necessity is straightforward and we only prove sufficiency here. Similar to (8.13), we have

$$\chi_i(t) \geq \chi_i(0)e^{-(\beta_i+\frac{\delta_i}{\theta_i})t} > 0,\ \forall t \geq 0.$$

Consider a Lyapunov candidate

$$F(x(t), \chi(t)) = V(x(t)) + \sum_{i=1}^{n}\chi_i(t),$$

where $\chi(t) = \mathrm{col}(\chi_1(t), \ldots, \chi_n(t))$. Then the derivative of $F(x(t), \chi(t))$ along the trajectories of the multi-agent system (8.1)–(8.3) and system (8.24) satisfies

$$\dot{F}(x(t), \chi(t)) = \dot{V}(x(t)) + \sum_{i=1}^{n}\dot{\chi}_i(t)$$

$$
\begin{aligned}
&\leq -\sum_{i=1}^{n}\frac{1}{2}\hat{q}_i(t) + \sum_{i=1}^{n}L_{ii}e_i^2(t) - \sum_{i=1}^{n}\beta_i\chi_i(t) + \sum_{i=1}^{n}\delta_i\Big(\frac{\sigma_i}{2}\hat{q}_i(t) - L_{ii}e_i^2(t)\Big)\\
&= -\sum_{i=1}^{n}\frac{1}{2}(1-\sigma_i)\hat{q}_i(t) - \sum_{i=1}^{n}\beta_i\chi_i(t) + \sum_{i=1}^{n}(\delta_i - 1)\Big(\frac{\sigma_i}{2}\hat{q}_i(t) - L_{ii}e_i^2(t)\Big)\\
&\leq -\sum_{i=1}^{n}\frac{1}{2}(1-\sigma_i)\hat{q}_i(t) - \sum_{i=1}^{n}\beta_i\chi_i(t) + \sum_{i=1}^{n}\frac{1-\delta_i}{\theta_i}\chi_i(t)\\
&= -\sum_{i=1}^{n}\frac{1}{2}(1-\sigma_i)\hat{q}_i(t) - \sum_{i=1}^{n}\Big(\beta_i - \frac{1-\delta_i}{\theta_i}\Big)\chi_i(t)\\
&\leq -(1-\sigma_{\max})\sum_{i=1}^{n}\frac{1}{2}\hat{q}_i(t) - k_d\sum_{i=1}^{n}\chi_i(t)\\
&= -\frac{1}{2}(1-\sigma_{\max})\hat{x}^\top(t)L\hat{x}(t) - k_d\sum_{i=1}^{n}\chi_i(t).
\end{aligned}
$$

Similar to the derivation process to get (8.22), we have

$$
\begin{aligned}
x^\top(t)Lx(t) &\leq 2\hat{x}^\top(t)L\hat{x}(t) + 2\|L\|\|e(t)\|^2\\
&\leq 2\hat{x}^\top(t)L\hat{x}(t) + \frac{\|L\|\sigma_{\max}}{\min_{i\in[n]}\{L_{ii}\}}\sum_{i=1}^{n}\hat{q}_i(t) + \frac{2\|L\|}{\min_{i\in[n]}\{\theta_i L_{ii}\}}\sum_{i=1}^{n}\chi_i(t)\\
&= \Big(2 + \frac{\|L\|\sigma_{\max}}{\min_{i\in[n]}\{L_{ii}\}}\Big)\hat{x}^\top(t)L\hat{x}(t) + \frac{2\|L\|}{\min_{i\in[n]}\{\theta_i L_{ii}\}}\sum_{i=1}^{n}\chi_i(t)\\
&\leq k_x\hat{x}^\top(t)L\hat{x}(t) + \frac{2\|L\|}{\min_{i\in[n]}\{\theta_i L_{ii}\}}\sum_{i=1}^{n}\chi_i(t),
\end{aligned}
$$

where

$$
k_x = \max\Big\{2 + \frac{\|L\|\sigma_{\max}}{\min_{i\in[n]}\{L_{ii}\}},\ \frac{2(1-\sigma_{\max})\|L\|}{k_d\min_{i\in[n]}\{\theta_i L_{ii}\}}\Big\}.
$$

Then,

$$
-\frac{1}{2}(1-\sigma_{\max})\hat{x}^\top(t)L\hat{x}(t) \leq -\frac{1}{2k_x}(1-\sigma_{\max})x^\top(t)Lx(t) + \frac{k_d}{2}\sum_{i=1}^{n}\chi_i(t).
$$

Thus,

$$
\begin{aligned}
\dot{F}(x(t),\chi(t)) &\leq -\frac{1}{2k_x}(1-\sigma_{\max})x^\top(t)Lx(t) - \frac{k_d}{2}\sum_{i=1}^{n}\chi_i(t)\\
&\leq -\frac{\rho_2(L)}{2k_x}(1-\sigma_{\max})x^\top(t)K_nx(t) - \frac{k_d}{2}\sum_{i=1}^{n}\chi_i(t)
\end{aligned}
$$

$$
\begin{aligned}
&= -\frac{\rho_2(L)}{k_x}(1 - \sigma_{\max})V(t) - \frac{k_d}{2}\sum_{i=1}^{n}\chi_i(t) \\
&\leq k_F F(x(t), \chi(t)),
\end{aligned}
$$

where

$$
k_F = \min\Big\{\frac{\rho_2(L)}{k_x}(1 - \sigma_{\max}),\ \frac{k_d}{2}\Big\}.
$$

Hence,

$$
V(x(t)) < F(x(t), \chi(t)) \leq F(x(0), \chi(0))e^{-k_F t},\ \forall t \geq 0. \tag{8.26}
$$

This implies that system (8.1)–(8.3) reaches average consensus exponentially.
(ii) The way to exclude Zeno behavior is the same as the proof of Theorem 8.1. □

Remark 8.8. *The triggering law (8.25) is dynamic and it is also distributed. One can easily check that every agent does not need to continuously access its neighbors' states when implementing the static triggering law (8.20) and dynamic triggering law (8.25). The static triggering law (8.20) can be seen as a limit case of the dynamic triggering law (8.25) when θ_i grows large. Thus, from the analysis in Remark 8.7, we can conclude that the distributed triggering law (6) proposed in [194] is a special case of the dynamic triggering law (8.25).*

If we choose β_i large enough, then $k_F = \frac{(1-\sigma_{\max})\min_{i\in[n]}\{L_{ii}\}}{2\min_{i\in[n]}\{L_{ii}\}+\|L\|\sigma_{\max}}\rho_2(L)$. Hence, in this case, from (8.23) and (8.26), we know that the trajectories of the multi-agent system (8.1)–(8.3) under static triggering law (8.20) and dynamic triggering law (8.25) have the same guaranteed decay rate given by (8.23).

Remark 8.9. *In [358], the authors propose three distributed triggering laws for multi-agent systems with event-triggered control and directed topologies. With some modifications, similar to this chapter, the three distributed triggering laws in [358] can be extended to dynamic triggering laws as the one in Theorems 8.1 and 8.2. In other words, the results in Theorems 8.1 and 8.2 can be extended to the case that the underlying graph is directed and has a directed spanning tree. Moreover, the results in Theorems 8.1 and 8.2 also can most likely be extended to general linear and even nonlinear multi-agent systems. However, in the general linear case, the triggering laws are not distributed anymore since global information, such as the eigenvalues of the Laplacian matrix, is needed. Actually, to the best of our knowledge, in all the existing studies that considered event-triggered control for general linear multi-agent systems, the use of the eigenvalues of the Laplacian matrix cannot be avoided. And for the nonlinear case, some standard continuity assumptions, such as upper and lower Lipschtiz continuity assumptions, for the nonlinear dynamics are normally required.*

8.4 Distributed self-triggered control algorithm

When applying the dynamic triggering law (8.25) in Theorem 8.2, although each agent avoids to continuously monitor its neighbors' states, agent i still needs to continuously sense its own state since it has to continuously monitor the triggering law (8.25) and continuously listen to $x_j(t_k^j)$, $k \in \mathbb{N}_+$, $j \in \mathcal{N}_i$, since it does not know the triggering times of its neighbors, t_k^j, $k \in \mathbb{N}_+$, $j \in \mathcal{N}_i$, in advance. The way to avoid continuous sensing is straightforward since the control input of each agent is piecewise constant and the state of each agent can be predicted by simple calculation as (8.27) in the following. The challenge is to avoid continuous listening. If every agent $i \in [n]$, at its current triggering time t_k^i, can predict (determine) its next triggering time t_{k+1}^i and broadcast it to its neighbors, then at time t_k^i agent i knows agent j's latest triggering time $t^j_{k_j(t_k^i)}$ which is before t_k^i and its next triggering time $t^j_{k_j(t_k^i)+1}$ which is after t_k^i, for $j \in \mathcal{N}_i$. In this case, agent i only needs listen to and receive information at $\{t_k^j\}_{k=1}^{\infty}$, $j \in \mathcal{N}_i$ since it knows these time instants in advance. In this case, each agent only needs to to sense its state information and broadcast its triggering information at its own triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. Inspired by this, in the following we will propose a self-triggered algorithm such that at time t_k^i each agent i could determine t_{k+1}^i in advance. The idea is explained below.

Denote $u_{ij}(t) = x_j(t^j_{k_j(t)}) - x_i(t^i_{k_i(t)})$, then we have

$$\dot{x}_i(t) = u_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t^j_{k_j(t)}) = -\sum_{j=1}^{n} L_{ij} u_{ij}(t).$$

Thus,

$$x_i(t) = x_i(t_k^i) + \int_{t_k^i}^{t} u_i(s)ds = x_i(t_k^i) - \int_{t_k^i}^{t} \sum_{j=1}^{n} L_{ij} u_{ij}(s)ds, \ t \in [t_k^i, t_{k+1}^i). \tag{8.27}$$

Then, for $t \in [t_k^i, t_{k+1}^i)$, we have

$$|e_i(t)| = |x_i(t_k^i) - x_i(t)| = \left| \sum_{j=1}^{n} \int_{t_k^i}^{t} L_{ij} u_{ij}(s)ds \right|. \tag{8.28}$$

Here we need to highlight that $u_{ij}(t)$ may not be a constant for all $t \in [t_k^i, t_{k+1}^i)$ since $x_j(t^j_{k_j(t)})$ may not be a constant for all $t \in [t_k^i, t_{k+1}^i)$. So at time t_k^i, we do not know the value of $|e_i(t)|$ for all $t \in (t_k^i, t_{k+1}^i)$ in advance. However, if at time t_k^i we can estimate the upper bound of $u_{ij}(t)$, then we can also estimate the upper bound of $|e_i(t)|$. In this case, we can estimate t_{k+1}^i at time t_k^i.

In order to estimate the upper bound of $u_{ij}(t)$, we first need to simplify the dynamic triggering laws (8.11) and (8.25) in Theorems 8.1 and 8.2. As Remark 8.5 pointed out, if we choose $\delta_i = 0$ in (8.10) and $\sigma_i = 0$ in (8.11), then $\eta_i(t) = \eta_i(0)e^{-\beta_i t}$ and now the inequality in (8.11) is $|e_i(t)| \geq \alpha_i e^{-\frac{\beta_i}{2}t}$ with $\alpha_i = \sqrt{\eta_i(0)}/\sqrt{\theta_i L_{ii}} > 0$. Here, α_i can be chosen as any positive real numbers since $\eta_i(0)$ can be chosen as any positive real numbers. Then from Theorem 8.1, we derive the following corollary[1].

Corollary 8.1. *Consider the multi-agent system (8.1)–(8.3). Suppose that the underlying graph $\mathcal{G}$ is undirected. Given $\alpha > 0$, $\beta > 0$ and the first triggering time $t_1^i = 0$, agent i determines the triggering times $\{t_k^i\}_{k=2}^{\infty}$ by*

$$t_{k+1}^i = \min\Big\{t: \ |e_i(t)| \geq \frac{\alpha}{\sqrt{L_{ii}}}e^{-\frac{\beta}{2}t}, \ t \geq t_k^i\Big\}. \tag{8.29}$$

Then,

(i) *average consensus is achieved exponentially if and only if $\mathcal{G}$ is connected;*

(ii) *there is no Zeno behavior.*

Remark 8.10. *The design parameters α and β can be distributively chosen for each agent in the above corollary, but their effects on inter-event times and decay rate are not clear in theory. The reason that we require every agent to choose the same design parameters here is that it is convenient to design the self-triggered algorithm in the following.*

Next, let us estimate the upper bound of $|x_i(t) - x_j(t)|$ which will be used later. From the way to determine the triggering times in (8.29), we have

$$|e_i(t)| \leq \frac{\alpha}{\sqrt{L_{ii}}}e^{-\frac{\beta}{2}t}, \ \forall t \geq 0. \tag{8.30}$$

From (8.5) and (8.30), we have

$$\begin{aligned}\dot{V}(x(t)) &\leq -\sum_{i=1}^{n}\frac{1}{2}q_i(t) + \sum_{i=1}^{n}L_{ii}e_i^2(t) \leq -\frac{1}{2}x^\top(t)Lx(t) + \sum_{i=1}^{n}\alpha^2 e^{-\beta t}\\ &\leq -\frac{1}{2}\rho_2(L)x^\top(t)K_n x(t) + n\alpha^2 e^{-\beta t} = -\rho_2(L)V(x(t)) + n\alpha^2 e^{-\beta t}.\end{aligned}$$

Then,

$$\frac{dV(t)e^{\rho_2(L)t}}{dt} \leq n\alpha^2 e^{(\rho_2(L)-\beta)t}.$$

Thus,

$$V(x(t)) \leq \begin{cases} V(0)e^{-\rho_2(L)t} + \frac{n\alpha^2}{\rho_2(L)-\beta}(e^{-\beta t} - e^{-\rho_2(L)t}), & \text{if } \rho_2(L) \neq \beta,\\ V(0)e^{-\rho_2(L)t} + n\alpha^2 t e^{-\rho_2(L)t}, & \text{if } \rho_2(L) = \beta.\end{cases}$$

[1] If we choose $\delta_i = 0$ in (8.24) and $\sigma_i = 0$ in (8.25), then Corollary 8.1 is also a special case of Theorem 8.2.

From the fact that for any given $\varepsilon > 0$, $e^{\varepsilon t} \geq 1 + \varepsilon t$ holds, we have

$$V(x(t)) \leq k_1 e^{-\rho_2(L)t} + k_2 e^{-k_3 t}, \ \forall t \geq 0,$$

where

$$k_1 = \begin{cases} V(x(0)) - \frac{n\alpha^2}{\rho_2(L)-\beta}, & \text{if } \rho_2(L) \neq \beta, \\ V(x(0)) - \frac{n\alpha^2}{\varepsilon}, & \text{if } \rho_2(L) = \beta, \end{cases}$$

$$k_2 = \begin{cases} \frac{n\alpha^2}{\rho_2(L)-\beta}, & \text{if } \rho_2(L) \neq \beta, \\ \frac{n\alpha^2}{\varepsilon}, & \text{if } \rho_2(L) = \beta, \end{cases}$$

$$k_3 = \begin{cases} \beta, & \text{if } \rho_2(L) \neq \beta, \\ \beta - \varepsilon, & \text{if } \rho_2(L) = \beta, \end{cases}$$

and $\varepsilon \in (0, \beta)$ is a design parameter. Then, from (8.4), we have

$$\sum_{i=1}^{n} |x_i(t) - \bar{x}(0)|^2 = 2V(x(t)) \leq 2(k_1 e^{-\rho_2(L)t} + k_2 e^{-k_3 t}), \ \forall t \geq 0.$$

Thus,

$$\begin{aligned} |x_i(t) - x_j(t)| &\leq |x_i(t) - \bar{x}(0)| + |x_j(t) - \bar{x}(0)| \\ &\leq \sqrt{2(|x_i(t) - \bar{x}(0)|^2 + |x_j(t) - \bar{x}(0)|^2)} \leq f^x(t), \ \forall t \geq 0, \end{aligned}$$

where

$$f^x(t) = 2\sqrt{k_1 e^{-\rho_2(L)t} + k_2 e^{-k_3 t}}.$$

Now, let us estimate the upper bound of $u_{ij}(t)$ as follows

$$\begin{aligned} |u_{ij}(t)| &= |x_j(t^j_{k_j(t)}) - x_i(t^i_{k_i(t)})| = |x_j(t^j_{k_j(t)}) - x_j(t) + x_j(t) - x_i(t) + x_i(t) - x_i(t^i_{k_i(t)})| \\ &\leq |x_j(t^j_{k_j(t)}) - x_j(t)| + |x_j(t) - x_i(t)| + |x_i(t) - x_i(t^i_{k_i(t)})| \\ &\leq \Big(\frac{\alpha}{\sqrt{L_{ii}}} + \frac{\alpha}{\sqrt{L_{jj}}}\Big) e^{-\frac{\beta}{2}t} + f^x(t), \ \forall t \geq 0. \end{aligned} \tag{8.31}$$

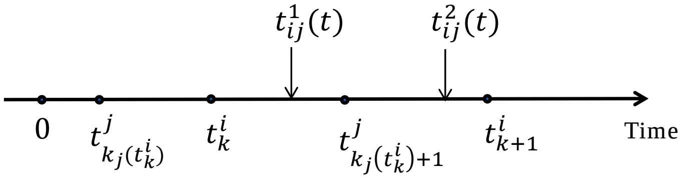
Finally, let us estimate the upper bound of $e_i(t)$. For $t \in [t^i_k, t^i_{k+1})$, denote

$$t^1_{ij}(t) = \min\Big\{t, \ t^j_{k_j(t^i_k)+1}\Big\}, \ t^2_{ij}(t) = \max\Big\{t, \ t^j_{k_j(t^i_k)+1}\Big\}. \tag{8.32}$$

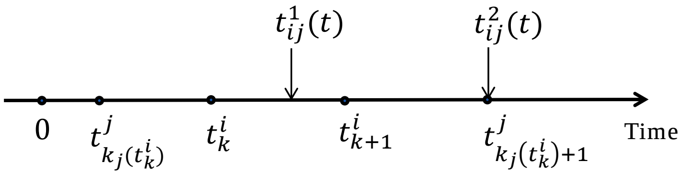
Figure 8.1 illustrates the relation of t^i_k, t^i_{k+1}, $t \in [t^i_k, t^i_{k+1})$, $t^j_{k_j(t^i_k)}$, $t^j_{k_j(t^i_k)+1}$, $t^1_{ij}(t)$ and $t^2_{ij}(t)$.

From the definition of $u_{ij}(t)$ and $t^1_{ij}(t)$, we know that $u_{ij}(t)$ is constant for all $t \in [t^i_k, t^1_{ij}(t)]$. And for $t > t^1_{ij}(t)$, $u_{ij}(t)$ can be bounded above by (8.31). Thus, from (8.28), for $t \in [t^i_k, t^i_{k+1})$ we have

$$|e_i(t)| = \Big|\sum_{j=1}^{n} \int_{t^i_k}^{t} L_{ij} u_{ij}(s) ds\Big| = \Big|\sum_{j=1}^{n} L_{ij} \Big(\int_{t^i_k}^{t^1_{ij}} u_{ij}(s) ds + \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}} u_{ij}(s) ds\Big)\Big| \leq g_i(t),$$



(a) The case of $t^j_{k_j(t^i_k)+1} < t^i_{k+1}$.



(b) The case of $t^j_{k_j(t^i_k)+1} \geq t^i_{k+1}$.

Figure 8.1: Illustration of the relation of t^i_k, t^i_{k+1}, $t \in [t^i_k, t^i_{k+1})$, $t^j_{k_j(t^i_k)}$, $t^j_{k_j(t^i_k)+1}$, $t^1_{ij}(t)$ and $t^2_{ij}(t)$.

where

$$g_i(t) = \Big|\sum_{j=1}^{n} L_{ij}(t^1_{ij} - t^i_k)u_{ij}(t^i_k)\Big| - \sum_{j=1, j\neq i}^{n} L_{ij} \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}} \Big(\Big(\frac{\alpha}{\sqrt{L_{ii}}} + \frac{\alpha}{\sqrt{L_{jj}}}\Big)e^{-\frac{\beta}{2}s} + f^x(s)\Big)ds.$$

Hence, one necessary condition to guarantee that the inequality in (8.29) holds, i.e.,

$$|e_i(t)| \geq \frac{\alpha}{\sqrt{L_{ii}}} e^{-\frac{\beta}{2}t}, \ \forall t \in [t^i_k, t^i_{k+1}),$$

is

$$g_i(t) \geq \frac{\alpha}{\sqrt{L_{ii}}} e^{-\frac{\beta}{2}t}, \ \forall t \in [t^i_k, t^i_{k+1}).$$

Noting that $\frac{\alpha}{\sqrt{L_{ii}}} e^{-\frac{\beta}{2}t}$ decreases with respect to t, $g_i(t)$ increases with respect to t during $[t^i_k, t^i_{k+1})$, and $g_i(t^i_k) = 0$, for given t^i_k, agent i can estimate t^i_{k+1} by solving

$$g_i(t) = \frac{\alpha}{\sqrt{L_{ii}}} e^{-\frac{\beta}{2}t}, \ t \geq t^i_k. \tag{8.33}$$

In other words, if at time t^i_k agent i knows $t^j_{k_j(t^i_k)}$, $t^j_{k_j(t^i_k)+1}$, $x_j(t^j_{k_j(t^i_k)})$, L_{jj}, $\forall j \in \mathcal{N}_i$, then it can determine its next triggering time t^i_{k+1} by solving (8.33). The above implement idea is summarized in Algorithm 8.1.

The following theorem proves that consensus is achieved exponentially and there is no Zeno behavior when every agent performs Algorithm 8.1.

Algorithm 8.1 Distributed Self-Triggered Control Algorithm

1: Choose $\alpha > 0, \beta > 0$ and $\varepsilon \in (0, \beta)$;
2: Agent $i \in [n]$ sends L_{ii} to its neighbors;
3: Agent i initializes $t_1^i = 0$ and $k = 1$;
4: At time $s = t_k^i$, agent i senses its own state $x_i(t_k^i)$, and updates its control input $u_i(t_k^i)$ by (8.3), and determines t_{k+1}^i by (8.33)[1], and broadcasts its triggering information $\{t_{k+1}^i, x_i(t_k^i)\}$ to its neighbors;
5: At agent i's neighbors' triggering times which are between $[t_k^i, t_{k+1}^i]$, agent i listens to and receives triggering information from its neighbors[2], and updates its control input $u_i(\cdot)$ by (8.3);
6: Agent i resets $k = k + 1$, and goes back to Step 4.

Theorem 8.3. *Consider the multi-agent system (8.1)–(8.3). Suppose that the underlying graph $\mathcal{G}$ is undirected. If all agents perform Algorithm 8.1, then*

(i) *average consensus is achieved exponentially if and only if $\mathcal{G}$ is connected;*

(ii) *there is no Zeno behavior.*

Proof. The necessity is straightforward.

Under Algorithm 8.1, we have $|e_i(t)| \leq \frac{\alpha}{\sqrt{L_{ii}}} e^{-\frac{\beta}{2}t}$ for all $i \in [n]$ and $t \geq 0$. Then from Corollary 8.1, we know that consensus is achieved exponentially.

The method of the exclusion of Zeno behavior is similar to the corresponding proof of Theorem 8.1. $\square$

Remark 8.11. *Self-triggered control approaches has also been proposed in [193, 200, 358, 366–369]. However, one potential drawback of these studies and other studies using a similar approach is that continuous listening is still needed. One can verify that continuous sensing, broadcasting, listening, and receiving are avoided under Algorithm 8.1. Although these are also avoided in [351, 355, 357, 370] by combining event-triggered control with periodic sampling, periodic sensing and listening are still needed. Moreover, it is not clear how to show that the average inter-event time is strictly larger than the required sampling period in theory. In order to perform Algorithm 8.1, the global parameters $V(x(0))$, n, and $\rho_2(L)$ are needed to be known in advance, which may be a drawback.*

Table 8.1 summarizes the communication requirements for agent $i \in [n]$ if the dynamic triggering laws (8.11) and (8.25), and Algorithm 8.1 are performed.

[1] Agent i uses $t^j_{k_j(t_k^i)}$ to replace $t^j_{k_j(t_k^i)+1}$ to determine t_{k+1}^i by (8.33) when $t_k^i = t^j_{k_j(t_k^i)}$.

[2] In other words, agent i onlys listen to incoming information at its neighbors' triggering times. Thus continuous listening is avoided.

Table 8.1: Summary of the communication requirements for agent i when dynamic triggering laws (8.11) and (8.25), and Algorithm 8.1 are performed.

	Law (8.11)	Law (8.25)	Algorithm 8.1
Broadcasting time	All $t \geq 0$	$\{t_k^i\}_{k=1}^{\infty}$	$\{t_k^i\}_{k=1}^{\infty}$
Listening time	All $t \geq 0$	All $t \geq 0$	$\{t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$
Receiving time	All $t \geq 0$	$\{t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$	$\{t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$
Information broadcasted	$\{x_i(t), t \geq 0\}$	$\{x_i(t_k^i)\}_{k=1}^{\infty}$	$\{t_{k+1}^i, x_i(t_k^i)\}_{k=1}^{\infty}$
Zeno behavior	No	No	No

8.5 Simulations

In this section, a numerical example is given to demonstrate the presented results. Consider a connected undirected graph in Figure 2.2 (a). We choose an arbitrary initial state $x(0) = [6.2945, 8.1158, -7.4603, 8.2675]^\top$. Then the average initial state is $\bar{x}(0) = 3.8044$.

Figure 8.2 (a) shows the state evolutions of the multi-agent system (8.1)–(8.3) under the static triggering law (8.7) with $\sigma_i = 0.5$. Figure 8.2 (b) shows the corresponding triggering times for each agent.

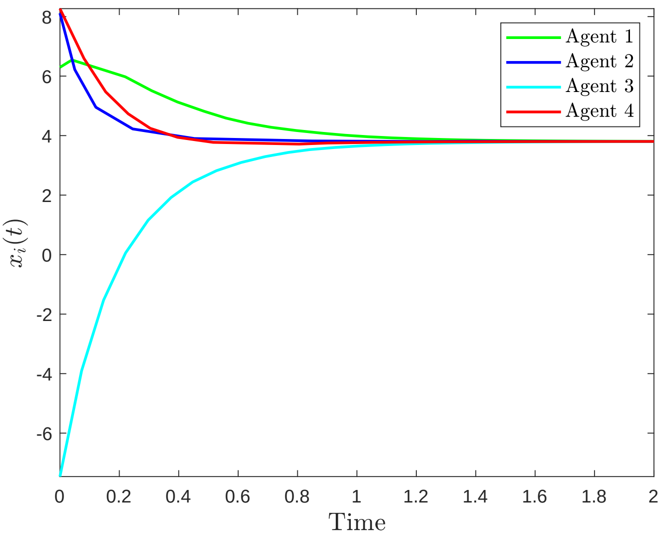
Figure 8.3 (a) shows the state evolutions of the multi-agent system (8.1)–(8.3) under the dynamic triggering law (8.11) with $\sigma_i = 0.5$, $\eta_i(0) = 10$, $\beta_i = 1$, $\delta_i = 1$ and $\theta_i = 1$. Figure 8.3 (b) shows the corresponding triggering times for each agent.

Figure 8.4 (a) shows the state evolutions of the multi-agent system (8.1)–(8.3) under the static triggering law (8.20) with $\sigma_i = 0.5$. Figure 8.4 (b) shows the corresponding triggering times for each agent.

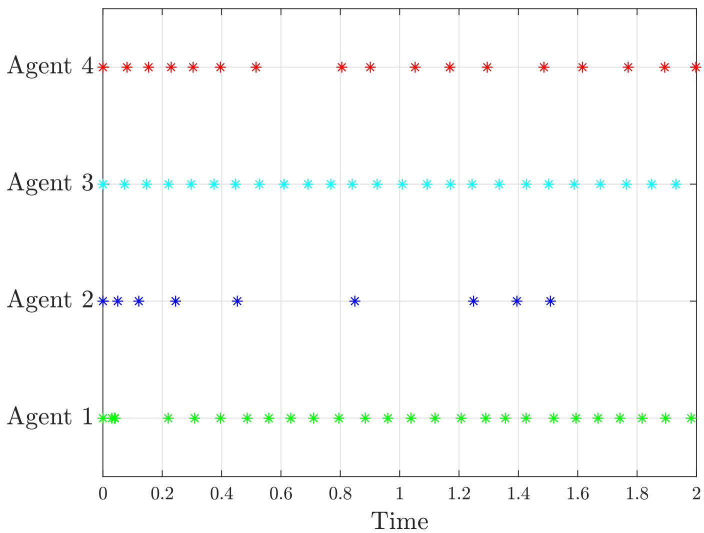
Figure 8.5 (a) shows the state evolutions of the multi-agent system (8.1)–(8.3) under the dynamic triggering law (8.25) with $\sigma_i = 0.5$, $\chi_i(0) = 10$, $\beta_i = 1$, $\delta_i = 1$ and $\theta_i = 1$. Figure 8.5 (b) shows the corresponding triggering times for each agent.

Figure 8.6 (a) shows the state evolutions of the multi-agent system (8.1)–(8.3) when each agent performs Algorithm 8.1 with $\alpha = 10, \beta = 1$ and $\varepsilon = \frac{\beta}{2}$. Figure 8.6 (b) shows the corresponding triggering times for each agent. And the smallest inter-event time is 0.009 in this simulation.

It can be seen that average consensus is achieved when performing the four triggering laws and Algorithm 8.1 proposed in this chapter. Moreover, as stated in Theorems 8.1–8.3, from the simulations we can also see that there is no Zeno behavior under the dynamic triggering laws (8.11) and (8.25) and Algorithm 8.1. It can also be seen that the average inter-event times under the dynamic triggering laws (8.11) and (8.25) are in general larger than these determined by the corresponding static triggering laws (8.7) and (8.20), respectively, and they are also larger than that determined by Algorithm 8.1. Although there is also no Zeno behavior under the static triggering laws (8.7) and (8.20) in the simulations, it is still not clear if this could be proved in theory.

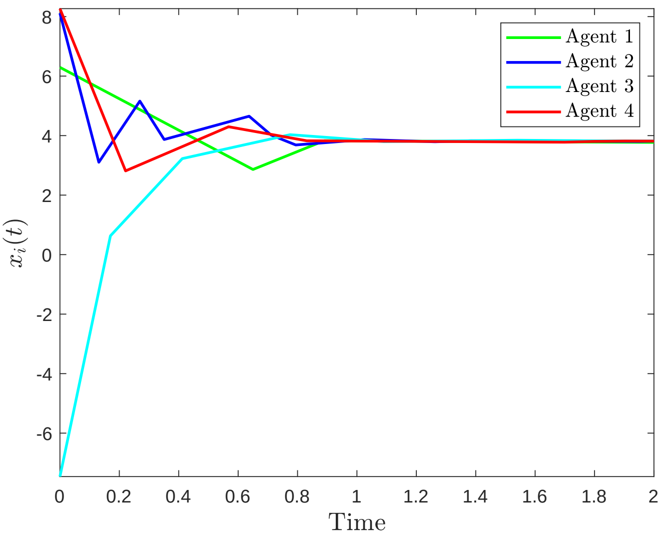


(a) The state evolutions of the multi-agent system (8.1)–(8.3) under the static triggering law (8.7).

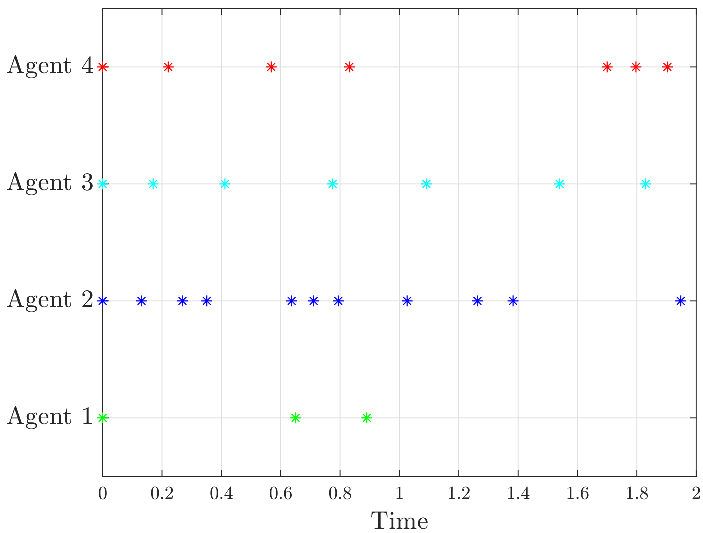


(b) The triggering times for each agent.

Figure 8.2: Performance of the distributed static event-triggered control algorithm with continuous broadcasting and receiving.

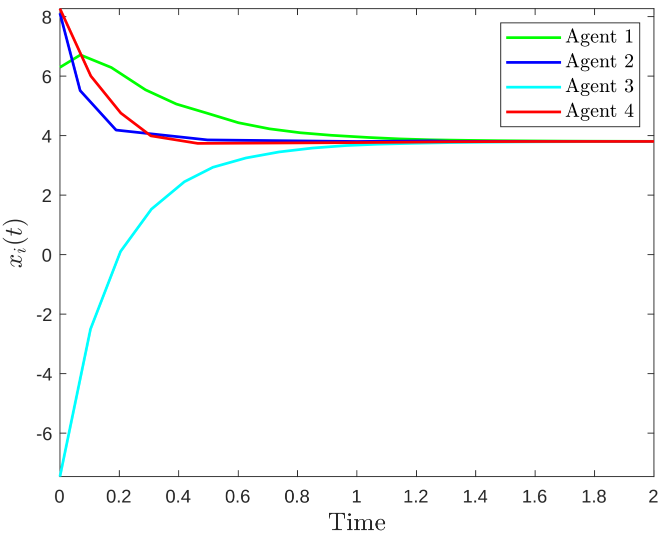


(a) The state evolutions of the multi-agent system (8.1)–(8.3) under the dynamic triggering law (8.11).

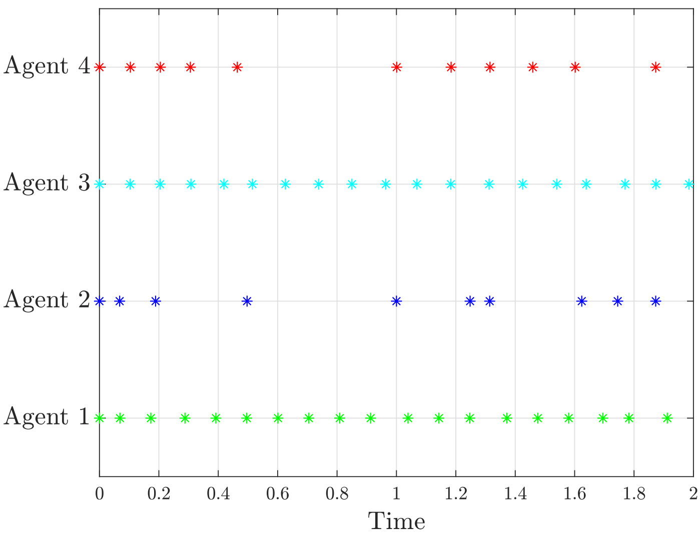


(b) The triggering times for each agent.

Figure 8.3: Performance of the distributed dynamic event-triggered control algorithm with continuous broadcasting and receiving.

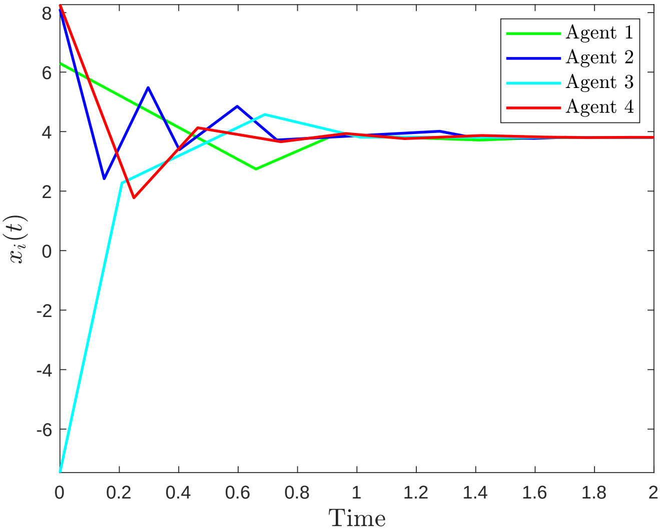


(a) The state evolutions of the multi-agent system (8.1)–(8.3) under the static triggering law (8.20).

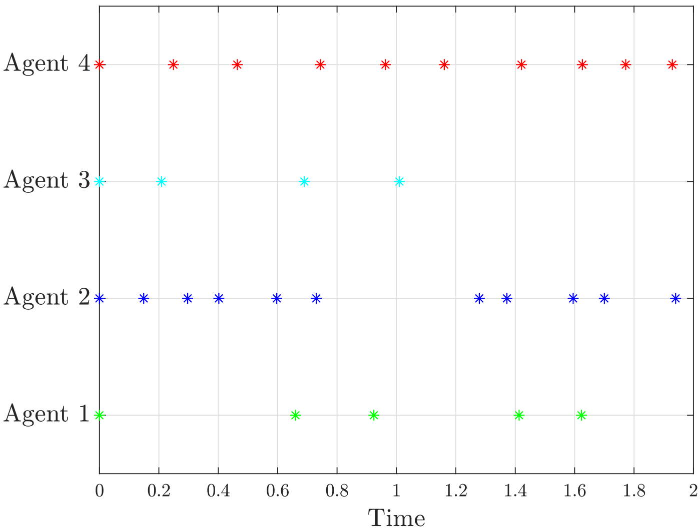


(b) The triggering times for each agent.

Figure 8.4: Performance of the distributed static event-triggered control algorithm with discontinuous broadcasting and receiving.

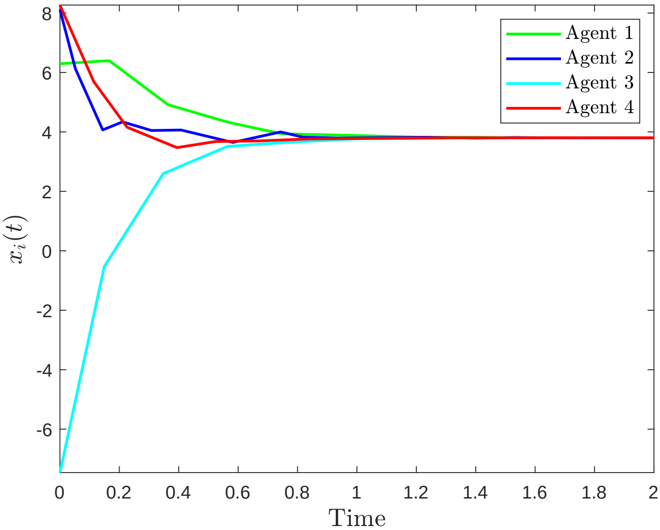


(a) The state evolutions of the multi-agent system (8.1)–(8.3) under the dynamic triggering law (8.25).

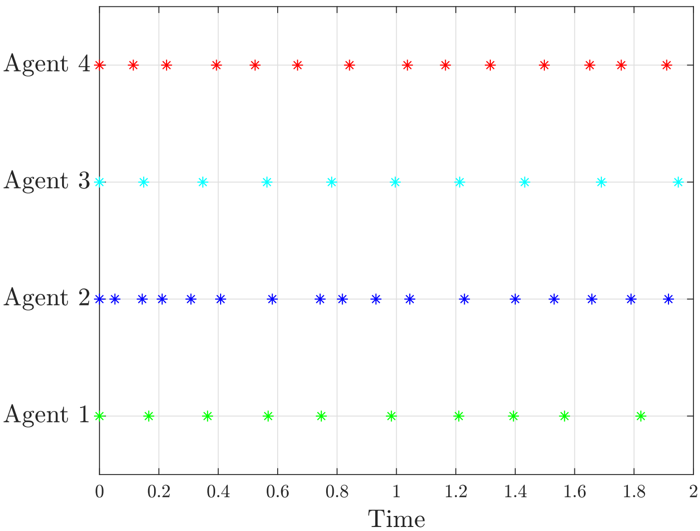


(b) The triggering times for each agent.

Figure 8.5: Performance of the distributed dynamic event-triggered control algorithm with discontinuous broadcasting and receiving.



(a) The state evolutions of the multi-agent system (8.1)–(8.3) when performing Algorithm 8.1.



(b) The triggering times for each agent.

Figure 8.6: Performance of the distributed self-triggered control algorithm.

8.6 Summary

In this chapter, we presented two dynamic triggering laws and one self-triggered algorithm for multi-agent systems with event-triggered control over undirected graphs. We showed that, some existing triggering laws are special cases of the proposed dynamic triggering laws and average consensus is achieved exponentially if and only if the communication graph is connected. In addition, Zeno behavior was excluded by proving that the triggering time sequence of each agent is divergent. Moreover, each agent only needs to sense and broadcast at its own triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. Thus continuous listening is avoided. Future research directions include considering the influence of parameters in the proposed dynamic triggering laws.

Chapter 9

Distributed event-triggered saturation control algorithms

In this chapter, the global consensus problem for first-order continuous-time multi-agent systems with input saturation is considered. We first show that the underlying directed graph having a directed spanning tree is a necessary and sufficient condition for global consensus; thus, this condition for consensus without input saturation extends to the case with saturation constraints. Moreover, in order to reduce the overall need of communication and system updates, we then propose an event-triggered consensus protocol and a triggering law, which do not require any a priori knowledge of global network parameters. Furthermore, in order to avoid continuous listening, we also propose a self-triggered algorithm. It is shown that Zeno behavior is excluded for these systems and that global consensus is achieved, again, if and only if the underlying directed graph has a directed spanning tree. We use a new Lyapunov function to show the sufficient condition and it inspires the triggering law. Numerical simulations are provided to illustrate the effectiveness of the theoretical results.

This chapter is organized as follows. Section 9.1 gives the background. Section 9.2 reviews the global consensus problem for the first-order continuous-time multi-agent systems with input saturation. Section 9.3 shows that the underlying digraph having a directed spanning tree is a necessary and sufficient condition for global consensus. Sections 9.4 and 9.5 use event- and self-triggered control to solve the same problem, respectively. Simulations are given in Section 9.6. The chapter is concluded in Section 9.7. Section 9.8 gives the proof of the main results.

9.1 Introduction

Physical systems are subject to physical constraints, such as input, output, communication, and sensor constraints. These constraints normally lead to nonlinearities in the closed-loop dynamics. Thus the behavior of each agent is affected and special attention to the constraints needs to be taken in order to understand their influence on the consensus convergence. Some recent investigations on this problem include, for example, [371] considered the global consensus problem for multi-agent systems with input saturation; [372] considered the leader-following consensus problem for multi-agent systems subject to input saturation; [373] studied global consensus for discrete-time multi-agent systems with input saturation constraint; [374, 375] investigated initial conditions for achieving consensus in the presence of output saturation; [371] shown that the distributed consensus protocol asymptotically leads to consensus, for multi-agent systems with input saturations and directed topologies; and [376] achieved the same result under a more general problem settings.

In almost all real applications, actuators have bounds. However, there are few event-triggered studies took saturation into consideration. In fact, even for a single-agent system with input saturation and event-triggered control, the stability problem is challenging. [377] addressed the influence of actuator saturation on event-triggered control. [378] studied a global stabilization of multiple integrator system using event-triggered bounded control. Consensus problem with input saturation and event-triggered control is challenging since the constraints lead to nonlinearities in the closed-loop dynamics. [379] proposed a distributed event-triggered control strategy to achieve consensus for multi-agent systems subject to input saturation through output feedback. Different from this chapter, the underlying graph they consider is undirected and they do not exclude Zeno behavior in their analysis. [380] investigated the event-triggered semi-global consensus problem for general linear multi-agent systems subject to input saturation. However, the underlying graph is assumed to be undirected and in order to determine the triggering times, each agent needs to continuously measure its neighbors' states, i.e., continuous communication is still needed.

In this chapter, we solve the global consensus problem for multi-agent systems with input saturation over digraphs. We have the following contributions.

(C9.1) We first show that the multi-agent systems achieve consensus if and only if the underlying digraph has a directed spanning tree. In other words, the existence of a directed spanning tree is a necessary and sufficient condition for consensus for both multi-agent systems with and without input saturation, despite that the saturation gives rise to a more complex nonlinear dynamic behavior.

(C9.2) We then consider event-triggered control and propose a distributed triggering law, which leads to global consensus under the same necessary and sufficient directed spanning tree condition. By distributed, we mean that the event-triggered control input together with the triggering law do not require any a priori knowledge of global network parameters. The triggering law is a special kind of dynamic triggering law, and is free from Zeno behavior, and is inspired by the Lyapunov function we use in the proof of the above consensus result. The Lyapunov function is different from the one in [371, 376]. As a result, continuous broadcasting, receiving, and updating are avoided.

(C9.3) Note that in the above distributed triggering law, continuous sensing is needed since each agent has to continuously monitor the triggering law and continuous listening is also needed since the triggering times are determined during runtime and not known in advance. Inspired by the idea of the self-triggered algorithm in Section 8.4, we also propose a self-triggered algorithm to avoid continuous sensing and listening.

9.2 Global consensus for multi-agent systems with input saturation

We consider a set of n agents modeled as single integrators with input saturation:

$$\dot{x}_i(t) = \mathrm{sat}_h(u_i(t)),\ i \in [n],\ t \geq 0, \tag{9.1}$$

where $x_i(t) \in \mathbb{R}^p$ and $u_i(t) \in \mathbb{R}^p$ are the state and the control input of agent i, respectively, $p > 0$ is the state dimension, and $\mathrm{sat}_h(\cdot)$ is the saturation function with h being a positive constant referred to as saturation level. For any $s = \mathrm{col}(s_1, \dots, s_p) \in \mathbb{R}^p$, the saturation function $\mathrm{sat}_h(s)$ is defined (with slight abuse of notation) as

$$\mathrm{sat}_h(s) = \mathrm{col}(\mathrm{sat}_h(s_1), \dots, \mathrm{sat}_h(s_p)), \tag{9.2}$$

where

$$\mathrm{sat}_h(s_i) = \begin{cases} h, & \text{if } s_i \geq h, \\ s_i, & \text{if } |s_i| < h, \\ -h, & \text{if } s_i \leq -h. \end{cases}$$

Remark 9.1. *For the ease of presentation, we focus on the case where all the agents have the same saturation level. The analysis can be readily extended to the case where the agents have different saturation levels.*

Definition 9.1 (Global consensus)**.** *We say global consensus for the multi-agent system (9.1) is achieved if*

$$\lim_{t\to\infty} \|x_i(t) - x_j(t)\| = 0,\ \forall i, j \in [n],\ \forall x_l(0) \in \mathbb{R}^p,\ l \in [n].$$

Our first goal in this chapter is to solve the following problem.

Problem 9.1. *Design control input for the saturated multi-agent system (9.1) such that global consensus is achieved.*

The following properties about the saturation function are useful for our analysis.

Lemma 9.1. *For any real constants a and b,*

$$\frac{1}{2}a^2 \geq \int_0^a sat_h(s)ds \geq \frac{1}{2}(sat_h(a))^2,\ (a-b)^2 \geq (sat_h(a) - sat_h(b))^2.$$

Lemma 9.2. *Suppose that L is the Laplacian matrix associated with a digraph $\mathcal{G}$ that has a directed spanning tree. For $x_1, \dots, x_n \in \mathbb{R}^p$, define $\pi_i = sat_h(-\sum_{j=1}^n L_{ij}x_j)$. Then $\pi_1 = \cdots = \pi_n$ if and only if $x_1 = \cdots = x_n$.*

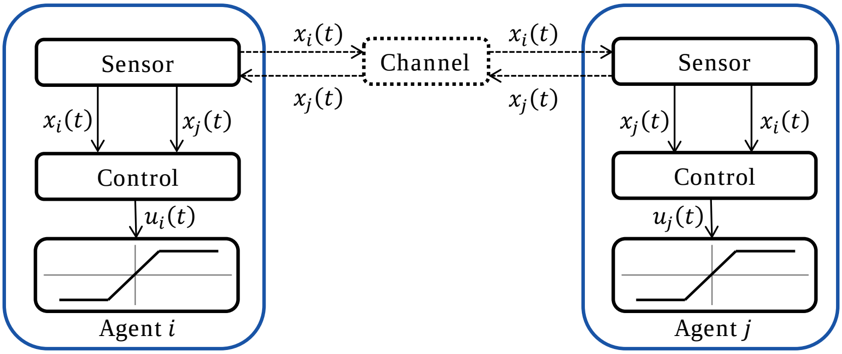


Figure 9.1: Illustration of how one agent communicates with another agent when the control input is saturated.

Proof. The sufficiency is straightforward. Let us show the necessity. Let $\mu_i = -\sum_{j=1}^{n} L_{ij}x_j$. From $\pi_1 = \cdots = \pi_n$, we know that for any $l = 1, \ldots, p$, $c_l(\mu_i) > 0$, $\forall i \in [n]$, or $c_l(\mu_i) < 0$, $\forall i \in [n]$, or $c_l(\mu_i) = 0$, $\forall i \in [n]$, where $c_l(\mu_i)$ is the l-th component of μ_i.

From Lemma 2 in [371], we know that neither $c_l(\mu_i) > 0$, $\forall i \in [n]$ nor $c_l(\mu_i) < 0$, $\forall i \in [n]$ holds. Thus $-\sum_{j=1}^{n} L_{ij}c_l(x_j) = c_l(\mu_i) = 0$, $\forall i \in [n]$. From Lemma 2.1, we know $\mathrm{rank}(L) = n-1$. Thus, we have $c_l(x_i) = c_l(x_j)$, $\forall i, j \in [n]$. Hence $x_1 = \cdots = x_n$. $\square$

9.3 Distributed continuous-time saturation control algorithm

In this section, we show that consensus is achieved by the classic distributed continuous-time consensus protocol even in the presence of input saturation if $\mathcal{G}$ has a directed spanning tree. The mathematical analysis is inspired by [381].

We consider the classic distributed continuous-time consensus protocol

$$u_i(t) = -\sum_{j=1}^{n} L_{ij}x_j(t), \tag{9.3}$$

where L_{ij} is the element of the Laplacian matrix L. In this chapter, we assume that the underlying graph $\mathcal{G}$ is directed. The communication in a multi-agent system described by (9.1) and (9.3) is illustrated in Figure 9.1. Note that the control signal is saturated before it is transmitted to the actuator..

In the following, we show a necessary and sufficient condition to consensus for system (9.1) and (9.3).

Theorem 9.1. *Consider the multi-agent system (9.1) and (9.3). Global consensus is achieved if and only if the digraph $\mathcal{G}$ has a directed spanning tree.*

Proof. The necessity in Theorem 9.1 is a direct result of Lemma 2.3. We illustrate the main idea of the proof of sufficiency here, while the detailed proof is given in Section 9.8.1. We first consider the case where $\mathcal{G}$ is strongly connected, i.e., $M = 1$ in (2.2), and show that consensus is achieved. We next consider the case where $\mathcal{G}$ has a directed spanning tree but it is not strongly connected, i.e., $M \geq 2$. From the first case ($M = 1$), it follows that all agents in SCC_M achieve consensus since SCC_M is either strongly connected or of dimension one. Then, we consider SCC_{M-1} and note that all agents in SCC_{M-1}, which is either strongly connected or of dimension one, achieve the same consensus value as those in SCC_M, since the agents in SCC_M and SCC_{M-1} are not influenced by $\mathrm{SCC}_1, \dots, \mathrm{SCC}_{M-2}$ and the consensus problem of this subsystem can be treated as a leader–follower problem where agents in SCC_M are leaders and agents in SCC_{M-1} are followers. Notice that $\mathrm{SCC}_1, \dots, \mathrm{SCC}_{M-2}$, are either strongly connected or of dimension one. By applying a similar analysis, consensus of $\mathrm{SCC}_m, \mathrm{SCC}_{m+1}, \dots, \mathrm{SCC}_M$ can be treated as a leader–follower consensus problem with agents in $\mathrm{SCC}_M, \mathrm{SCC}_{M-1}, \dots, \mathrm{SCC}_{m+1}$ being leaders and agents in SCC_m being followers. Therefore, the result follows. □

Remark 9.2. *The proof of Theorem 9.1 is based on the Lyapunov function*

$$V(x(t)) = \sum_{i=1}^{n} \xi_i \sum_{l=1}^{p} \int_0^{-\sum_{j=1}^{n} L_{ij} c_l(x_j(t))} sat_h(s) ds, \tag{9.4}$$

where $x(t) = \mathrm{col}(x_1(t), \dots, x_n(t))$ and $\xi = \mathrm{col}(\xi_1, \dots, \xi_n)$ is the vector defined in Lemma 2.1. It is different from the one used in [371]. In addition, our Lyapunov function facilitates the design of event-triggered control as shown in Section 9.4.

Remark 9.3. *When $h \to \infty$, i.e., the multi-agent system is free from saturation, Theorem 9.1 corresponds to the well-known result for the consensus problem of multi-agent systems without saturation [334, 335]. The main differences between the case with and without saturation are the convergence speed and the consensus value. For the saturated case, the convergence speed is slower and the consensus value is not fully determined by the Laplacian matrix L and the initial states of the agents. From the proof of Theorem 9.1, we know that the saturation is no longer active after a finite time $T_2 \geq 0$ which depends on the initial value of each agent, the saturation level, and the communication network. Thus after T_2 the convergence speed is exponential and the consensus value is determined by the state of each agent at T_2.*

9.4 Distributed event-triggered saturation control algorithm

To avoid continuous exchange of information among agents and update of actuators, we equip the consensus protocol (9.3) with an event-triggered communication scheme. The control signal is only updated when the triggering condition is satisfied. It results in the following multi-agent system with input saturation and event-triggered control input

$$\dot{x}_i(t) = \mathrm{sat}_h(\hat{u}_i(t)), \ i \in [n], \ t \geq 0, \tag{9.5}$$

$$\hat{u}_i(t) = -\sum_{j=1}^{n} L_{ij} x_j(t^j_{k_j(t)}). \tag{9.6}$$

Note that the consensus protocol (9.6) only updates at the triggering times and is constant between two consecutive triggering times. For simplicity, let $\hat{x}_i(t) = x_i(t^i_{k_i(t)})$, and $e_i(t) = \hat{x}_i(t) - x_i(t)$.

Our second goal in this chapter is to solve the following problem.

Problem 9.2. *Propose methods to determine the triggering times such that consensus is reached, while continuous exchange of information, continuous update of actuators, and Zeno behavior are avoided.*

This problem is solved by the following theorem.

Theorem 9.2. *Consider the multi-agent system (9.5)–(9.6). Given $\alpha_i > 0$, $\beta_i > 0$ and the first triggering time $t^i_1 = 0$, agent i determines the triggering times $\{t^i_k\}^{\infty}_{k=2}$ by*

$$t^i_{k+1} = \min\{t : \ \|e_i(t)\|^2 \geq \alpha_i e^{-\beta_i t}, \ t \geq t^i_k\}. \tag{9.7}$$

Then,

(i) *there is no Zeno behavior;*

(ii) *global consensus is achieved iff the underlying digraph $\mathcal{G}$ has a directed spanning tree.*

Proof. The proof is given in Section 9.8.2. □

Remark 9.4. *The event-triggered control input (9.6) together with the triggering law (9.7) is fully distributed. That is, each agent only requires its own state information and its neighbors' state information, without any a priori knowledge of any global parameter, such as the eigenvalue of the Laplacian matrix. This is different from [195, 366].*

9.5 Distributed self-triggered saturation control algorithm

When performing the event-triggered control input (9.6) together with the triggering law (9.7), each agent needs to broadcast its state to its neighbors at its triggering times, and to receive and to update its input at its neighbors' triggering times. Thus, continuous broadcasting, receiving, and updating are avoided. However, continuous sensing is needed since each agent has to continuously monitor the triggering law and continuous listening is also needed since the triggering times are determined during runtime and not known in advance. Inspired by the idea of self-triggered algorithm in Section 8.4, if each agent can predict its next triggering time and broadcast it to its neighbors at the current triggering time, then each agent only needs to sense and broadcast at its own triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. In the following we will propose a self-triggered algorithm such that at time t^i_k each agent i could estimate t^i_{k+1}. The idea is illustrated as follows.

From $\dot{x}_i(t) = \mathrm{sat}_h(\hat{u}_i(t))$, we have

$$x_i(t) = x_i(t_k^i) + \int_{t_k^i}^{t} \mathrm{sat}_h(\hat{u}_i(s))ds, \ t \in [t_k^i, t_{k+1}^i].$$

Thus for $t \in [t_k^i, t_{k+1}^i)$, we have

$$\|e_i(t)\| = \|x_i(t_k^i) - x_i(t)\| = \left\| \int_{t_k^i}^{t} \mathrm{sat}_h(\hat{u}_i(s))ds \right\|.$$

Here we need to highlight that $\mathrm{sat}_h(\hat{u}_i(t))$ may be not a constant vector for all $t \in [t_k^i, t_{k+1}^i)$ since $x_j(t^j_{k_j(t)})$ may be not a constant vector for all $t \in [t_k^i, t_{k+1}^i)$ which is due to that agent j may trigger at some time instants in this interval. So at time t_k^i we do not know what is the value of $\|e_i(t)\|$ for all $t \in (t_k^i, t_{k+1}^i)$. However, we know $\mathrm{sat}_h(\hat{u}_i(t))$ is a constant vector for $t \in [t_k^i, T_i^1(t_k^i))$, where

$$T_i^1(t_k^i) = \min\left\{t^i_{k_j(t_k^i)+1}, \ j \in \mathcal{N}_i\right\},$$

i.e., $T_i^1(t_k^i)$ is the first triggering time of all agent i's neighbors after time t_k^i. Although, at time t_k^i, agent i does not know $\mathrm{sat}_h(\hat{u}_i(t))$ for $t > T_i^1(t_k^i)$, it knows $|c_l(\mathrm{sat}_h(\hat{u}_i(t)))| \leq h$, $l = 1 \ldots, p$. Hence

$$\|e_i(t)\| = \left\| \int_{t_k^i}^{t} \mathrm{sat}_h(\hat{u}_i(s))ds \right\| = \left\| \int_{t_k^i}^{T_i^2(t)} \mathrm{sat}_h(\hat{u}_i(s))ds + \int_{T_i^2(t)}^{t} \mathrm{sat}_h(\hat{u}_i(s))ds \right\| \leq \varrho_i(t),$$

where

$$T_i^2(t) = \min\left\{T_i^1(t_k^i), \ t\right\}, \ \text{for } t \in [t_k^i, t_{k+1}^i),$$

and

$$\varrho_i(t) = (T_i^2(t) - t_k^i)\|\mathrm{sat}_h(\hat{u}_i(t_k^i))\| + (t - T_i^2(t))h\sqrt{p}, \ \text{for } t \in [t_k^i, t_{k+1}^i).$$

Then, a necessary condition to guarantee that the inequality in (9.7) holds, i.e.,

$$\|e_i(t)\|^2 \geq \alpha_i e^{-\beta_i t}, \ \forall t \in [t_k^i, t_{k+1}^i),$$

is

$$\varrho_i(t) \geq \sqrt{\alpha_i} e^{-\frac{\beta_i}{2}t}, \ \forall t \in [t_k^i, t_{k+1}^i).$$

Noting that $\sqrt{\alpha_i} e^{-\frac{\beta_i}{2}t}$ decreases with respect to t, $\varrho_i(t)$ increases with respect to t during $[t_k^i, t_{k+1}^i)$, and $\varrho_i(t_k^i) = 0$, for given t_k^i, agent i can estimate t_{k+1}^i by solving

$$\varrho_i(t) = \sqrt{\alpha_i} e^{-\frac{\beta_i}{2}t}, \ t \geq t_k^i. \tag{9.8}$$

Algorithm 9.1 Distributed Self-Triggered Saturation Control Algorithm

1: Agent $i \in [n]$ chooses $\alpha_i > 0$ and $\beta_i > 0$;
2: Agent i initializes $t_1^i = 0$ and $k = 1$;
3: At time $s = t_k^i$, agent i senses $x_i(t_k^i)$, and updates $u_i(t_k^i)$ by (9.6), and determines t_{k+1}^i by (9.8)[1], and broadcasts its triggering information $\{t_{k+1}^i,\ x_i(t_k^i)\}$ to its neighbors;
4: At agent i's neighbors' triggering times which are between $[t_k^i, t_{k+1}^i]$, agent i listens to and receives triggering information from its neighbors[2], and updates its $u_i(\cdot)$ by (9.6);
5: Agent i resets $k = k + 1$, and goes back to Step 3.

In other words, if at time t_k^i agent i knows $t^j_{k_j(t_k^i)}$, $t^j_{k_j(t_k^i)+1}$, $x_j(t^j_{k_j(t_k^i)})$, $\forall j \in \mathcal{N}_i$, then it can estimate its next triggering time t_{k+1}^i by solving (9.8). The above implement idea is summarized in Algorithm 9.1.

The following theorem shows that consensus is achieved and there is no Zeno behavior when every agent performs Algorithm 9.1.

Theorem 9.3. *Consider the multi-agent system (9.5)–(9.6). If all agents perform Algorithm 9.1, then,*

(i) *there is no Zeno behavior;*

(ii) *global consensus is achieved iff the underlying digraph $\mathcal{G}$ has a directed spanning tree.*

Proof. The method of the exclusion of Zeno behavior is similar to the way in the proof of Theorem 9.2. Under Algorithm 9.1, we have $\|e_i(t)\|^2 \leq \alpha_i e^{-\beta_i t}$ for all $i \in [n]$ and $t \geq 0$. Then from Theorem 9.2, we know that consensus is achieved. $\square$

Remark 9.5. *In order to perform Algorithm 9.1, no global parameters are used, i.e., Algorithm 9.1 is distributed.*

9.6 Simulations

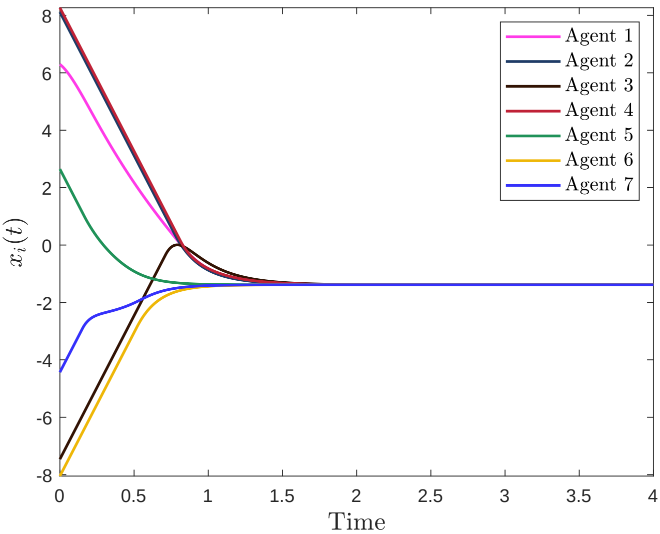
In this section, simulations are given to demonstrate the theoretical results. Consider again the digraph and the corresponding multi-agent system in Figure 2.1. Let the saturation level be $h = 10$. We choose an arbitrary initial state $x(0) = [6.2945, 8.1158, -7.4603, 8.2675, 2.6472, -8.0492, -4.4300]^\top$.

Figure 9.2 (a) shows the state evolutions of the multi-agent system (9.1)–(9.3) and Figure 9.2 (b) shows the saturated input of each agent. We see that consensus is achieved, even if some agents are saturated initially.

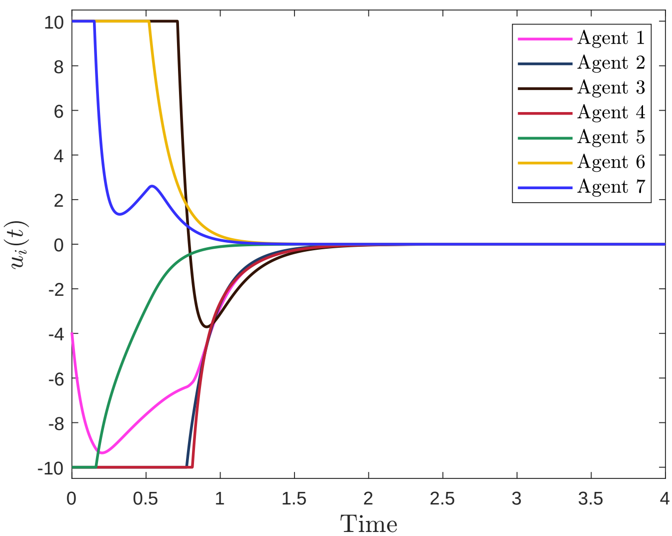
We next consider the case with event-triggered control input. Figure 9.3 (a) shows the state evolutions of the multi-agent system (9.5)–(9.6) under the triggering law (9.7) with

[1]Agent i uses $t^j_{k_j(t_k^i)}$ to replace $t^j_{k_j(t_k^i)+1}$ to determine t_{k+1}^i by (9.8) when $t_k^i = t^j_{k_j(t_k^i)}$.

[2]In other words, agent i only listen to incoming information at its neighbors' triggering times. Thus continuous listening is avoided.

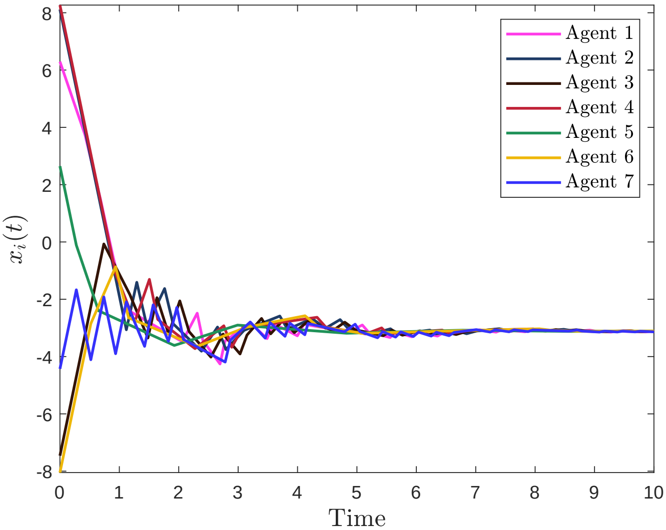


(a) The state evolutions of the multi-agent system (9.1)–(9.3).

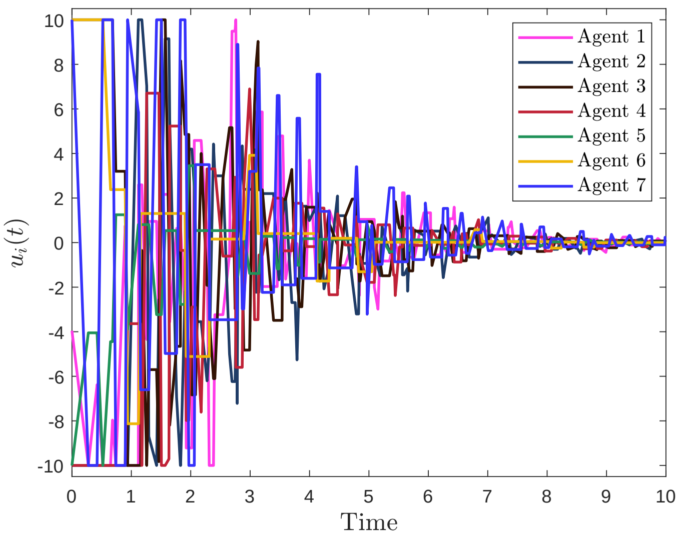


(b) The saturated input of each agent.

Figure 9.2: Performance of the distributed continuous-time saturation control algorithm.



(a) The state evolutions of the multi-agent system (9.5)–(9.6) under the triggering law (9.7).



(b) The saturated input of each agent.

Figure 9.3: Performance of the distributed event-triggered saturation control algorithm.

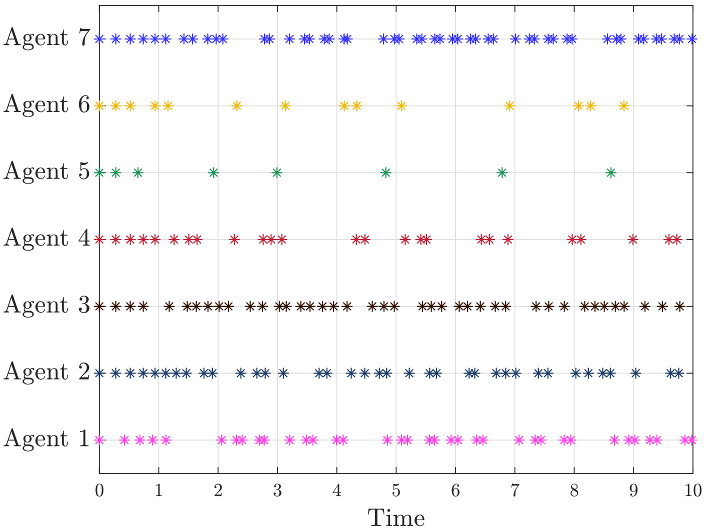


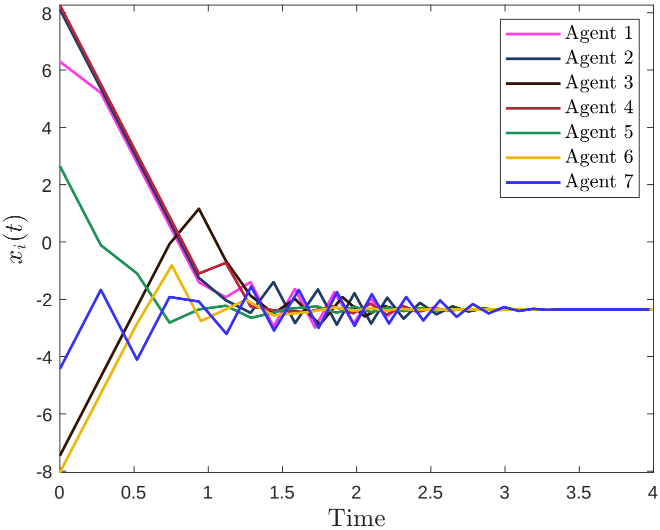
Figure 9.4: The triggering times for each agent determined by the distributed event-triggered saturation control algorithm.

$\alpha_i = 10$ and $\beta_i = 1$. Figure 9.3 (b) shows the saturated input of each agent. Figure 9.4 shows the corresponding triggering times for each agent. We see that consensus is achieved also in this case. Moreover, from Figure 9.4, we see that each agent only needs to broadcast its state to its neighbors at its triggering times. Thus continuous broadcasting and receiving are avoided.

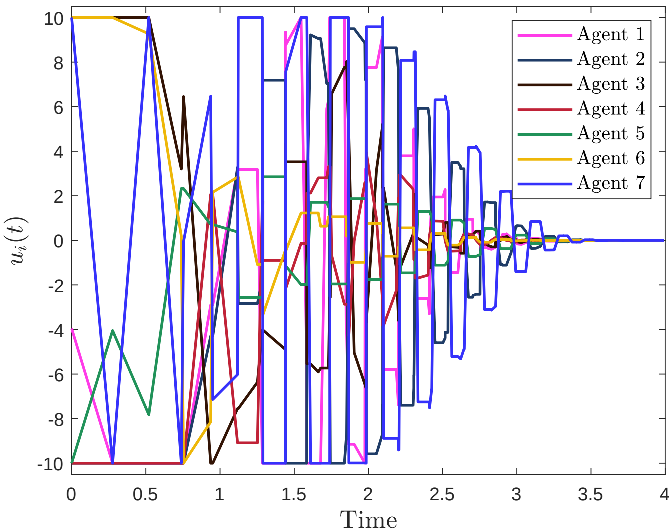
Figure 9.5 (a) shows the state evolutions of the multi-agent system (9.5)–(9.6) when each agent performs Algorithm 9.1 with $\alpha_i = 10$ and $\beta_i = 1$. Figure 9.5 (b) shows the saturated input of each agent. Figure 9.6 shows the corresponding triggering times for each agent. From Figure 9.5 (a) and (b), we see that consensus is achieved and $\text{sat}_h(u_i(t))$ is within the saturation level. Moreover, from Figure 9.6, we see that each agent only needs to sense and broadcast at its triggering times. Thus continuous sensing, broadcasting, receiving, and listening are avoided. Note however that both the event-triggered control and self-triggered control give rise to a less smooth state evolutions because of the large variability in the control input.

9.7 Summary

In this chapter, we studied the global consensus problem for multi-agent systems with input saturation constraints over digraphs. We showed that global consensus is achieved if and only if the underlying directed communication network has a directed spanning tree by using a Laypunov function. Moreover, we considered event-triggered control and presented



(a) The state evolutions of the multi-agent system (9.5)–(9.6) when each agent performs Algorithm 9.1.



(b) The saturated input of each agent.

Figure 9.5: Performance of the distributed self-triggered saturation control algorithm.

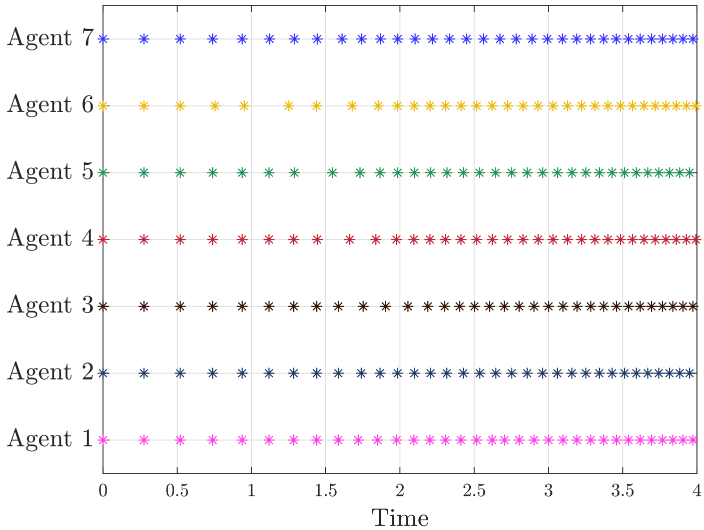


Figure 9.6: The triggering times for each agent determined by the distributed self-triggered saturation control algorithm.

a distributed triggering law and a self-triggered algorithm to reduce the overall need of communication and system updates. We showed that global consensus is still achieved under the same connectivity condition. Furthermore, Zeno behavior was excluded. Future research directions include considering more general systems such as double integrator systems and comparing the convergence speed between the saturation and non-saturation cases.

9.8 Proofs

9.8.1 Proof of sufficiency of Theorem 9.1

The proof of sufficiency follows the structure outlined after the theorem stated in Section 9.3. More specifically, we first show consensus for the case where $M = 1$ in (2.2) which corresponds to only one SCC. Then, we consider the case $M = 2$ in (2.2), and show that the agents in SCC_1 and SCC_2 reach consensus. We finally argue that the general case where $M > 2$ follows in a similar way.

(i) In this step, we consider the situation where $\mathcal{G}$ is strongly connected, i.e., $M = 1$ in (2.2).

We first prove that consensus is achieved. Consider the Lyapunov candidate (9.4) introduced in Remark 9.2. From Lemma 2.1, we have $\xi_i > 0$, $i \in [n]$, since $\mathcal{G}$ is strongly connected. From Lemma 9.1, we know that

$$V_{il}(x(t)) := \int_0^{-\sum_{j=1}^n L_{ij}c_l(x_j(t))} \mathrm{sat}_h(s)ds \geq 0,$$

and $V_{il}(x(t)) = 0$ if and only if $-\sum_{j=1}^n L_{ij}c_l(x_j(t)) = 0$. Then, we know that

$$V(x(t)) = \sum_{i=1}^n \xi_i \sum_{l=1}^p V_{il}(x) \geq 0,$$

and $V(x(t)) = 0$ if and only if $-\sum_{j=1}^n L_{ij}c_l(x_j(t)) = 0$ for all $i \in [n]$ and $l \in [p]$. This is furthermore equivalent to $x_1(t) = \cdots = x_n(t)$ due to $\mathrm{rank}(L) = n-1$. Hence, we have $V(x(t)) \geq 0$ and $V(x(t)) = 0$ if and only if $x_1(t) = \cdots = x_n(t)$.

The derivative of $V(x(t))$ along the trajectories of (9.1)–(9.3) is

$$\begin{aligned}
\dot{V}(x(t)) &= \sum_{i=1}^n \xi_i \sum_{l=1}^p \mathrm{sat}_h\Big(-\sum_{j=1}^n L_{ij}c_l(x_j(t))\Big)\Big(-\sum_{j=1}^n L_{ij}c_l(\dot{x}_j(t))\Big) \\
&= \sum_{i=1}^n \xi_i \sum_{l=1}^p \mathrm{sat}_h(c_l(u_i(t)))\Big(-\sum_{j=1}^n L_{ij}\mathrm{sat}_h(c_l(u_j(t)))\Big) \\
&= \sum_{i=1}^n \xi_i(\mathrm{sat}_h(u_i(t)))^\top \sum_{j=1}^n -L_{ij}\mathrm{sat}_h(u_j(t)) \\
&= -\sum_{i=1}^n \xi_i q_i^s(t),
\end{aligned} \tag{9.9}$$

where

$$q_i^s(t) = -\frac{1}{2}\sum_{j=1}^n L_{ij}\|\mathrm{sat}_h(u_j(t)) - \mathrm{sat}_h(u_i(t))\|^2 \geq 0,$$

and the last equality of (9.9) holds since

$$\begin{aligned}
-\sum_{i=1}^n \xi_i q_i^s(t) &= \sum_{i=1}^n \frac{1}{2}\sum_{j=1}^n \xi_i L_{ij}\|\mathrm{sat}_h(u_j(t)) - \mathrm{sat}_h(u_i(t))\|^2 \\
&= \sum_{i=1}^n \frac{1}{2}\sum_{j=1}^n \xi_i L_{ij}(\|\mathrm{sat}_h(u_j(t))\|^2 + \|\mathrm{sat}_h(u_i(t))\|^2) \\
&\quad - \sum_{i=1}^n \sum_{j=1}^n \xi_i L_{ij}(\mathrm{sat}_h(u_j(t)))^\top \mathrm{sat}_h(u_i(t)) \\
&= \frac{1}{2}\sum_{j=1}^n \|\mathrm{sat}_h(u_j(t))\|^2 \sum_{i=1}^n \xi_i L_{ij} + \frac{1}{2}\sum_{i=1}^n \xi_i\|\mathrm{sat}_h(u_i(t))\|^2 \sum_{j=1}^n L_{ij}
\end{aligned}$$

$$-\sum_{i=1}^{n}\sum_{j=1}^{n}\xi_i L_{ij}(\mathrm{sat}_h(u_j(t)))^\top \mathrm{sat}_h(u_i(t))$$

$$=-\sum_{i=1}^{n}\sum_{j=1}^{n}\xi_i L_{ij}(\mathrm{sat}_h(u_j(t)))^\top \mathrm{sat}_h(u_i(t)), \tag{9.10}$$

where we have used $\xi^\top L = \mathbf{0}_n$ and $L\mathbf{1}_n = \mathbf{0}_n$ in (9.10).

From (9.9), we know that $\dot{V}(x(t)) \leq 0$ and $\dot{V}(x(t)) = 0$ if and only if $\mathrm{sat}_h(u_i(t)) = \mathrm{sat}_h(u_j(t))$, $\forall i, j \in [n]$. It follows from Lemma 9.2 that, this is equivalent to $x_i(t) = x_j(t)$, $\forall i, j \in [n]$. Thus, by LaSalle Invariance Principle [382], we have

$$\lim_{t\to\infty}\|x_i(t) - x_j(t)\| = 0,\ \forall i, j \in [n], \tag{9.11}$$

i.e., consensus is achieved.

We next show that the input of each agent enters into the saturation level in finite time.

Noting that $-\sum_{j=1}^{n} L_{ij}c_l(x_j(t))$, $i \in [n]$, $l \in [p]$ are continuous with respect to t, it follows from (9.11) that there exists a constant $T_1 \geq 0$ such that

$$|c_l(u_i(t))| = \Big|-\sum_{j=1}^{n} L_{ij}c_l(x_j(t))\Big| \leq h,\ \forall t \geq T_1.$$

In other words the saturation function in (9.1) is not active after T_1. Thus,

$$\dot{x}_i(t) = -\sum_{j=1}^{n} L_{ij}x_j(t),\ t \geq T_1. \tag{9.12}$$

Finally, we estimate the convergence speed, which will be used later. Consider the function

$$\tilde{V}(x(t)) = \frac{1}{2}x^\top(t)(U \otimes I_p)x(t). \tag{9.13}$$

From Lemma 2.2, we know that $\tilde{V}(x(t)) \geq 0$. The derivative of $\tilde{V}(x(t))$ along the trajectories of system (9.12) satisfies

$$\begin{aligned}\dot{\tilde{V}}(x(t)) &= x^\top(t)(U\otimes I_p)\dot{x}(t) = x^\top(t)(U\otimes I_p)(-L\otimes I_p)x(t) = -x^\top(t)(R\otimes I_p)x(t)\\ &\leq -\frac{\rho_2(R)}{\rho(U)}x^\top(t)(U\otimes I_p)x(t) = -2\frac{\rho_2(R)}{\rho(U)}\tilde{V}(x(t)),\ \forall t \geq T_1.\end{aligned}$$

Thus,

$$\tilde{V}(x(t)) \leq \tilde{V}(x(T_1))e^{-2\frac{\rho_2(R)}{\rho(U)}(t-T_1)},\ \forall t \geq T_1.$$

Noting that $\tilde{V}(x(t))$ is continuous with respect to t, there exists a positive constant C_1 such that

$$\tilde{V}(x(t)) \leq C_1,\ \forall t \in [0, T_1].$$

Then,

$$\tilde{V}(x(t)) \le C_2 e^{-2\frac{\rho_2(R)}{\rho(U)}t}, \ \forall t \ge 0, \tag{9.14}$$

where $C_2 = \max\{\tilde{V}(x(T_1)),\ C_1 e^{2\frac{\rho_2(R)}{\rho(U)}T_1}\}$.

Moreover, from Lemma 2.2, we know that

$$\begin{aligned}\sum_{j=1}^{n} \|u_j(t)\|^2 &= x^\top(t)(L^\top L \otimes I_p)x(t) \le \frac{\rho(L^\top L)}{\rho_2(U)} x^\top(t)(U \otimes I_p)x(t) \\ &= 2\frac{\rho(L^\top L)}{\rho_2(U)}\tilde{V}(x(t)) \le 2\frac{\rho(L^\top L)}{\rho_2(U)} C_2 e^{-2\frac{\rho_2(R)}{\rho(U)}t}, \ \forall t \ge 0. \end{aligned} \tag{9.15}$$

(ii) In this step, we consider the case where $M \ge 2$, but we first introduce some notations which will be used later.

Let $N_0 = 0$, $N_l = \sum_{m=1}^{l} n_m$, $l \in [M]$, where n_m is the dimension of $L^{m,m}$. Then the i-th agent in SCC_m is the $N_{m-1} + i$-th agent of the whole graph. In the following, we exchangeably use v_i^m and $v_{N_{m-1}+i}$ to denote this agent. Accordingly, denote $x_i^m(t) = x_{N_{m-1}+i}(t)$, $\hat{x}_i^m(t) = \hat{x}_{N_{m-1}+i}(t)$, $u_i^m(t) = u_{N_{m-1}+i}(t)$ and $u^m(t) = \mathrm{col}(u_1^m(t), \dots, u_{n_m}^m(t))$.

In the following we only consider the case where $M = 2$. The case where $M > 2$ can be treated in a similar manner, as discussed in the proof sketch in Section 9.3.

First, note that the agents in SCC_2 do not depend on any agents in SCC_1. Thus, SCC_2 can be treated as a strongly connected digraph. Then, from the analysis in (i), we have

$$\lim_{t\to+\infty} \|x_i^2(t) - x_j^2(t)\| = 0, \ i, j \in [n_2],$$

and that there exists a constant $T_2 \ge 0$ such that

$$|c_l(u_i^2(t))| = \Big| -\sum_{j=1}^{n_2} L_{ij}^{2,2} c_l(x_j^2(t))\Big| \le h, \ \forall t \ge T_2. \tag{9.16}$$

In addition, similar to (9.15), we have

$$\|u^2(t)\|^2 = \sum_{j=1}^{n_2} \|u_j^2(t)\|^2 \le C_3 e^{-C_4 t}, \ t \ge 0,$$

where C_3 and C_4 are two positive constants.

Second, let us consider SCC_1. Similar to $V(x)$ defined in (9.4), define

$$V_1(x(t)) = \sum_{i=1}^{n_1} \xi_i^1 \sum_{l=1}^{p} \int_0^{c_l(u_i^1(t))} \mathrm{sat}_h(s)ds, \tag{9.17}$$

$$V_2(x(t)) = \sum_{i=1}^{n_2} \xi_i^2 \sum_{l=1}^{p} \int_0^{c_l(u_i^2(t))} \mathrm{sat}_h(s)ds. \tag{9.18}$$

From the definition of the component operator $c_l(\cdot)$, we know $c_l(u_i^1(t)) = -\sum_{j=1}^{n_1} L_{ij}^{1,1} c_l(x_i^1(t)) - \sum_{j=1}^{n_2} L_{ij}^{1,2} c_l(x_i^2(t))$ and $c_l(u_i^2(t)) = -\sum_{j=1}^{n_2} L_{ij}^{2,2} c_l(x_i^2(t))$. From Lemma 9.1, we have $V_1(x) \geq 0$ and $V_2(x) \geq 0$.

Similar to the way to get (9.9), we have

$$\dot{V}_2(x(t)) = \sum_{i=1}^{n_2} -\xi_i^2 q_i^2(t),$$

where

$$q_i^2(t) = -\frac{1}{2}\sum_{j=1}^{n} L_{ij}^{2,2} \|\mathrm{sat}_h(u_j^2(t)) - \mathrm{sat}_h(u_i^2(t))\|^2 \geq 0.$$

Moreover, similar to the analysis of $\dot{V}(x(t))$ in (i), we know that $\dot{V}_2(x(t)) = 0$ if and only if $x_i^2(t) = x_j^2(t),\ \forall i, j \in [n_2]$.

The derivative of $V_1(x(t))$ along the trajectories of (9.1)–(9.3) satisfies

$$\begin{aligned}
\dot{V}_1(x(t)) &= \sum_{i=1}^{n_1} \xi_i^1 \sum_{l=1}^{p} \mathrm{sat}_h(c_l(u_i^1(t))) c_l(\dot{u}_i^1(t)) \\
&= \sum_{i=1}^{n_1} \xi_i^1 \sum_{l=1}^{p} c_l(\mathrm{sat}_h(u_i^1(t)))\Big(-\sum_{j=1}^{n_1} L_{ij}^{1,1} c_l(\mathrm{sat}_h(u_j^1(t))) - \sum_{j=1}^{n_2} L_{ij}^{1,2} c_l(\mathrm{sat}_h(u_j^2(t)))\Big) \\
&= \sum_{i=1}^{n_1} \xi_i^1 (\mathrm{sat}_h(u_i^1(t)))^\top \Big(-\sum_{j=1}^{n_1} L_{ij}^{1,1} \mathrm{sat}_h(u_j^1(t)) - \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(u_j^2(t))\Big) \\
&= -(\mathrm{sat}_h(u^1(t)))^\top (Q^1 \otimes I_p) \mathrm{sat}_h(u^1(t)) - \sum_{i=1}^{n_1} \xi_i^1 (\mathrm{sat}_h(u_i^1(t)))^\top \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(u_j^2(t)) \\
&\leq -\rho_2(Q^1)\|\mathrm{sat}_h(u^1(t))\|^2 + \frac{\rho_2(Q^1)}{2} \sum_{i=1}^{n_1} \|\mathrm{sat}_h(u_i^1(t))\|^2 \\
&\quad + \frac{1}{2\rho_2(Q^1)} \sum_{i=1}^{n_1} \Big\|\xi_i^1 \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(u_j^2(t))\Big\|^2 \\
&\leq -\frac{\rho_2(Q^1)}{2}\|\mathrm{sat}_h(u^1(t))\|^2 + \frac{n_1 n_2 \max_{i\in[n_1], j\in[n_2]}\{(L_{ij}^{1,2})^2\}}{2\rho_2(Q^1)} \|\mathrm{sat}_h(u^2(t))\|^2 \\
&\leq -\frac{\rho_2(Q^1)}{2}\|\mathrm{sat}_h(u^1(t))\|^2 + \frac{n_1 n_2 \max_{i\in[n_1], j\in[n_2]}\{(L_{ij}^{1,2})^2\}}{2\rho_2(Q^1)} C_3 e^{-C_4 t},\ t \geq 0,
\end{aligned}$$

where the first inequality holds due to $Q^1 > 0$ which is stated in Lemma 2.4.

Let us treat $y_i(t) = e^{-C_4 t},\ t \geq 0,\ i \in [n]$, as an additional state of each agent, and let $y(t) = \mathrm{col}(y_1(t), \ldots, y_n(t))^\top$. Consider a Lyapunov candidate:

$$V_3(x(t), y(t)) = V_1(x(t)) + V_2(x(t)) + \frac{2 n_1 n_2 \max_{i\in[n_1], j\in[n_2]}\{(L_{ij}^{1,2})^2\}}{2\rho_2(Q^1) C_4 n} C_3 \sum_{i=1}^{n} y_i(t).$$

The derivative of $V_3(x(t), y(t))$ along the trajectories of (9.1)–(9.3) is

$$\dot{V}_3(x(t), y(t)) = \dot{V}_1(x(t)) + \dot{V}_2(x(t)) - \frac{2n_1 n_2 \max_{i\in[n_1], j\in[n_2]}\{(L_{ij}^{1,2})^2\}}{2\rho_2(Q^1)n} C_3 \sum_{i=1}^{n} y_i(t).$$

Then, we have

$$\begin{aligned}\dot{V}_3(x(t), y(t)) \le & -\frac{\rho_2(Q^1)}{2}\|\mathrm{sat}_h(u^1(t))\|^2 + \sum_{i=1}^{n_2} -\xi_i^2 q_i^2(t) \\ & - \frac{n_1 n_2 \max_{i\in[n_1], j\in[n_2]}\{(L_{ij}^{1,2})^2\}}{2\rho_2(Q^1)n} C_3 \sum_{i=1}^{n} y_i(t),\ t \ge 0.\end{aligned}$$

By LaSalle Invariance Principle, similar to the analysis in (i), we have

$$\lim_{t\to\infty} \|x_j(t) - x_i(t)\| = 0,\ \forall i, j \in [n].$$

Thus, consensus is achieved. Moreover, similar to the analysis in (i), we can show that after a finite time $T_2 \ge 0$ the saturation is no longer active.

9.8.2 Proof of Theorem 9.2

(i) Similar to the proof of excluding Zeno behavior in Theorem 8.1, we prove that there is no Zeno behavior by contradiction. Suppose there exists Zeno behavior. Then there exists an agent i, such that $\lim_{k\to\infty} t_k^i = T_0$ for some constant T_0. Let $\varepsilon_0 = \frac{\sqrt{\alpha_i}}{2\sqrt{p}h} e^{-\frac{1}{2}\beta_i T_0} > 0$. Then from the property of limits, there exists a positive integer $N(\varepsilon_0)$ such that

$$t_k^i \in [T_0 - \varepsilon_0, T_0],\ \forall k \ge N(\varepsilon_0). \tag{9.19}$$

Also noting $\|\mathrm{sat}_h(s)\| \le h\sqrt{p}$ for any $s \in \mathbb{R}^p$, we have

$$\|\mathrm{sat}_h(\hat{u}_i(t))\| \le h\sqrt{p}.$$

Noting

$$\left|\frac{d\|e_i(t)\|}{dt}\right| \le \|\dot{x}_i(t)\| = \|\mathrm{sat}_h(\hat{u}_i(t))\| \le h\sqrt{p},$$

and $\|\hat{x}_i(t_k^i) - x_i(t_k^i)\| = 0$ for any triggering time t_k^i, we conclude that one necessary condition to guarantee $\|e_i(t)\|^2 \ge \alpha_i e^{-\beta_i t}, t \ge t_k^i$ is

$$(t - t_k^i)h\sqrt{p} \ge \sqrt{\alpha_i} e^{-\frac{1}{2}\beta_i t},\ t \ge t_k^i.$$

Then, similar to (8.17), we have

$$t_{N(\varepsilon_0)+1}^i - t_{N(\varepsilon_0)}^i \ge \frac{\sqrt{\alpha_i}}{\sqrt{p}h} e^{-\frac{1}{2}\beta_i t_{N(\varepsilon_0)+1}^i} \ge \frac{\sqrt{\alpha_i}}{\sqrt{p}h} e^{-\frac{1}{2}\beta_i T_0} = 2\varepsilon_0,$$

which contradicts (9.19). Therefore, there is no Zeno behavior.
(ii) (Necessity) Necessity follows from Lemma 2.3.
(Sufficiency) (ii-1) In this step, we consider the situation where $\mathcal{G}$ is strongly connected, i.e., $M = 1$ in (2.2).

We first show that consensus is achieved. Let $f_i(t) = \mathrm{sat}_h(\hat{u}_i(t)) - \mathrm{sat}_h(u_i(t))$. We have

$$\begin{aligned}
-\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\xi_i L_{ij}\|f_j(t)\|^2 &= -\sum_{i=1}^{n}\Big(-\xi_i L_{ii}\|f_i(t)\|^2 + \sum_{j=1}^{n}\xi_i L_{ij}\|f_j(t)\|^2\Big)\\
&= \sum_{i=1}^{n}\xi_i L_{ii}\|f_i(t)\|^2 - \sum_{j=1}^{n}\sum_{i=1}^{n}\xi_i L_{ij}\|f_j(t)\|^2\\
&\overset{*}{=} \sum_{i=1}^{n}\xi_i L_{ii}\|f_i(t)\|^2,
\end{aligned}\tag{9.20}$$

where the equality denoted by $\overset{*}{=}$ holds due to $\xi^\top L = \mathbf{0}_n$.

We have

$$\begin{aligned}
&-\sum_{i=1}^{n}\sum_{j=1}^{n}\xi_i L_{ij}[f_j(t)]^\top \mathrm{sat}_h(u_i(t))\\
&\overset{*}{=} -\sum_{i=1}^{n}\sum_{j=1}^{n}\xi_i L_{ij}[f_j(t)]^\top[\mathrm{sat}_h(u_i(t)) - \mathrm{sat}_h(u_j(t))]\\
&= -\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\xi_i L_{ij}[f_j(t)]^\top[\mathrm{sat}_h(u_i(t)) - \mathrm{sat}_h(u_j(t))]\\
&\le -\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\xi_i L_{ij}\Big(\|f_j(t)\|^2 + \frac{1}{4}\|\mathrm{sat}_h(u_i(t)) - \mathrm{sat}_h(u_j(t))\|^2\Big)\\
&= -\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\xi_i L_{ij}\|f_j(t)\|^2 - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\xi_i L_{ij}\|\mathrm{sat}_h(u_i(t)) - \mathrm{sat}_h(u_j(t))\|^2\\
&= -\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\xi_i L_{ij}\|f_j(t)\|^2 - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\xi_i L_{ij}\|\mathrm{sat}_h(u_i(t)) - \mathrm{sat}_h(u_j(t))\|^2\\
&\overset{**}{=} \sum_{i=1}^{n}\xi_i L_{ii}\|f_i(t)\|^2 + \frac{1}{2}\sum_{i=1}^{n}\xi_i q_i^s(t),
\end{aligned}\tag{9.21}$$

where the equality denoted by $\overset{*}{=}$ holds due to $\xi^\top L = \mathbf{0}_n$; the inequality holds due to the Cauchy-Schwarz inequality; and the equality denoted by $\overset{**}{=}$ holds due to (9.20) and the definition of $q_i^s(t)$.

The derivative of $V(x)$, as defined in (9.4), but along the trajectories of (9.5)–(9.6), satisfies

$$\begin{aligned}
\dot{V}(x(t)) &= \sum_{i=1}^{n} \xi_i \sum_{l=1}^{p} \mathrm{sat}_h\Big(-\sum_{j=1}^{n} L_{ij} c_l(x_j(t))\Big)\Big(-\sum_{j=1}^{n} L_{ij} c_l(\dot{x}_j(t))\Big) \\
&= \sum_{i=1}^{n} \xi_i \sum_{l=1}^{p} \mathrm{sat}_h(c_l(u_i(t)))\Big(-\sum_{j=1}^{n} L_{ij} \mathrm{sat}_h(c_l(\hat{u}_j(t)))\Big) \\
&= -\sum_{i=1}^{n} \xi_i (\mathrm{sat}_h(u_i(t)))^\top \sum_{j=1}^{n} L_{ij} \mathrm{sat}_h(\hat{u}_j(t)) \\
&= -\sum_{i=1}^{n} \xi_i (\mathrm{sat}_h(u_i(t)))^\top \sum_{j=1}^{n} L_{ij} (\mathrm{sat}_h(u_j(t)) + f_j(t)) \\
&= -\sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i L_{ij} (\mathrm{sat}_h(u_i(t)))^\top \mathrm{sat}_h(u_j(t)) - \sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i L_{ij} (f_j(t))^\top \mathrm{sat}_h(u_i(t)) \\
&\overset{*}{\le} -\sum_{i=1}^{n} \xi_i q_i^s(t) + \sum_{i=1}^{n} \xi_i L_{ii} \|f_i(t)\|^2 + \frac{1}{2} \sum_{i=1}^{n} \xi_i q_i^s(t) \\
&= -\sum_{i=1}^{n} \frac{\xi_i}{2} q_i^s(t) + \sum_{i=1}^{n} \xi_i L_{ii} \|f_i(t)\|^2 \\
&= -\sum_{i=1}^{n} \frac{\xi_i}{2} q_i^s(t) + \sum_{i=1}^{n} \xi_i L_{ii} \|\mathrm{sat}_h(\hat{u}_i(t)) - \mathrm{sat}_h(u_i(t))\|^2 \\
&\overset{**}{\le} -\sum_{i=1}^{n} \frac{\xi_i}{2} q_i^s(t) + \sum_{i=1}^{n} \xi_i L_{ii} \|\hat{u}_i(t) - u_i(t)\|^2 \\
&= -\sum_{i=1}^{n} \frac{\xi_i}{2} q_i^s(t) + \sum_{i=1}^{n} \xi_i L_{ii} \Big\| \sum_{j=1}^{n} L_{ij} e_j(t) \Big\|^2 \\
&\le -\sum_{i=1}^{n} \frac{\xi_i}{2} q_i^s(t) + \max_{i\in[n]}\{\xi_i L_{ii}\} e^\top(t)(L^\top L \otimes I_p) e(t) \\
&\le -\sum_{i=1}^{n} \frac{\xi_i}{2} q_i^s(t) + \max_{i\in[n]}\{\xi_i L_{ii}\} \rho(L^\top L) \sum_{i=1}^{n} \|e_i(t)\|^2,
\end{aligned} \tag{9.22}$$

where the inequality denoted by $\overset{*}{\le}$ holds due to (9.10) and (9.21); and the inequality denoted by $\overset{**}{\le}$ holds due to Lemma 9.1.

Let us treat $z_i(t) = e^{-\beta_i t}$, $t \ge 0$ as an additional state to agent i, $i \in [n]$, and let $z(t) = \mathrm{col}(z_1(t), \dots, z_n(t))$. Consider a Lyapunov candidate:

$$W(x(t), z(t)) = V(x(t)) + 2 \max_{i\in[n]}\{\xi_i L_{ii}\} \rho(L^\top L) \sum_{i=1}^{n} \frac{\alpha_i}{\beta_i} z_i(t).$$

The derivative of $W(x(t), z(t))$ along the trajectories of (9.5)–(9.6) and $\dot{z}_i(t) = -\beta_i z_i(t)$ is

$$
\begin{aligned}
\dot{W}(x(t), z(t)) &= \dot{V}(x) - 2\max_{i\in[n]}\{\xi_i L_{ii}\}\rho(L^\top L)\sum_{i=1}^{n}\alpha_i e^{-\beta_i t} \\
&\leq -\sum_{i=1}^{n}\frac{\xi_i}{2}q_i^s(t) + \max_{i\in[n]}\{\xi_i L_{ii}\}\rho(L^\top L)\sum_{i=1}^{n}\|e_i(t)\|^2 \\
&\quad - 2\max_{i\in[n]}\{\xi_i L_{ii}\}\rho(L^\top L)\sum_{i=1}^{n}\alpha_i e^{-\beta_i t} \\
&\leq -\sum_{i=1}^{n}\frac{\xi_i}{4}q_i^s(t) - \max_{i\in[n]}\{\xi_i L_{ii}\}\rho(L^\top L)\sum_{i=1}^{n}\alpha_i e^{-\beta_i t} \leq 0.
\end{aligned}
$$

By LaSalle Invariance Principle, similar to the proof of Theorem 9.1, we have

$$
\lim_{t\to\infty}\|x_j(t) - x_i(t)\| = 0, \ i, j \in [n], \tag{9.23}
$$

i.e., consensus is achieved.

We next show that the input of each agent enters into the saturation level in finite time.

Noting that $c_l(\hat{u}_i(t)) = -\sum_{j=1}^{n} L_{ij}c_l(x_j(t)) - \sum_{j=1}^{n} L_{ij}c_l(e_j(t))$, (9.7), $-\sum_{j=1}^{n} L_{ij}c_l(x_j(t))$, $i \in [n]$, $l \in [p]$ are continuous with respect to t, it follows from (9.23) that there exists a constant $T_3 \geq 0$ such that

$$
|c_l(\hat{u}_i(t))| \leq \Big| -\sum_{j=1}^{n} L_{ij}c_l(x_j(t))\Big| + \Big| -\sum_{j=1}^{n} L_{ij}c_l(e_j(t))\Big| \leq h, \ \forall t \geq T_3.
$$

In other words, the saturation function in (9.5) is no longer active after T_3. Thus, the multi-agent system (9.5) with event-triggered control input (9.6) reduces to

$$
\dot{x}_i(t) = -\sum_{j=1}^{n} L_{ij}\hat{x}_j(t), \ t \geq T_3.
$$

Finally, we estimate the convergence speed, which will be used later. Similar to the proof of Theorem 2 in [358], we conclude that there exist $C_5 > 0$ and $C_6 > 0$ such that

$$
\tilde{V}(x(t)) \leq C_5 e^{-C_6 t}, \ \forall t \geq T_3,
$$

where $\tilde{V}(x(t))$ is defined in (9.13). Similar to (9.14), we have

$$
\tilde{V}(x(t)) \leq C_7 e^{-C_6 t}, \ \forall t \geq 0,
$$

where C_7 is a positive constant.

Moreover, similar to the analysis for obtaining (9.15), we have

$$
\sum_{i=1}^{n}\|\hat{u}_i(t)\|^2 = \sum_{i=1}^{n}\|u_i(t) - \sum_{j=1}^{n} L_{ij}e_j(t)\|^2
$$

$$\leq 2\sum_{i=1}^{n}\|u_i(t)\|^2 + 2\rho(L^\top L)\sum_{i=1}^{n}\|e_i(t)\|^2 \leq C_9 e^{-C_8 t},\ \forall t \geq 0, \tag{9.24}$$

where C_9 and C_8 are two positive constants.

(ii-2) In this step, we consider the situation where $\mathcal{G}$ has a directed spanning tree but it is not strongly connected, i.e., $M \geq 2$ in (2.2). For simplicity, we only consider the case where $M = 2$. The general case can be treated in a similar manner. We use the same notation as in the proof of Theorem 9.1. For simplicity, let $\hat{u}_i^m(t) = \hat{u}_{N_{m-1}+i}(t)$, $e_i^m(t) = e_{N_{m-1}+i}(t)$, $f_i^m(t) = f_{N_{m-1}+i}(t)$, $\alpha_i^m = \alpha_{N_{m-1}+i}$, $\beta_i^m = \beta_{N_{m-1}+i}$, and $\hat{u}^m(t) = \mathrm{col}(\hat{u}_1^m(t), \dots, \hat{u}_{n_m}^m(t))$.

First, let us consider SCC_2 and note that no agent in SCC_2 is dependent on any agent in SCC_1. Thus, SCC_2 can be treated as a strongly connected digraph. Then, from the analysis in (ii-1), we have that

$$\lim_{t\to\infty}\|x_i^2(t) - x_j^2(t)\| = 0,\ i, j \in [n_2],$$

and that there exists a constant $T_4 \geq 0$ such that

$$|c_l(\hat{u}_i^2(t))| = \Big| -\sum_{j=1}^{n_2} L_{ij}^{2,2} c_l(\hat{x}_j^2(t))\Big| \leq h,\ \forall t \geq T_4.$$

In addition, similar to (9.24), we have

$$\|\hat{u}^2(t)\|^2 = \sum_{j=1}^{n_2}\|\hat{u}_j^2(t)\|^2 \leq C_{11}e^{-C_{10}t},\ t \geq 0,$$

where C_{11} and C_{10} are two positive constants.

Second, let us consider SCC_1. Similar to (9.22), the derivative of $V_2(x(t))$, as defined in (9.18), but along the trajectories of system (9.5)–(9.6), satisfies

$$\dot{V}_2(x(t)) \leq -\sum_{i=1}^{n_2}\frac{\xi_i^2}{2}q_i^2(t) + d_1\sum_{i=1}^{n_2}\|e_i^2(t)\|^2,$$

where

$$d_1 = \max_{i\in[n]}\{\xi_i^2 L_{ii}^{2,2}\}\rho((L^{2,2})^\top L^{2,2}).$$

The derivative of $V_1(x(t))$, as defined in (9.17), but along the trajectories of system (9.5)–(9.6), satisfies

$$\begin{aligned}
\dot{V}_1(x(t)) &= \sum_{i=1}^{n_1}\xi_i^1\sum_{l=1}^{p}\mathrm{sat}_h(c_l(u_i^1(t)))c_l(\dot{u}_i^1(t)) \\
&= \sum_{i=1}^{n_1}\xi_i^1\sum_{l=1}^{p}c_l(\mathrm{sat}_h(u_i^1(t)))\Big(-\sum_{j=1}^{n_1}L_{ij}^{1,1}c_l(\mathrm{sat}_h(\hat{u}_j^1(t))) - \sum_{j=1}^{n_2}L_{ij}^{1,2}c_l(\mathrm{sat}_h(\hat{u}_j^2(t)))\Big) \\
&= \sum_{i=1}^{n_1}\xi_i^1(\mathrm{sat}_h(u_i^1(t)))^\top\Big(-\sum_{j=1}^{n_1}L_{ij}^{1,1}\mathrm{sat}_h(\hat{u}_j^1(t)) - \sum_{j=1}^{n_2}L_{ij}^{1,2}\mathrm{sat}_h(\hat{u}_j^2(t))\Big)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n_1} \xi_i^1 (\mathrm{sat}_h(\hat{u}_i^1(t)) - f_i^1(t))^\top \Big(- \sum_{j=1}^{n_1} L_{ij}^{1,1} \mathrm{sat}_h(\hat{u}_j^1(t)) - \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(\hat{u}_j^2(t)) \Big) \\
&= -(\mathrm{sat}_h(\hat{u}^1(t)))^\top (Q^1 \otimes I_p) \mathrm{sat}_h(\hat{u}^1(t)) + \sum_{i=1}^{n_1} \xi_i^1 (\mathrm{sat}_h(\hat{u}_i^1(t)))^\top \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(\hat{u}_j^2(t)) \\
&\quad + \sum_{i=1}^{n_1} \xi_i^1 (f_i^1(t))^\top \Big(\sum_{j=1}^{n_1} L_{ij}^{1,1} \mathrm{sat}_h(\hat{u}_j^1(t)) + \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(\hat{u}_j^2(t)) \Big) \\
&\le -\rho_2(Q^1) \|\mathrm{sat}_h(\hat{u}^1(t))\|^2 + \frac{\rho_2(Q^1)}{4} \sum_{i=1}^{n_1} \|\mathrm{sat}_h(\hat{u}_i^1(t))\|^2 \\
&\quad + \frac{1}{\rho_2(Q^1)} \sum_{i=1}^{n_1} \Big\| \xi_i^1 \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(\hat{u}_j^2(t)) \Big\|^2 + \frac{\rho_2(Q^1)}{4} \sum_{j=1}^{n_1} \|\mathrm{sat}_h(\hat{u}_j^1(t))\|^2 \\
&\quad + \frac{1}{\rho_2(Q^1)} \sum_{j=1}^{n_1} \Big\| \sum_{i=1}^{n_1} \xi_i^1 L_{ij}^{1,1} f_i^1(t) \Big\|^2 + \sum_{i=1}^{n_1} \frac{1}{4} \|f_i^1(t)\|^2 + \sum_{i=1}^{n_1} \Big\| \xi_i^1 \sum_{j=1}^{n_2} L_{ij}^{1,2} \mathrm{sat}_h(\hat{u}_j^2(t)) \Big\|^2 \\
&\le -\frac{\rho_2(Q^1)}{2} \|\mathrm{sat}_h(\hat{u}^1(t))\|^2 + d_2 \sum_{i=1}^{n_1} \|f_i^1(t)\|^2 + d_3 \|\mathrm{sat}_h(\hat{u}^2(t))\|^2,
\end{aligned} \tag{9.25}
$$

where

$$
d_2 = \frac{1}{4} + (n_1)^2 \max_{i,j \in [n_1]} \{(\xi_i^1 L_{ij}^{1,1})^2\} \frac{1}{\rho_2(Q^1)}, \ d_3 = 2 n_1 n_2 \max_{i \in [n_1], j \in [n_2]} \{(\xi_i^1 L_{ij}^{1,2})^2\} \Big(\frac{1}{\rho_2(Q^1)} + 1 \Big).
$$

Similar to the analysis to get (9.22), from (9.25), we have

$$
\dot{V}_1(x(t)) \le -\frac{\rho_2(Q^1)}{2} \|\mathrm{sat}_h(\hat{u}^1(t))\|^2 + d_4 \sum_{i=1}^{n_1} \|e_i^1(t)\|^2 + d_4 \sum_{i=1}^{n_2} \|e_i^2(t)\|^2 + d_3 \|\mathrm{sat}_h(\hat{u}^2(t))\|^2,
$$

where

$$
d_4 = d_2 \rho(L^\top L).
$$

Let us treat $\eta_i^r(t) = e^{-\beta_i^r y}, \ t \ge 0$, as an additional state of agent $v_i^r, \ r = 1, 2, \ i \in [n_2]$, $\theta_i^2(t) = e^{-C_{10} t}, \ t \ge 0$, as an additional state of agent $v_i^2, \ i \in [n_2]$, and $\theta_i^1(t) = 0, \ t \ge 0$, as an additional state of agent $v_i^1, \ i \in [n_1]$. Let $\eta(t) = \mathrm{col}(\eta_1^1(t), \ldots, \eta_{n_1}^1(t), \eta_1^2(t), \ldots, \eta_{n_2}^1(t))$ and $\theta(t) = \mathrm{col}(\theta_1^1(t), \ldots, \theta_{n_1}^1(t), \theta_1^2(t), \ldots, \theta_{n_2}^1(t))$.

Consider the Lyapunov candidate

$$
\begin{aligned}
W_r(x(t), \eta(t), \theta(t)) = V_1(x(t)) + V_2(x(t)) + 2 \frac{C_{11}}{C_{10}} d_3 \sum_{i=1}^{n_2} \theta_i^2(t) \\
+ 2 \sum_{i=1}^{n_2} \frac{(d_1 + d_4) \alpha_i^2}{\beta_i^2} \eta_i^2(t) + 2 \sum_{i=1}^{n_1} \frac{d_4 \alpha_i^1}{\beta_i^1} \eta_i^1(t).
\end{aligned}
$$

The derivative of $W_r(x(t), \eta(t), \theta(t))$ along the trajectories of system (9.5)–(9.6) satisfies

$$\begin{aligned}\dot{W}_r(x(t), \eta(t), \theta(t)) = \dot{V}_1(x(t)) + \dot{V}_2(x(t)) - 2C_{11}d_3 \sum_{i=1}^{n_2} \theta_i^2(t) \\ - 2\sum_{i=1}^{n_2}(d_1 + d_4)\alpha_i^2\eta_i^2(t) - 2\sum_{i=1}^{n_1} d_4\alpha_i^1\eta_i^1(t).\end{aligned}$$

Then, for any $t \geq T_4$, we have

$$\begin{aligned}\dot{W}_r(x(t), \eta(t), \theta(t)) \leq -\frac{\rho_2(Q^1)}{2}\|\mathrm{sat}_h(u^1(t))\|^2 + \sum_{i=1}^{n_2} -\frac{\xi_i^2}{2}q_i^2(t) \\ - C_{11}d_3\sum_{i=1}^{n_2}\theta_i^2(t) - \sum_{i=1}^{n_2}(d_1 + d_4)\alpha_i^2\eta_i^2(t) - \sum_{i=1}^{n_1} d_4\alpha_i^1\eta_i^1(t).\end{aligned}$$

By LaSalle Invariance Principle again, we have

$$\lim_{t\to\infty} \|x_j(t) - x_i(t)\| = 0, \ \ i, j \in [n].$$

Thus, consensus is achieved. Moreover, similar to the analysis in (ii-1), we can show that after a finite time the saturation is no longer active.

Chapter 10

Distributed event-triggered formation control algorithms

In this chapter, event- and self-triggered control algorithms are proposed to establish prespecified formations with connectivity preservation. Each agent only needs to update its control input by sensing the relative state to its neighbors and to broadcast its triggering information at its own triggering times. The agents listen to and receive neighbors' triggering information at their triggering times. Two types of system dynamics, single and double integrators, are considered. It is shown that all agents converge to the prespecified formation exponentially with connectivity preservation and exclusion of Zeno behavior. Numerical simulations are provided to illustrate the effectiveness of the theoretical results.

The rest of this chapter is organized as follows. Section 10.1 gives the background. Section 10.2 introduces the formation control problem. Section 10.3 provides event-triggered formation control algorithms for first-order continuous-time multi-agent systems with connectivity preservation. Section 10.4 extends the results to second-order systems. Simulations are given in Section 10.5. This chapter is concluded in Section 10.6. Proofs can be found in Section 10.7.

10.1 Introduction

Generally speaking, formation control for a multi-agent system is about making the agents move to a desired geometric shape. In the survey paper [383], the authors categorized the existing results on formation control into position-, displacement-, and distance-based control according to types of sensed and controlled variables, as summarized in Table 10.1. In position-based control, agents sense their own positions with respect to a global coordinate system. They actively control their own positions to achieve the desired formation, which is prescribed by desired positions with respect to the global coordinate system. This kind of work can be found in [384–387]. In displacement-based control, agents actively control displacements of their neighboring agents to achieve the desired formation, which is specified by the desired displacements with respect to a global coordinate system under the assumption that each agent is able to sense relative positions

Table 10.1: Summary of formation control principles.

	Position-based	Displacement-based	Distance-based
Sensors	Positions	Relative positions	Relative positions
Controls	Positions	Relative positions	Inter-agent distances
Coordinates	Global coordinate system	Orientation aligned local coordinate systems	Local coordinate systems
Interactions	Usually not required	Existence of a spanning tree	Rigidity or persistence

to its neighboring agents with respect to the global coordinate system. This implies that the agents need to know the orientation of the global coordinate system. However, the agents require neither knowledge on the global coordinate system itself nor their positions with respect to the coordinate system. This kind of work can be found in [388–392]. In distance-based control, inter-agent distances are actively controlled to achieve the desired formation, which is given by the desired inter-agent distances. Individual agents are assumed to be able to sense relative positions to their neighboring agents with respect to their own local coordinate systems. The orientations of local coordinate systems are not necessarily aligned with each other. This kind of work can be found in [393–396].

In the study of distributed coordination, such as consensus and formation control, one vital assumption is that the associated communication graph is connected or has a directed spanning tree, at least in some average sense. However, in realistic applications, it is difficult to guarantee this assumption. For example, in mobile robot networks with limited communication range, connectivity of the initial deployment of the robots do not guarantee connectivity in the future.

Motivated by this, many researchers have studied connectivity preservation for multi-agent systems. In particular, the control should ensure that the associated communication graph remains connected during the evolution of the system. For instance, in [397], the authors presented a geometric analysis of wireless connectivity in vehicle networks. In [398], the authors proposed a decentralized control strategy that drives a system of multiple nonholonomic kinematic unicycles to agreement and maintains at the same time the connectivity properties of the initially formed communication graph. In [388], the authors designed nonlinear control input based on an edge-tension function to solve the formation control problem while ensuring connectedness. In [399], the authors proposed a centralized feedback control framework based on artificial potential fields to maintain graph connectivity. In [400], the authors introduced a general class of distributed potential functions guaranteeing connectivity for single-integrator agents. In [401], based on the navigation function formalism, the authors developed a decentralized controller to enable a group of agents to achieve a desired global configuration while maintaining global network connectivity. In [402], the authors provided a decentralized robust control approach, which guarantees that connectivity is maintained when certain bounded input terms are added to the control law.

In this chapter, we study formation control for multi-agent systems with connectivity preservation and event-triggered control. We have the following contributions.

(C10.1) We propose distributed triggering laws for agents to determine their triggering times and one corresponding algorithm for each agent to avoid continuous monitoring of its own triggering law. The advantages of this algorithm are that absolute measurements of states are avoided and it is only at its triggering times that each agent needs to update its control input by sensing the relative states, to broadcast its triggering information, including current triggering time and control input at this time, to its neighbors. The main disadvantage is that continuous listening is still needed. To overcome this, we then present two self-triggered algorithms.

(C10.2) Two types of system dynamics, single integrators and double integrators, are considered. We show that under the proposed event- and self-triggered algorithms all agents converge to prespecified formations exponentially with connectivity preservation. In addition, Zeno behavior can be excluded by proving that the inter-event times are lower bounded by a positive constant for single integrators and the triggering time sequence of each agent is divergent for double integrators. Two related existing studies are [403], [404]. However, [403] does not explicitly exclude Zeno behavior, but it is well known that such behavior can be problematic, see [196]. And it is under the assumption that no agent exhibits Zeno behavior, that [404] proves asymptotic rendezvous can be achieved.

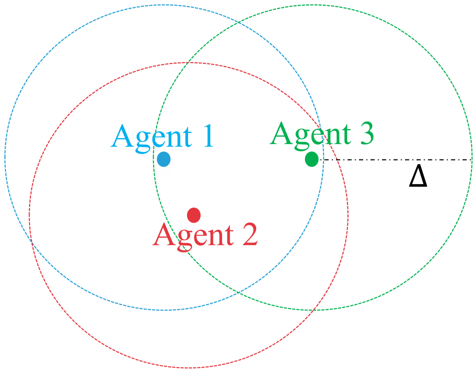
10.2 Formation control for multi-agent systems with connectivity preservation

Consider a connected and undirected graph $\mathcal{G}$ with n vertices and n_e edges. Let $B(\mathcal{G})$ denotes its incidence matrix which is defined in Section 2.2 and $d_{ij} \in \mathbb{R}^p$ the desired internode displacement of edge $(i, j) \in \mathcal{E}(\mathcal{G})$. Denote $\Phi = \{\mathrm{col}(\tau_1, \dots, \tau_n) \in \mathbb{R}^{np} : \tau_i - \tau_j = d_{ij}, \forall (i, j) \in \mathcal{E}(\mathcal{G})\}$. We call the set of desired internode displacements $\{d_{ij}, (i, j) \in \mathcal{E}(\mathcal{G})\}$ a formation associated with $\mathcal{G}$ and we say it is feasible if $\Phi \neq \emptyset$.

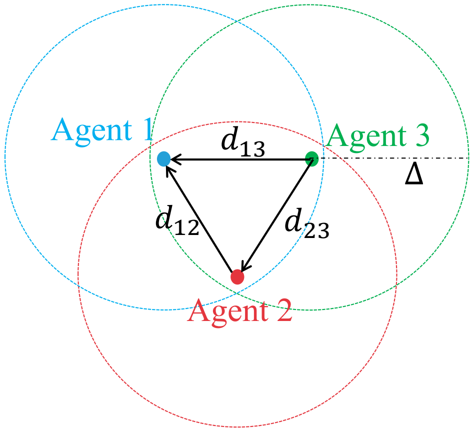
Definition 10.1 (Achieving desired formation)**.** *Consider a multi-agent system with n agents whose underlying graph is $\mathcal{G}$. Let $x_i(t) \in \mathbb{R}^p$ denotes the position of agent i at time $t \geq 0$. The multi-agent system converges to a desired formation $\{d_{ij}, (i, j) \in \mathcal{E}(\mathcal{G})\}$ if*

$$\lim_{t\to\infty}(x_i(t) - x_j(t)) = d_{ij}, \ \forall (i, j) \in \mathcal{E}(\mathcal{G}).$$

In practice, agents normally have limited communication capabilities and one agent cannot exchange information with the agents that outside its communication radius. For simplicity we assume all agents have the same communication radius $\Delta > 0$. Figure 10.1 (a) shows the initial positions of three agents and each agent has the same communication radius Δ; and Figure 10.1 (b) shows the desired formation $\{d_{12}, d_{13}, d_{23}\}$. We say the graph $\mathcal{G}$ and the multi-agent system are consistent if $\|x_i(t) - x_j(t)\| \leq \Delta$ for all $(i, j) \in \mathcal{E}(\mathcal{G})$ and all times $t \geq 0$. Namely, the communication channels are kept for all time. Notice here that we assume the following.



(a) The initial positions of three agents.



(b) The desired formation $\{d_{12}, d_{13}, d_{23}\}$.

Figure 10.1: Illustration of formation control.

Assumption 10.1. *The desired formation* $\{d_{ij},\ (i,j) \in \mathcal{E}(\mathcal{G})\}$ *is feasible and* $\|d_{ij}\| < \Delta$, $\forall (i,j) \in \mathcal{E}(\mathcal{G})$.

Definition 10.2 (Achieving desired formation with connectivity preservation)**.** *A group of agents are said to converge to the desired formation with connectivity preservation if they converge to the desired formation while the graph* $\mathcal{G}$ *remains consistent with their dynamics.*

Note that we do not assume new edges are created, while we only show that old edges are maintained. Our goal in this chapter is to solve the following problem.

Problem 10.1. *Propose distributed event-triggered control input and determine the corresponding triggering times for first- and second-order multi-agent systems such that the desired formation is achieved with connectivity preservation, while continuous exchange of information, continuous update of actuators, and Zeno behavior are avoided.*

10.3 Distributed event-triggered formation control for single integrators

In this section, we consider the case that the dynamics of agents are modeled as single integrators

$$\dot{x}_i(t) = u_i(t),\ i \in [n],\ t \geq 0, \tag{10.1}$$

where $x_i(t) \in \mathbb{R}^p$ is the position and $u_i(t) \in \mathbb{R}^p$ is the control input of agent i with $p > 0$ being the dimension.

From Assumption 10.1, we know $\Phi \neq \emptyset$. Choose any $\mathrm{col}(\tau_1, \dots, \tau_n) \in \Phi$. Let $y_i(t) = x_i(t) - \tau_i$ for $i \in [n]$ and $y(t) = \mathrm{col}(y_1(t), \dots, y_n(t))$. Then, we can rewrite the above multi-agent system as

$$\dot{y}_i(t) = u_i(t), \; i \in [n], \; t \geq 0. \tag{10.2}$$

At time t, for $\|y_i(t) - y_j(t)\| < \Delta - \|d_{ij}\|$, the edge-tension function ν_{ij} (introduced in [388]) is defined as

$$\nu_{ij}(\Delta, y(t)) = \begin{cases} \frac{\|y_i(t)-y_j(t)\|^2}{\Delta-\|d_{ij}\|-\|y_i(t)-y_j(t)\|}, & \text{if } (i,j) \in \mathcal{E}(\mathcal{G}), \\ 0, & \text{otherwise} \end{cases}$$

with

$$\frac{\partial \nu_{ij}(\Delta, y(t))}{\partial y_i} = \begin{cases} \frac{2\Delta-2\|d_{ij}\|-\|y_i(t)-y_j(t)\|}{(\Delta-\|d_{ij}\|-\|y_i(t)-y_j(t)\|)^2}(y_i(t) - y_j(t)), & \text{if } (i,j) \in \mathcal{E}(\mathcal{G}), \\ 0, & \text{otherwise.} \end{cases}$$

We denote as $\omega_{ij}(t)$ the weight coefficient of the partial derivative of ν_{ij} with respect to y_i as above, i.e.,

$$\omega_{ij}(t) = \begin{cases} \frac{2\Delta-2\|d_{ij}\|-\|y_i(t)-y_j(t)\|}{(\Delta-\|d_{ij}\|-\|y_i(t)-y_j(t)\|)^2}, & \text{if } (i,j) \in \mathcal{E}(\mathcal{G}), \\ 0, & \text{otherwise.} \end{cases}$$

Note that $\omega_{ij}(t)$ can also be written as a function of $x_i(t)$ and $x_j(t)$ due to $y_i(t) - y_j(t) = x_i(t) - x_j(t) - d_{ij}$.

Let L_ω denotes the Laplacian matrix associated with $\mathcal{G}$ after assigning the above weight $\omega_{ij}(t)$ to edge $(i, j) \in \mathcal{E}(\mathcal{G})$. Then, from Lemma 2.6, we have

$$L_\omega = B(\mathcal{G})\Omega(\mathcal{G})B(\mathcal{G})^\top,$$

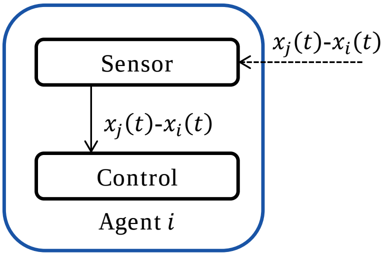
where $\Omega(\mathcal{G}) = \mathrm{Diag}([\omega(e_1), \cdots, \omega(e_{n_e})])$, where $\omega(e_k) = \omega_{ij}$ with e_k being the label of edge (i, j).

In order to reduce the overall need of communication and system updates, we use the event-triggered control input

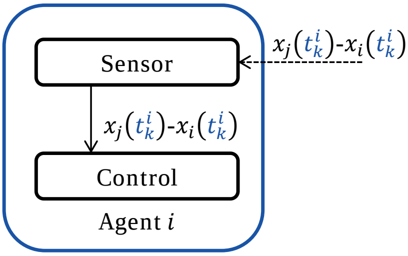
$$u_i(t) = \sum_{j \in N_i} -\omega_{ij}(t^i_{k_i(t)})(y_i(t^i_{k_i(t)}) - y_j(t^i_{k_i(t)})) \tag{10.3}$$

$$= \sum_{j \in N_i} -\omega_{ij}(t^i_{k_i(t)})(x_i(t^i_{k_i(t)}) - x_j(t^i_{k_i(t)}) - d_{ij}). \tag{10.4}$$

One can see that the above control input uses relative state information and only updates at the triggering times. Figure 10.2 illustrates how one agent gathers relative state information. Specifically, Figure 10.2 (a) shows that each agent continuously sense the relative state information between itself and its neighbors and use such information to generate its control input. Figure 10.2 (b) shows a similar process except that each agent only senses the relative state information at discrete time instants $\{t^i_1, t^i_2, \dots\}$.



(a) Continuous-time control input.



(b) Event-triggered control input.

Figure 10.2: Illustration of how one agent gathers relative state information.

Remark 10.1. *The control input (10.4) is constant during each interval $[t_k^i, t_{k+1}^i)$. In other words, the control input (10.4) of each agent is not affected by its neighbors during $[t_k^i, t_{k+1}^i)$. On the contrary, the control input (1.10) is not necessarily a constant during $[t_k^i, t_{k+1}^i)$ since $x_j(t^j_{k_j(t)})$ normally is not a constant for all $t \in [t_k^i, t_{k+1}^i)$. In other words, the control input (1.10) of each agent is affected by its neighbors during each interval $[t_k^i, t_{k+1}^i)$. Another difference between (10.4) and (1.10) is that the (weighted) summation of the control input (1.10) is zero, which does not present in (10.4).*

10.3.1 Distributed event-triggered formation control algorithm

In the following theorem, we will give triggering laws to determine the triggering times such that the formation with connectivity preservation can be established and Zeno behavior can be excluded.

Theorem 10.1. *Given a graph $\mathcal{G}$ which is undirected and connected, and a desired formation associated with $\mathcal{G}$ which satisfies Assumption 10.1. Consider the multi-agent system (10.1) with event-triggered control input (10.4) associated with $\mathcal{G}$. Assume that at the initial time,*

$$\|x_i(0) - x_j(0) - d_{ij}\| = \|y_i(0) - y_j(0)\| < \Delta - \|d_{ij}\|, \ \forall (i, j) \in \mathcal{E}(\mathcal{G}). \tag{10.5}$$

Given $\alpha > 0$, $0 < \beta < \beta_0$ with $\beta_0 = \frac{\rho_2(B(\mathcal{G})B(\mathcal{G})^\top)}{\Delta_0}$ and $\Delta_0 = \max_{(i,j)\in\mathcal{E}(\mathcal{G})} \Delta - \|d_{ij}\|$, and given the first triggering time $t_1^i = 0$, agent i determines the triggering times $\{t_k^i\}_{k=2}^\infty$ by

$$t_{k+1}^i = \min\{t : \ \|e_i(t)\| \geq \alpha e^{-\beta t}, \ t \geq t_k^i\}, \tag{10.6}$$

where

$$e_i(t) = \sum_{j\in N_i} \omega_{ij}(t)(x_i(t) - x_j(t) - d_{ij}) - \sum_{j\in N_i} \omega_{ij}(t^i_{k_i(t)})(x_i(t^i_{k_i(t)}) - x_j(t^i_{k_i(t)}) - d_{ij}).$$

Then,

Algorithm 10.1 Distributed Event-Triggered Formation Control Algorithm for Single Integrators

1: Choose $\alpha > 0$ and $0 < \beta < \beta_0$;
2: Agent $i \in [n]$ sends $\{d_{ij},\ (i, j) \in \mathcal{E}(\mathcal{G})\}$ to its neighbors;
3: Agent i initializes $t_1^i = 0$ and $k = 1$;
4: At time $s = t_k^i$, agent i senses the relative position $x_i(s) - x_j(s)$ and predicts future relative position $x_i(t) - x_j(t),\ t \geq s,\ \forall j \in \mathcal{N}_i$ by (10.7);
5: Agent i substitutes these relative positions into $e_i(t)$ and finds out τ_{k+1}^i which is the smallest solution to equation $\|e_i(t)\| = \alpha e^{-\beta t},\ t \geq s$;
6: Agent i continuously listens to whether there is broadcasting from its neighbors and receives the broadcasted information if it occurs;
7: **if** there is broadcasting from its neighbors at $t_0 \in (s, \tau_{k+1}^i)$, i.e., there exists $j \in \mathcal{N}_i$ such that agent j broadcasts its triggering information at $t_0 \in (s, \tau_{k+1}^i)$[1] **then**
8: agent i receives information at t_0, and updates $s = t_0$, and goes back to Step 4;
9: **else**
10: agent i determines $t_{k+1}^i = \tau_{k+1}^i$, and updates its control input $u_i(t_{k+1}^i)$ by sensing the relative positions to its neighbors, and broadcasts its triggering information $\{t_{k+1}^i,\ u_i(t_{k+1}^i)\}$ to its neighbors, and resets $k = k + 1$, and goes back to Step 4;
11: **end if**

(i) $\|x_i(t) - x_j(t)\| \leq \Delta,\ \forall (i, j) \in \mathcal{E}(\mathcal{G}),\ \forall t \geq 0$;

(ii) $\lim_{t\to\infty}(x_i(t) - x_j(t)) = d_{ij},\ \forall (i, j) \in \mathcal{E}(\mathcal{G})$, *exponentially;*

(iii) *there exists a constant* $\epsilon_i > 0$*, such that* $t_{k+1}^i - t_k^i \geq \epsilon_i,\ \forall i \in [n],\ \forall k \in \mathbb{N}_+$.

Proof. The proof is given in Section 10.7.1. □

Apparently, in order to monitor the inequality in the triggering law (10.6), each agent needs to continuously sense the relative positions to its neighbors. This may be a drawback. In the following we will give an event-triggered algorithm to avoid this. In other words, the following algorithm is an implementation of Theorem 10.1, but it only requires agents to sense, broadcast and receive at the triggering times. The idea is illustrated as follows.

Each agent $i \in [n]$, at any time $s \geq 0$, knows its last triggering time $t_{k_i(s)}^i$ and its control input $u_i(s) = u_i(t_{k_i(s)}^i)$ which is a constant until it determines its next triggering time. If agent i also knows the relative position $x_i(s) - x_j(s)$ and $u_j(s) = u_j(t_{k_j(s)}^j)$ which is a constant until agent j determines its next triggering time, for $j \in \mathcal{N}_i$, then agent i can predict

$$x_i(t) - x_j(t) = x_i(s) - x_j(s) + (t - s)(u_i(t_{k_i(s)}^i) - u_j(t_{k_j(s)}^j)),\ t \geq s, \tag{10.7}$$

until $t \leq \min\{t_{k_i(s)+1}^i,\ t_{k_j(s)+1}^j\}$. This means continuous sensing, broadcasting and receiving are not needed any more. The above implement idea is summarized in Algorithm 10.1.

[1]This kind of situation can only occur at most finite times during (s, τ_{k+1}^i) since $|\mathcal{N}_i|$ is finite and there is no Zeno behavior.

Remark 10.2. *In order to implement Algorithm 10.1, β_0 should be known first. However β_0 is a global parameter since it relates to $\rho_2(B(\mathcal{G})B(\mathcal{G})^\top)$ and Δ_0. We can lower bound β_0 by $\frac{4}{n(n-1)\Delta}$ due to $\Delta_0 < \Delta$ and $\rho_2(B(\mathcal{G})B(\mathcal{G})^\top) \geq \frac{4}{n(n-1)}$, see [405].*

10.3.2 Distributed self-triggered formation control algorithms

When applying Algorithm 10.1, although continuous broadcasting and sensing are avoided, each agent still needs to continuously listen to incoming information from its neighbors since the triggering times are determined during runtime and not known in advance. If every agent $i \in [n]$, at its current triggering time t^i_k, can predict its next triggering time t^i_{k+1} and broadcast it to its neighbors, then at time t^i_k agent i knows agent j's latest triggering time $t^j_{k_j(t^i_k)}$ which is before t^i_k and its next triggering time $t^j_{k_j(t^i_k)+1}$ which is after t^i_k, for $j \in \mathcal{N}_i$. In this case, agent i only needs to listen to and receive information at $\{t^j\}_{k=1}^{\infty}$, $j \in \mathcal{N}_i$ since it knows these time instants in advance. Thus, each agent only needs to sense and broadcast at its own triggering times, and to listen to and receive the incoming information from its neighbors at their triggering times. Inspired by this, in the following we will propose two self-triggered algorithms such that at time t^i_k each agent i could estimate t^i_{k+1} in a more precise way than $t^i_k + \epsilon_i$. The idea is explained below.

From (10.44) and (10.50), we have

$$\|y_i(t) - y_j(t)\| < \hat{k}_{ij}(t),\ \forall (i,j) \in \mathcal{E}(\mathcal{G}),\ \forall t \geq 0, \tag{10.8}$$

where

$$\hat{k}_{ij}(t) = \min\{k_{ij},\ 2\sqrt{k_V} e^{-\beta t}\}.$$

Then, from (10.53), we have

$$\|u_i(t)\| = \|\dot{y}_i(t)\| \leq \theta_i(t),\ \forall i \in [n],\ \forall t \geq 0, \tag{10.9}$$

where

$$\theta_i(t) = \alpha e^{-\beta t} + \sum_{j \in N_i} f_{ij}(\hat{k}_{ij}(t))\hat{k}_{ij}(t).$$

From (10.2), we have $\dot{y}_i(t) - \dot{y}_j(t) = u_i(t) - u_j(t)$. Then,

$$y_i(t) - y_j(t) = y_i(t^i_k) - y_j(t^i_k) + \int_{t^i_k}^{t} (u_i(s) - u_j(s))ds,\ t \geq t^i_k.$$

Agent i can determine $y_i(t^i_k) - y_j(t^i_k) = x_i(t^i_k) - x_j(t^i_k) - d_{ij}$ for $j \in \mathcal{N}_i$ by sensing the relative position to its neighbors at time t^i_k.

The control input $u_i(s)$ is a constant during $[t^i_k, t^i_{k+1})$ and $u_j(s)$ is a constant during $[t^j_{k_j(t^i_k)}, t^j_{k_j(t^i_k)+1})$. At time t^i_k, agent i already knows $t^j_{k_j(t^i_k)}$ and $u_j(t^j_{k_j(t^i_k)})$, for $j \in \mathcal{N}_i$. If at time t^i_k, agent i also knows $t^j_{k_j(t^i_k)+1}$, then at time t^i_k it knows $u_j(s) \equiv u_j(t^j_{k_j(t^i_k)})$, for $s \in [t^i_k, t^j_{k_j(t^i_k)+1})$. In other words, same as (8.32), for $t \in [t^i_k, t^i_{k+1})$, if denote

$$t^1_{ij}(t) = \min\left\{t,\ t^j_{k_j(t^i_k)+1}\right\},\ t^2_{ij}(t) = \max\left\{t,\ t^j_{k_j(t^i_k)+1}\right\}, \tag{10.10}$$

then at time t^i_k, agent i knows $u_j(s) \equiv u_j(t^j_{k_j(t^i_k)})$, for $s \in [t^i_k, t^1_{ij}(t))$ but does not know $u_j(s)$, for $s \geq t^2_{ij}(t)$. Figure 8.1 illustrates the relation of t^i_k, t^i_{k+1}, $t \in [t^i_k, t^i_{k+1})$, $t^j_{k_j(t^i_k)}$, $t^j_{k_j(t^i_k)+1}$, $t^1_{ij}(t)$ and $t^2_{ij}(t)$. Then,

$$y_i(t) - y_j(t) = z_{ij}(t^i_k, t) - \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(t)} u_j(s)ds, \ \forall (i,j) \in \mathcal{E}(\mathcal{G}), \ t \in [t^i_k, t^i_{k+1}), \tag{10.11}$$

where

$$z_{ij}(t^i_k, t) = y_i(t^i_k) - y_j(t^i_k) + (t - t^i_k)u_i(t^i_k) - (t^1_{ij}(t) - t^i_k)u_j(t^j_{k_j(t^i_k)}).$$

Thus

$$\|y_i(t) - y_j(t)\| \leq \|z_{ij}(t^i_k, t)\| + \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(t)} \|u_j(s)\|ds, \ \forall (i,j) \in \mathcal{E}(\mathcal{G}), \ t \in [t^i_k, t^i_{k+1}).$$

Then, from (10.9), we have

$$\|y_i(t) - y_j(t)\| \leq \check{k}_{ij}(t), \ \forall (i,j) \in \mathcal{E}(\mathcal{G}), \ t \in [t^i_k, t^i_{k+1}), \tag{10.12}$$

where

$$\check{k}_{ij}(t) = \|z_{ij}(t^i_k, t)\| + \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(t)} \theta_j(s)ds, \ \forall (i,j) \in \mathcal{E}(\mathcal{G}), \ t \in [t^i_k, t^i_{k+1}).$$

Then, from (10.8) and (10.12), we have

$$\|y_i(t) - y_j(t)\| \leq \tilde{k}_{ij}(t), \ \forall (i,j) \in \mathcal{E}(\mathcal{G}), \ t \in [t^i_k, t^i_{k+1}), \tag{10.13}$$

where

$$\tilde{k}_{ij}(t) = \min\{\hat{k}_{ij}(t), \ \check{k}_{ij}(t)\}, \ t \in [t^i_k, t^i_{k+1}). \tag{10.14}$$

Thus, from (10.55), (10.56), (10.9), (10.11) and (10.13), we have

$$\|e_i(t)\| \leq \varphi_i(t), \ t \in [t^i_k, t^i_{k+1}),$$

where

$$\begin{aligned}\varphi_i(t) = \Big\| \sum_{j \in N_i} \int_{t^i_k}^{t^1_{ij}(t)} \Big(& h_{ij}(\|z_{ij}(t^i_k, s)\|) \frac{(z_{ij}(t^i_k, s))^\top}{\|z_{ij}(t^i_k, s)\|} \Big(u_i(t^i_k) - u_j\Big(t^j_{k_j(t^i_k)}\Big)\Big) z_{ij}(t^i_k, s) \\ & + f_{ij}(\|z_{ij}(t^i_k, s)\|) \Big(u_i(t^i_k) - u_j\Big(t^j_{k_j(t^i_k)}\Big)\Big)\Big) ds \Big\|\end{aligned}$$

Algorithm 10.2 Distributed Self-Triggered Formation Control Algorithm for Single Integrators

1: Choose $\alpha > 0$ and $0 < \beta < \beta_0$;
2: Agent $i \in [n]$ sends $\{d_{ij},\ (i, j) \in \mathcal{E}(\mathcal{G})\}$ to its neighbors;
3: Agent i initializes $t_1^i = 0$ and $k = 1$;
4: At time $s = t_k^i$, agent i updates its control input $u_i(t_k^i)$ by sensing the relative positions to its neighbors, and determines t_{k+1}^i by (10.16)[1], and broadcasts its triggering information $\{t_{k+1}^i,\ u_i(t_k^i)\}$ to its neighbors;
5: At agent i's neighbors' triggering times which are between $[t_k^i, t_{k+1}^i]$, agent i listens to and receives triggering information from its neighbors[2];
6: Agent i resets $k = k + 1$, and goes back to Step 4.

$$
\begin{aligned}
&+\sum_{j\in N_i}\int_{t_{ij}^1(t)}^{t} g_{ij}(\tilde{k}_{ij}(s))\|u_i(t_k^i)\|ds+\sum_{j\in N_i}\int_{t^j_{k_j(t_k^i)+1}}^{t_{ij}^2(t)} g_{ij}(\tilde{k}_{ij}(s))\theta_j(s)ds\\
=&\left\|\sum_{j\in N_i}\Big(f_{ij}(\|z_{ij}(t_k^i,t_{ij}^1(t))\|)z_{ij}(t_k^i,t_{ij}^1(t))-f_{ij}(\|z_{ij}(t_k^i,t_k^i)\|)z_{ij}(t_k^i,t_k^i)\Big)\right\|\\
&+\sum_{j\in N_i}\int_{t_{ij}^1(t)}^{t} g_{ij}(\tilde{k}_{ij}(s))\|u_i(t_k^i)\|ds+\sum_{j\in N_i}\int_{t^j_{k_j(t_k^i)+1}}^{t_{ij}^2(t)} g_{ij}(\tilde{k}_{ij}(s))\theta_j(s)ds,\ t\in[t_k^i,t_{k+1}^i).
\end{aligned}
\tag{10.15}
$$

Hence, a necessary condition to guarantee that the inequality in (10.6) holds, i.e.,

$$\alpha e^{-\beta t} \le \|e_i(t)\|,\ \forall t \in [t_k^i, t_{k+1}^i),$$

is

$$\alpha e^{-\beta t} \le \varphi_i(t),\ \forall t \in [t_k^i, t_{k+1}^i).$$

Noting that $\alpha e^{-\beta t}$ decreases with respect to t, $\varphi_i(t)$ increases with respect to t during $[t_k^i, t_{k+1}^i)$, and $\varphi_i(t_k^i) = 0$, for given t_k^i, agent i can estimate t_{k+1}^i by the solution to

$$\alpha e^{-\beta t} = \varphi_i(t),\ t \ge t_k^i. \tag{10.16}$$

In conclusion, if at time t_k^i agent i knows $u_i(t_k^i)$, $t^j_{k_j(t_k^i)}$, $t^j_{k_j(t_k^i)+1}$, $u_j(t^j_{k_j(t_k^i)})$, $\forall j \in \mathcal{N}_i$, then it can predict its next triggering time t_{k+1}^i by solving (10.16). The above implement idea is summarized in Algorithm 10.2.

[1] Agent i uses $t^j_{k_j(t_k^i)}$ to replace $t^j_{k_j(t_k^i)+1}$ to determine t_{k+1}^i by (10.16) when $t_k^i = t^j_{k_j(t_k^i)}$, i.e., when agent i does not know $t^j_{k_j(t_k^i)+1}$ at time t_k^i. This situation could occur, for example when two adjacent agents trigger at the same time.

[2] In other words, agent i only listens to incoming information at its neighbors' triggering times. Thus continuous listening is avoided. This is the main difference with Algorithm 10.1.

Algorithm 10.3 Distributed Self-Triggered Formation Control Algorithm for Single Integrators (Sensing Only)

1: Choose $\alpha > 0$ and $0 < \beta < \beta_0$;
2: Agent $i \in [n]$ sends $\{d_{ij},\ (i, j) \in \mathcal{E}(\mathcal{G})\}$ to its neighbors;
3: Agent i initializes $t_1^i = 0$ and $k = 1$;
4: At time $s = t_k^i$, agent i updates its control input $u_i(t_k^i)$ by sensing the relative positions to its neighbors, and determines t_{k+1}^i by (10.18), and resets $k = k + 1$, and repeats this step.

Actually, broadcasting, receiving and listening can be ruled out except at the beginning, and each agent only needs to sense the relative positions to its neighbors and update its control input at its triggering times. The idea is illustrated as follows.

From (10.8), (10.53) and (10.57), we have

$$\frac{d\|e_i(t)\|}{dt} < \hat{c}_i(t), \ \forall t \geq 0, \tag{10.17}$$

where

$$\hat{c}_i(t) = \sum_{j \in N_i} g_{ij}(\hat{k}_{ij}(t))\Big(2\alpha + \sum_{l \in N_i} f_{il}(\hat{k}_{il}(t))\hat{k}_{il}(t) + \sum_{l \in N_j} f_{jl}(\hat{k}_{jl}(t))\hat{k}_{jl}(t)\Big).$$

Then, similar to the way to determine ϵ_i in (10.60), if t_k^i is known, then agent i can estimate t_{k+1}^i by

$$\int_{t_k^i}^{t_{k+1}^i} \hat{c}_i(t)dt = \alpha e^{-\beta t_{k+1}^i}. \tag{10.18}$$

The above implement idea is summarized in Algorithm 10.3.

The following theorem shows that the formation with connectivity preservation can be established and Zeno behavior can be excluded.

Theorem 10.2. *Under the same settings as Theorem 10.1. All agents perform Algorithm 10.2 or 10.3, then the multi-agent system (10.1) with event-triggered control input (10.4) converges to the formation exponentially with connectivity preservation, and there is no Zeno behavior.*

Proof. Under both Algorithms 10.2 and 10.3, $\|e_i(t)\| \leq \alpha e^{-\beta t}$ holds for all $i \in [n]$ and $t \geq 0$. Then from Theorem 10.1, we know that the formation is achieved exponentially and the connectivity is preserved. The method of the exclusion of Zeno behavior is similar to the way in the proof of Theorem 10.1. □

Remark 10.3. *In order to perform Algorithms 10.2 and 10.3, the global parameters n, β_0, k_v defined in (10.43) and k_V defined in (10.49) are needed to be known in advance. Firstly,* *from Remark 10.2, we can estimate β_0 by $\frac{4}{n^2\Delta}$. Secondly, one way to avoid using k_v is by choosing an arbitrary small $\varepsilon > 0$. Then, from (10.45), we have*

$$\hat{k}_{ij}(\varepsilon) := \Delta - \|d_{ij}\| - \varepsilon \geq k_{ij}.$$

Thus, $\hat{k}_{ij}(\varepsilon)$ can be used to replace k_{ij} since $f_{ij}(\cdot)$ defined in (10.47) and $g_{ij}(\cdot)$ defined in (10.51) are increasing functions. Thirdly, k_V can be estimated if we know the upper bound of $V(y(0))$ defined in (10.46). From the underlying graph $\mathcal{G}$ is connected, we have $\|y_i(0) - y_j(0)\| < (n-1)\Delta$, $\forall i, j \in [n]$. Then $\|y_i(0) - \bar{y}(0)\| < \Delta$, $\forall i \in [n]$. Hence $V(y(0)) < \frac{1}{2}n\Delta^2$. Thus, the only global parameter that is needed to perform Algorithms 10.2 and 10.3 is the number of agents n.

The comparison of the inter-event times determined by Algorithms 10.1– 10.3 is shown as below.

Property 10.1. *Consider the multi-agent system (10.1) with event-triggered control input (10.4). For agent i, assume t_k^i has been determined, let $t_{k+1}^{i,E1}$, $t_{k+1}^{i,S1}$ and $t_{k+1}^{i,S2}$ be the next triggering time determined by Algorithms 10.1–10.3 respectively, then $t_{k+1}^{i,E1} \geq t_{k+1}^{i,S2} \geq t_k^i + \epsilon_i$ and $t_{k+1}^{i,S1} \geq t_{k+1}^{i,S2} \geq t_k^i + \epsilon_i$.*

Proof. From (10.59) and (10.17), we know $c_i e^{-\beta t} \geq \hat{c}_i(t), \forall t \geq 0$ since (10.8), and $f_{ij}(\cdot)$ defined in (10.47) and $g_{ij}(\cdot)$ defined in (10.51) are increasing functions. Thus $t_{k+1}^{i,S2} \geq t_k^i + \epsilon_i$.

From (10.15) and (10.17), we know $\varphi_i(t) \leq \int_{t_k^i}^t \hat{c}_i(s)ds$, for $t \geq t_k^i$ since (10.14), and $f_{ij}(\cdot)$ and $g_{ij}(\cdot)$ are increasing functions. Thus $t_{k+1}^{i,S1} \geq t_{k+1}^{i,S2}$.

From (10.17), we know $t_{k+1}^{i,E1} \geq t_{k+1}^{i,S2}$. □

Remark 10.4. *Property 10.1 has to be considered carefully, since it only shows that for given t_k^i, the next triggering time determined by Algorithm 10.1 or 10.2 is larger than that determined by Algorithm 10.3. However, we cannot say anything on further triggering times because generally $t_{k+1}^{i,E1} \neq t_{k+1}^{i,S2}$ and $t_{k+1}^{i,S1} \neq t_{k+1}^{i,S2}$, and thus we cannot apply this property again. Moreover, we cannot to compare $t_{k+1}^{i,E1}$ and $t_{k+1}^{i,S1}$ since $u_j(\cdot)$ are different when we perform Algorithms 10.1 and 10.2.*

Table 10.2 summarizes the communication requirements for agent $i \in [n]$ when Algorithms 10.1–10.3 are performed.

10.4 Distributed event-triggered formation control for double integrators

In this section, we extend the results in above section to the case where the dynamics of agents are modeled as double integrators

$$\begin{cases} \dot{x}_i(t) = r_i(t), \\ \dot{r}_i(t) = u_i^d(t), \ i \in [n], \ t \geq 0, \end{cases} \tag{10.19}$$

Table 10.2: Summary of the communication requirements for agent i when Algorithms 10.1–10.3 are performed.

	Algorithm 10.1	Algorithm 10.2	Algorithm 10.3
Sensing time	$\{t_k^i, t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$	$\{t_k^i\}_{k=1}^{\infty}$	$\{t_k^i\}_{k=1}^{\infty}$
Broadcasting time	$\{t_k^i\}_{k=1}^{\infty}$	$\{t_k^i\}_{k=1}^{\infty}$	$t_1^i = 0$
Listening time	All $t \geq 0$	$\{t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$	$t_1^i = 0$
Receiving time	$\{t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$	$\{t_k^j, j \in \mathcal{N}_i\}_{k=1}^{\infty}$	$t_1^i = 0$
Information sensed	$\{x_i(t_k^i) - x_j(t_k^i)\}_{k=1}^{\infty}$, $\{x_i(t_k^j) - x_j(t_k^j), j \in \mathcal{N}_i\}_{k=1}^{\infty}$	$\{x_i(t_k^i) - x_j(t_k^i)\}_{k=1}^{\infty}$	$\{x_i(t_k^i) - x_j(t_k^i)\}_{k=1}^{\infty}$
Information broadcasted	$\{t_k^i, u_i(t_k^i)\}_{k=1}^{\infty}, d_{ij}, j \in \mathcal{N}_i$	$\{t_k^i, u_i(t_k^i)\}_{k=1}^{\infty}, d_{ij}, j \in \mathcal{N}_i$	$d_{ij}, j \in \mathcal{N}_i$
Zeno behavior	No	No	No

where $x_i(t) \in \mathbb{R}^p$ still denotes the position of agent i at time t, $r_i(t) \in \mathbb{R}^p$ denotes the speed and $u_i^d(t) \in \mathbb{R}^p$ is the control input. Recall that $y_i(t) = x_i(t) - \tau_i$, so we can rewrite (10.19) as

$$\begin{cases} \dot{y}_i(t) = r_i(t), \\ \dot{r}_i(t) = u_i^d(t), \ i \in [n], \ t \geq 0. \end{cases} \tag{10.20}$$

Denote

$$B_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ z_i(t) = \begin{bmatrix} y_i(t) \\ r_i(t) \end{bmatrix},$$

then we can rewrite (10.20) as

$$\dot{z}_i(t) = (B_1 \otimes I_p)z_i(t) + (B_2 \otimes I_p)u_i^d(t). \tag{10.21}$$

It can be derived that (B_1, B_2) is controllable and (I_2, B_1) is observable. Hence, from [406], we know that there exist positive constants k_0, k_1 and k_2 such that

$$P > 0, \ \frac{1}{2}(PB_1 + B_1^\top P) - \beta_1 PB_2B_2^\top P + 2I_2 \leq 0, \tag{10.22}$$

where $P = \begin{bmatrix} k_0 & k_1 \\ k_1 & k_2 \end{bmatrix}$ and $0 < \beta_1 \leq \beta_0$. Similar to (2.6), we have

$$\rho(P)I_2 \geq P \geq \rho_2(P)I_2. \tag{10.23}$$

Similar to the event-triggered control input (10.4), we use the event-triggered control input

$$u_i^d(t) = -k_1 \sum_{j \in N_i} \omega_{ij}(t_{k_i(t)}^i)(y_i(t_{k_i(t)}^i) - y_j(t_{k_i(t)}^i))$$

$$- k_2 \sum_{j\in N_i} \omega_{ij}(t^i_{k_i(t)})(r_i(t^i_{k_i(t)}) - r_j(t^i_{k_i(t)})) - k_3 r_i(t^i_{k_i(t)}) \tag{10.24}$$

$$= -k_1 \sum_{j\in N_i} \omega_{ij}(t^i_{k_i(t)})(x_i(t^i_{k_i(t)}) - x_j(t^i_{k_i(t)}) - d_{ij})$$

$$- k_2 \sum_{j\in N_i} \omega_{ij}(t^i_{k_i(t)})(r_i(t^i_{k_i(t)}) - r_j(t^i_{k_i(t)})) - k_3 r_i(t^i_{k_i(t)}), \tag{10.25}$$

where k_3 is a constant which will be determined later. Here we should highlight that this control input needs absolute speed information because of the term $k_3 r_i(t^i_{k_i(t)})$. Later we will show that no agent needs to sense absolute speed if each agent knows its initial speed.

10.4.1 Distributed event-triggered formation control algorithm

Similar to Theorem 10.1, we know that the multi-agent system (10.19) with event-triggered control input (10.25) converges to the formation exponentially with connectivity preservation, and there is no Zeno behavior as stated in the following theorem.

Theorem 10.3. *Given a graph $\mathcal{G}$ which is undirected and connected, and a desired formation associated with $\mathcal{G}$ which satisfies Assumption 10.1. Given $0 < \beta_1 \leq \beta_0$ with β_0 defined in Theorem 10.1, determine the matrix P by (10.22). Consider the multi-agent system (10.19) with event-triggered control input (10.25) associated with $\mathcal{G}$. Assume the initial position satisfies (10.5) for all $(i, j) \in \mathcal{E}(\mathcal{G})$ and every agent knows its initial speed*[1]*. Given $0 < k_3 < 4/(k_2 + (k_1^2 + k_2^2)^{1/2})$, $\alpha_d > 0$, $0 < \beta_d < (2 - k_4)/\rho(P)$ with $k_4 = k_3(k_2 + (k_1^2 + k_2^2)^{1/2})/2 < 2$, and the first triggering time $t^i_1 = 0$, agent i determines the triggering times $\{t^i_k\}_{k=2}^{\infty}$ by*

$$t^i_{k+1} = \min\{t : \ \|E_i(t)\| \geq \alpha_d e^{-\beta_d t}, \ t \geq t^i_k\}, \tag{10.26}$$

where

$$E_i(t) = k_1 e_i(t) + k_2 e^r_i(t) + k_3(r_i(t) - r_i(t^i_{k_i(t)})),$$
$$e^r_i(t) = \sum_{j\in N_i} \omega_{ij}(t)(r_i(t) - r_j(t)) - \sum_{j\in N_i} \omega_{ij}(t^i_{k_i(t)})(r_i(t^i_{k_i(t)}) - r_j(t^i_{k_i(t)})),$$

and $e_i(t)$ is given in Theorem 10.1. Then,

(i) $\|x_i(t) - x_j(t)\| \leq \Delta, \ \forall (i, j) \in \mathcal{E}(\mathcal{G}), \ \forall t \geq 0$*;*

(ii) $\lim_{t\to\infty}(x_i(t) - x_j(t)) = d_{ij}, \ \forall (i, j) \in \mathcal{E}(\mathcal{G})$*, exponentially;*

(iii) *there is no Zeno behavior.*

Proof. The proof is given in Section 10.7.2. □

[1] In real applications, initial speed normally is zero.

Similar to the analysis after Theorem 10.1, in order to monitor the inequality in the triggering law (10.26), each agent needs to continuously sense its absolute speed, the relative positions and speeds to its neighbors. In the following we will give an event-triggered algorithm to implement Theorem 10.3 and at the same time to avoid continuous sensing by using the similar idea as Algorithm 10.1.

Noting that it is assumed that every agent knows its initial speed, each agent $i \in [n]$ knows $\{r_i(t_k^i)\}_{k=1}^{\infty}$ through iterative computation as follows

$$r_i(t_{k+1}^i) = r_i(t_k^i) + (t_{k+1}^i - t_k^i)u_i^d(t_k^i). \tag{10.27}$$

Thus, at any time $s \geq 0$, agent i can predict

$$r_i(t) = r_i(t_{k_i(s)}^i) + (t - t_{k_i(s)}^i)u_i^d(t_{k_i(s)}^i), \ \forall t \geq s. \tag{10.28}$$

This means that no agent needs to sense absolute speed.

Each agent $i \in [n]$, at any time $s \geq 0$, knows its last triggering time $t_{k_i(s)}^i$ and control input $u_i^d(t_{k_i(s)}^i)$ which is a constant until it determines its next triggering time. If agent i also knows the relative position $x_i(s) - x_j(s)$, relative speed $r_i(s) - r_j(s)$ and $u_j^d(s) = u_j^d(t_{k_j(s)}^j)$ which is a constant until agent j determines its next triggering time, for $j \in \mathcal{N}_i$, then agent i can predict

$$x_i(t) - x_j(t) = x_i(s) - x_j(s) + (t - s)(r_i(s) - r_j(s)) + \frac{1}{2}(t - s)^2(u_i^d(t_{k_i(s)}^i) - u_j^d(t_{k_j(s)}^j)), \tag{10.29}$$

$$r_i(t) - r_j(t) = r_i(s) - r_j(s) + (t - s)(u_i^d(t_{k_i(s)}^i) - u_j^d(t_{k_j(s)}^j)), \ t \geq s, \tag{10.30}$$

until $t \leq \min\{t_{k_i(s)+1}^i, \ t_{k_j(s)+1}^j\}$. This means that continuous sensing, broadcasting and receiving are not needed any more.

The above implement idea is summarized in Algorithm 10.4.

10.4.2 Distributed self-triggered formation control algorithms

As noted earlier, each agent still needs to continuously listen to incoming information from its neighbors. In order to avoid this, in the following we will first give a self-triggered algorithm which is similar to Algorithm 10.2 such that each agent only needs to listen at its neighbors' triggering times. Then, we will give another self-triggered algorithm which is similar to Algorithm 10.3 such that broadcasting, receiving, and listening only occur at the beginning.

From (10.61) and (10.64), we have

$$\|y_i(t) - y_j(t)\| < \hat{k}_{ij}^y(t), \ \forall (i, j) \in \mathcal{E}(\mathcal{G}), \ \forall t \geq 0, \tag{10.31}$$

where

$$\hat{k}_{ij}^y(t) = \min\left\{k_{ij}^d, \ \sqrt{k_V^d} e^{-\beta_d t}\right\}.$$

Algorithm 10.4 Distributed Event-Triggered Formation Control Algorithm for Double Integrators

1: Choose $0 < \beta_1 \leq \beta_0$ and determine P by (10.22);
2: Choose $0 < k_3 < \frac{4}{k_2 + \sqrt{k_1^2 + k_2^2}}$, $\alpha_d > 0$ and $0 < \beta_d < \frac{2 - k_4}{\rho(P)}$;
3: Agent $i \in [n]$ sends $\{d_{ij},\ (i, j) \in \mathcal{E}(\mathcal{G})\}$ to its neighbors;
4: Agent i initializes $t_1^i = 0$ and $k = 1$;
5: At time $s = t_k^i$, agent i senses the relative position $x_i(s) - x_j(s)$ and relative speed $r_i(s) - r_j(s)$, and predicts future relative position $x_i(t) - x_j(t)$, future relative speed $r_i(t) - r_j(t)$, $\forall j \in \mathcal{N}_i$, and its future speed $r_i(t)$, $t \geq s$ by (10.29), (10.30) and (10.28), respectively;
6: Agent i substitutes these into $E_i(t)$ and finds out τ_{k+1}^i which is the smallest solution to equation $\|E_i(t)\| = \alpha_d e^{-\beta_d t}$, $t \geq s$;
7: Agent i continuously listens to whether there is broadcasting from its neighbors and receives the broadcasted information if it occurs;
8: **if** there is broadcasting from its neighbors at $t_0 \in (s, \tau_{k+1}^i)$, i.e., there exists $j \in \mathcal{N}_i$ such that agent j broadcasts its triggering information at $t_0 \in (s, \tau_{k+1}^i)$ **then**
9: agent i receives information at t_0, and updates $s = t_0$, and goes back to Step 5;
10: **else**
11: agent i determines $t_{k+1}^i = \tau_{k+1}^i$, and gets $r_i(t_{k+1}^i)$ by (10.27), and updates $u_i^d(t_{k+1}^i)$ by sensing the relative positions and speeds to its neighbors, and broadcasts its triggering information $\{t_{k+1}^i,\ u_i^d(t_{k+1}^i)\}$ to its neighbors, and resets $k = k + 1$, and goes back to Step 5;
12: **end if**

From (10.64) and (10.65), we have

$$\|r_i(t) - r_j(t)\| < \hat{k}_{ij}^r(t),\ \forall i, j \in [n],\ \forall t \geq 0. \tag{10.32}$$

where

$$\hat{k}_{ij}^r(t) = \min\Big\{\sqrt{k_V^d} e^{-\beta_d t},\ \|r_i(0)\| + \|r_j(0)\| + \frac{c_i^r + c_j^r}{\beta_d}\Big\}.$$

Then, similar to (10.68), we have

$$\|\dot{r}_i(t) - \dot{r}_j(t)\| = \|u_i^d(t) - u_j^d(t)\| < \theta_{ij}^d(t),\ \forall (i, j) \in \mathcal{E}(\mathcal{G}),\ \forall t \geq 0, \tag{10.33}$$

where

$$\begin{aligned}\theta_{ij}^d(t) = 2\alpha_d e^{-\beta_d t} &+ \sum_{l \in N_i} f_{il}(\hat{k}_{il}^y(t))\Big(k_1 \hat{k}_{il}^y(t) + k_2 \hat{k}_{il}^r(t)\Big)\\ &+ \sum_{l \in N_j} f_{jl}(\hat{k}_{jl}^y(t))\Big(k_1 \hat{k}_{jl}^y(t) + k_2 \hat{k}_{jl}^r(t)\Big) + k_3 \hat{k}_{ij}^r(t).\end{aligned}$$

Then, similar to (10.11), we have

$$r_i(t) - r_j(t) = z^r_{ij}(t^i_k, t) + \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(t)} (u^d_i(s) - u^d_j(s))ds,\ \forall (i, j) \in \mathcal{E}(\mathcal{G}),\ t \in [t^i_k, t^i_{k+1}), \tag{10.34}$$

where $t^1_{ij}(t)$ and $t^2_{ij}(t)$ defined in (10.10), and

$$z^r_{ij}(t^i_k, t) = r_i(t^i_k) - r_j(t^i_k) + (t^1_{ij}(t) - t^i_k)(u^d_i(t^i_k) - u^d_j(t^j_{k_j(t^i_k)})).$$

Thus

$$\|r_i(t) - r_j(t)\| \le \|z^r_{ij}(t^i_k, t)\| + \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(t)} \|u^d_i(s) - u^d_j(s)\|ds \le \check{k}^r_{ij}(t),\ t \in [t^i_k, t^i_{k+1}),$$

where

$$\check{k}^r_{ij}(t) = \|z^r_{ij}(t^i_k, t)\| + \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(t)} \theta^d_{ij}(s)ds,\ t \in [t^i_k, t^i_{k+1}). \tag{10.35}$$

Hence, then, from (10.32) and (10.35), we have

$$\|r_i(t) - r_j(t)\| \le \tilde{k}^r_{ij}(t),\ \forall (i, j) \in \mathcal{E}(\mathcal{G}),\ t \in [t^i_k, t^i_{k+1}), \tag{10.36}$$

where

$$\tilde{k}^r_{ij}(t) = \min\left\{\hat{k}^r_{ij}(t),\ \check{k}^r_{ij}(t)\right\},\ t \in [t^i_k, t^i_{k+1}).$$

From $\dot{y}_i(t) = r_i(t)$ and (10.34), we have

$$\begin{aligned} y_i(t) - y_j(t) &= y_i(t^i_k) - y_j(t^i_k) + \int_{t^i_k}^{t} [r_i(s) - r_j(s)]ds \\ &= z^y_{ij}(t^i_k, t) + \int_{t^i_k}^{t} \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(r)} (u^d_i(s) - u^d_j(s))dsdr,\ t \in [t^i_k, t^i_{k+1}), \end{aligned} \tag{10.37}$$

where

$$z^y_{ij}(t^i_k, t) = y_i(t^i_k) - y_j(t^i_k) + (t^1_{ij}(t) - t^i_k)(r_i(t^i_k) - r_j(t^i_k)) + \frac{1}{2}(t^1_{ij}(t) - t^i_k)^2(u^d_i(t^i_k) - u^d_j(t^j_{k_j(t^i_k)})).$$

Thus

$$\begin{aligned} \|y_i(t) - y_j(t)\| &\le \|z^y_{ij}(t^i_k, t)\| + \int_{t^i_k}^{t} \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(r)} \|u^d_i(s) - u^d_j(s)\|dsdr \\ &\le \check{k}^y_{ij}(t),\ \forall (i, j) \in \mathcal{E}(\mathcal{G}),\ t \in [t^i_k, t^i_{k+1}), \end{aligned} \tag{10.38}$$

where

$$\check{k}^y_{ij}(t) = \|z^y_{ij}(t^i_k, t)\| + \int_{t^i_k}^{t} \int_{t^j_{k_j(t^i_k)+1}}^{t^2_{ij}(r)} \theta^d_{ij}(s) ds dr.$$

Hence, then, from (10.31) and (10.38), we have

$$\|y_i(t) - y_j(t)\| \le \tilde{k}^y_{ij}(t),\ \forall (i,j) \in \mathcal{E}(\mathcal{G}),\ t \in [t^i_k, t^i_{k+1}), \tag{10.39}$$

where

$$\tilde{k}^y_{ij}(t) = \min\left\{\hat{k}^y_{ij}(t),\ \check{k}^y_{ij}(t)\right\},\ t \in [t^i_k, t^i_{k+1}).$$

Then from (10.69)–(10.72), (10.33), (10.34), (10.36), (10.37), and (10.39), we have

$$\|E_i(t)\| \le \varphi^d_i(t),\ t \in [t^i_k, t^i_{k+1}),$$

where

$$\begin{aligned}
\varphi^d_i(t) = {} & \Bigg\| \sum_{j\in N_i} \int_{t^i_k}^{t^1_{ij}(t)} \Bigg(k_1 h_{ij}(\|z^y_{ij}(t^i_k, s)\|) \frac{(z^y_{ij}(t^i_k, s))^\top}{\|z^y_{ij}(t^i_k, s)\|} z^r_{ij}(t^i_k, s) z^y_{ij}(t^i_k, s) \\
& + k_1 f_{ij}(\|z^y_{ij}(t^i_k, s)\|)(z^r_{ij}(t^i_k, s)) + k_2 h_{ij}(\|z^y_{ij}(t^i_k, s)\|) \frac{(z^y_{ij}(t^i_k, s))^\top}{\|z^y_{ij}(t^i_k, s)\|} z^r_{ij}(t^i_k, s) z^r_{ij}(t^i_k, s) \\
& + k_2 f_{ij}(\|z^y_{ij}(t^i_k, s)\|)\Big(u^d_i(t^i_k) - u^d_j\Big(t^j_{k_j(t^i_k)}\Big)\Big)\Bigg) ds + k_3(t - t^i_k) u^d_i(t^i_k) \Bigg\| \\
& + \sum_{j\in N_i} \int_{t^1_{ij}(t)}^{t} \Big(k_1 g_{ij}(\tilde{k}^y_{ij}(s)) \tilde{k}^r_{ij}(s) + k_2 h_{ij}(\tilde{k}^y_{ij}(s))(\tilde{k}^r_{ij}(s))^2 + k_2 f_{ij}(\tilde{k}^y_{ij}(s)) \theta^d_{ij}(s) \Big) ds \\
= {} & \Bigg\| \sum_{j\in N_i} \Big(f_{ij}(\|z^y_{ij}(t^i_k, t^1_{ij}(t))\|)(k_1 z^y_{ij}(t^i_k, t^1_{ij}(t)) + k_2 z^r_{ij}(t^i_k, t^1_{ij}(t))) \\
& - f_{ij}(\|z^y_{ij}(t^i_k, t^i_k)\|)(k_1 z^y_{ij}(t^i_k, t^i_k) + k_2 z^r_{ij}(t^i_k, t^i_k)) \Big) + k_3(t - t^i_k) u^d_i(t^i_k) \Bigg\| \\
& + \sum_{j\in N_i} \int_{t^1_{ij}(t)}^{t} \Big(k_1 g_{ij}(\tilde{k}^y_{ij}(s)) \tilde{k}^r_{ij}(s) + k_2 h_{ij}(\tilde{k}^y_{ij}(s))(\tilde{k}^r_{ij}(s))^2 \\
& + k_2 f_{ij}(\tilde{k}^y_{ij}(s)) \theta^d_{ij}(s) \Big) ds,\ t \in [t^i_k, t^i_{k+1}).
\end{aligned}$$

Hence, a necessary condition to guarantee that the inequality in (10.26) holds, i.e.,

$$\alpha_d e^{-\beta_d t} \le \|E_i(t)\|,\ \forall t \in [t^i_k, t^i_{k+1}),$$

is

$$\alpha_d e^{-\beta_d t} = \varphi^d_i(t),\ \forall t \in [t^i_k, t^i_{k+1}).$$

Algorithm 10.5 Distributed Self-Triggered Formation Control Algorithm for Double Integrators

1: Choose $0 < \beta_1 \leq \beta_0$ and determine P by (10.22);
2: Choose $0 < k_3 < \frac{4}{k_2+\sqrt{k_1^2+k_2^2}}$, $\alpha_d > 0$ and $0 < \beta_d < \frac{2-k_4}{\rho(P)}$;
3: Agent $i \in [n]$ sends $\{d_{ij},\ (i,j) \in \mathcal{E}(\mathcal{G}),\ r_i(0)\}$ to its neighbors;
4: Agent i initializes $t_1^i = 0$ and $k = 1$;
5: At time $s = t_k^i$, agent i gets $r_i(t_k^i)$ by (10.27), and updates $u_i^d(t_k^i)$ by sensing the relative positions and speeds to its neighbors, and determines t_{k+1}^i by (10.40)[1], and broadcasts its triggering information $\{t_{k+1}^i,\ u_i^d(t_k^i)\}$ to its neighbors;
6: At agent i's neighbors' triggering times which are between $[t_k^i, t_{k+1}^i]$, agent i listens to and receives triggering information from its neighbors[2];
7: Agent i resets $k = k + 1$, and goes back to Step 5.

Noting that $\alpha_d e^{-\beta_d t}$ decreases with respect to t, $\varphi_i^d(t)$ increases with respect to t during $[t_k^i, t_{k+1}^i)$, and $\varphi_i^d(t_k^i) = 0$, for given t_k^i, agent i can estimate t_{k+1}^i by the solution to

$$\alpha_d e^{-\beta_d t} = \varphi_i^d(t),\ t \geq t_k^i. \tag{10.40}$$

In other words, if at time t_k^i agent i knows $t^j_{k_j(t_k^i)}$, $t^j_{k_j(t_k^i)+1}$, $u_j^d(t^j_{k_j(t_k^i)})$, $\forall j \in \mathcal{N}_i$, then it can estimate its next triggering time t_{k+1}^i by solving (10.40). The above implement idea is summarized in Algorithm 10.5.

Similar to the single integrators case, broadcasting, receiving and listening can be ruled out except at the beginning, and each agent only needs to sense the relative positions to its neighbors and to update its control input at its triggering times. The idea is illustrated as follows.

From (10.31), (10.32), (10.67) and (10.72), we have

$$\frac{d\|E_i(t)\|}{dt} \leq \hat{c}_i^d(t),\ \forall t \geq 0,$$

where

$$\hat{c}_i^d(t) = \sum_{j \in N_i} \Big(k_1 g_{ij}(\hat{k}_{ij}^y(t))\hat{k}_{ij}^r(t) + k_2 h_{ij}(\hat{k}_{ij}^y(t))(\hat{k}_{ij}^r(t))^2 + k_2 f_{ij}(\hat{k}_{ij}^y(t))\theta_{ij}^d(t) \Big) + k_3\|u_i^d(t)\|.$$

If t_k^i is known, then agent i can estimate t_{k+1}^i by

$$\int_{t_k^i}^{t_{k+1}^i} \hat{c}_i^d(t)dt = \alpha_d e^{-\beta_d t_{k+1}^i}. \tag{10.41}$$

The above implement idea is summarized in Algorithm 10.6.

[1]Agent i uses $t^j_{k_j(t_k^i)}$ to replace $t^j_{k_j(t_k^i)+1}$ to determine t_{k+1}^i by (10.40) when $t_k^i = t^j_{k_j(t_k^i)}$.

[2]In other words, agent i only listen to incoming information at its neighbors' triggering times. Thus continuous listening is avoided.

Algorithm 10.6 Distributed Self-Triggered Formation Control Algorithm for Double Integrators (Sensing Only)

1: Choose $0 < \beta_1 \leq \beta_0$ and determine P by (10.22);
2: Choose $0 < k_3 < \frac{4}{k_2+\sqrt{k_1^2+k_2^2}}$, $\alpha_d > 0$ and $0 < \beta_d < \frac{2-k_4}{\rho(P)}$;
3: Agent $i \in [n]$ sends $\{d_{ij},\ (i,j) \in \mathcal{E}(\mathcal{G}),\ r_i(0)\}$ to its neighbors;
4: Agent i initializes $t_1^i = 0$ and $k = 1$;
5: At time $s = t_k^i$, agent i gets $r_i(t_k^i)$ by (10.27), and updates $u_i^d(t_k^i)$ by sensing the relative positions and speeds to its neighbors, and determines t_{k+1}^i by (10.41), and resets $k = k+1$, and repeats this step.

The following theorem shows that the formation with connectivity preservation can be established and Zeno behavior can be excluded.

Theorem 10.4. *Under the same settings as Theorem 10.3. All agents perform Algorithm 10.5 or 10.6, then the multi-agent system (10.19) with event-triggered control input (10.25) converges to the formation exponentially with connectivity preservation, and there is no Zeno behavior.*

Proof. Under both Algorithms 10.5 and 10.6, $\|E_i(t)\| \leq \alpha_d e^{-\beta_d t}$ holds for all $i \in [n]$ and $t \geq 0$. Then from Theorem 10.3, we know that the formation is achieved exponentially and the connectivity is preserved. The method of the exclusion of Zeno behavior is similar to the way in the proof of Theorem 10.3. □

Remark 10.5. *In real applications, it is reasonable to assume the initial speed of each agent is zero. By this assumption and Remark 10.3, we know that the only global parameter that is needed to perform Algorithms 10.5 and 10.6 is the number of agents n.*

Similar to Table 10.2, we can summarizes what and when information should be exchanged by each agent when Algorithms 10.4–10.6 are performed. Since it is similar to Table 10.2, we omit it here. Moreover, the comparison of the inter-event times determined by Algorithms 10.4–10.6 is similar to Property 10.1. The absolute measurements of positions and speeds are not needed when Algorithms 10.1–10.6 are performed.

10.5 Simulations

In this section, two numerical examples are given to demonstrate the effectiveness of the presented algorithms.

Consider a network of $n = 3$ agents in $\mathbb{R}^2$ whose Laplacian matrix is given by

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}.$$

The three agents are trying to establish a right triangle formation with

$$d_{12} = \begin{bmatrix} 0 \\ -3 \end{bmatrix}, \; d_{13} = \begin{bmatrix} -4 \\ 0 \end{bmatrix}, \; d_{23} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}.$$

The communication radius is $\Delta = 20$. We have $\beta_0 = 0.1765$.

Firstly, we consider the situation that the three agents are modeled as single integrators. The initial positions of agents can be randomly selected as long as the initial condition (10.5) is satisfied. Here, the initial positions of agents are chosen as

$$x_1(0) = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \; x_2(0) = \begin{bmatrix} 3.5 \\ 7 \end{bmatrix}, \; x_3(0) = \begin{bmatrix} 4.5 \\ 5.5 \end{bmatrix}.$$

One can easily check that both Assumption 10.1 and initial condition (10.5) hold. Choose $\alpha = 100$ and $\beta = \frac{\beta_0}{50}$, by applying the Algorithm 10.2 to the multi-agent system (10.1) with event-triggered control input (10.4), we show the evolutions of the formation in Figure 10.3. Figure 10.4 (a) shows the position evolutions of the multi-agent system (10.1) with event-triggered control input (10.4) when performing Algorithm 10.2, where "circles" denote the initial positions and "triangle" denotes the desired formation, and the triggering times for each agent are shown in Figure 10.4 (b). When every agent performs Algorithm 10.3, Figures 10.5 (a) and (b) show the position evolutions and the triggering times, respectively.

Secondly, we consider the situation that the three agents are modeled as double integrators. The initial positions of agents are chosen as before. The initial speeds of agents can be randomly selected and here we choose

$$r_1(0) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; r_2(0) = \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \; r_3(0) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}.$$

Moreover, we choose $\beta_1 = \beta_0$. Then, from (10.22), we have

$$P = \begin{bmatrix} 5.0237 & 1.1547 \\ 1.1547 & 1.4502 \end{bmatrix}.$$

Thus, $k_1 = 1.1547$, $k_2 = 1.4502$, and $\rho(P) = 5.3643$. Choose $k_3 = \frac{2}{k_2 + \sqrt{k_1^2 + k_2^2}} = 0.6053$, $\alpha_d = 100$, and $\beta_d = \frac{(2-k_4)}{10\rho(P)}$, by applying the Algorithm 10.5, we show the evolutions of the position in Figure 10.6 (a), where "circles" denote the initial positions and "triangle" denotes the desired formation, and show the triggering times for each agent in Figure 10.6 (b). When every agent performs Algorithm 10.6, Figures 10.7 (a) and (b) show the position evolutions and the triggering times, respectively.

It can be seen that the formation is achieved when any one of the four self-triggered algorithms is performed, but the formation could be achieved in different positions. It can also be seen that the average inter-event time determined by Algorithm 10.2 is greater than that determined by Algorithm 10.3. However, just as Table 10.2 summarized, when performing Algorithm 10.3, each agent only need to sense the relative positions to its neighbors at its triggering times. This is simpler than the case when performing Algorithm 10.2. Similar comparison can be made between Algorithms 10.5 and 10.6. Moreover, we can see that double integrators have more smooth trajectories compared with single integrators.

10.6 Summary

In this chapter, formation control for multi-agent systems with limited communication, including sensing, broadcasting, receiving and listening, was addressed. We first considered the situation that agents are modeled as single integrators. An event-triggered algorithm and two self-triggered algorithms, to avoid continuous communication and using absolute measurements of states, were proposed. It was shown that each agent only updates its control input by sensing the relative state to its neighbors and broadcasts its triggering information at its triggering times, and listens to and receives its neighbors' triggering information at their triggering times. Moreover, the desired formation was established exponentially with connectivity preservation and exclusion of Zeno behavior. Then, these results were extended to double integrators. Future research directions of this work include taking input saturation into account since the proposed event-triggered control input could be very large, which is unrealistic.

10.7 Proofs

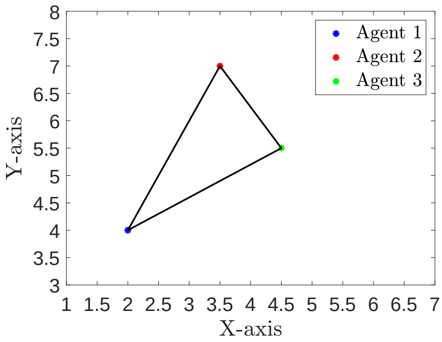
10.7.1 Proof of Theorem 10.1

(i) We define the total tension energy of $\mathcal{G}$ as

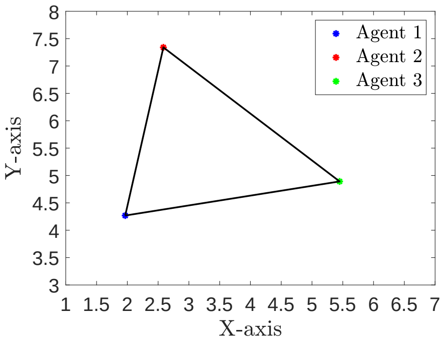
$$v(\Delta, y(t)) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in N_i} v_{ij}(\Delta, y(t)).$$

The time derivative of $v(\Delta, y(t))$ along the trajectories of the multi-agent system (10.2)–(10.3) is

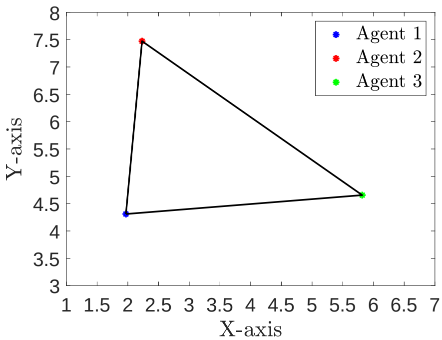
$$\begin{aligned}
\dot{v}(\Delta, y(t)) &= \sum_{i=1}^{n}\sum_{j\in N_i}\Big(\frac{\partial v_{ij}(\Delta, y)}{\partial y_i}\Big)^\top\Big|_{y=y(t)}\dot{y}_i(t)\\
&= \sum_{i=1}^{n}\sum_{j\in N_i}(\omega_{ij}(t)(y_i(t)-y_j(t)))^\top\Big(\sum_{j\in N_i}-\omega_{ij}(t^i_{k_i(t)})(y_i(t^i_{k_i(t)})-y_j(t^i_{k_i(t)}))\Big)\\
&= \sum_{i=1}^{n}\sum_{j\in N_i}(\omega_{ij}(t)(y_i(t)-y_j(t)))^\top\Big(e_i(t)-\sum_{j\in N_i}\omega_{ij}(t)(y_i(t)-y_j(t))\Big)\\
&= \sum_{i=1}^{n}\sum_{j\in N_i}(\omega_{ij}(t)(y_i(t)-y_j(t)))^\top\Big(-\sum_{j\in N_i}\omega_{ij}(t)(y_i(t)-y_j(t))\Big)
\end{aligned}$$



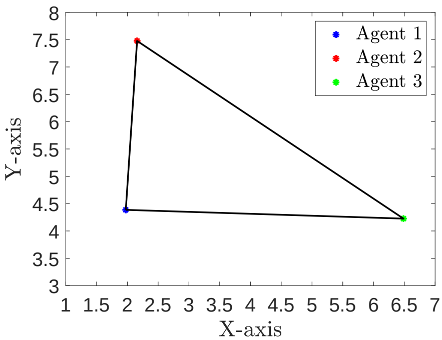
(a) At $t = 0$.



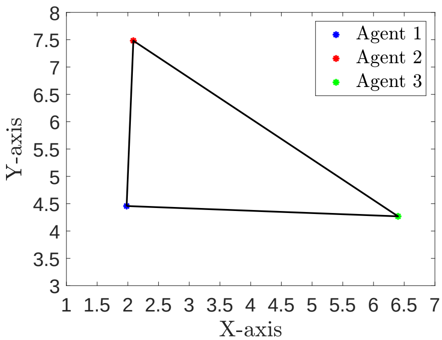
(b) At $t = 1.2$.



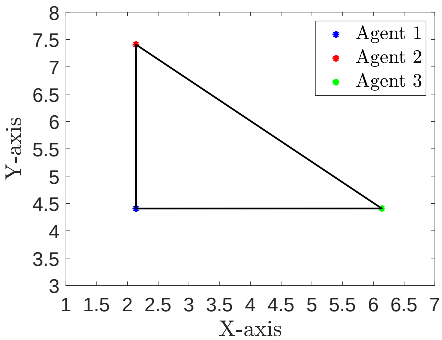
(c) At $t = 1.6$.



(d) At $t = 2.4$.

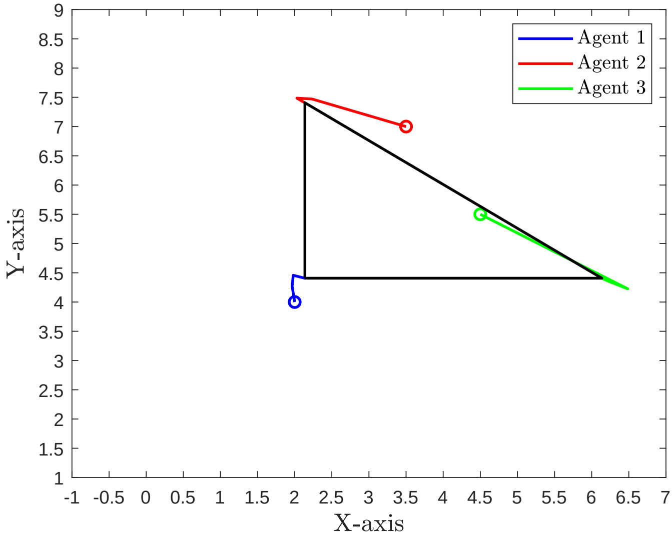


(e) At $t = 3.2$.

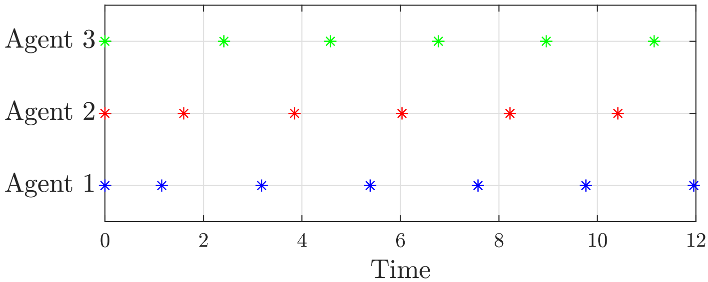


(f) At $t = 10.4$.

Figure 10.3: Evolutions of the formation process when performing Algorithm 10.2 for single integrators.

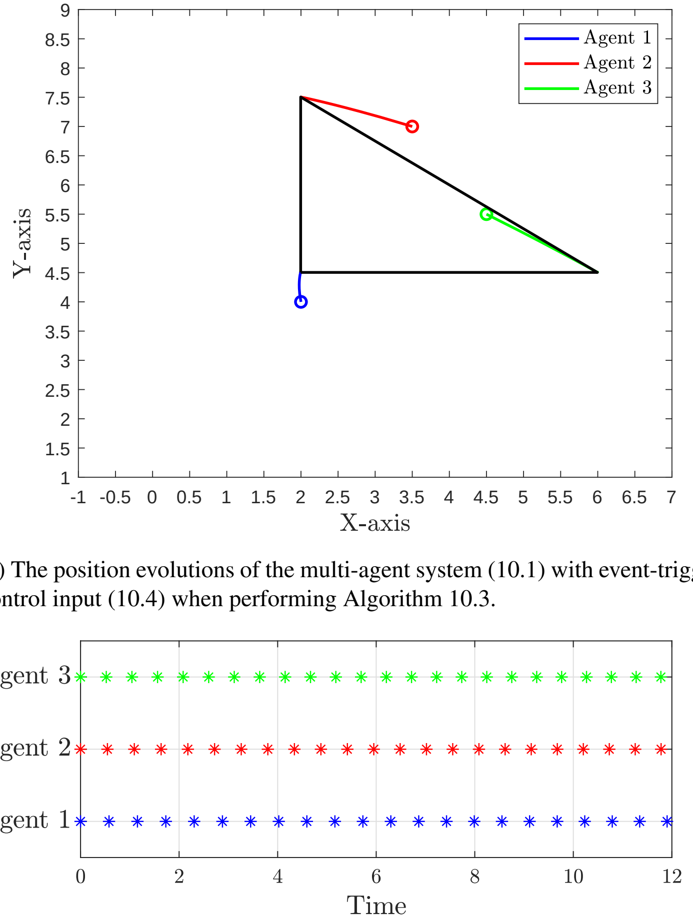


(a) The position evolutions of the multi-agent system (10.1) with event-triggered control input (10.4) when performing Algorithm 10.2.



(b) The triggering times for each agent.

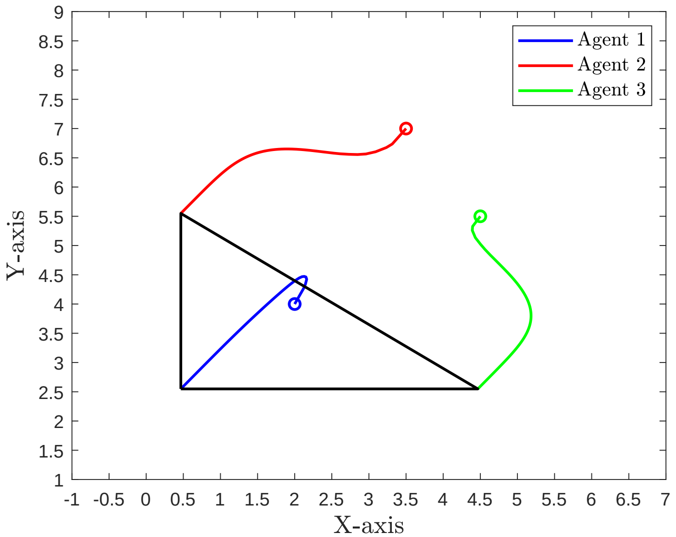
Figure 10.4: Performance of the distributed self-triggered formation control algorithm for single integrators.



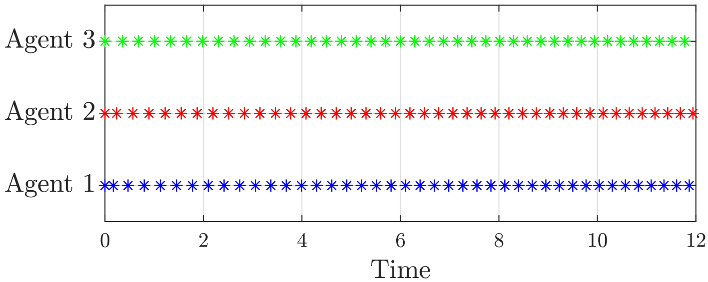
(a) The position evolutions of the multi-agent system (10.1) with event-triggered control input (10.4) when performing Algorithm 10.3.

(b) The triggering times for each agent.

Figure 10.5: Performance of the distributed self-triggered formation control algorithm for single integrators (sensing only).

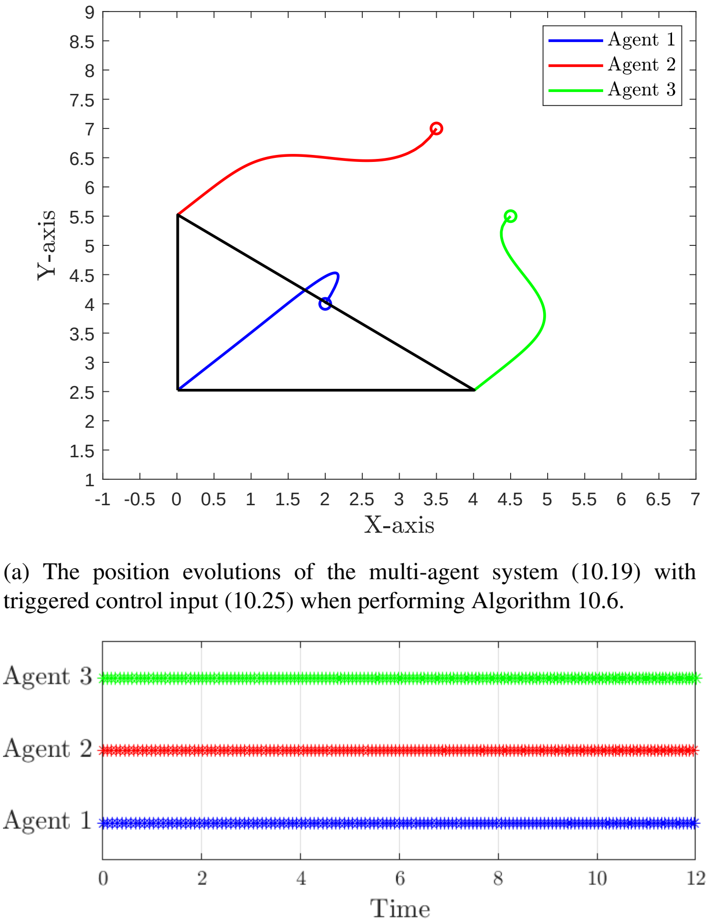


(a) The position evolutions of the multi-agent system (10.19) with event-triggered control input (10.25) when performing Algorithm 10.5.



(b) The triggering times for each agent.

Figure 10.6: Performance of the distributed self-triggered formation control algorithm for double integrators.



(a) The position evolutions of the multi-agent system (10.19) with event-triggered control input (10.25) when performing Algorithm 10.6.

(b) The triggering times for each agent.

Figure 10.7: Performance of the distributed self-triggered formation control algorithm for double integrators (sensing only).

$$
\begin{aligned}
&+ \sum_{i=1}^{n} \sum_{j \in N_i} (\omega_{ij}(t)(y_i(t) - y_j(t)))^\top e_i(t) \\
&\le -\|L_\omega y(t)\|^2 + \sum_{i=1}^{n} \Big\| \sum_{j \in N_i} \omega_{ij}(t)(y_i(t) - y_j(t)) \Big\|^2 + \frac{1}{4} \sum_{i=1}^{n} \|e_i(t)\|^2 \\
&= \frac{1}{4} \sum_{i=1}^{n} \|e_i(t)\|^2,
\end{aligned}
$$

From (10.6), we know that

$$
\|e_i(t)\| \le \alpha e^{-\beta t}, \ \forall t \ge 0. \tag{10.42}
$$

Hence

$$
\dot{v}(\Delta, y(t)) \le \frac{n\alpha^2}{4} e^{-2\beta t}, \ \forall t \ge 0.
$$

Thus

$$
v(\Delta, y(t)) \le v(\Delta, y(0)) + \frac{n\alpha^2}{8\beta}(1 - e^{-2\beta t}) \le k_v, \ \forall t \ge 0,
$$

where

$$
k_v = v(\Delta, y(0)) + \frac{n\alpha^2}{8\beta} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \frac{\|x_i(0) - x_j(0) - d_{ij}\|^2}{\Delta - \|d_{ij}\| - \|x_i(0) - x_j(0) - d_{ij}\|} + \frac{n\alpha^2}{8\beta}. \tag{10.43}
$$

Then, for any $(i, j) \in \mathcal{E}(\mathcal{G})$ and $t \ge 0$, we have

$$
v_{ij}(\Delta, y(t)) = \frac{\|y_i(t) - y_j(t)\|^2}{\Delta - \|d_{ij}\| - \|y_i(t) - y_j(t)\|} \le 2v(\Delta, y(t)) \le 2k_v.
$$

Hence

$$
\|y_i(t) - y_j(t)\| \le k_{ij}, \ \forall t \ge 0, \tag{10.44}
$$

where

$$
k_{ij} = -k_v + \sqrt{k_v^2 + 2k_v(\Delta - \|d_{ij}\|)} < \Delta - \|d_{ij}\|. \tag{10.45}
$$

Then, we have

$$
\begin{aligned}
\|x_i(t) - x_j(t)\| &= \|x_i(t) - \tau_i - (x_j(t) - \tau_j) + d_{ij}\| \\
&= \|y_i(t) - y_j(t) + d_{ij}\| \le \|y_i(t) - y_j(t)\| + \|d_{ij}\| \le k_{ij} + \|d_{ij}\| < \Delta, \ \forall t \ge 0,
\end{aligned}
$$

and thus connectivity maintenance is established.

(ii) Let $e(t) = \mathrm{col}(e_1(t), \dots, e_n(t))$, $\bar{y}(t) = \frac{1}{n}\sum_{i=1}^{n} y_i(t)$ and $\delta(t) = y(t) - \mathbf{1}_n \otimes \bar{y}(t) = (K_n \otimes I_p)y(t)$. We consider the Lyapunov candidate

$$V(y(t)) = \frac{1}{2}\delta^\top(t)\delta(t) = \frac{1}{2}y^\top(t)(K_n \otimes I_p)y(t). \tag{10.46}$$

Then its derivative along the trajectories of the multi-agent system (10.2)–(10.3) is

$$\begin{aligned}\dot{V}(y(t)) &= y^\top(t)(K_n \otimes I_p)\dot{y}(t) = y^\top(t)(K_n \otimes I_p)(-(L_\omega \otimes I_p)y(t) + e(t))\\ &= -y^\top(t)(L_\omega \otimes I_p)y(t) + \delta^\top(t)e(t).\end{aligned}$$

For $(i, j) \in \mathcal{E}(\mathcal{G})$, define

$$f_{ij}(l) = \frac{2\Delta - 2\|d_{ij}\| - l}{(\Delta - \|d_{ij}\| - l)^2},\ l \in [0, \Delta - \|d_{ij}\|). \tag{10.47}$$

We can easily check that $f_{ij}(l)$ is an increasing function on $[0, \Delta - \|d_{ij}\|)$. Then from (10.44) and (10.45) we have

$$\omega_{ij}(t) \le f_{ij}(k_{ij}),\ \omega_{ij}(t) \ge f_{ij}(0) = \frac{2}{\Delta - \|d_{ij}\|} \ge \frac{2}{\Delta_0},\ \forall (i, j) \in \mathcal{E}(\mathcal{G}),\ \forall t \ge 0. \tag{10.48}$$

Then,

$$\Omega(\mathcal{G}) = \mathrm{Diag}([\omega(e_1), \cdots, \omega(e_m)]) \ge \frac{2}{\Delta_0}I_m,$$

and

$$L_\omega = B(\mathcal{G})\Omega(\mathcal{G})B(\mathcal{G})^\top \ge \frac{2}{\Delta_0}B(\mathcal{G})I_mB(\mathcal{G})^\top \ge \frac{2\rho_2(B(\mathcal{G})B(\mathcal{G})^\top)}{\Delta_0}K_n = 2\beta_0K_n.$$

Thus

$$\begin{aligned}\dot{V}(y(t)) &= -y^\top(t)(L_\omega \otimes I_p)y(t) + \delta^\top(t)e(t)\\ &\le -2\beta_0 y^\top(t)(K_n \otimes I_p)y(t) + \beta_0\delta^\top(t)\delta(t) + \frac{1}{4\beta_0}\|e(t)\|^2\\ &= -2\beta_0V(y(t)) + \frac{1}{4\beta_0}\|e(t)\|^2 \le -2\beta_0V(y(t)) + \frac{n\alpha^2}{4\beta_0}e^{-2\beta t},\end{aligned}$$

where the first inequality holds due to $L_\omega \ge 2\beta_0K_n$ and the second inequality holds due to (10.42). Hence

$$V(y(t)) \le V(y(0))e^{-2\beta_0 t} + \frac{n\alpha^2}{8\beta_0(\beta_0 - \beta)}(e^{-2\beta t} - e^{-2\beta_0 t}) < k_Ve^{-2\beta t},$$

where

$$k_V = V(y(0)) + \frac{n\alpha^2}{8\beta_0(\beta_0 - \beta)}. \tag{10.49}$$

Thus

$$\|y_i(t) - y_j(t)\|^2 \leq 2\|y_i(t) - \bar{y}(t)\|^2 + 2\|\bar{y}(t) - y_j(t)\|^2$$
$$\leq 4V(y(t)) < 4k_V e^{-2\beta t}, \ \forall t \geq 0, \ \forall i, j \in [n]. \tag{10.50}$$

Hence

$$\lim_{t\to\infty}(x_i(t) - x_j(t)) = \lim_{t\to\infty}(y_i(t) - \tau_i - (y_j(t) - \tau_j)) = d_{ij},$$

exponentially.
(iii) For $(i, j) \in \mathcal{E}(\mathcal{G})$, define

$$g_{ij}(l) = \frac{2(\Delta - \|d_{ij}\|)^2}{(\Delta - \|d_{ij}\| - l)^3}, \ l \in [0, \Delta - \|d_{ij}\|), \tag{10.51}$$
$$h_{ij}(l) = \frac{3\Delta - 3\|d_{ij}\| - l}{(\Delta - \|d_{ij}\| - l)^3}, \ l \in [0, \Delta - \|d_{ij}\|). \tag{10.52}$$

We can easily check that both $g_{ij}(l)$ and $h_{ij}(l)$ are increasing functions on $[0, \Delta - \|d_{ij}\|)$. From (10.48), we have

$$\|\dot{y}_i(t)\| = \Big\|e_i(t) - \sum_{j\in N_i} \omega_{ij}(t)(y_i(t) - y_j(t))\Big\| \leq \|e_i(t)\| + \sum_{j\in N_i} \omega_{ij}(t)\|(y_i(t) - y_j(t))\| \tag{10.53}$$
$$< \alpha e^{-\beta t} + \sum_{j\in N_i} 2f_{ij}(k_{ij})\sqrt{k_V}e^{-\beta t}. \tag{10.54}$$

From

$$\begin{aligned}\dot{e}_i(t) &= \sum_{j\in N_i}(\dot{\omega}_{ij}(t)(y_i(t) - y_j(t)) + \omega_{ij}(t)(\dot{y}_i(t) - \dot{y}_j(t)))\\ &= \sum_{j\in N_i}\Big(h_{ij}(\|y_i(t) - y_j(t)\|)\frac{(y_i(t) - y_j(t))^\top}{\|y_i(t) - y_j(t)\|}(\dot{y}_i(t) - \dot{y}_j(t))(y_i(t) - y_j(t))\\ &\quad + \omega_{ij}(t)(\dot{y}_i(t) - \dot{y}_j(t))\Big),\end{aligned} \tag{10.55}$$

we have

$$\begin{aligned}\frac{d\|e_i(t)\|}{dt} &\leq \|\dot{e}_i(t)\|\\ &\leq \sum_{j\in N_i}\Big(\Big\|h_{ij}(\|y_i(t) - y_j(t)\|)\frac{(y_i(t) - y_j(t))^\top}{\|y_i(t) - y_j(t)\|}(\dot{y}_i(t) - \dot{y}_j(t))(y_i(t) - y_j(t))\Big\|\\ &\quad + \|\omega_{ij}(t)(\dot{y}_i(t) - \dot{y}_j(t))\|\Big)\\ &\leq \sum_{j\in N_i}\Big(h_{ij}(\|y_i(t) - y_j(t)\|)\|\dot{y}_i(t) - \dot{y}_j(t)\|\|y_i(t) - y_j(t)\| + \omega_{ij}(t)\|\dot{y}_i(t) - \dot{y}_j(t)\|\Big)\end{aligned}$$

$$= \sum_{j\in N_i} g_{ij}(\|y_i(t) - y_j(t)\|)\|\dot y_i(t) - \dot y_j(t)\| \tag{10.56}$$

$$\le \sum_{j\in N_i} g_{ij}(\|y_i(t) - y_j(t)\|)(\|\dot y_i(t)\| + \|\dot y_j(t)\|) \tag{10.57}$$

$$\le \sum_{j\in N_i} g_{ij}(k_{ij})(\|\dot y_i(t)\| + \|\dot y_j(t)\|) < c_i e^{-\beta t}, \tag{10.58}$$

where

$$c_i = \sum_{j\in N_i} g_{ij}(k_{ij})\Big(2\alpha + \sum_{l\in N_i} 2 f_{il}(k_{il})\sqrt{k_V} + \sum_{l\in N_j} 2 f_{jl}(k_{jl})\sqrt{k_V}\Big). \tag{10.59}$$

Thus, a necessary condition to guarantee that the inequality in (10.6) holds, i.e.,

$$\alpha e^{-\beta t} \le \|e_i(t)\| = \int_{t_k^i}^{t} \frac{d\|e_i(s)\|}{ds} ds,\ \forall t \in [t_k^i, t_{k+1}^i),$$

is

$$\begin{aligned}
&\alpha e^{-\beta t} \le \int_{t_k^i}^{t} c_i e^{-\beta s} ds = \frac{c_i}{\beta}(e^{-\beta t_k^i} - e^{-\beta t}) \\
&\Leftrightarrow (c_i + \alpha\beta)e^{-\beta t} \le c_i e^{-\beta t_k^i} \Leftrightarrow (c_i + \alpha\beta)e^{-\beta(t - t_k^i)} \le c_i \\
&\Rightarrow (c_i + \alpha\beta)(1 - \beta(t - t_k^i)) \le c_i \Leftrightarrow t - t_k^i \ge \epsilon_i,
\end{aligned}$$

where

$$\epsilon_i = \frac{\alpha}{c_i + \alpha\beta} > 0. \tag{10.60}$$

In other words, for all $t \in [t_k^i, t_k^i + \epsilon_i]$, $\|e_i(t)\| \le \alpha e^{-\beta t}$ holds. Hence $t_{k+1}^i \ge t_k^i + \epsilon_i$.

10.7.2 Proof of Theorem 10.3

(i) We define the total tension energy of $\mathcal{G}$ as

$$\nu_d(\Delta, y(t)) = k_1 \nu(\Delta, y(t)) + \frac{1}{2}\sum_{i=1}^{n} \|r_i(t)\|^2.$$

Then time derivative of $\nu_d(\Delta, y(t))$ along the trajectories of the multi-agent system (10.20) with (10.24) is

$$\begin{aligned}
\dot\nu_d(\Delta, y(t)) &= k_1 \sum_{i=1}^{n}\sum_{j\in N_i}\Big(\frac{\partial \nu_{ij}(\Delta, y)}{\partial y_i}\Big)^\top\Big|_{y=y(t)} \dot y_i(t) + \sum_{i=1}^{n} r_i^\top(t)\dot r_i(t) \\
&= \sum_{i=1}^{n} r_i^\top(t)\Big(k_1 \sum_{j\in N_i}(\omega_{ij}(t)(y_i(t) - y_j(t))) + u_i^d(t)\Big)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} r_i^\top(t)\Big(E_i(t) - k_2 \sum_{j\in N_i} \omega_{ij}(t)(r_i(t) - r_j(t)) - k_3 r_i(t)\Big) \\
&\le \frac{1}{4k_3}\sum_{i=1}^{n} \|E_i(t)\|^2 - \sum_{i=1}^{n} r_i^\top(t) k_2 \sum_{j\in N_i} \omega_{ij}(t)(r_i(t) - r_j(t)) \\
&= \frac{1}{4k_3}\sum_{i=1}^{n} \|E_i(t)\|^2 - k_2 r^\top(t) L_\omega r(t).
\end{aligned}
$$

From (10.26), we know that

$$\|E_i(t)\| \le \alpha_d e^{-\beta_d t}, \ \forall t \ge 0.$$

Hence

$$\dot{v}_d(\Delta, y(t)) \le \frac{n\alpha_d^2}{4k_3} e^{-2\beta_d t}, \ \forall t \ge 0.$$

Thus

$$v_d(\Delta, y(t)) \le v_d(\Delta, y(0)) + \frac{n\alpha_d^2}{8k_3\beta_d}(1 - e^{-2\beta_d t}) \le k_v^d, \ \forall t \ge 0,$$

where

$$
\begin{aligned}
k_v^d &= v_d(\Delta, y(0)) + \frac{n\alpha_d^2}{8k_3\beta_d} \\
&= \frac{k_1}{2}\sum_{i=1}^{n}\sum_{j\in N_i} \frac{\|x_i(0) - x_j(0) - d_{ij}\|^2}{\Delta - \|d_{ij}\| - \|x_i(0) - x_j(0) - d_{ij}\|} + \frac{1}{2}\sum_{i=1}^{n}\|r_i(0)\|^2 + \frac{n\alpha_d^2}{8k_3\beta_d}.
\end{aligned}
$$

Then, for any $(i, j) \in \mathcal{E}(\mathcal{G})$ and $t \ge 0$, we have

$$v_{ij}(\Delta, y(t)) = \frac{\|y_i(t) - y_j(t)\|^2}{\Delta - \|d_{ij}\| - \|y_i(t) - y_j(t)\|} \le \frac{2}{k_1} v_d(\Delta, y(t)) \le \frac{2}{k_1} k_v^d.$$

Hence

$$\|y_i(t) - y_j(t)\| \le k_{ij}^d, \tag{10.61}$$

where

$$k_{ij}^d = -\frac{k_v^d}{k_1} + \Big(\Big(\frac{k_v^d}{k_1}\Big)^2 + 2\frac{k_v^d}{k_1}(\Delta - \|d_{ij}\|)\Big)^{\frac{1}{2}} < \Delta - \|d_{ij}\|.$$

Then, we have

$$\|x_i(t) - x_j(t)\| = \|x_i(t) - \tau_i - (x_j(t) - \tau_j) + d_{ij}\| = \|y_i(t) - y_j(t) + d_{ij}\|$$

$$\leq \|y_i(t) - y_j(t)\| + \|d_{ij}\| \leq k_{ij}^d + \|d_{ij}\| < \Delta, \ \forall t \geq 0,$$

and thus connectivity maintenance is guaranteed.
(ii) Note $B_2^\top P = [k_1 \ k_2]$, then we can rewrite the control input (10.24) as

$$u_i^d(t) = -(B_2^\top P \otimes I_p) \sum_{j \in N_i} \omega_{ij}(t)(z_i(t) - z_j(t)) + E_i(t) - k_3(B_2^\top \otimes I_p) z_i(t).$$

Let $z(t) = \mathrm{col}(z_1(t), \ldots, z_n(t))$ and $\bar{z}(t) = \frac{1}{n} \sum_{i=1}^n z_i(t)$. We consider the Lyapunov candidate

$$V_d(z(t)) = \frac{1}{2}(z(t) - \mathbf{1}_n \bar{z}(t))^\top (I_n \otimes P \otimes I_p)(z(t) - \mathbf{1}_n \bar{z}(t)) = \frac{1}{2} z^\top(t)(K_n \otimes P \otimes I_p) z(t).$$

The last equality holds since

$$z^\top(t)(K_n \otimes I_2 \otimes I_p) z(t) = (z(t) - \mathbf{1}_n \bar{z}(t))^\top (I_n \otimes I_2 \otimes I_p)(z(t) - \mathbf{1}_n \bar{z}(t)).$$

Then the derivative of $V_d(z(t))$ along the trajectories of (10.21) is

$$\begin{aligned}
\dot{V}_d(z(t)) &= z^\top(t)(K_n \otimes P \otimes I_p)\dot{z}(t) \\
&= z^\top(t)(K_n \otimes P \otimes I_p)((I_n \otimes B_1 \otimes I_p)z(t) + (I_n \otimes B_2 \otimes I_p)u^d(t)) \\
&= z^\top(t)(K_n \otimes P \otimes I_p)((I_n \otimes B_1 \otimes I_p)z(t) \\
&\quad + (I_n \otimes B_2 \otimes I_p)(-(L_\omega \otimes B_2^\top P \otimes I_p)z(t) + E(t) - k_3(I_n \otimes B_2^\top \otimes I_p)z(t))) \\
&= \frac{1}{2} z^\top(t)(K_n \otimes (PB_1 + B_1^\top P) \otimes I_p)z(t) - z^\top(t)(L_\omega \otimes PB_2 B_2^\top P \otimes I_p)z(t) \\
&\quad - k_3 z^\top(t)(K_n \otimes PB_2 B_2^\top \otimes I_p)z(t) + z^\top(t)(K_n \otimes PB_2 \otimes I_p)E(t),
\end{aligned}$$

where $u^d(t) = \mathrm{col}(u_1^d(t), \ldots, u_n^d(t))$ and $E(t) = \mathrm{col}(E_1(t), \ldots, E_n(t))$. From $PB_2 B_2^\top P \geq 0$ and $L_\omega \geq 2\beta_0 K_n \geq 2\beta_1 K_n$ due to (2.6), we have

$$-z^\top(t)(L_\omega \otimes PB_2 B_2^\top P \otimes I_p)z(t) \leq -2\beta_1 z^\top(t)(K_n \otimes PB_2 B_2^\top P \otimes I_p)z(t). \tag{10.62}$$

Noting

$$\frac{PB_2 B_2^\top + B_2 B_2^\top P}{2} = \begin{bmatrix} 0 & \frac{k_1}{2} \\ \frac{k_1}{2} & k_2 \end{bmatrix},$$

one can easily check that $\rho(\frac{1}{2}(PB_2 B_2^\top + B_2 B_2^\top P)) = \frac{1}{2}(k_2 + (k_1^2 + k_2^2)^{\frac{1}{2}})$. Noting $k_4 = \frac{1}{2} k_3 (k_2 + (k_1^2 + k_2^2)^{\frac{1}{2}})$, we have

$$-k_3 z^\top(t)(K_n \otimes PB_2 B_2^\top \otimes I_p)z(t) \leq k_4 z^\top(t)(K_n \otimes I_2 \otimes I_p)z(t). \tag{10.63}$$

Then from (10.62), (10.63) and the inequality

$$z^\top(t)(K_n \otimes PB_2 \otimes I_p)E(t) \leq \beta_1 z^\top(t)(K_n \otimes PB_2 B_2^\top P \otimes I_p)z(t) + \frac{1}{4\beta_1}\|E(t)\|^2,$$

we get

$$\begin{aligned}
\dot{V}_d(z(t)) &\le \frac{1}{2} z^\top(t)\Big(K_n \otimes (PB_1 + B_1^\top P - 2\beta_1 PB_2B_2^\top P) \otimes I_p\Big) z(t) \\
&\quad + k_4 z^\top(t)(K_n \otimes I_2 \otimes I_p) z(t) + \frac{1}{4\beta_1}\|E(t)\|^2 \\
&\le -(2-k_4) z^\top(t)(K_n \otimes I_2 \otimes I_p) z(t) + \frac{1}{4\beta_1}\|E(t)\|^2 \\
&\le -\frac{2(2-k_4)}{\rho(P)} V_d(z(t)) + \frac{n\alpha_d^2}{4\beta_1} e^{-2\beta_d t},
\end{aligned}$$

where the second inequality holds due to (10.22) and the last inequality holds due to (10.23). Hence

$$V_d(z(t)) \le V_d(z(0)) e^{-\frac{2(2-k_4)}{\rho(P)}t} + \frac{\rho(P) n\alpha_d^2 (e^{-2\beta_d t} - e^{-\frac{2(2-k_4)}{\rho(P)}t})}{8\beta_1(2-k_4-\beta_d\rho(P))}.$$

Thus

$$\begin{aligned}
\|y_i(t) - y_j(t)\|^2 + \|r_i(t) - r_j(t)\|^2 = \|z_i(t) - z_j(t)\|^2 &\le 2\|z_i(t) - \bar{z}(t)\|^2 + 2\|\bar{z}(t) - z_j(t)\|^2 \\
&\le \frac{4}{\rho_2(P)} V_d(z(t)) < k_V^d e^{-2\beta_d t}, \qquad (10.64)
\end{aligned}$$

where

$$k_V^d = \frac{4V_d(z(0))}{\rho_2(P)} + \frac{\rho(P) n\alpha_d^2}{2\rho_2(P)\beta_1(2-k_4-\beta_d\rho(P))}.$$

Hence

$$\lim_{t\to\infty}(x_i(t) - x_j(t)) = \lim_{t\to\infty}(y_i(t) - \tau_i - (y_j(t) - \tau_j)) = d_{ij},$$

and

$$\lim_{t\to\infty}(r_i(t) - r_j(t)) = 0,$$

exponentially.
(iii) From

$$\dot{r}_i(t) = u_i^d(t) = E_i(t) - k_1 \sum_{l\in N_i} \omega_{il}(t)(y_i(t) - y_l(t)) - k_2 \sum_{l\in N_i} \omega_{il}(t)(r_i(t) - r_l(t)) - k_3 r_i(t),$$

we have

$$\frac{de^{k_3 t} r_i(t)}{dt} = \Big(E_i(t) - k_1 \sum_{l\in N_i} \omega_{il}(t)(y_i(t) - y_l(t)) - k_2 \sum_{l\in N_i} \omega_{il}(t)(r_i(t) - r_l(t))\Big) e^{k_3 t}.$$

Then, similar to (10.54), we have

$$\begin{aligned}
&\frac{d\|e^{k_3t}r_i(t)\|}{dt} \le \left\|\frac{de^{k_3t}r_i(t)}{dt}\right\| \\
&= \left\|E_i(t) - k_1\sum_{l\in N_i}\omega_{il}(t)(y_i(t)-y_l(t)) - k_2\sum_{l\in N_i}\omega_{il}(t)(r_i(t)-r_l(t))\right\|e^{k_3t} \\
&\le c_i^r e^{(k_3-\beta_d)t},
\end{aligned}$$

where

$$c_i^r = \alpha_d + (k_1+k_2)\sum_{l\in N_i} f_{il}(k_{il}^d)\sqrt{k_V^d}.$$

From

$$e^{k_3t}\frac{d\|r_i(t)\|}{dt} \le e^{k_3t}\frac{d\|r_i(t)\|}{dt} + k_3e^{k_3t}\|r_i(t)\| = \frac{de^{k_3t}\|r_i(t)\|}{dt} = \frac{d\|e^{k_3t}r_i(t)\|}{dt},$$

we have

$$\frac{d\|r_i(t)\|}{dt} \le c_i^r e^{-\beta_d t},\ \forall t\ge 0.$$

Thus

$$\|r_i(t)\| \le \|r_i(0)\| + \frac{c_i^r}{\beta_d},\ \forall t\ge 0, \tag{10.65}$$

and

$$\begin{aligned}
&\|u_i^d(t)\| = \|\dot r_i(t)\| \\
&= \left\|E_i(t) - k_1\sum_{l\in N_i}\omega_{il}(t)(y_i(t)-y_l(t)) - k_2\sum_{l\in N_i}\omega_{il}(t)(r_i(t)-r_l(t)) - k_3r_i(t)\right\| \\
&\le c_i^r e^{-\beta_d t} + k_3\Big(\|r_i(0)\| + \frac{c_i^r}{\beta_d}\Big).
\end{aligned} \tag{10.66}$$

Again, similar to (10.54), we have

$$\begin{aligned}
\|\dot r_i(t) - \dot r_j(t)\| = \Big\|&E_i(t) - E_j(t) - k_1\sum_{l\in N_i}\omega_{il}(t)(y_i(t)-y_l(t)) \\
&- k_2\sum_{l\in N_i}\omega_{il}(t)(r_i(t)-r_l(t)) + k_1\sum_{l\in N_j}\omega_{jl}(t)(y_j(t)-y_l(t))\Big\| \\
&+ k_2\sum_{l\in N_j}\omega_{jl}(t)(r_j(t)-r_l(t)) - k_3(r_i(t)-r_j(t))\| \qquad (10.67)\\
< c_{ij}^r &e^{-\beta_d t}, \qquad (10.68)
\end{aligned}$$

where $c_{ij}^r = 2\alpha_d + ((k_1 + k_2)(\sum_{l\in N_i} f_{il}(k_{il}^d) + \sum_{l\in N_j} f_{jl}(k_{jl}^d)) + k_3)\sqrt{k_V^d}$

Similar to (10.55), we have

$$\begin{aligned}
\dot{e}_i(t) &= \sum_{j\in N_i}(\dot{\omega}_{ij}(t)(y_i(t) - y_j(t)) + \omega_{ij}(t)(\dot{y}_i(t) - \dot{y}_j(t))) \\
&= \sum_{j\in N_i}\Big(h_{ij}(\|y_i(t) - y_j(t)\|)\frac{(y_i(t) - y_j(t))^\top}{\|y_i(t) - y_j(t)\|}(r_i(t) - r_j(t))(y_i(t) - y_j(t)) \\
&\quad + \omega_{ij}(t)(r_i(t) - r_j(t))\Big), \qquad (10.69)
\end{aligned}$$

and

$$\begin{aligned}
\dot{e}_i^r(t) &= \sum_{j\in N_i}(\dot{\omega}_{ij}(t)(r_i(t) - r_j(t)) + \omega_{ij}(t)(\dot{r}_i(t) - \dot{r}_j(t))) \\
&= \sum_{j\in N_i}\Big(h_{ij}(\|y_i(t) - y_j(t)\|)\frac{(y_i(t) - y_j(t))^\top}{\|y_i(t) - y_j(t)\|}(r_i(t) - r_j(t))(r_i(t) - r_j(t)) \\
&\quad + \omega_{ij}(t)(\dot{r}_i(t) - \dot{r}_j(t))\Big). \qquad (10.70)
\end{aligned}$$

Similar to (10.58), from (10.61), (10.66), and (10.68), we have

$$\begin{aligned}
\frac{d\|E_i(t)\|}{dt} &= \frac{d\|k_1 e_i(t) + k_2 e_i^r(t) + k_3(r_i(t) - r_i(t_{k_i(t)}^i))\|}{dt} \\
&\le \|k_1\dot{e}_i(t) + k_2\dot{e}_i^r(t) + k_3\dot{r}_i(t)\| \qquad (10.71)\\
&\le k_1\|\dot{e}_i(t)\| + k_2\|\dot{e}_i^r(t)\| + k_3\|\dot{r}_i(t)\| \\
&\le \sum_{j\in N_i}\Big(k_1 g_{ij}(\|y_i(t) - y_j(t)\|)\|r_i(t) - r_j(t)\| \\
&\quad + k_2 h_{ij}(\|y_i(t) - y_j(t)\|)\|r_i(t) - r_j(t)\|^2 + k_2\omega_{ij}(t)(\|\dot{r}_i(t) - \dot{r}_j(t)\|)\Big) \\
&\quad + k_3\|u_i^d(t)\| \qquad (10.72)\\
&\le \sum_{j\in N_i} k_1 g_{ij}(k_{ij}^d)\|r_i(t) - r_j(t)\| + k_2 h_{ij}(k_{ij}^d)\|r_i(t) - r_j(t)\|^2 \\
&\quad + k_2 f_{ij}(k_{ij}^d)(\|\dot{r}_i(t) - \dot{r}_j(t)\|) + k_3\Big(c_i^q e^{-\beta_d t} + k_3\Big(\|r_i(0)\| + \frac{c_i^r}{\beta_d}\Big)\Big) \\
&< c_i^d e^{-\beta_d t} + k_3\Big(c_i^r e^{-\beta_d t} + k_3\Big(\|r_i(0)\| + \frac{c_i^r}{\beta_d}\Big)\Big),
\end{aligned}$$

where $c_i^d = \sum_{j\in N_i}(k_1 g_{ij}(k_{ij}^d)\sqrt{k_V^d} + k_2 h_{ij}(k_{ij}^d)k_V^d + k_2 f_{ij}(k_{ij}^d)c_{ij}^r)$. Thus

$$\frac{d\|E_i(t)\|}{dt} < c_i^e,\ \forall t \ge 0, \qquad (10.73)$$

where $c_i^e = c_i^d + k_3(c_i^r + k_3(\|q_i(0)\| + \frac{c_i^r}{\beta_d}))$.

From (10.73), similar to the way to exclude Zeno behavior in the proof of Theorem 8.1 or 9.2, we can prove that there is no Zeno behavior by contradiction.

Chapter 11

Conclusions and future research

In this chapter, we summarize this thesis and discuss possible directions for future research.

11.1 Summary

The main results of this thesis were presented in Chapters 3–10 and divided into three parts.

Distributed nonconvex optimization

In this part, we proposed distributed algorithms to solve nonconvex optimization problems under different information feedback settings. We showed convergence properties, such as linear convergence and linear speedup, of these algorithms under weaker assumptions on the underlying communication network and cost functions than existing results in the literature.

In Chapter 3, we proposed three algorithms: a distributed primal–dual FO algorithm, a distributed ADMM algorithm, and a distributed linearized ADMM algorithm, to solve the nonconvex optimization problem with full-information feedback. We derived their convergence rates. More specifically, the classic $\mathcal{O}(1/T)$ convergence rate was achieved when each local cost function is smooth, and linear convergence was established when the global cost function satisfies the P–Ł condition in addition, which relaxes the standard strong convexity condition in the literature. One immediate future research direction is to show other distributed optimization algorithms, such as distributed heavy-ball and adaptive momentum algorithms, also achieve linear convergence under the P–Ł condition.

In Chapter 4, we studied distributed nonconvex optimization with stochastic gradient feedback. We proposed a distributed primal–dual SGD algorithm which is suitable for arbitrarily connected communication networks and any smooth cost functions. We showed that the linear speedup convergence rate $\mathcal{O}(1/\sqrt{nT})$ was established for smooth nonconvex cost functions. The convergence rate was improved to the linear speedup convergence rate $\mathcal{O}(1/(nT))$ when the global cost function satisfies the P–Ł condition in addition. It was also shown that the output of the proposed algorithm with constant parameters linearly

Table 11.1: Summary of the results in Part I of this thesis and comparison with the literature.

	Extra assumption?	Linear convergence?	Linear speedup?
[68–74, 76–91]	No	Strongly convex	Not applicable
Chapter 3	No	P–Ł condition	Not applicable
[31–33, 132, 133, 135–137, 140, 141, 251, 255, 257–259]	Yes	Not applicable	Yes
Chapter 4	No	Not applicable	Yes
[147–149, 151–155]	No	Not applicable	No
Chapter 5	No	Not applicable	Yes

converges to a neighborhood of a global optimum. With some modifications, we believe that the results in this chapter still hold for the distributed primal SGD and stochastic gradient tracking algorithms.

In Chapter 5, we investigated distributed nonconvex optimization with ZO oracle feedback. We first proposed a distributed primal–dual DZO algorithm to solve this problem when DZO oracle feedback is available. We showed that it has the same convergence properties as its FO counterpart under the same conditions. We then proposed two distributed SZO algorithms to solve this problem when SZO oracle feedback is available. We showed that the linear speedup convergence rate $O(\sqrt{p/(nT)})$ was established for smooth nonconvex cost functions under arbitrarily connected communication networks. The convergence rate was improved to $O(p/(nT))$ when the global cost function satisfies the P–Ł condition in addition. It was also shown that the output of the these two algorithms linearly converges to a neighborhood of a global optimum. One immediate future research direction is to achieve faster convergence with reduced sampling complexity by using variance reduction techniques.

We summarize some aspects of the results in this part in Table 11.1 and compare them with the literature. The columns list some specific properties of distributed optimization algorithms. Strong convexity is needed by existing full-information based distributed optimization algorithms to obtain linear convergence, whereas this condition has been relaxed by the P–Ł condition in Chapter 3, see the second and third rows in Table 11.1. Existing SGD algorithms that obtained linear speedup require extra assumptions on the communication network and cost functions, such as star graph, bounded gradients of the local cost functions, and/or bounded difference between the gradients of the local and global cost functions, but Chapter 4 does not, see the fourth and fifth rows in Table 11.1. None of existing distributed ZO algorithms achieve linear speedup and most of them do not consider the SZO oracle feedback setting either, while Chapter 5 does both, see the last two rows in Table 11.1.

Distributed online convex optimization

In this part, we proposed distributed online algorithms to solve convex optimization problems with time-varying coupled inequality constraints under different information feedback settings. We showed that the proposed algorithms achieve comparable and sometimes better performance than existing (centralized) algorithms in the literature measured by regret and constraint violation under weaker assumptions.

In Chapter 6, we considered an online convex optimization problem with time-varying coupled inequality constraints. To the best of our knowledge, no existing studies considered this problem before. We proposed a distributed online primal–dual dynamic mirror descent algorithm to solve this problem. This algorithm does not require knowledge of the total number of rounds or any other parameters related to the loss or constraint functions. We derived regret and constraint violation bounds for the algorithm and showed how they depend on the stepsize sequences, the accumulated dynamic variation of the comparator sequence, the number of agents, and the network connectivity. We proved that the algorithm achieves sublinear regret and constraint violation for both convex and strongly convex objective functions. Compared with existing literature, this chapter achieved better results under much weaker assumptions. With some modifications, we believe that the results in this chapter can be extended to the situation where stochastic gradient information is available. Furthermore, the results also can most likely be extended to time-varying unbalanced directed communication networks.

In Chapter 7, we considered the distributed bandit online convex optimization problem with time-varying coupled inequality constraints. To the best of our knowledge, no existing studies considered this problem before. There are even no studies considered the centralized bandit online convex optimization problem with time-varying inequality constraints in the one-point bandit feedback setting. We proposed distributed bandit online algorithms with one- and two-point bandit feedback, which do not require knowledge of the total number of rounds or any other parameters related to the loss or constraint functions. We showed that sublinear expected regret and constraint violation can be achieved by both algorithms, which recover the bounds achieved by existing centralized bandit algorithms. With some modifications, we believe that the results in this chapter still hold when considering sampling noise. Furthermore, the results also most likely hold under time-varying unbalanced directed communication networks.

We summarize some aspects of the results in this part in Table 11.2 and compare them with the literature. The columns list some specific properties of (bandit) online convex optimization algorithms. Most of existing algorithms require assumptions, such as knowing the total number of rounds or any other parameters related to the loss or constraint functions. Moreover, most are centralized and do not consider time-varying constraints.

Distributed event-triggered control

In this part, we proposed distributed dynamic event-triggered control algorithms for multi-agent systems to reduce the amount of information exchanged and system update in general. In particular, the three problems of average consensus for single-integrator agents,

Table 11.2: Summary of the results in Part II of this thesis and comparison with the literature.

	Extra assumption?	Distributed setting?	Time-varying constraints?	Bandit feedback?
[169, 170, 172]	Yes	No	No	No
[173–175]	Yes	No	Yes	No
[190, 191]	No	Yes	No	No
Chapter 6	No	Yes	Yes	No
[169, 212, 280, 324, 325]	Yes	No	No	Yes
[328, 329]	Yes	No	Yes	Yes
Chapter 7	No	Yes	Yes	Yes

global consensus for single-integrator agents with input saturation, and formation control for single- and double-integrator agents with connectivity preservation were solved.

In Chapter 8, we first proposed two dynamic event-triggered control algorithms for first-order continuous-time multi-agent systems to solve average consensus problem. Compared with existing event-triggered control algorithms, our dynamic event-triggered control algorithms involve internal dynamic variables which play an essential role in guaranteeing that the triggering time sequence does not exhibit Zeno behavior. Some of the existing event-triggered control algorithms are special cases of our strategies. We proved that average consensus is achieved exponentially if and only if the communication graph is connected, and Zeno behavior was excluded by proving that the triggering time sequence of each agent is divergent. Then, we proposed a self-triggered control algorithm to avoid continuous listening over the network. As a result, each agent only needs to sense and broadcast at its triggering times, and to listen to and receive incoming information from its neighbors at their triggering times. Thus continuous listening is avoided. With some modifications, the results in this chapter can be extended to the cases that the underlying graph is directed and has a directed spanning tree. Furthermore, the results also can most likely be extended to general linear and nonlinear multi-agent systems with standard controllability assumptions for linear dynamics and standard continuity assumptions for the nonlinear dynamics.

In Chapter 9, we extended the results above to multi-agent systems with input saturation constraints over digraphs. We first showed that global consensus is achieved if and only if the underlying directed communication network has a directed spanning tree. We then considered event-triggered control and presented a distributed triggering law to reduce the overall need of communication and system updates. The triggering law was a special kind of dynamic triggering and was inspired by a Lyapunov function we used in the proof of the first result. We showed that consensus is achieved for the event-triggered control under the same connectivity condition, and the triggering law was proven to be free of Zeno behavior. Moreover, we presented a self-triggered algorithm to avoid continuous listening. With some modifications, we believe that the results in this chapter can be extended to multi-agent systems with output saturation constraints and even

Table 11.3: Summary of the results in Part III of this thesis and comparison with the literature.

	Continuous broadcasting?	Continuous listening?	State information?	Avoiding Zeno?
[193–195, 358, 366, 379, 403, 408]	No	Yes	Absolute	?
[200, 367–369, 380, 404, 407]	No	Yes	Relative	?
Chapter 8	No	No	Absolute	Yes
Chapter 9	No	No	Absolute	Yes
Chapter 10	No	No	Relative	Yes

nonlinear multi-agent systems with standard continuity assumptions.

In Chapter 10, formation control for multi-agent systems with limited communication was addressed. We first considered the situation that agents are modeled as single integrators and designed distributed event-triggered control. An event-triggered algorithm and two self-triggered algorithms were proposed. It was shown that each agent only updates its control input by sensing the relative state to its neighbors and broadcasts its triggering information at its triggering times, and listens to and receives its neighbors' triggering information at their triggering times. The desired formation was shown to be established exponentially with connectivity preservation and exclusion of Zeno behavior. Then, these results were extended to double integrators. With some modifications, we think the results in this chapter can be extended to position- and distance-based formation control, and can most likely be extended to systems with input saturation.

We summarize some aspects of the results in this part in Table 11.3 and compare them with the literature. The columns list some specific properties of distributed event-triggered control algorithms. None of the listed work assume continuous broadcasting of the agents' state to its neighbors, but it is common in the literature to assume continuous listening. None of results in Chapters 8–10 require agents to continuously listen to their neighbors. The table specifies if the considered control laws are based on absolute state information or relative state information. Finally, as shown in the thesis it is important to exclude Zeno behavior. In the literature, this issue has not always been carefully investigated. In particular, references [193, 194, 367, 369, 379, 403, 404, 407] do not strictly show that Zeno behavior is excluded, while [195, 200, 358, 366, 368, 380, 408] do.

11.2 Future research directions

There are several interesting research directions that can be based on the work of this thesis. Some of the immediate ones were mentioned above. Other extensions are discussed in this section.

Distributed optimization with adversarial agents

A common assumption in the distributed optimization literature and Chapters 3–5 is that all agents cooperate to learn the optimal solution. However, in networked cyber-physical systems, some agents may become adversarial. Therefore, it is important to investigate the performance of existing distributed optimization algorithms in the presence of adversarial agents. Although distributed resilient consensus has been quite well studied, e.g., [409–411], distributed resilient optimization with adversarial agents is a more open problem, e.g., [412–414]. Existing results establish sufficient and/or necessary conditions under which the proposed distributed algorithms ensure that the non-adversarial agents converge to the convex hull of the local minimizers even in the presence of adversarial agents. However, these results focus on distributed algorithms with diminishing stepsizes and thus the convergence rate is slow. It is interesting to develop distributed resilient optimization algorithms with fixed stepsizes such that faster convergence rate can be achieved.

Distributed online convex optimization with aggregated variables

Distributed online convex optimization literature and Chapters 6–7 focus on the case where each local loss and constraint functions depend only on local decision variables. However, in many applications, the local loss and constraint functions depend also on other agents' decision variables. For example, the target surrounding problem, in which a collection of agents desire to form a circular formation enclosing a moving target in dynamic environments, can be formulated as an online optimization problem with each local loss function resting not only on each agent's own decision variable (such as position), but also on the average of all agents' decision variables. There are only few results on this direction, e.g., [415]. It is challenging to develop distributed online algorithms such that sublinear regret can be achieved.

Distributed event-triggered control with limited data rate

In Chapters 8–10, we showed that when agents use dynamic event-triggered strategies the overall need of communication and system updates are reduced. It would be interesting to quantify this reduction systematically and compare it with other event- and time-triggered strategies. One specific problem is to determine the number of triggering times that are needed to guarantee that all agents reach a ball of given radius centered at the average of all agents' states. It would be also interesting to find the minimum communication rate between agents to guarantee that desired properties still can be achieved. Such minimum rate question is well studied for single-agent systems, e.g., [416–420]. Although there are some results also for multi-agent systems, e.g., [421, 422], this direction is far from being complete. For example, it would be relevant to establish bit rate conditions under which desired properties for multi-agent systems based on event-triggered control can be guaranteed.

Bibliography

[1] C. H. Botts, J. C. Spall, and A. J. Newman, "Multi-agent surveillance and tracking using cyclic stochastic gradient," in *American Control Conference*, 2016, pp. 270–275.

[2] E. Shakshuki and M. Reid, "Multi-agent system applications in healthcare: Current technology and future roadmap," *Procedia Computer Science*, vol. 52, pp. 252–261, 2015.

[3] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, and P. Van den Hof, "Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges," *Annual Reviews in Control*, vol. 43, pp. 1–64, 2017.

[4] C. Giannella, R. Bhargava, and H. Kargupta, "Multi-agent systems and distributed data mining," in *International Workshop on Cooperative Information Agents*. Springer, 2004, pp. 1–15.

[5] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous Agents and Multi-agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[6] D. Meng and K. L. Moore, "Learning to cooperate: Networks of formation agents with switching topologies," *Automatica*, vol. 64, pp. 278–293, 2016.

[7] M. Wooldridgey and P. Ciancarini, "Agent-oriented software engineering: The state of the art," in *International Workshop on Agent-Oriented Software Engineering*. Springer, 2000, pp. 55–82.

[8] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *Power Systems Conference and Exposition*, 2009, pp. 1–8.

[9] F. Dörfler, J. W. Simpson-Porco, and F. Bullo, "Breaking the hierarchy: Distributed control and economic optimality in microgrids," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 3, pp. 241–253, 2015.

[10] X. F. Xie, S. F. Smith, and G. J. Barlow, "Schedule-driven coordination for real-time traffic network control," in *International Conference on Automated Planning and Scheduling*, 2012, pp. 323–331.

[11] T. Máhr, J. Srour, M. de Weerdt, and R. Zuidwijk, "Can agents measure up? A comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 1, pp. 99–119, 2010.

[12] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. MIT Press, 2016.

[13] A. Antoniadis, I. Gijbels, and M. Nikolova, "Penalized likelihood regression for generalized linear models with non-quadratic penalties," *Annals of the Institute of Statistical Mathematics*, vol. 63, no. 3, pp. 585–615, 2011.

[14] R. Gerehberg and W. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, no. 2, pp. 237–246, 1972.

[15] J. R. Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Optics Letters*, vol. 3, no. 1, pp. 27–29, 1978.

[16] ——, "Phase retrieval algorithms: A comparison," *Applied Optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[17] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[18] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[19] T. Goldstein and C. Studer, "Phasemax: Convex phase retrieval via basis pursuit," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2675–2689, 2018.

[20] S. Bahmani and J. Romberg, "Phase retrieval meets statistical learning theory: A flexible convex relaxation," in *Artificial Intelligence and Statistics*, 2017, pp. 252–260.

[21] R. Chandra, T. Goldstein, and C. Studer, "Phasepack: A phase retrieval library," in *International Conference on Sampling Theory and Applications*, 2019, pp. 1–5.

[22] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: A contemporary overview," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[23] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[24] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[25] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, “On optimization methods for deep learning,” in *International Conference on Machine Learning*, 2011.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[27] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, “Large scale distributed deep networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[28] J. K. Kim, Q. Ho, S. Lee, X. Zheng, W. Dai, G. A. Gibson, and E. P. Xing, “STRADS: A distributed framework for scheduled model parallel machine learning,” in *European Conference on Computer Systems*, 2016, pp. 1–16.

[29] V. Hegde and S. Usmani, “Parallel and distributed deep learning,” in *Technical report, Stanford University*, 2016.

[30] J. Hermans, “On scalable deep learning and parallelizing gradient descent,” Master’s thesis, Maastricht University, 2017.

[31] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, “Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[32] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, “D^2: Decentralized training over decentralized data,” in *International Conference on Machine Learning*, 2018, pp. 4848–4856.

[33] H. Yu, R. Jin, and S. Yang, “On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization,” in *International Conference on Machine Learning*, 2019, pp. 7184–7193.

[34] N. Narodytska and S. Kasiviswanathan, “Simple black-box adversarial attacks on deep neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1310–1318.

[35] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *IEEE European Symposium on Security and Privacy*, 2016, pp. 372–387.

[36] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[37] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[38] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[39] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations*, 2017.

[40] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, 2017, pp. 39–57.

[41] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *IEEE Security and Privacy Workshops*, 2018, pp. 36–42.

[42] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Caffe model description: VGG_ CNN_ S," *https://gist.github.com/ksimonyan/fd8800eeb36e276cd6f9*, 2014.

[43] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*, 2019.

[44] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018, pp. 274–283.

[45] C. Finlay, A. M. Oberman, and B. Abbasi, "Improved robustness to adversarial examples using lipschitz regularization of the loss," *arXiv preprint arXiv:1810.00953*, 2018.

[46] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 15–26.

[47] A. N. Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *European Conference on Computer Vision*, 2018, pp. 158–174.

[48] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 934–10 944.

[49] R. Lasseter, A. Akhil, C. Marnay, J. Stephens, J. Dagle, R. Guttromsom, A. S. Meliopoulous, R. Yinger, and J. Eto, “Integration of distributed energy resources. The CERTS Microgrid Concept,” Lawrence Berkeley National Laboratory, Tech. Rep., 2002.

[50] F. Rahimi and A. Ipakchi, “Demand response as a market resource under the smart grid paradigm,” *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 82–88, 2010.

[51] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, “Coordinated scheduling of residential distributed energy resources to optimize smart home energy services,” *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 134–143, 2010.

[52] A. Abdelaziz, E. Ali, and S. A. Elazim, “Combined economic and emission dispatch solution using flower pollination algorithm,” *International Journal of Electrical Power and Energy Systems*, vol. 80, pp. 264–274, 2016.

[53] C. Sabol, R. Burns, and C. A. McLaughlin, “Satellite formation flying design and evolution,” *Journal of Spacecraft and Rockets*, vol. 38, no. 2, pp. 270–278, 2001.

[54] S. Persson, S. Veldman, and P. Bodin, “PRISMA—a formation flying project in implementation phase,” *Acta Astronautica*, vol. 65, no. 9, pp. 1360–1374, 2009.

[55] R. Faller, A. Ohndorf, B. Schlepp, and S. Eberle, “Preparation, handover, and conduction of prisma mission operations at gsoc,” in *IAF International Astronautical Congress*, 2012, pp. 1–11.

[56] A. Alam, B. Besselink, V. Turri, J. Martensson, and K. H. Johansson, “Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency,” *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, Dec 2015.

[57] R. Ringbräck, “Multi-agent autonomous target tracking using distance-based formations,” Master’s thesis, KTH Royal Institute of Technology, 2017.

[58] J. N. Tsitsiklis, “Problems in decentralized decision making and computation,” Ph.D. dissertation, MIT, Cambridge, MA, 1984.

[59] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[60] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[61] A. Nedić and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[62] Y. Nesterov, *Lectures on Convex Optimization*, 2nd ed. Springer International Publishing, 2018.

[63] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *IEEE Conference on Decision and Control*, 2008, pp. 4185–4190.

[64] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2011.

[65] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *IEEE Conference on Decision and Control*, 2012, pp. 5453–5458.

[66] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[67] T. Yang, J. Lu, D. Wu, J. Wu, G. Shi, Z. Meng, and K. H. Johansson, "A distributed algorithm for economic dispatch over time-varying directed networks with delays," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5095–5106, 2017.

[68] J. Lu and C. Y. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2348–2354, 2012.

[69] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[70] D. Jakovetić, J. M. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented Lagrangian algorithms," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, 2015.

[71] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[72] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[73] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, 2017.

[74] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3141–3155, 2018.

[75] H. Li and Z. Lin, "Revisiting extra for smooth distributed optimization," *SIAM Journal on Optimization*, vol. 30, no. 3, pp. 1795–1821, 2020.

[76] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5158–5173, 2016.

[77] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[78] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, "Geometrically convergent distributed optimization with uncoordinated step-sizes," in *American Control Conference*, 2017, pp. 3950–3955.

[79] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2566–2581, 2020.

[80] ——, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

[81] D. Jakovetić, "A unification and generalization of exact distributed first-order methods," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 31–46, 2019.

[82] F. Mansoori and E. Wei, "A flexible framework of first-order primal–dual algorithms for distributed optimization," *arXiv preprint arXiv:1912.07526*, 2019.

[83] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1329–1339, 2018.

[84] S. Pu, W. Shi, J. Xu, and A. Nedić, "A push-pull gradient method for distributed optimization in networks," in *IEEE Conference on Decision and Control*, 2018, pp. 3385–3390.

[85] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 325–330, 2018.

[86] R. Xin, C. Xi, and U. A. Khan, "FROST–Fast row-stochastic optimization with uncoordinated step-sizes," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, p. 1, 2019.

[87] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 434–448, 2018.

[88] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 994–1009, 2016.

[89] F. Saadatniaki, R. Xin, and U. A. Khan, "Optimization over time-varying directed graphs with row and column-stochastic matrices," *IEEE Transactions on Automatic Control*, 2020.

[90] M. Maros and J. Jaldén, "On the Q-linear convergence of distributed generalized ADMM under non-strongly convex function components," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 3, pp. 442–453, 2019.

[91] Y. Tian, Y. Sun, and G. Scutari, "ASY-SONATA: Achieving linear convergence in distributed asynchronous multiagent optimization," in *Annual Allerton Conference on Communication, Control, and Computing*, 2018, pp. 543–551.

[92] B. Yang and M. Johansson, "Distributed optimization and games: A tutorial overview," in *Networked Control Systems*. Springer, 2010, pp. 109–148.

[93] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[94] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[95] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[96] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[97] A. H. Sayed *et al.*, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[98] A. Nedić, "Convergence rate of distributed averaging dynamics and optimization in networks," *Foundations and Trends in Systems and Control*, vol. 2, no. 1, pp. 1–100, 2015.

[99] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[100] W. Ren and Y. Cao, *Distributed Coordination of Multi-Agent Networks: Emergent Problems, Models, and Issues*. Springer Science & Business Media, 2010.

[101] M. Zhu and S. Martínez, *Distributed Optimization-Based Control of Multi-Agent Networks in Complex Environments*. Springer, 2019.

[102] F. Bullo, *Lectures on Network Systems*. Kindle Direct Publishing, 2019, with contributions by J. Cortes, F. Dörfler, and S. Martinez. [Online]. Available: http://motion.me.ucsb.edu/book-lns

[103] P. Giselsson and A. Rantzer, *Large-Scale and Distributed Optimization*. Springer, 2018.

[104] J. Guo, G. Hug, and O. K. Tonguz, "A case for nonconvex distributed optimization in large-scale power systems," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3842–3851, 2016.

[105] G. Tychogiorgos, A. Gkelias, and K. K. Leung, "A non-convex distributed optimization framework and its application to wireless ad-hoc networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4286–4296, 2013.

[106] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[107] P. Bianchi, G. Fort, and W. Hachem, "Performance of a distributed stochastic approximation algorithm," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7405–7418, 2013.

[108] H.-T. Wai, T.-H. Chang, and A. Scaglione, "A consensus-based decentralized algorithm for non-convex optimization with application to dictionary learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 3546–3550.

[109] P. Xu, F. Roosta, and M. W. Mahoney, "Newton-type methods for non-convex optimization under inexact hessian information," *Mathematical Programming*, pp. 1–36, 2017.

[110] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2834–2848, 2018.

[111] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[112] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal–dual algorithm for fast distributed nonconvex optimization and learning over networks," in *International Conference on Machine Learning*, 2017, pp. 1529–1538.

[113] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," *arXiv preprint arXiv:1804.02729*, 2018.

[114] ——, “Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms,” *IEEE Transactions on Signal Processing*, vol. 67, no. 22, pp. 5912–5928, 2019.

[115] D. Hajinezhad and M. Hong, “Perturbed proximal primal–dual algorithm for nonconvex nonsmooth optimization,” *Mathematical Programming*, vol. 176, no. 1-2, pp. 207–245, 2019.

[116] A. Daneshmand, G. Scutari, and V. Kungurtsev, “Second-order guarantees of gradient algorithms over networks,” in *Annual Allerton Conference on Communication, Control, and Computing*, 2018, pp. 359–365.

[117] M. Hong, M. Razaviyayn, and J. Lee, “Gradient primal–dual algorithm converges to second-order stationary solution for nonconvex distributed optimization over networks,” in *International Conference on Machine Learning*, 2018, pp. 2009–2018.

[118] B. Swenson, R. Murray, H. V. Poor, and S. Kar, “Distributed gradient descent: Nonconvergence to saddle points and the stable-manifold theorem,” in *Annual Allerton Conference on Communication, Control, and Computing*, 2019, pp. 595–601.

[119] S. Vlaski and A. H. Sayed, “Distributed learning in non-convex environments–part I: Agreement at a linear rate,” *arXiv preprint arXiv:1907.01848*, 2019.

[120] ——, “Distributed learning in non-convex environments–part II: Polynomial escape from saddle-points,” *arXiv preprint arXiv:1907.01849*, 2019.

[121] P. Bianchi and J. Jakubowicz, “Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization,” *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2012.

[122] M. Zhu and S. Martínez, “An approximate dual subgradient algorithm for multi-agent non-convex optimization,” *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1534–1539, 2012.

[123] P. Di Lorenzo and G. Scutari, “NEXT: In-network nonconvex optimization,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[124] H.-T. Wai, J. Lafond, A. Scaglione, and E. Moulines, “Decentralized Frank–Wolfe algorithm for convex and nonconvex problems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5522–5537, 2017.

[125] N. Chatzipanagiotis and M. M. Zavlanos, “On the convergence of a distributed augmented Lagrangian method for nonconvex optimization,” *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4405–4420, 2017.

[126] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, "A distributed asynchronous method of multipliers for constrained nonconvex optimization," *Automatica*, vol. 103, pp. 243–253, 2019.

[127] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Mathematical Programming*, vol. 176, no. 1-2, pp. 497–544, 2019.

[128] Y. Sun, A. Daneshmand, and G. Scutari, "Convergence rate of distributed optimization algorithms based on gradient tracking," *arXiv preprint arXiv:1905.02637*, 2019.

[129] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, "Collaborative deep learning in fixed topology networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 5904–5914.

[130] J. George, T. Yang, H. Bai, and P. Gurram, "Distributed stochastic gradient method for non-convex problems with applications in supervised learning," in *IEEE Conference on Decision and Control*, 2019, pp. 5538–5543.

[131] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *International Conference on Machine Learning*, 2018, pp. 3043–3052.

[132] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *International Conference on Machine Learning*, 2019, pp. 344–353.

[133] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Advances in Neural Information Processing Systems*, 2018, pp. 7652–7662.

[134] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 8386–8397.

[135] H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Quantized push-sum for gossip and decentralized optimization over directed graphs," in *International Conference on Machine Learning*, 2020.

[136] N. Singh, D. Data, J. George, and S. Diggavi, "SQuARM-SGD: Communication-efficient momentum SGD for decentralized optimization," *arXiv preprint arXiv:2005.07041*, 2020.

[137] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," *arXiv preprint arXiv:1808.07576*, 2018.

[138] S. Lu, X. Zhang, H. Sun, and M. Hong, "GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization," in *IEEE Data Science Workshop*, 2019, pp. 315–321.

[139] J. Zhang and K. You, "Decentralized stochastic gradient tracking for empirical risk minimization," *arXiv preprint arXiv:1909.02712*, 2019.

[140] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *International Conference on Machine Learning*, 2019, pp. 3478–3487.

[141] A. Olshevsky, I. C. Paschalidis, and S. Pu, "A non-asymptotic analysis of network independence for distributed stochastic gradient descent," *arXiv preprint arXiv:1906.02702*, 2019.

[142] S. Ghadimi and G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[143] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *International Conference on Machine Learning*, 2012, pp. 1571–1578.

[144] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM Philadelphia, 2009.

[145] C. Audet and W. Hare, *Derivative-Free and Blackbox Optimization*. Springer, 2017.

[146] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404, 2019.

[147] D. Yuan and D. W. Ho, "Randomized gradient-free method for multiagent optimization over time-varying networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1342–1347, 2014.

[148] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, "Distributed zeroth order optimization over random networks: A Kiefer–Wolfowitz stochastic approximation approach," in *IEEE Conference on Decision and Control*, 2018, pp. 4951–4958.

[149] Y. Wang, W. Zhao, Y. Hong, and M. Zamani, "Distributed subgradient-free stochastic optimization algorithm for nonsmooth convex functions over time-varying networks," *SIAM Journal on Control and Optimization*, vol. 57, no. 4, pp. 2821–2842, 2019.

[150] Y. Pang and G. Hu, "Randomized gradient-free distributed optimization methods for a multi-agent system with unknown cost function," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 333–340, 2020.

[151] Y. Tang, J. Zhang, and N. Li, “Distributed zero-order algorithms for nonconvex multi-agent optimization,” *arXiv preprint arXiv:1908.11444v3*, 2020.

[152] D. Yuan, S. Xu, and J. Lu, “Gradient-free method for distributed multi-agent optimization via push-sum algorithms,” *International Journal of Robust and Nonlinear Control*, vol. 25, no. 10, pp. 1569–1580, 2015.

[153] Z. Yu, D. W. Ho, and D. Yuan, “Distributed randomized gradient-free mirror descent algorithm for constrained optimization,” *arXiv preprint arXiv:1903.04157*, 2019.

[154] A. Beznosikov, E. Gorbunov, and A. Gasnikov, “Derivative-free method for composite optimization with applications to decentralized distributed optimization,” *arXiv preprint arXiv:1911.10645v4*, 2020.

[155] D. Hajinezhad, M. Hong, and A. Garcia, “ZONE: Zeroth-order nonconvex multiagent optimization over networks,” *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3995–4010, 2019.

[156] T. Yang, J. George, J. Qin, X. Yi, and J. Wu, “Distributed finite-time least squares solver for network linear equations,” *Automatica*, vol. 113, 2020.

[157] S. Shalev-Shwartz *et al.*, “Online learning and online convex optimization,” *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[158] X. Zhou, E. Dall’Anese, L. Chen, and A. Simonetto, “An incentive-based online optimization framework for distribution grids,” *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2019–2031, 2017.

[159] S. Shahrampour and A. Jadbabaie, “Distributed online optimization in dynamic environments using mirror descent,” *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2017.

[160] N. Cesa-Bianchi, P. M. Long, and M. K. Warmuth, “Worst-case quadratic loss bounds for prediction using linear functions and gradient descent,” *IEEE Transactions on Neural Networks*, vol. 7, no. 3, pp. 604–619, 1996.

[161] C. Gentile and M. K. Warmuth, “Linear hinge loss and average margin,” in *Advances in Neural Information Processing Systems*, 1999, pp. 225–231.

[162] G. J. Gordon, “Regret bounds for prediction problems,” in *Conference on Learning Theory*, 1999, pp. 29–40.

[163] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *International Conference on Machine Learning*, 2003, pp. 928–936.

[164] A. Agarwal, O. Dekel, and L. Xiao, “Optimal algorithms for online convex optimization with multi-point bandit feedback.” in *Conference on Learning Theory*, 2010, pp. 28–40.

[165] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[166] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.

[167] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *IEEE Conference on Decision and Control*, 2016, pp. 7195–7201.

[168] L. Zhang, T. Yang, J. Yi, J. Rong, and Z.-H. Zhou, "Improved dynamic regret for non-degenerate functions," in *Advances in Neural Information Processing Systems*, 2017, pp. 732–741.

[169] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2503–2528, 2012.

[170] R. Jenatton, J. Huang, and C. Archambeau, "Adaptive algorithms for online convex optimization with long-term constraints," in *International Conference on Machine Learning*, 2016, pp. 402–411.

[171] H. Yu and M. J. Neely, "A low complexity algorithm with $O(\sqrt{T})$ regret and finite constraint violations for online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 1–24, 2020.

[172] J. Yuan and A. Lamperski, "Online convex optimization for cumulative constraints," in *Advances in Neural Information Processing Systems*, 2018, pp. 6140–6149.

[173] W. Sun, D. Dey, and A. Kapoor, "Safety-aware algorithms for adversarial contextual bandit," in *International Conference on Machine Learning*, 2017, pp. 3280–3288.

[174] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6350–6364, 2017.

[175] M. J. Neely and H. Yu, "Online convex optimization with time-varying constraints," *arXiv preprint arXiv:1702.04783*, 2017.

[176] H. Yu, M. Neely, and X. Wei, "Online convex optimization with stochastic constraints," in *Advances in Neural Information Processing Systems*, 2017, pp. 1428–1438.

[177] D. Mateos-Núnez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 1, pp. 23–37, 2014.

[178] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 417–428, 2017.

[179] A. Nedić, S. Lee, and M. Raginsky, "Decentralized online optimization with global objectives and local communication," in *American Control Conference*, 2015, pp. 4497–4503.

[180] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[181] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018.

[182] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed convex optimization on dynamic networks." *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3545–3550, 2016.

[183] S. Mannor, J. N. Tsitsiklis, and J. Y. Yu, "Online learning with sample path constraints," *Journal of Machine Learning Research*, vol. 10, pp. 569–590, 2009.

[184] S. Lee, A. Nedić, and M. Raginsky, "Stochastic dual averaging for decentralized online optimization on time-varying communication graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6407–6414, 2017.

[185] ——, "Coordinate dual averaging for decentralized online optimization with nonseparable global objectives," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 34–44, 2018.

[186] K. I. Tsianos and M. G. Rabbat, "Distributed strongly convex optimization," in *Annual Allerton Conference on Communication, Control, and Computing*, 2012, pp. 593–600.

[187] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2013.

[188] D. Yuan, A. Proutiere, and G. Shi, "Distributed online linear regression," *arXiv preprint arXiv:1902.04774*, 2019.

[189] D. Yuan, D. W. Ho, and G.-P. Jiang, "An adaptive primal–dual subgradient algorithm for online distributed constrained optimization," *IEEE Transactions on Cybernetics*, 2017.

[190] S. Lee and M. M. Zavlanos, "On the sublinear regret of distributed primal–dual algorithms for online constrained optimization," *arXiv preprint arXiv:1705.11128*, 2017.

[191] Z. Li, Z. Ding, J. Sun, and Z. Li, "Distributed adaptive convex optimization on directed graphs via continuous-time algorithms," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, 2018.

[192] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*. Courier Corporation, 2013.

[193] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[194] E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer, "Decentralised event-triggered cooperative control with limited communication," *International Journal of Control*, vol. 86, no. 9, pp. 1479–1488, 2013.

[195] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[196] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry, "On the regularization of Zeno hybrid automata," *Systems & Control Letters*, vol. 38, no. 3, pp. 141–150, 1999.

[197] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.

[198] W. Lu and T. Chen, "New approach to synchronization analysis of linearly coupled ordinary differential systems," *Physica D: Nonlinear Phenomena*, vol. 213, no. 2, pp. 214–230, 2006.

[199] ——, "A new approach to synchronization analysis of linearly coupled map lattices," *Chinese Annals of Mathematics, Series B*, vol. 28, no. 2, pp. 149–160, 2007.

[200] X. Yi, W. Lu, and T. Chen, "Pull-based distributed event-triggered consensus for multi-agent systems with directed topologies," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 1, pp. 71–79, 2017.

[201] C. W. Wu, *Synchronization in Complex Networks of Nonlinear Dynamical Systems*. World Scientific, 2007.

[202] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005.

[203] C. W. Wu, "Synchronization in networks of nonlinear dynamical systems coupled via a directed graph," *Nonlinearity*, vol. 18, no. 3, pp. 1057–1064, 2005.

[204] X. Yi, L. Yao, T. Yang, J. George, and K. H. Johansson, "Distributed optimization for second-order multi-agent systems with dynamic event-triggered communication," in *IEEE Conference on Decision and Control*, 2018, pp. 3397–3402.

[205] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[206] S. Boyd, J. Duchi, and L. Vandenberghe, "Subgradients," *https://stanford.edu/class/ee364b/lectures/subgradients_notes.pdf*, 2018.

[207] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and Trends in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[208] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer-Verlag, New York, 2007.

[209] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016, pp. 795–811.

[210] H. Zhang and L. Cheng, "Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization," *Optimization Letters*, vol. 9, no. 5, pp. 961–979, 2015.

[211] S. Shalev-Shwartz and Y. Singer, "Logarithmic regret algorithms for strongly convex repeated games," *The Hebrew University*, 2007.

[212] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 385–394.

[213] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[214] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.

[215] X. Gao, B. Jiang, and S. Zhang, "On the information-adaptive variants of the ADMM: An iteration complexity perspective," *Journal of Scientific Computing*, vol. 76, no. 1, pp. 327–363, 2018.

[216] J. Kiefer, J. Wolfowitz *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[217] S. Kar, J. M. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[218] I. Necoara, Y. Nesterov, and F. Glineur, “Linear convergence of first order methods for non-strongly convex optimization,” *Mathematical Programming*, vol. 175, no. 1-2, pp. 69–107, 2019.

[219] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[220] J. Zeng and W. Yin, “Extrapush for convex smooth decentralized optimization over directed networks,” *Journal of Computational Mathematics*, vol. 35, no. 4, pp. 383–396, 2017.

[221] C. Xi and U. A. Khan, “DEXTRA: A fast algorithm for optimization over directed graphs,” *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, 2017.

[222] S. Liang, L. Y. Wang, and G. Yin, “Exponential convergence of distributed primal–dual convex optimization algorithm without strong convexity,” *Automatica*, vol. 105, pp. 298–306, 2019.

[223] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, “Exponential convergence for distributed smooth optimization under the restricted secant inequality condition,” in *IFAC World Congress*, 2020.

[224] R. Zhang and J. Kwok, “Asynchronous distributed ADMM for consensus optimization,” in *International Conference on Machine Learning*, 2014, pp. 1701–1709.

[225] W. Deng and W. Yin, “On the global and linear convergence of the generalized alternating direction method of multipliers,” *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.

[226] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, “Distributed robust multicell coordinated beamforming with imperfect CSI: An ADMM approach,” *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2988–3003, 2012.

[227] V. Kekatos and G. B. Giannakis, “Distributed robust power system state estimation,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1617–1626, 2012.

[228] B. O’Donoghue, G. Stathopoulos, and S. Boyd, “A splitting method for optimal control,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2432–2442, 2013.

[229] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2012.

[230] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *IEEE Conference on Decision and Control*, 2012, pp. 5445–5450.

[231] A. Makhdoumi and A. Ozdaglar, "Broadcast-based distributed alternating direction method of multipliers," in *Annual Allerton Conference on Communication, Control, and Computing*, 2014, pp. 270–277.

[232] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 892–904, 2015.

[233] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2016.

[234] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 5–20, 2017.

[235] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored ADMM for decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2565–2579, 2019.

[236] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, 2020.

[237] R. Carli and M. Dotoli, "Distributed alternating direction method of multipliers for linearly constrained optimization over a network," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 247–252, 2019.

[238] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[239] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization—part I: Algorithm and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3118–3130, 2016.

[240] M. Hong, "A distributed, asynchronous, and incremental algorithm for nonconvex optimization: An ADMM approach," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 935–945, 2017.

[241] M. Schuresko and J. Cortés, "Distributed motion constraints for algebraic connectivity of robotic networks," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1-2, pp. 99–126, 2009.

[242] T. M. D. Tran and A. Y. Kibangou, "Consensus-based distributed estimation of laplacian eigenvalues of undirected graphs," in *European Control Conference*, 2013, pp. 227–232.

[243] Y. Tang and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," in *Annual Allerton Conference on Communication, Control, and Computing*, 2019, pp. 781–786.

[244] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *Journal of Machine Learning Research*, vol. 10, pp. 777–801, 2009.

[245] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2011, pp. 693–701.

[246] C. M. De Sa, C. Zhang, K. Olukotun, and C. Ré, "Taming the wild: A unified analysis of hogwild-style algorithms," in *Advances in Neural Information Processing Systems*, 2015, pp. 2674–2682.

[247] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 2737–2745.

[248] X. Lian, H. Zhang, C.-J. Hsieh, Y. Huang, and J. Liu, "A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order," in *Advances in Neural Information Processing Systems*, 2016, pp. 3054–3062.

[249] Z. Zhou, P. Mertikopoulos, N. Bambos, P. Glynn, Y. Ye, L.-J. Li, and F.-F. Li, "Distributed asynchronous optimization with unbounded delays: How slow can you go?" in *International Conference on Machine Learning*, 2018, pp. 5970–5979.

[250] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*, 2018, pp. 560–569.

[251] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Advances in Neural Information Processing Systems*, 2018, pp. 2525–2536.

[252] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 2021–2031.

[253] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 668–14 679.

[254] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," in *Conference on Machine Learning and Systems*, 2019.

[255] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5693–5700.

[256] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe, "Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization," in *International Conference on Machine Learning*, 2019, pp. 2545–2554.

[257] ——, "Local SGD with periodic averaging: Tighter analysis and adaptive synchronization," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 080–11 092.

[258] H. Yu and R. Jin, "On the computation and communication complexity of parallel SGD with dynamic batch sizes for stochastic non-convex optimization," in *International Conference on Machine Learning*, 2019, pp. 7174–7183.

[259] S. U. Stich, "Local SGD converges fast and communicates little," in *International Conference on Learning Representations*, 2019.

[260] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2015, pp. 517–520.

[261] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," *Mathematical Programming*, pp. 1–48, 2018.

[262] D. Yuan, Y. Hong, D. W. Ho, and G. Jiang, "Optimal distributed stochastic mirror descent for strongly convex optimization," *Automatica*, vol. 90, pp. 196–203, 2018.

[263] D. Jakovetic, D. Bajovic, A. K. Sahu, and S. Kar, "Convergence rates for distributed stochastic optimization over random networks," in *IEEE Conference on Decision and Control*, 2018, pp. 4238–4245.

[264] A. Fallah, M. Gurbuzbalaban, A. Ozdaglar, U. Simsekli, and L. Zhu, "Robust distributed accelerated stochastic gradient methods for multi-agent networks," *arXiv preprint arXiv:1910.08701*, 2019.

[265] S. Pu and A. Garcia, "Swarming for faster convergence in stochastic optimization," *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2997–3020, 2018.

[266] S. Pu and A. Nedić, "A distributed stochastic gradient tracking method," in *IEEE Conference on Decision and Control*, 2018, pp. 963–968.

[267] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *IEEE Conference on Decision and Control*, 2019, pp. 8353–8358.

[268] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *Available: http://yann. lecun. com/exdb/mnist*, 2010.

[269] L. Bottou, “Stochastic gradient descent tricks,” in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 421–436.

[270] R. Hooke and T. A. Jeeves, ““Direct search” solution of numerical and statistical problems,” *Journal of the ACM*, vol. 8, no. 2, pp. 212–229, 1961.

[271] J. Matyas, “Random optimization,” *Automation and Remote Control*, vol. 26, no. 2, pp. 246–253, 1965.

[272] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[273] E. H. Bergou, E. Gorbunov, and P. Richtarik, “Stochastic three points method for unconstrained smooth minimization,” *arXiv preprint arXiv:1902.03591*, 2019.

[274] A. Bibi, E. H. Bergou, O. Sener, B. Ghanem, and P. Richtarik, “A stochastic derivative-free optimization method with importance sampling: Theory and learning to control,” *arXiv preprint arXiv:1902.01272*, 2019.

[275] E. Gorbunov, A. Bibi, O. Sener, E. H. Bergou, and P. Richtárik, “A stochastic derivative free optimization method with momentum,” in *International Conference on Learning Representations*, 2020.

[276] D. Golovin, J. Karro, G. Kochanski, C. Lee, X. Song *et al.*, “Gradientless descent: High-dimensional zeroth-order optimization,” *arXiv preprint arXiv:1911.06317*, 2019.

[277] M. Marazzi and J. Nocedal, “Wedge trust region methods for derivative free optimization,” *Mathematical Programming*, vol. 91, no. 2, pp. 289–305, 2002.

[278] A. R. Conn, K. Scheinberg, and L. N. Vicente, “Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points,” *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 387–415, 2009.

[279] K. Scheinberg and P. L. Toint, “Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization,” *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3512–3532, 2010.

[280] O. Shamir, “An optimal algorithm for bandit and zero-order convex optimization with two-point feedback,” *Journal of Machine Learning Research*, vol. 18, no. 52, pp. 1–11, 2017.

[281] ——, “On the complexity of bandit and derivative-free stochastic convex optimization,” in *Conference on Learning Theory*, 2013, pp. 3–24.

[282] F. Bach and V. Perchet, “Highly-smooth zero-th order online optimization,” in *Conference on Learning Theory*, 2016, pp. 257–283.

[283] H. Ye, Z. Huang, C. Fang, C. J. Li, and T. Zhang, “Hessian-aware zeroth-order optimization for black-box adversarial attack,” *arXiv preprint arXiv:1812.11377*, 2018.

[284] E.-V. Vlatakis-Gkaragkounis, L. Flokas, and G. Piliouras, “Efficiently avoiding saddle points with zero order methods: No gradients required,” in *Advances in Neural Information Processing Systems*, 2019, pp. 10 066–10 077.

[285] D. Kozak, S. Becker, A. Doostan, and L. Tenorio, “A stochastic subspace approach to gradient-free optimization in high dimensions,” *arXiv preprint arXiv:2003.02684*, 2020.

[286] S. Liu, X. Li, P.-Y. Chen, J. Haupt, and L. Amini, “Zeroth-order stochastic projected gradient descent for nonconvex optimization,” in *IEEE Global Conference on Signal and Information Processing*, 2018, pp. 1179–1183.

[287] S. Liu, P.-Y. Chen, X. Chen, and M. Hong, “signSGD via zeroth-order oracle,” in *International Conference on Learning Representations*, 2019.

[288] Y. Zhang, Y. Zhou, K. Ji, and M. M. Zavlanos, “Improving the convergence rate of one-point zeroth-order optimization using residual feedback,” *arXiv preprint arXiv:2006.10820*, 2020.

[289] K. Balasubramanian and S. Ghadimi, “Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates,” in *Advances in Neural Information Processing Systems*, 2018, pp. 3455–3464.

[290] C. Jin, L. T. Liu, R. Ge, and M. I. Jordan, “On the local minima of the empirical risk,” in *Advances in Neural Information Processing Systems*, 2018, pp. 4896–4905.

[291] S. Ghadimi, G. Lan, and H. Zhang, “Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization,” *Mathematical Programming*, vol. 155, no. 1-2, pp. 267–305, 2016.

[292] E. Kazemi and L. Wang, “A proximal zeroth-order algorithm for nonconvex nonsmooth problems,” in *Annual Allerton Conference on Communication, Control, and Computing*, 2018, pp. 64–71.

[293] B. Gu, Z. Huo, C. Deng, and H. Huang, “Faster derivative-free stochastic algorithm for shared memory machines,” in *International Conference on Machine Learning*, 2018, pp. 1812–1821.

[294] C. Fang, C. J. Li, Z. Lin, and T. Zhang, “Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator,” in *Advances in Neural Information Processing Systems*, 2018, pp. 689–699.

[295] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, “Zeroth-order stochastic variance reduction for nonconvex optimization,” in *Advances in Neural Information Processing Systems*, 2018, pp. 3727–3737.

[296] E. Gorbunov, P. Dvurechensky, and A. Gasnikov, "An accelerated method for derivative-free smooth stochastic convex optimization," *arXiv preprint arXiv:1802.09022*, 2018.

[297] L. Liu, M. Cheng, C.-J. Hsieh, and D. Tao, "Stochastic zeroth-order optimization via variance reduction method," *arXiv preprint arXiv:1805.11811*, 2018.

[298] F. Huang, B. Gu, Z. Huo, S. Chen, and H. Huang, "Faster gradient-free proximal stochastic methods for nonconvex nonsmooth optimization," in *AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1503–1510.

[299] K. Ji, Z. Wang, Y. Zhou, and Y. Liang, "Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization," in *International Conference on Machine Learning*, 2019, pp. 3100–3109.

[300] F. Huang, L. Tao, and S. Chen, "Accelerated stochastic gradient-free and projection-free methods," in *International Conference on Machine Learning*, 2020.

[301] Y. Chen, A. Orvieto, and A. Lucchi, "An accelerated DFO algorithm for finite-sum convex functions," in *International Conference on Machine Learning*, 2020.

[302] H. Gao and H. Huang, "Can stochastic zeroth-order Frank–Wolfe method converge faster for non-convex problems?" in *International Conference on Machine Learning*, 2020.

[303] H. Cai, D. Mckenzie, W. Yin, and Z. Zhang, "Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling," *arXiv preprint arXiv:2003.13001*, 2020.

[304] P. Nazari, D. A. Tarzanagh, and G. Michailidis, "Adaptive first- and zeroth-order methods for weakly convex stochastic optimization problems," *arXiv preprint arXiv:2005.09261*, 2020.

[305] A. K. Sahu, M. Zaheer, and S. Kar, "Towards gradient free and projection free stochastic optimization," in *International Conference on Artificial Intelligence and Statistics*, 2019, pp. 3468–3477.

[306] Y. Wang, S. Du, S. Balakrishnan, and A. Singh, "Stochastic zeroth-order optimization in high dimensions," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1356–1365.

[307] X. Chen, S. Liu, K. Xu, X. Li, X. Lin, M. Hong, and D. Cox, "ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization," in *Advances in Neural Information Processing Systems*, 2019, pp. 7204–7215.

[308] F. Huang, S. Gao, S. Chen, and H. Huang, "Zeroth-order stochastic alternating direction method of multipliers for nonconvex nonsmooth optimization," in *International Conference on Artificial Intelligence and Statistics*, 2019, pp. 2549–2555.

[309] F. Huang, S. Gao, J. Pei, and H. Huang, “Nonconvex zeroth-order stochastic ADMM methods with lower function query complexity,” *arXiv preprint arXiv:1907.13463*, 2019.

[310] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[311] T. Chen and G. B. Giannakis, “Bandit convex optimization for scalable and dynamic IoT management,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1276–1286, 2019.

[312] S. Paternain and A. Ribeiro, “Online learning of feasible strategies in unknown environments,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2807–2822, 2017.

[313] E. C. Hall and R. M. Willett, “Online convex optimization in dynamic environments,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[314] N. Littlestone and M. K. Warmuth, “The weighted majority algorithm,” *Information and Computation*, vol. 108, no. 2, pp. 212–261, 1994.

[315] S. Paternain, S. Lee, M. M. Zavlanos, and A. Ribeiro, “Distributed constrained online learning,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 3486–3499, 2020.

[316] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, “Composite objective mirror descent.” in *Conference on Learning Theory*, 2010, pp. 14–26.

[317] V. Dani, S. M. Kakade, and T. P. Hayes, “The price of bandit information for online optimization,” in *Advances in Neural Information Processing Systems*, 2008, pp. 345–352.

[318] J. D. Abernethy, E. Hazan, and A. Rakhlin, “Competing in the dark: An efficient algorithm for bandit linear optimization,” in *Conference on Learning Theory*, 2008, pp. 263–273.

[319] ——, “Interior-point methods for full-information and bandit online learning,” *IEEE Transactions on Information Theory*, vol. 58, no. 7, pp. 4164–4175, 2012.

[320] A. Saha and A. Tewari, “Improved regret guarantees for online smooth convex optimization with bandit feedback,” in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 636–642.

[321] E. Hazan and K. Levy, “Bandit convex optimization: Towards tight bounds,” in *Advances in Neural Information Processing Systems*, 2014, pp. 784–792.

[322] S. Bubeck, O. Dekel, T. Koren, and Y. Peres, “Bandit convex optimization: $\sqrt{T}$ regret in one dimension,” in *Conference on Learning Theory*, 2015, pp. 266–278.

[323] S. Bubeck and R. Eldan, “Multi-scale exploration of convex functions and bandit convex optimization,” in *Conference on Learning Theory*, 2016, pp. 583–589.

[324] E. Hazan and Y. Li, “An optimal algorithm for bandit convex optimization,” *arXiv preprint arXiv:1603.04350*, 2016.

[325] T. Yang, L. Zhang, R. Jin, and J. Yi, “Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient,” in *International Conference on Machine Learning*, 2016, pp. 449–457.

[326] T. Tatarenko and M. Kamgarpour, “Minimizing regret in bandit online optimization in unconstrained and constrained action spaces,” *arXiv preprint arXiv:1806.05069*, 2018.

[327] I. Shames, D. Selvaratnam, and J. H. Manton, “Online optimization using zeroth order oracles,” *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 31–36, 2019.

[328] T. Chen and G. B. Giannakis, “Bandit convex optimization for scalable and dynamic IoT management,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1276–1286, 2018.

[329] X. Cao and K. R. Liu, “Online convex optimization with time-varying constraints and bandit feedback,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2665–2680, 2019.

[330] D. Yuan, D. W. Ho, Y. Hong, and G. Jiang, “Online bandit convex optimization over a network,” in *Chinese Control Conference*, 2016, pp. 8090–8095.

[331] D. Yuan, A. Proutiere, and G. Shi, “Distributed online optimization with long-term constraints,” *arXiv preprint arXiv:1912.09705*, 2019.

[332] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1996.

[333] W. Ren, R. W. Beard, and E. M. Atkins, “A survey of consensus problems in multi-agent coordination,” in *American Control Conference*, 2005, pp. 1859–1864.

[334] W. Ren and R. W. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[335] M. Cao, A. S. Morse, and B. D. Anderson, “Agreeing asynchronously,” *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1826–1838, 2008.

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

[337] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[338] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 213–224, 2010.

[339] W. Lu and T. Chen, "Synchronization analysis of linearly coupled networks of discrete time systems," *Physica D: Nonlinear Phenomena*, vol. 198, no. 1, pp. 148–168, 2004.

[340] T. Yang, S. Roy, Y. Wan, and A. Saberi, "Constructing consensus controllers for networks with identical general linear agents," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1237–1256, 2011.

[341] B. Liu, W. Lu, and T. Chen, "Consensus in networks of multiagents with switching topologies modeled as adapted stochastic processes," *SIAM Journal on Control and Optimization*, vol. 49, no. 1, pp. 227–253, 2011.

[342] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[343] K. You and L. Xie, "Network topology and communication data rate for consensusability of discrete-time multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2262–2275, 2011.

[344] J. Wei, X. Yi, H. Sandberg, and K. H. Johansson, "Nonlinear consensus protocols with applications to quantized communication and actuation," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 598–608, 2018.

[345] K. J. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," in *IFAC World congress*, vol. 11, 1999, pp. 301–306.

[346] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[347] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conference on Decision and Control*, 2012, pp. 3270–3285.

[348] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1992–1997, 2015.

[349] S. Hu, D. Yue, X. Yin, X. Xie, and Y. Ma, "Adaptive event-triggered control for nonlinear discrete-time systems," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 18, pp. 4104–4125, 2016.

[350] C. De Persis and P. Frasca, "Robust self-triggered coordination with ternary controllers," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3024–3038, 2013.

[351] X. Ge and Q.-L. Han, "Distributed formation control of networked multi-agent systems using a dynamic event-triggered communication mechanism," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8118–8127, 2017.

[352] V. Dolk, D. P. Borgers, and W. Heemels, "Output-based and decentralized dynamic event-triggered control with guaranteed $\mathcal{L}_p$-gain performance and Zeno-freeness," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 34–49, 2017.

[353] V. Dolk, P. Tesi, C. De Persis, and W. Heemels, "Event-triggered control systems under denial-of-service attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 93–105, 2017.

[354] V. Dolk and M. Heemels, "Event-triggered control systems under packet losses," *Automatica*, vol. 80, pp. 143–155, 2017.

[355] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125–2132, 2013.

[356] W. Zhu, Z.-P. Jiang, and G. Feng, "Event-based consensus of multi-agent systems with general linear models," *Automatica*, vol. 50, no. 2, pp. 552–558, 2014.

[357] C. Nowzari and J. Cortés, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.

[358] X. Yi, W. Lu, and T. Chen, "Distributed event-triggered consensus for multi-agent systems with directed topologies," in *Chinese Control and Decision Conference*, 2016, pp. 807–813.

[359] Z. Sun, N. Huang, B. D. Anderson, and Z. Duan, "A new distributed Zeno-free event-triggered algorithm for multi-agent consensus," in *IEEE Conference on Decision and Control*, 2016, pp. 3444–3449.

[360] X. Meng, L. Xie, and Y. C. Soh, "Event-triggered output regulation of heterogeneous multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4429–4434, 2018.

[361] M. Velasco, J. Fuertes, and P. Marti, "The self triggered task model for real-time control systems," in *Work-in-Progress Session of the 24th IEEE Real-Time Systems Symposium*, vol. 384, 2003, pp. 67–70.

[362] X. Wang and M. D. Lemmon, "Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 452–467, 2009.

[363] A. Anta and P. Tabuada, “To sample or not to sample: Self-triggered control for nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[364] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, “Multi-agent trajectory tracking with self-triggered cloud access,” in *IEEE Conference on Decision and Control*, 2016, pp. 2207–2214.

[365] Y. Wang, W. X. Zheng, and H. Zhang, “Dynamic event-based control of nonlinear stochastic systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6544–6551, 2017.

[366] D. Yang, W. Ren, X. Liu, and W. Chen, “Decentralized event-triggered consensus for linear multi-agent systems under general directed graphs,” *Automatica*, vol. 69, pp. 242–249, 2016.

[367] Y. Fan, G. Feng, Y. Wang, and C. Song, “Distributed event-triggered control of multi-agent systems with combinational measurements,” *Automatica*, vol. 49, no. 2, pp. 671–675, 2013.

[368] Y. Fan, L. Liu, G. Feng, and Y. Wang, “Self-triggered consensus for multi-agent systems with Zeno-free triggers,” *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2779–2784, 2015.

[369] W. Hu, L. Liu, and G. Feng, “Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy,” *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1914–1924, 2017.

[370] X. Meng, L. Xie, Y. C. Soh, C. Nowzari, and G. J. Pappas, “Periodic event-triggered average consensus over directed graphs,” in *IEEE Conference on Decision and Control*, 2015, pp. 4151–4156.

[371] Y. Li, J. Xiang, and W. Wei, “Consensus problems for linear time-invariant multi-agent systems with saturation constraints,” *IET Control Theory & Applications*, vol. 5, no. 6, pp. 823–829, 2011.

[372] Z. Meng, Z. Zhao, and Z. Lin, “On global leader-following consensus of identical linear dynamic systems subject to actuator saturation,” *Systems & Control Letters*, vol. 62, no. 2, pp. 132–142, 2013.

[373] T. Yang, Z. Meng, D. V. Dimarogonas, and K. H. Johansson, “Global consensus for discrete-time multi-agent systems with input saturation constraints,” *Automatica*, vol. 50, no. 2, pp. 499–506, 2014.

[374] Q. Wang and C. Sun, “Conditions for consensus in directed networks of agents with heterogeneous output saturation,” *IET Control Theory & Applications*, vol. 10, no. 16, pp. 2119–2127, 2016.

[375] Y.-H. Lim and H.-S. Ahn, "Consensus with output saturations," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5388–5395, 2017.

[376] J. Wei, A. R. Everts, M. K. Camlibel, and A. J. van der Schaft, "Consensus dynamics with arbitrary sign-preserving nonlinearities," *Automatica*, vol. 83, pp. 226–233, 2017.

[377] G. A. Kiener, D. Lehmann, and K. H. Johansson, "Actuator saturation and anti-windup compensation in event-triggered control," *Discrete Event Dynamic Systems*, vol. 24, no. 2, pp. 173–197, 2014.

[378] Y. Xie and Z. Lin, "Event-triggered global stabilization of multiple integrator systems with bounded controls," in *American Control Conference*, 2017.

[379] X. Wu and T. Yang, "Distributed constrained event-triggered consensus: $\mathcal{L}_2$ gain design result," in *Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 5420–5425.

[380] X. Wang, H. Su, X. Wang, and G. Chen, "Fully distributed event-triggered semiglobal consensus of multi-agent systems with input saturation," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5055–5064, 2017.

[381] T. Chen, X. Liu, and W. Lu, "Pinning complex networks by a single controller," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1317–1326, 2007.

[382] H. K. Khalil, *Nonlinear Systems, 3rd.* Prentice-Hall, New Jersey, 2002.

[383] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[384] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10-11, pp. 1002–1033, 2007.

[385] W. Dong and J. A. Farrell, "Cooperative control of multiple nonholonomic mobile agents," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1434–1448, 2008.

[386] T. H. Van den Broek, N. van de Wouw, and H. Nijmeijer, "Formation control of unicycle mobile robots: A virtual structure approach," in *IEEE Conference on Decision and Control, held jointly with Chinese Control Conference*, 2009, pp. 8328–8333.

[387] J. Li, W. Ren, and S. Xu, "Distributed containment control with multiple dynamic leaders for double-integrator dynamics using only position measurements," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1553–1559, 2012.

[388] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[389] K.-K. Oh and H.-S. Ahn, "Formation control of mobile agents based on inter-agent distance dynamics," *Automatica*, vol. 47, no. 10, pp. 2306–2312, 2011.

[390] G. Wen, Z. Duan, W. Ren, and G. Chen, "Distributed consensus of multi-agent systems with general linear node dynamics and intermittent communications," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2438–2457, 2014.

[391] D. V. Dimarogonas and K. J. Kyriakopoulos, "On the rendezvous problem for multiple nonholonomic agents," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 916–922, 2007.

[392] ——, "A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems," *Automatica*, vol. 44, no. 10, pp. 2648–2654, 2008.

[393] D. V. Dimarogonas and K. H. Johansson, "On the stability of distance-based formation control," in *IEEE Conference on Decision and Control*, 2008, pp. 1200–1205.

[394] ——, "Further results on the stability of distance-based multi-robot formations," in *American Control Conference*, 2009, pp. 2972–2977.

[395] ——, "Stability analysis for multi-agent systems using the incidence matrix: Quantized communication and formation control," *Automatica*, vol. 46, no. 4, pp. 695–700, 2010.

[396] K.-K. Oh and H.-S. Ahn, "Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1809–1820, 2014.

[397] D. P. Spanos and R. M. Murray, "Robust connectivity of networked vehicles," in *IEEE Conference on Decision and Control*, vol. 3, 2004, pp. 2893–2898.

[398] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectivity preserving state agreement for multiple unicycles," in *American Control Conference*, 2007, pp. 1179–1184.

[399] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 812–816, 2007.

[400] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2828–2833, 2010.

[401] Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1827–1832, 2012.

[402] D. Boskos and D. V. Dimarogonas, "Robust connectivity analysis for multi-agent systems," in *IEEE Conference on Decision and Control*, 2015, pp. 6767–6772.

[403] H. Yu and P. J. Antsaklis, "Formation control of multi-agent systems with connectivity preservation by using both event-driven and time-driven communication," in *IEEE Conference on Decision and Control*, 2012, pp. 7218–7223.

[404] Y. Fan and G. Hu, "Connectivity-preserving rendezvous of multi-agent systems with event-triggered controllers," in *IEEE Conference on Decision and Control*, 2015, pp. 234–239.

[405] B. Mohar, "Eigenvalues, diameter, and mean distance in graphs," *Graphs and Combinatorics*, vol. 7, no. 1, pp. 53–64, 1991.

[406] V. Kucera, "A contribution to matrix quadratic equations," *IEEE Transactions on Automatic Control*, vol. 17, no. 3, pp. 344–347, 1972.

[407] H. Li, X. Liao, T. Huang, and W. Zhu, "Event-triggering sampling based leader-following consensus in second-order multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1998–2003, 2015.

[408] C. Nowzari and J. Cortés, "Zeno-free, distributed event-triggered communication and control for multi-agent average consensus," in *American Control Conference*. IEEE, 2014, pp. 2148–2153.

[409] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2010.

[410] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2011.

[411] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

[412] L. Su and N. Vaidya, "Multi-agent optimization in the presence of Byzantine adversaries: Fundamental limits," in *American Control Conference*, 2016, pp. 7183–7188.

[413] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

[414] C. Zhao, J. He, and Q.-G. Wang, "Resilient distributed optimization algorithm against adversarial attacks," *IEEE Transactions on Automatic Control*, 2019.

[415] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *arXiv preprint arXiv:2007.06844*, 2020.

[416] P. Tallapragada and J. Cortés, "Event-triggered stabilization of linear systems under bounded bit rates," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1575–1589, 2015.

[417] J. Pearson, J. P. Hespanha, and D. Liberzon, "Control with minimal cost-per-symbol encoding and quasi-optimality of event-based encoders," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2286–2301, 2016.

[418] Q. Ling, "Bit rate conditions to stabilize a continuous-time scalar linear system based on event triggering," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4093–4100, 2017.

[419] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[420] M. J. Khojasteh, P. Tallapragada, J. Cortés, and M. Franceschetti, "The value of timing information in event-triggered control," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 925–940, 2019.

[421] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[422] P. Yu and D. V. Dimarogonas, "Explicit computation of sampling period in periodic event-triggered multi-agent control under limited data rate," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 4, pp. 1366–1378, 2019.