Energy Management in Wireless Communications: From Convex Optimization to Machine Learning

by

Yanjie Dong

M.A.Sc., The University of British Columbia, Canada, 2016
B.Eng., Xidian University, P.R. China, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

Energy Management in Wireless Communications: From Convex Optimization to Machine Learning

submitted by Yanjie Dong in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

in Electrical and Computer Engineering

Examining Committee:

Victor C. M. Leung, Department of Electrical and Computer Engineering, UBC
Co-supervisor

Julian Cheng, School of Engineering, UBC, Okanagan Campus
Co-supervisor

Md. Jahangir Hossain, School of Engineering, UBC, Okanagan Campus
Supervisory Committee Member

Vijay K. Bhargava, Department of Electrical and Computer Engineering, UBC
University Examiner

Michael F. Friedlander, Department of Computer Science and Department of Mathematics, UBC
University Examiner

Additional Supervisory Committee Members:

Lutz Lampe, Department of Electrical and Computer Engineering, UBC
Supervisory Committee Member

Abstract

Ever-increasing energy consumption of network infrastructures motivates wireless operators to exploit renewable energy resources (e.g., sunlight and wind) to network infrastructure. When solely powered by weather-dependent renewable energy, a base station can experience power outages. Therefore, a smart-grid powered communication system (SGPCS) is proposed to avoid the power outage at base stations. To successfully apply renewable energy to an SGPCS, wireless operators need to develop energy management algorithms that can handle the unpredictable and intermittent arrival of renewable energy. Since machine learning algorithms are inherently designed for problems with random sources (i.e., stochastic optimization problems), we start by investigating machine learning algorithms for different stochastic optimization problems. Then, we adapt the potential machine learning algorithms to the long-term grid-energy expenditure minimization problem under various practical constraints.

Using the finite-sample analytical methods, we quantify the convergence rates of the proposed offline learning and online learning algorithms. Based on the derived convergence rates, we have the following findings. When faulty users exist in the federated learning framework, our proposed fault-resilient proximal gradient and local fault-resilient proximal gradient algorithms require fewer communication rounds than the state-of-the-art benchmarks. Therefore, they are more energy-efficient than the benchmarks. The proposed linear function approximation based decentralized Q-learning converges as fast as the tabular Q-learning while retaining robustness to the large state and action spaces. Based on Lyapunov learning algorithms, we can successfully integrate the renewable energy in single-cell and multi-cell SGPCSs. Moreover, our proposed two time-scale resource allocation algorithm can trade the grid-energy expenditure for access delay of user equipments in single-cell SGPCS. Our proposed two time-scale resource allocation algorithm can trade grid-energy expenditure for the end-to-end delay of user equipments in multi-cell SGPCS.

Lay Summary

Wireless operators have deployed a massive number of base stations, which then causes their energy bills to surge. Therefore sustainable solutions to integrate renewable energy and smart grid into the network infrastructure are urgently needed. This thesis investigates machine learning algorithms and applies them to address several practical problems in smart-grid powered communication systems so that renewable energy can be used efficiently with less waste. This thesis suggests the applicable scenarios of proposed algorithms and reveals that the tradeoff relations between grid-energy expenditure and users' delay. In other words, wireless operators can use our proposed algorithms to reduce the grid-energy expenditure at the expense of users' delay.

Preface

The thesis is based on the research works conducted under the supervision of Professor Victor C. M. Leung and Professor Julian Cheng.

For all the chapters, I performed literature review, problem formulation, theoretical analysis, and numerical results. Moreover, I prepared related manuscripts. My supervisors helped me to validate the analytical and numerical results, and improved the presentation of the manuscripts. All the manuscripts were also co-authored by Professor Md. Jahangir Hossain, who is also a member of my supervisory committee. Professor Md. Jahangir Hossain helped me to improve the system models as well as the overall presentation of the manuscripts. Dr. Tianyi Chen and Professor Georgios B. Giannakis joined the discussion on research issues and improved the technical presentations of [J1]. Dr. Gang Wang and Professor Georgios B. Giannakis provided suggestions during the derivation of theoretical results and polished the presentations of [C1].

Following is the list of the journal and conference publications related to each chapter of the thesis. Both published and submitted articles are provided in this list.

Publication Related to Chapter 3

J1: Y. Dong, G. B. Giannakis, T. Chen, J. Cheng, M. J. Hossain, and V. C. M. Leung, "Communication-efficient fault-resilient federated learning over heterogeneous datasets," *submitted for possible publication*, Aug. 2020.

J2: Y. Dong, J. Cheng, M. J. Hossain, and V. C. M. Leung, "Secure distributed on-device learning networks with Byzantine adversaries," *IEEE Network*, vol. 33, no. 6, pp. 180–187, Nov.–Dec. 2019.

Publication Related to Chapter 4

C1: Y. Dong, G. Wang, J. Cheng, M. J. Hossain, G. B. Giannakis, and V. C. M. Leung, "A unifying finite-sample analysis of decentralized Q-learning with linear function approximation," *submitted for possible publication*, June 2020.

Publication Related to Chapter 5

J3: Y. Dong, M. J. Hossain, J. Cheng and V. C. M. Leung, "Dynamic cross-layer beamforming in hybrid powered communication systems with harvest-use-trade strategy," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8011–8025, Dec. 2017.

Publication Related to Chapter 6

J4: Y. Dong, M. J. Hossain, J. Cheng, and V. C. M. Leung, "Cross-layer scheduling and beamforming in smart-grid powered cellular networks with heterogeneous energy coordination," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2711–2725, May 2020.

C2: Y. Dong, M. J. Hossain, J. Cheng and V. C. M. Leung, "Cross-layer scheduling and beamforming in smart grid powered small-cell networks," in *Proceedings of IEEE International Conference on Communications*, Shanghai, China, pp. 1–6, June 2019.

Table of Contents

List of Figures

List of Symbols

Operator Descriptions

$g(\cdot)$	Auxiliary Function
$\boldsymbol{\Lambda}$	Averaging Operator
$\mathbb{B}[\cdot]$	Bellman Operator
$\boldsymbol{X}^{\mathrm{H}}$	Conjugate-Transpose of Matrix $\boldsymbol{X}$
$\boldsymbol{\Delta}$	Difference Operator
$\mathbb{E}[\cdot]$	Expectation Over All Random Sources
$\mathbb{E}_X[\cdot]$	Expectation Over Random Variable X
∇	Gradient Operator
$\langle \cdot, \cdot \rangle$	Inner Product Operator
$\otimes$	Kronecker Product
$\|\cdot\|$	ℓ_2-norm of a vector
$\mathbf{1}_{N\times N}$	N-Dimension All-One Matrix
$\boldsymbol{I}_N$	N-Dimension Identity Matrix
$O(\cdot)$	Polynomial of Input
$\mathrm{proj}_{\mathcal{Q}}\{\cdot\}$	Projection Over Subspace $\mathcal{Q}$
$\mathrm{vec}(\cdot)$	Stacking Input to a Vector
$[x]^+$	$\max\{x, 0\}$

Functions

$F(\cdot)$	Loss Function (e.g. Grid-Energy Expenditure)
$\mathbb{C}(\cdot)$	Lyapunov function
$f_0(\cdot)$	Regularization Function of Server
$f_n(\cdot)$	Loss Function of UE n
$p_n(\cdot)$	Penalty Function of UE n

$P(\cdot)$	Probability Function
$Q(\cdot)$	Q-Function
$r(\cdot)$	Reward Function (e.g. Packet Rate in Chapter 4 or Data Rate in Chapter 5)

Constants

k and K	Index of Slot and Total Slots
i and I	Index of Frame and Total Frames
m and j	Indices of Base Stations
n and l	Indices of (Wireless) UEs
$\boldsymbol{B}$	Mixing Matrix
$\boldsymbol{\Phi}$	Feature Matrix
$\boldsymbol{g}$	Auxiliary Gradient
$\boldsymbol{u}$ and $\boldsymbol{v}$	Auxiliary Vectors
$\boldsymbol{w}$	Optimization Variable (e.g. Beamforming Vector)
$\mathcal{A}_n$ and $\mathcal{A}$	Action Space of UE n and Joint Action Space
$\mathcal{N}_m^{\mathrm{ACT}}$	Set of Active Wireless UE of BST m
$\mathcal{Q}$	Linear Subspace Induced by Feature Matrix $\boldsymbol{\Phi}$
$\mathcal{S}$	State Space
$A_{n,k}$ and A_k	Random Action of UE n and Joint Action Per Slot k
G	Bound of Gradient
V	Lyapunov Control Parameter
L	Lipschitz Constant
M	Number of Base Stations
N	Number of (Wireless) UEs
N_{B}	Number of Byzantine UEs
N_{R}	Number of Reliable UEs
N_m	Number of Wireless UEs Connected to BST m
$P^{\max}$	Maximum Transmit Power of BST in mW
P^{SP}	Power of Based Band Processing in mW
S_k	Random State Per Slot k

T	Number of Slots Per Frame
C	Number of bits per packet
c	Auxiliary Constant Specified in Each Chapter
d	Dimensional of Optimization Variable
f_c	Carrier Frequency in GHz
q^{A}	Backlog of Access Queue
q^{U}	Backlog of Processing Queue
α and β	Stepsizes
γ_b and γ_s	Prices of Purchasing and Selling a Unit Energy in Cents/slot/mW
γ_n^{REQ}	Required Signal-to-Interference-Plus-Noise Ratio in dB
δ	Strongly Convex Constant
ϵ	Small Positive Constant
ζ	Gradient Noise
η	Efficiency of Power Amplifier
ι	Set of Random Sources
κ	Link Distance in meters
λ	Auxiliary Polynomial Specified in Each Chapter
μ_n and $\boldsymbol{\mu}$	Behavior Policy of UE n and Joint Behavior Policy
ν_n	Traffic Arrival Rate of Wireless UE n
π	Stationary Distribution of Markov Decision Process
$\varpi_{m\to l}$	Exchanged Amount of Energy From BST m to l
σ^2	Noise Power in mW
τ	Index of Iteration in a Slot
$\boldsymbol{\phi}(s,\boldsymbol{a})$	Feature Vector for State s and Joint Action $\boldsymbol{a}$
ω	Pathloss in dB

List of Abbreviations

Abbreviations	Definitions
AWGN	Additive White Gaussian Noise
BST	Base Station
CSI	Channel State Information
CSCG	Circularly Symmetric Complex Gaussian
ESI	Energy State Information
FRPG	Fault Resilient Proximal Gradient
5G	Fifth Generation
GeoMed	Geometric Median
GEE	Grid Energy Expenditure
KKT	Karush-Khun-Tucker
LAG	Lazy Aggregation
LFA	Linear Function Approximation
LFRPG	Local Fault-Resilient Proximal Gradient
MDP	Markov Decision Process
MCS	Modulation and Coding Scheme
MARL	Multi-Agent Reinforcement Learning
MISO	Multiple-Input-Single-Output
QoS	Quality-of-Service
RSA	Robust Stochastic Aggregation
RHS	Right-Hand Side
SINR	Signal-to-Interference-Plus-Noise Ratio
SGPCS	Smart-Grid Powered Communication System
SARSA	State-Action-Reward-State-Action

SGD	Stochastic Gradient Descent
SABF	Successive Approximation Beamforming
TD	Temporal Difference
TAEGEE	Time-Average Expectation of Grid-Energy Expenditure
TSUBE	Two Time-Scale UE Scheduling, Beamforming and Energy Trading
UE	User Equipment
WOLPE	Without Local Power Exchanging
ZFBF	Zero-Forcing Beamforming

Acknowledgements

I want to devote the first paragraph of acknowledgment to The University of British Columbia (UBC) for granting me an opportunity to pursue my Ph.D. degree and for helping me to work with world-renowned professors and researchers.

I owe my deepest gratitude to my supervisor Professor Victor C. M. Leung for his invaluable insight on my thesis topic and his tireless encouragement throughout my Ph.D. journey. His thoughtfulness, dedication, and guidance have made this dissertation a reality. He provided me with the freedom to pursue research topics of my interest and helped me grow up from a wandering graduate student to a professional researcher. I have been truly fortunate to have Professor Victor C. M. Leung as my supervisor, and it has been a great pleasure working under his supervision.

I want to express my gratitude to my co-supervisor Professor Julian Cheng for his collaboration and suggestions through my research. Moreover, I would like to express my appreciation to Professor Md. Jahangir Hossain, who is one of my supervisory committee members and was also my co-supervisor during my M.A.Sc. I gained many helpful insights and analytical perspectives from them. Besides, their critical thinking on research and scientific writing will continue to influence me in my future career. I would thank Dr. Tianyi Chen, Dr. Gang Wang, Dr. Ahmed El Shafie, Professor Naofal Al-Dhahir, and Professor Georgios B. Giannakis for their warm hospitality and constructive suggestions during our discussions.

I would like to thank Professor Lutz Lampe and Professor David Michelson for their willingness to serve on my departmental examination committee, and for providing me their valuable suggestions and critical comments during my departmental oral examinations. I am sincerely thankful to Professor Wei Yu for providing several insightful comments as the external examiner of my thesis. I would also like to thank Professor Vijay K. Bhargava, Professor Michael F. Friedlander, and Professor Ryozo Nagamune for their willingness to participate in my thesis examination

committee.

My life at UBC would never have been so memorable without the brilliant colleagues and friends. I am especially thankful to the friends in room X327 of ICICS Building and room 2255 of EME Building in UBC for both technical and non-technical conversations. I am deeply indebted to my parents for their endless love and unconditional support. I owe everything to them, and none of my dreams could be achieved without their belief, encouragement, and sacrifice during all these years. I would not be where I am today without them.

This work was supported in part by a Four-Year Fellowship of UBC, in part by the Natural Sciences and Engineering Research Council of Canada, and in part by Mitacs.

Chapter 1

Introduction

Wireless traffic is estimated to exceed 131 exabytes per month by 2024 [1]. The huge volume of wireless traffic requires a dense deployment of base stations (BSTs) [2]. It was reported that the number of the fifth-generation (5G) BSTs reached 130,000 by the end of 2019 in China[1]. The tremendous number of BSTs leads to a significant amount of greenhouse gas emissions and surging energy bills for wireless operators [3]. These emerging issues motivate wireless operators and equipment manufacturers to search for green communication solutions [4–6]. In a typical wireless communication system, BSTs consume 50% to 60% of energy [3]. Therefore, an eco-friendly and cost-effective solution to reduce the energy bills is to utilize renewable energy resources (e.g., sunlight and wind) at BSTs [7–10]. For example, Huawei and Telefonica have installed solar-powered BSTs in central Chile [11]. Using energy harvesters (e.g., residential-level photovoltaic panels and miniature wind turbines), each BST can scavenge energy from renewable energy resources such that the surging energy bills can be reduced [7–14]. Nevertheless, the availability of renewable energy is highly weather-dependent and space-varying [7]; therefore, it is challenging to maintain an acceptable communication quality-of-service (QoS) when BSTs are solely powered by renewable energy [9].

1.1 Energy Management in Smart-Grid Powered Communication Systems

A hybrid-powered communication system is preferred to reduce the energy bills of wireless operators and guarantee communication QoS. While an energy harvester allows each BST to scavenge energy from renewable energy resources, strategies need to be carefully developed to

[1] https://techblog.comsoc.org/2019/11/22/2019-world-5g-convention-in-beijing-china-has-built-113000-5g-base-stations-130000-by-the-end-of-2019/

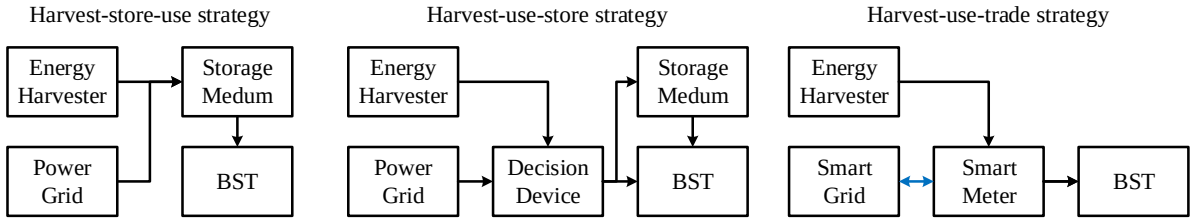


Figure 1.1: Different strategies of using renewable energy.

improve the utilization efficiency of harvested energy. Generally, popular usage strategies can be classified into the following three categories.

- **Harvest-use-store strategy.** As shown in Fig. 1.1, the harvest-store-use strategy requires a storage medium per BST to store harvested energy and to provide energy for a BST [12–14]. The harvested energy in a storage medium can also be used when energy-harvesting opportunities do not exist (or when energy demand of a BST has to be increased to support communication QoS). However, nearly all practical storage media suffer energy loss and leakage to varying extents, ranging from 10% to 30% [9].

- **Harvest-store-use strategy.** To mitigate energy loss and leakage, wireless operators prefer the harvest-use-store strategy [15, 16]. As shown in Fig. 1.1, a decision device is included in deciding whether the harvested energy per slot[2] is used to operate a BST or is stored for future usage. Unlike the harvested-use-store strategy, the harvested energy per slot is prioritized for usage before being stored in a storage medium. Therefore, the storage loss of the harvest-use-store strategy can be reduced compared with the harvest-store-use strategy.

- **Harvest-use-trade strategy.** Harvest-store-use and harvest-use-store strategies require an energy-storage medium, which can suffer from imperfections such as energy loss and leakage [9, 15–17]. When the traditional power grid is upgraded to a smart grid, the two-way energy trading feature of the smart grid (as shown in Fig. 1.1) will benefit wireless operators [18]. In a smart-grid powered communication system (SGPCS), the harvest-use-trade strategy is preferred to avoid the imperfections of storage media and to improve the utiliza-

[2]We assume that the investigated hybrid-powered communication systems operate in discrete-time mode with the minimum interval defined as a slot.

tion efficiency of renewable energy [19, 20]. Also, the two-way energy trading feature allows wireless operators to generate revenue by selling surplus harvested energy to the power grid.

1.2 Machine Learning for Energy Management in SGPCSs

When applying the harvest-use-trade strategy to an SGPCS, unpredictable and intermittent characteristics of renewable energy need to be harnessed by energy management algorithms. An important first step is to choose the proper tools for designing energy management algorithms. One potential tool is convex optimization that has been successfully applied to wireless communications [21]. Many optimization problems in SGPCSs can be formulated as (or converted to) convex ones in wireless communications [19, 22]. For convex problems, a local optimum is also a global optimum. Therefore, the optimal solutions to convex problems can be efficiently obtained via existing toolboxes [23]. A non-convex problem can also be recast into a set of convex subproblems by successive convex approximation [24] and semi-definite relaxation [25]. Besides, the optimality conditions and duality theory of convex optimization also provide the foundations for other optimization algorithms. When unknown random sources are considered in an optimization problem, the objective values are unknown to a convex optimization algorithm [26]. Therefore, convex optimization algorithms cannot directly solve problems with unknown random sources (i.e., stochastic optimization problems). When dealing with the unknown random sources, machine learning algorithms are good candidates to solve stochastic optimization problems [27–30]. In this thesis, we start by investigating machine learning algorithms for different stochastic optimization problems. We then adapt the potential machine learning algorithms to the long-term grid-energy expenditure minimization problem in SGPCSs under various practical constraints.

A machine learning algorithm can transform the experience (e.g., data samples) into expertise (e.g., model parameters[3]). Here, we briefly introduce the taxonomies of machine learning algorithms that are related to the thesis, e.g., centralized learning versus distributed (decentralized) learning, supervised learning versus unsupervised learning, and offline learning versus online learning. Note that there are several taxonomies of machine learning algorithms, and we refer interested readers to [27] for details.

[3]For example, the model parameters are employed to provide a mapping between the data samples and labels in classification problems.

Centralized learning, distributed learning, and decentralized learning. When machine learning algorithms need to process datasets at a central server, they are classified as centralized learning algorithms. However, copies of data at a central server render centralized learning algorithms vulnerable to privacy leakage [31–33]. Besides, uploading datasets to a central server also requires high bandwidth. Distributed learning algorithms have emerged to alleviate these concerns based on the ever-improving computational capability of network-edge devices [33–35]. A distributed learning can be implemented when the parameter server only exchanges the model parameters with associated agents[4]. Since the datasets are kept at the agents, the concerns of privacy and bandwidth demand are alleviated. While centralized learning and distributed learning algorithms require a central server to coordinate the learning process, they may experience failures of the central server. Therefore, decentralized learning algorithms are proposed so that each agent is allowed to exchange information with other agents within certain communication distance (i.e., neighboring agents). In decentralized learning algorithms, since each agent only communicates with its neighboring agents, each agent can continue the learning process when the central node has failed.

Supervised learning and unsupervised learning. Supervised learning algorithms require labeled training datasets, while unsupervised learning algorithms do not require labeled training datasets. In other words, supervised learning is instructed by the environment that sets labels for training datasets. In contrast, unsupervised learning aims at generating some summaries of unlabeled datasets. Clustering data samples into subsets of similar instances is a typical example of unsupervised learning.

Offline learning and online learning. Based on the data-collecting process, machine learning algorithms can be classified into two categories: offline learning algorithms [27, 28] and online learning algorithms [29, 30]. In wireless networks, offline learning is usually performed after a dataset has been obtained by an agent. Offline learning algorithms have seen increasing wireless applications, such as channel estimation [36] and automatic modulation classification [37]. In automatic modulation classification [37], an offline learning algorithm allows an agent to learn model parameters of a deep neural network based on the received signals and their modulation modes. Using the obtained model parameter, the agent can predict the modulation

[4]In this thesis, an agent can be either a user equipment or a BST based on the application scenario.

modes of incoming signals. Note that offline learning algorithms aim at optimizing the expectation of objective functions with unknown-distributed random sources. However, optimizing a time-average expectation[5] over an infinite horizon is a sequential decision-making process.

When the data-collecting process occurs simultaneously as the decision-making process, the obtained decisions can be correlated. By ignoring the correlated decisions, offline learning algorithms cannot handle time-average expectations over infinite horizons. The issue induced by correlated decisions is solved since online learning allows an agent to make decisions based on the accumulated impacts of previous decisions. Therefore, online learning algorithms are suitable to optimize time-average expectations over infinite horizons.

An online learning algorithm is executed while an agent is collecting data. We review two types of online learning algorithms, namely reinforcement learning[6] and Lyapunov learning. A reinforcement learning algorithm [29] allows an agent to learn optimal decisions under different environment states such that a time-average expectation is optimized. Since the environmental dynamic is unknown, the agent must discover the optimal decisions via sequential trial-and-error experiments. Besides, each decision affects the current objective value and subsequent objective values during the interactions between the agents and the environment. The optimization of time-average expectation boils down to optimal control of a dynamic system when the dynamic environment is formulated by a Markov decision process (MDP) [38]. Recent research has shown that reinforcement learning algorithms can achieve human-level performance in several tasks, such as video games [39], autonomous driving [40], and robotic control, when combined with deep neural networks [41].

Another category of online learning algorithms is Lyapunov learning algorithms [30]. While reinforcement learning algorithms are mainly designed for unconstrained optimization problems, Lyapunov learning algorithms provide a flexible framework to handle long-term and short-term constraints. After the original problem has been transformed into a set of per-slot subproblems, Lyapunov learning algorithms can handle constrained optimization problems having time-average expectations in the objective function and constraints. Therefore, Lyapunov learning algorithms

[5] In this thesis, a time-average expectation is obtained in two steps: taking the statistical expectation of a metric and taking the time average of the obtained statistical expectations.

[6] When we discuss reinforcement learning in the thesis, we refer to online reinforcement learning algorithms. Offline reinforcement learning algorithms have also been investigated to extract expertise from a fixed dataset.

are more suitable to minimize the time-average expectation of grid-energy expenditure (TAEGEE) in SGPCSs.

1.3 Related Works and Motivations

In this thesis, our primary goal is to solve the TAEGEE minimization problem in SGPCSs. The primary goal will be realized via the following three steps. Our first step is to investigate optimization problems with expectations in the objective function. In particular, we investigate a secure federated learning problem. The proposed algorithms can find potential applications in SGPCSs. Our second step is to investigate optimization problems with time-average expectations in the objective function. During the second step, we propose a memory-efficient reinforcement learning algorithm, namely the linear-approximate decentralized Q-learning. In the context of SGPCSs, we also suggest potential problems that the linear-approximate decentralized Q-learning can solve. Our final step is to investigate optimization problems with time-average expectations in constraints and objective functions. During the final step, we leverage Lyapunov learning algorithms to solve the TAEGEE minimization problem in SGPCSs. In the remaining part of this section, we perform a detailed literature review to motivate the research gaps that will be filled by this thesis.

1.3.1 Communication-Efficient Robust Federated Learning Over Heterogeneous Datasets

Traditional machine learning algorithms require centralized data processing at a server cloud. However, copies of data in a cloud make cloud-centric learning vulnerable to privacy leakage [31–33]. Leveraging the ever-improving computational capability of network-edge devices, distributed on-device learning has emerged to alleviate these privacy concerns. As an implementation of distributed learning, federated learning has attracted growing attention from both industry and academia [33–35]. In typical federated learning, multiple agents perform local training based on local datasets, and a parameter server collects and processes the local training results for further usage. Specifically, the parameter server updates and broadcasts global model parameters based on local messages (e.g., local gradients and local model parameters). After receiving global model

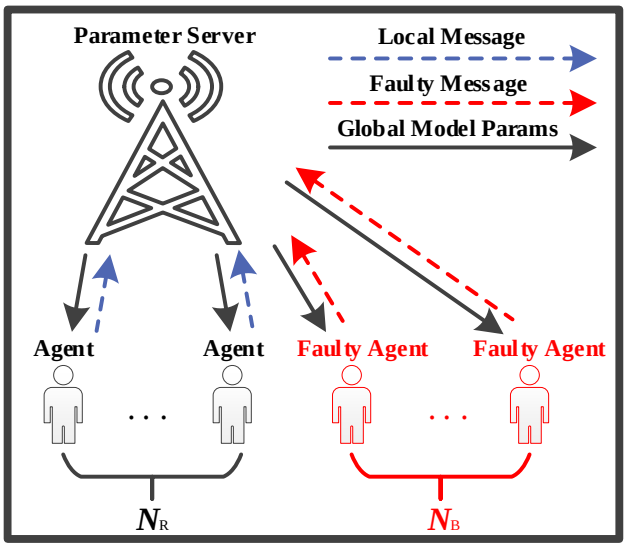


Figure 1.2: Illustration of a federated learning framework. The federated learning framework consists of a parameter server, N_R reliable agents, and N_B faulty agents.

parameters and local datasets, agents compute local messages in parallel. Since datasets are kept at agents in federated learning, the risk of privacy leakage is reduced. However, federated learning is susceptible to faulty or malicious agents. When a faulty agent uploads an unreliable message to the parameter server, the vanilla gradient descent algorithm and its stochastic version (Stochastic Gradient Descent, SGD[7]) fail to converge [42]. Besides, the seminal work by [43] also shows that a faulty agent can arbitrarily corrupt global gradients generated by the SGD algorithm. Therefore, it is crucial to deal with faulty agents. Here, the faulty agents can perform arbitrarily bad actions [42, 43]. The objective is to secure the federated learning framework in Fig. 1.2 via a secure federated learning algorithm. Secure federated learning algorithms need to consider two major topics: fault resilience and communication efficiency. Our first step is built on this research front.

Fault-resilient federated learning. When dealing with fault resilience in the federated learning framework, recent research has reported some learning approaches based on the full gradient per update [42, 44–46]. For strongly convex loss functions, the geometric median (GeoMed) algorithm [42] converges to a near-optimal solution when less than 50% of the local messages

[7]In a distributed setup, the vanilla gradient descent and SGD algorithms [27, (14.1)] require the parameter server to update model parameter based on the gradients uploaded by the agents.

from the agents are unreliable. For (strongly) convex and smooth non-convex loss functions, component-wise median and component-wise trimmed mean algorithms [44] can secure parameter updates at the server in the presence of faulty agents. However, the component-wise median and component-wise trimmed mean algorithms may converge to a saddle point that is far away from a local minimizer for non-convex loss functions. As a remedy, a Byzantine perturbed gradient algorithm [45] is proposed to obtain an approximate local minimizer of non-convex loss functions.

For large datasets, evaluating full gradient per iteration is computationally prohibitive. Computational efficiency was investigated in different fault-resilient stochastic federated learning algorithms, such as Krum [43], Bulyan [47], Byzantine SGD [48], Zeno [49], and DRACO [50]. After obtaining the dissimilarity scores of local gradients, the Krum algorithm sets the global gradient as the local gradient with the smallest score. When less than 50% of the agents are unreliable, the Krum algorithm converges to a near-optimal solution [43]. When less than 25% of the agents are unreliable, the Buylan algorithm [47] can reduce the radius of the neighborhood obtained by the Krum algorithm. Using historical gradients, the Byzantine SGD algorithm [48] allows the parameter server to remove the faulty local gradients before performing gradient aggregation. The Zeno algorithm [49] ranks the reliability of local gradients based on the weighted descent value and magnitude of local gradients. The Zeno algorithm can still work when there is at least one reliable agent by averaging the top-ranked local gradients. Based on coding theory and sample redundancy (i.e., multiple copies of a data sample across different agents), the DRACO algorithm [50] converges when there is at least one reliable agent.

The works above deal with homogeneous datasets in which samples from different agents are independent and identically distributed. In several practical settings, agents can collect non-identically distributed datasets (i.e., heterogeneous datasets) from the other agents. For example, different YouTube subscribers are provided with different categories of advertisements and video clips based on their search history. As a result, developing fault-resilient federated learning algorithms over heterogeneous datasets has emerged as an important research task. Given heterogeneous datasets and faulty agents, a robust stochastic aggregation (RSA) framework was investigated [51]. The proposed RSA algorithm converges to a near-optimal solution at a rate $O(\log(k)/\sqrt{k})$, where k is the number of communication rounds.

Communication-efficient federated learning. Frequent communications between the

server and agents are inevitable in federated learning. Since bandwidth is a scarce resource for the parameter server, the communication overhead becomes a bottleneck [52, 53]. A line of research focuses on skipping the unnecessary communication rounds to reduce communication overhead [54, 55], where the lazy aggregation (LAG) algorithm [54] avoids redundant information exchanges and can be extended to use quantized gradients [55]. Compared with the vanilla gradient descent algorithm, LAG has comparable convergence rate at reduced communication overhead; see also [56–58] that leverage local SGD to allow intermittent server-agent exchanges.

Since the convergence rate $\mathcal{O}(\log(k)/\sqrt{k})$ of the RSA algorithm is relatively slow, we will use Nesterov's acceleration technique to improve the convergence rate. Besides, the fault-resilience issue of LAG and local SGD algorithms are as yet unexplored. Motivated by the local SGD algorithm, we allow the agents to communicate with the server periodically such that the communication overhead is reduced. We will analyze the convergence rates when Nesterov's acceleration technique and periodic communication are used. Such an analysis will provide insights into the fault resilience of federated learning.

1.3.2 Linear-Approximate Decentralized Q-Learning

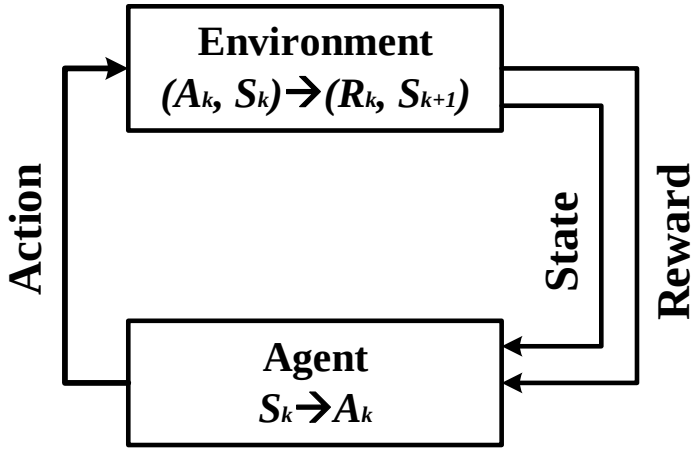


Figure 1.3: Illustration of a single-agent reinforcement learning process.

Reinforcement learning enables agents to make sequential decisions by interacting with an unknown environment [29, 38]. With the environment state evolving as an MDP, the objective of reinforcement learning is to maximize the long-term reward as shown in Fig. 1.3. In slot k, an agent performs random action A_k based on the environment state S_k. Given the action A_k and environment state S_k, the environment moves to state S_{k+1} and reveals a reward R_k for the

agent. When combined with deep neural networks, reinforcement learning algorithms can achieve human-level performance in many tasks, such as video games [39], autonomous driving [40], and robotic control [41].

Recently, multi-agent reinforcement learning (MARL) features increased computational and statistical efficiencies and enhanced privacy-preserving properties based on the parallel computation and experience-sharing among agents. Despite the success of single-agent reinforcement learning [29, 39, 41], algorithmic and theoretical developments of MARL remain challenging and limited. This is mainly because each agent interacts with the environment and other agents in a decentralized manner [59]. In general, MARL can be grouped into cooperative (or collaborative) MARL [59–63], competitive MARL [64–67], and mixed (neither collaborative nor competitive) MARL [68]. In particular, collaborative MARL is usually modeled as a multi-agent MDP, where agents can share the same (or have private) rewards, while their collective goal is to learn optimal policies to maximize the long-term team reward [62, 69, 70]. Competitive MARL is often studied under the framework of Markov games, especially the two-player Markov games [64–67].

We investigate collaborative MARL with private rewards. For model representation, we use the collaborative multi-agent MDP model [62, 70]. In this model, each agent selects an individual action in a particular state by following a local policy; the resulting joint actions of all agents determine the state transition, while each agent receives a private reward that may differ from those of other agents. The global reward (or team reward) is the average of all individual rewards. The private rewards distinguish our model from several other multi-agent models, where the global reward is observable by all agents [69, 71]. In collaborative MDP, the goal of the agents is still to optimize the global reward. Such a collaborative MARL task emerges naturally in several applications, including autonomous driving [40], control of robotic swarms [72], and wireless sensor networks [73].

Many model-free collaborative MARL algorithms have been proposed for systems with unknown transition probabilities, and agents are tasked with estimating one (or more) of the quantities: state-value function, action-value function (i.e., Q-function), and policy [62, 69–71]. Praised as one of the breakthroughs in reinforcement learning, tabular Q-learning[8] was introduced in

[8] A tabular Q-learning algorithm requires each agent to maintain a lookup table for values of Q-function. See Chapter 2 for details.

[74] for model-free control, which is central to modern artificial intelligence [39, 41]. Asymptotic convergence of tabular Q-learning was shown under different assumptions [74–77], whose non-asymptotic performances were studied in [78–81]. In an MARL setting, a decentralized tabular Q-learning algorithm and its asymptotic convergence were provided in [61]. To handle large state and action spaces, Q-learning with function approximation has become the workhorse [39, 82]. The asymptotic performance of linear-approximate Q-learning was analyzed under a stability condition on the behavior policy in [82].

In the Big Data era, recent interests have shifted toward understanding the data utilization efficiency of machine learning algorithms—which is often characterized by the finite-sample error bounds. In the context of reinforcement learning, finite-sample analysis of temporal-difference (TD) learning, Q-learning, and state-action-reward-state-action (SARSA) learning algorithms with linear-function approximation (LFA) have been recently studied [83–90]. In particular, it was shown [87] that linear-approximate Q-learning using decaying stepsizes converges at rate $\mathcal{O}(\log(k)/k)$, which is slower than the best known rate $\mathcal{O}(1/k)$ of tabular Q-learning [81]. Despite the existence of such exciting theoretical developments in single-agent reinforcement learning, the finite-sample analyses of MARL algorithms are under-exploited. For MARL algorithms, finite-sample analyses have recently attracted research attentions [62, 63, 91, 92]. For example, finite-sample error bounds were provided for decentralized batch Q-learning with function approximation [62] and for decentralized TD learning [91, 92]. However, conditions for convergence and convergence rates remain unknown for the linear-approximate decentralized Q-learning. To shed light on the unknown convergence conditions and convergence rates, we perform a finite-sample analysis on the linear-approximate decentralized Q-learning.

1.3.3 Lyapunov-Learning Based TAEGEE Minimization

The research on hybrid-powered communication systems can be classified into three categories: grid energy expenditure (GEE) minimization [12–14, 16, 17, 93–96], throughput maximization [15, 97, 98], and energy efficiency maximization [99, 100]. In particular, GEE minimization problems have been investigated in point-to-point systems [12, 13, 15, 16, 93], single-cell systems [14, 17, 95, 99] and multi-cell systems [93, 98, 100]. GEE can be minimized under energy-harvesting and outage constraints [16] or prior knowledge of hourly-varying energy price

[94] when the non-causal channel state information (CSI) and non-causal energy state information (ESI) are considered over multiple fading slots. Data-rate maximization algorithms were developed for point-to-point hybrid-powered communication systems [13, 101] and cognitive radio systems [97]. The energy-efficiency maximization algorithm [99] was developed for the single-cell hybrid-powered communication systems. When the two-way energy trading feature is considered, the design of SGPCS has become a topic of practical importance. However, the previous works [12–17, 94, 95, 97–100] focus on the traditional grid and ignore the two-way energy trading feature in a smart grid [102]. The two-way energy trading feature provides another dimension to reduce the energy bills of hybrid-powered BSTs (i.e., SGPCSs) [19].

Several works have investigated the framework of SGPCSs [18, 102–106] and resource allocation algorithms [19, 22, 107–109]. More specifically, resource allocation algorithms in SGPCSs can be classified into three categories: one-shot algorithms [19, 22], offline algorithms [107, 110], and online algorithms [108, 109]. The one-shot algorithms in [19, 22] are applicable to scenarios where the resources are independently allocated in each slot of SGPCS. The offline algorithms [107, 110] were proposed to allocate jointly the SGPCS resources over a finite number of slots. In contrast, the online algorithms [108, 109] were tailored to handle the volatility of renewable energy arrivals in an SGPCS over infinite slots. Practical SGPCSs operate over infinite-time horizons, and the statistical distributions of CSI and ESI are difficult to obtain. Therefore, online resource allocation algorithms are preferred.

Single-Cell SGPCS. Different from the previous works, we study TAEGEE minimization in a single-cell SGPCS while considering the packet rate of user equipments (UEs). Instead of using the well-known log-concave function for data rate [17, 19, 97, 99, 103, 107, 108, 110, 111], we formulate the packet rate as a sigmoid function, which captures the effects of packet transmission failure and data rate. Therefore, algorithm design based on the sigmoidal packet rate provides a realistic insight into the practical systems. However, the non-convexity of the sigmoidal packet rate renders the design of beamforming algorithms to a cross-layer design problem, and no previous work has developed algorithms to handle the sigmoidal packet rate. Therefore, we adopt Lyapunov learning to propose an online cross-layer beamforming framework to minimize the long-term GEE of a single-cell SGPCS.

Multi-Cell SGPCS. The aforementioned online resource allocation algorithms [108, 109]

allocate resources over a single time scale. The scheduled UE indicators need to be reallocated over several slots in practical systems since the frequent scheduling of the UEs can cause reliability issues. Moreover, the frame-scale user scheduling has a more accurate characterization of end-to-end delay than the slot-scale scheduling when the UEs can tolerate delay. Few studies have investigated the two time-scale resource allocation schemes. Based on the two time-scale energy merchandising, a dynamic beamforming algorithm was proposed to minimize the long-term GEE for a single-cell SGPCS [112]. The proposed two time-scale algorithm [112] allocates the ahead-of-time energy-trading amount in each frame and the real-time energy-trading amount and beamforming in each slot. Since the ahead-of-time energy-trading amount is a continuous variable, it can be obtained by the subgradient method. Since the scheduled UE indicators are binary variables, the proposed schemes in [108, 109, 112] cannot obtain the optimal indicators of scheduled UE. Exhaustive search is used in [113] to solve the network selection subproblem, and greedy selection is used to address the subchannel allocation subproblem. However, exhaustive search is computationally expensive, and greedy selection can lead to suboptimal solutions for the scheduled UE indicators when the number of UEs is large. Besides, the proposed algorithms in [112, 113] are not applicable when heterogeneous energy coordination and proportional-rate constraints are considered in multi-cell SGPCSs. Compared with Lyapunov learning [106, 108, 112, 113], reinforcement learning methods [98, 114] can also be used to develop online optimization algorithms. However, reinforcement learning requires the proper development of a function estimator to deal with continuous states and continuous actions. Besides, it is more challenging to solve constrained optimization problems via reinforcement learning algorithms. Therefore, we are motivated to extend Lyapunov learning to a two time-scale optimization framework. Using the proposed optimization framework, wireless operators can minimize the long-term GEE of a multi-cell SGPCS by scheduling UEs in each frame, and calculating beamforming vectors and exchanging renewable energy in each slot.

1.3.4 Objectives of the Thesis

Based on previous discussions, we briefly summarize the research objectives of the thesis as follows.

- We start by developing offline federated learning algorithms to handle random sources with unknown distributions. When faulty agents coexist with reliable agents, obtaining a reliable training model is important. An application of such algorithms can be energy planning for a set of BSTs in SGPCSs, where some of BSTs may not honestly upload local training results due to noisy channels or malicious functions. Besides, the communication overhead needs to be reduced for such an application to save cost on bandwidth. Therefore, our first step is to propose offline federated learning algorithms that are expected to reduce the communication rounds of federated learning algorithms while retaining robustness to faulty agents.

- Certain practical applications require agents to make sequential decisions. Besides, decisions in consecutive slots can be correlated to optimize long-term performance metrics. An example of such applications can be cooperative spectrum sensing in SGPCSs, where the spatially dispersed agents collaboratively detect spectrum vacancy and decide the usage of renewable energy. Another potential application is cooperative spectrum sharing in SGPCSs, where the agents need to use renewable energy and spectrum bands collaboratively. Therefore, we are motivated to investigate decentralized Q-learning that considers random sources with unknown distributions and correlation for consecutive decisions. The major issue of decentralized Q-learning is the prohibitive memory requirement for large state and action spaces. Hence, our second step is to improve the memory efficiency of the tabular decentralized Q-learning algorithm by using LFA technique. Moreover, the unknown convergence conditions and convergence rates are expected to be revealed for LFA based decentralized Q-learning algorithm.

- While the proposed Q-learning algorithm can optimize long-term performance metrics, it requires complex modifications to handle long-term and short-term constraints. Therefore, our third step aims at adapting Lyapunov learning algorithms to SGPCSs. Given different practical constraints, our last step is to minimize the TAEGEE in single-cell and multi-cell SGPCSs.

1.4 Thesis Structure and Contributions

The thesis is organized into six chapters. Chapter 1 presents a brief review of machine learning algorithms and describes the applications in the energy management of wireless communication systems. Besides, this chapter also provides a detailed literature review for the remaining chapters. Chapter 2 provides some preliminary information on secure federated learning, Q-learning, and Lyapunov learning. The technical chapters focus on two objectives: designing energy-efficient machine learning algorithms (Chapters 3 and 4) and applying machine learning algorithms to minimize long-term GEE of SGPCSs (Chapters 5 and 6).

Chapter 3 investigates fault-resilient federated learning when heterogeneous datasets are collected by agents, and the number of faulty agents is unknown to the central server. Different from the state-of-the-art fault-resilient algorithms, the proposed fault-resilient proximal gradient (FRPG) and local FRPG (LFRPG) algorithms do not have a limitation on the number of faulty agents. The theoretical results reveal that the proposed FRPG and LFRPG algorithms have convergence rate $\mathcal{O}(1/k)$. Hence, the FRPG and LFRPG algorithms converge faster when compared with the convergence rate $\mathcal{O}(\log(k)/\sqrt{k})$ of the RSA algorithm [51]. Numerical results performed on various real datasets confirm that FRPG and LFRPG algorithms converge faster than the state-of-the-art fault-resilient algorithms (i.e., RSA [51], Krum [43], and GeoMed [42]).

Chapter 4 investigates a linear-approximate decentralized Q-learning algorithm for collaborative multi-agent reinforcement learning tasks with private rewards. Given behavior policies, a group of agents collaboratively learn an optimal Q-function in a fully decentralized manner. When the state-action pairs are sampled from an MDP, "stochastic gradients" used in the updates are correlated and biased, making a convergence analysis challenging. Chapter 4 establishes convergence rates for the proposed decentralized Q-learning algorithm in the cases of both decaying and constant stepsizes. Under an appropriate assumption on the multi-agent joint behavior policy, our results show that the convergence rates of the decentralized Q-learning algorithm match that of tabular Q-learning, while the former gains scalability, privacy, and parallel computation when handling large state-action spaces.

Chapter 5 considers a single-cell SGPCS with a harvest-use-trade strategy that can reduce the GEE and improve the utilization efficiency of renewable energy. In this chapter, the investigated

TAEGEE minimization problem considers the joint effects of packet failure, the data rates of UEs, and unknown-distributed ESI and CSI. Using Lyapunov learning, we reformulate the TAEGEE minimization problem into per-slot subproblems. In this case, a BST can design the beamforming vectors based on the current state (i.e., ESI, CSI, and backlogs of UEs) of a single-cell SGPCS. Since each per-slot subproblem is non-convex, two suboptimal algorithms are proposed based on the successive approximation beamforming (SABF) and zero-forcing beamforming (ZFBF) techniques. The convergence properties are established for the proposed suboptimal algorithms, and the corresponding computational complexities are analyzed.

Chapter 6 investigates user scheduling, beamforming, and energy coordination in multi-cell SGPCSs, where the BSTs are powered by a smart grid and natural renewable energy sources. Heterogeneous energy coordination (i.e., energy merchandizing with the smart grid and energy exchanging among BSTs) is considered in multi-cell SGPCSs. A TAEGEE minimization problem with proportional-rate constraints is formulated for multi-cell SGPCSs. Since scheduled UE variables are coupled with the beamforming vectors, the formulated problem is challenging to handle via standard convex optimization methods. In practice, the beamforming vectors need to be updated over each slot according to the channel variations. User scheduling needs to be updated over several slots (which constitute a frame) since the frequent scheduling of UEs can cause reliability issues. Therefore, Lyapunov learning is used to decouple the problem. A two time-scale algorithm is proposed to schedule users in each frame and obtain beamforming and energy-exchanging variables in each slot. We prove that the proposed two time-scale algorithm can asymptotically achieve the optimal solutions via tuning a control parameter.

Chapter 7 concludes the thesis and suggests several future research directions.

Chapter 2

Background

In this chapter, we discuss the fundamental concepts on secure federated learning, reinforcement learning, and Lyapunov learning.

2.1 Fundamental Concept in Secure Federated Learning

Traditional machine learning algorithms require centralized data processing at a server cloud. However, copies of data in a cloud render cloud-centric learning vulnerable to privacy leakage [31–33]. Leveraging the ever-improving computational capability of network-edge devices, distributed on-device learning has emerged to alleviate these privacy concerns. As an implementation of distributed on-device learning, federated learning has attracted growing attention from both industry and academia [33–35]. A popular realization of federated learning uses a parameter server to interact with multiple agents, which are network-edge devices in practical systems. Specifically, the parameter server updates and broadcasts global model parameters based on local messages (e.g., local gradients and local model parameters) from/to agents. After receiving global model parameters and local datasets, reliable agents compute local messages in parallel. Since datasets are kept at agents in federated learning, the risk of privacy leakage is reduced. However, federated learning may fail when faulty agents upload corrupted local messages. Therefore, a secure federated learning algorithm is required to mitigate the undesirable effects (e.g., a divergence of algorithm and degradation of prediction accuracy) that are caused by faulty agents. Three components are required to implement a secure federated learning algorithm, i.e., a parameter server, a set of reliable agents, and a set of faulty agents. Their functions are defined as follows.

- A parameter server updates and broadcasts global model parameter to all agents.
- A reliable agent calculates the local information based on local datasets, and uploads local

information (e.g., local gradient and local model parameter) to the parameter server.

- A faulty agent performs adverse actions by uploading faulty model parameters.

The objective of a secure federated learning algorithm is to obtain a convergent and optimal solution to the following problem [42–44, 47]

$$\min_{\boldsymbol{w}} \sum_{n=1}^{N_{\mathrm{R}}} f_n(\boldsymbol{w}) \tag{2.1}$$

where $\boldsymbol{w}$ is the model parameter, and $f_n(\boldsymbol{w})$ is the loss function of reliable agent n.

Different methods have been proposed to obtain an optimal solution to (2.1) in the presence of faulty agents. Current methods can be classified into four categories: secure-aggregation based federated learning, secure-model based federated learning, fault-detection based federated learning, and preprocessing based federated learning.

- **Secure-aggregation based federated learning.** The Krum algorithm [43] can generate a sequence of global gradients, and each global gradient in each iteration has the smallest sum Euclidean distance to its first $N_{\mathrm{R}} - 2$ closest local gradients. When the fraction of faulty agents is less than 50% in each iteration, each global gradient can approximate the true global gradient. Therefore, the negative effects of faulty agents are mitigated. Using the fact that the median cannot be skewed by a small proportion of extremely large or small values, two secure aggregation rules are proposed, namely GeoMed [42] and component-wise median [44]. Here, Krum is applicable to non-convex loss functions. GeoMed is applicable to strongly convex loss functions. The component-wise median is applicable to (strongly) convex and non-convex loss functions.

- **Secure-model based federated learning.** Secure federated learning algorithms can be obtained based on problem formulation. For example, a penalty function is introduced to (2.1). Instead of uploading the local model parameter, each agent uploads a gradient of penalty function. When the gradient of a penalty function is discrete, the obtained algorithm is robust to faulty agents [51]. Moreover, a quantitative analysis was performed in [51] to reveal the relation between optimality and the number of faulty agents.

- **Fault-detection based federated learning.** Fault detection can be included to secure the aggregation of local gradients. For example, the Byzantine stochastic gradient descent algorithm uses a two-criterion approach to detect faulty agents. Local gradients of reliable agents can introduce a limited variation of time-average local gradients and limited fluctuation of time-average inner products of local gradient and model parameter. When either one of the limitations is violated, the agent is detected as a faulty one.

- **Preprocessing based federated learning.** Using the sample redundancy, DRACO [50] can remove some undesirable effects of faulty agents. Based on the information theory, the optimal gradient can be recovered when reliable agents report sufficient information to a parameter server. Another preprocessing method is named as Bulyan [47]. Bulyan uses Krum [43] to refine a subset of uploaded local gradients. The parameter server constructs a global gradient by taking the component-wise average to the refined subset of local gradients. Since Bulyan is a refinement of Krum by removing several unnecessary local gradients, it has a more strict limitation on the number of faulty agents. Specifically, the number of faulty agents is limited to less than 25% of the number of agents.

2.2 Fundamental Concepts in Q-Learning

Q-learning provides a flexible framework for agents to make sequential decisions in an unknown environment [29]. The goal of agents is to maximize the long-term reward by interacting with the environment [38]. In the sequel, we provide a brief review of several fundamental concepts of Q-learning.

2.2.1 Markov Decision Process and Tabular Q-Learning

Denote an MDP by a quintuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r)$. The state space is $\mathcal{S}$ of size $|\mathcal{S}|$, and the action space is $\mathcal{A}$ of size $|\mathcal{A}|$. The set of action-dependent transition probability matrices is denoted by $\mathcal{P} = \{[p_a^{s,s'}] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|} | s, s' \in \mathcal{S}, a \in \mathcal{A}\}$ where $p_a^{s,s'} = P(s'|s,a)$ is the transition probability from state s to s' by using the action a. The constant γ is a discounting factor. Performing action a under state s, the agent obtains a local reward $r(s,a)$ that is upper-bounded by $r_{\max}$.

Let $\boldsymbol{\mu} = \left\{[\mu_a^s] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|} | \mu_a^s = P(a|s), s \in \mathcal{S}, a \in \mathcal{A}\right\}$ denote a behavior policy of the agent.

Therefore, a Q-function is defined in terms of the discounted reward under the behavior policy $\boldsymbol{\mu}$ as

$$Q(s,a) = \mathbb{E}\left[\sum_{k=1}^{K} \gamma^k r(S_k, A_k) \middle| S_1 = s, A_1 = a\right] \tag{2.2}$$

where S_k is a random state in slot k, A_k is a random action in slot k following the behavior policy $\boldsymbol{\mu}$.

The theory of dynamic programming [115] guarantees that there exists at least one optimal behavior policy $\boldsymbol{\mu}^*$, under which the optimal Q-function satisfies the Bellman equation

$$\begin{aligned} Q^*(s,a) =& \mathbb{B}[Q^*(s,a)] \\ =& \sum_{s' \in \mathcal{S}} p_a^{s,s'} \left(r(s,a) + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a') \right) \end{aligned} \tag{2.3}$$

where $\mathbb{B}[\cdot]$ is a Bellman operator.

It has been demonstrated [76, 80, 81] that the Bellman operator $\mathbb{B}[\cdot]$ is a contraction mapping with respect to the ℓ_∞-norm [76, 80, 81]. Therefore, the optimal Q-function can be obtained through a fixed-point iteration as

$$Q_{k+1}(S_k, A_k) = (1 - \alpha_k) Q_k(S_k, A_k) + \alpha_k \mathbb{B}[Q_k(S_k, A_k)] \tag{2.4}$$

where α_k is the stepsize.

When the action-dependent transition probability $\mathcal{P}$ is unknown, a stochastic approximation of the fixed-point iteration (2.4) is used. Therefore, the tabular Q-learning algorithm is proposed as [74]

$$Q_{k+1}(S_k, A_k) = (1 - \alpha_k) Q_k(S_k, A_k) + \alpha_k \left(r(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} Q_k(S_{k+1}, a') \right). \tag{2.5}$$

Since the iteration in (2.5) works without the action-dependent transition probability matrices in $\mathcal{P}$, the tabular Q-learning is a model-free algorithm. Based on (2.5), the tabular Q-learning uses an off-policy[9] method to generate data samples.

[9]The behavior policy $\boldsymbol{\mu}$ is unchanged in tabular Q-learning.

2.2.2 Q-Learning With Linear Functional Approximation

The iteration in (2.5) becomes intractable when the number of state-action pairs is large, also known as curse of dimensionality [38, 82]. Therefore, the linear function approximation method is proposed to obtain the optimal Q-function effectively [116].

Denote the basis function for the initial state s and action a by $\boldsymbol{\phi}(s,a) = [\phi_1(s,a), \ldots, \phi_d(s,a)]^\top \in \mathbb{R}^d$. Hence, the approximated Q-function with the initial state s and action a is given by

$$Q(s,a) \approx \tilde{Q}(s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{w} \rangle \tag{2.6}$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is the weight vector, and $\|\boldsymbol{\phi}(s,a)\| \leq 1$.

To approximate the Q-function initialized from all the states and actions, one can define a feature matrix by stacking all the basis functions as

$$\boldsymbol{\Phi} = [\ldots, \boldsymbol{\phi}(s,a), \ldots]^\top \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d} \tag{2.7}$$

where $d \ll |\mathcal{S}|\,|\mathcal{A}|$. The induced linear space of feature matrix $\boldsymbol{\Phi}$ is obtained as $\mathcal{Q} = \left\{ \boldsymbol{\Phi}\boldsymbol{w} \,\middle|\, \boldsymbol{w} \in \mathbb{R}^d \right\}$, and the feature matrix $\boldsymbol{\Phi}$ is full rank.

Since the optimal Q-function may not belong to the linear space $\mathcal{Q}$, a projection step is required to obtain the fixed point of approximated Q-function $\tilde{Q}(s,a)$. Therefore, the projected Bellman equation for the approximate Q-function $\tilde{Q}(s,a)$ is given by

$$\tilde{Q}(s,a) = \mathrm{proj}_{\mathcal{Q}} \left\{ \mathbb{B}\left[\tilde{Q}(s,a) \right] \right\}. \tag{2.8}$$

The equivalent form of (2.8) is obtained as [87, Appendix A]

$$\mathbb{E}\left[\boldsymbol{\phi}(s,a) \left(r(s,a) + \gamma \max_{a' \in \mathcal{A}} \langle \boldsymbol{\phi}(s',a'), \boldsymbol{w} \rangle - \langle \boldsymbol{\phi}(s,a), \boldsymbol{w} \rangle \right) \right] = 0 \tag{2.9}$$

where $\boldsymbol{w}^*$ is the optimal weight vector.

Mimicking the tabular Q-learning, a stochastic approximation of (2.9) is used. Therefore, the

weight vector $\boldsymbol{w}$ is updated via

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \alpha_k \boldsymbol{\phi}(S_k, A_k)\Big(r(S_k, A_k) + \gamma \max_{a' \in \mathcal{A}} \langle \boldsymbol{\phi}(S_{k+1}, a'), \boldsymbol{w}\rangle - \langle \boldsymbol{\phi}(S_k, A_k), \boldsymbol{w}\rangle\Big). \tag{2.10}$$

Using the feature matrix $\boldsymbol{\Phi}$ and non-summable and square-summable stepsize α_k, one obtains the optimal weight vector $\boldsymbol{w}^*$ via (2.10) [87, Theorem 3.4]. With optimal $\boldsymbol{w}^*$, the optimal Q-function is obtained via (2.6).

2.2.3 Generation of Feature Vectors

In this section, we briefly introduce a method to generate feature vectors in this thesis. We use the tabular representation to maintain a feature vector for each state-action pair. For discrete state spaces, the number of feature vectors equals the number of state-action pairs. For continuous state spaces, the state space needs to be discretized by separating equally the state space into discrete levels. Then, the number of feature vectors equals the number of discrete levels. Since a linear function is used to approximate the Q-function, the value of Q-function is represented by an inner product of feature vector $\boldsymbol{\phi}(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and parameter vector $\boldsymbol{w} \in \mathbb{R}^d$, i.e., $Q(s, a) = \boldsymbol{\phi}^\top(s, a)\boldsymbol{w}$. Each entry of $\boldsymbol{\phi}(s, a)$ can be generated by cosine function or binary function [92].

The set of feature vectors appropriate for a given problem is usually chosen by the experts from the respective discipline. In general, there is no "one size fits all" solution to choose feature vectors, and the generation of feature vectors is still an open problem within the machine learning community. Please refer to [117, Chapter 3] for detailed information on feature vectors.

2.3 Fundamental Concepts in Lyapunov Learning

Lyapunov learning uses the Lyapunov drift function and penalty function to handle the environment dynamics [30]. Suppose a system has N traffic queues for N UEs. Let $q_{n,k}$ be the backlog of each queue n in slot k. A Lyapunov function $\mathbb{C}(\boldsymbol{q}_k)$ is defined as a scalar function of queue vector $\boldsymbol{q}_k = [q_{1,k}, \ldots, q_{N,k}]$. When the queue vector $\boldsymbol{q}_k$ is unstable, the value of the Lyapunov function $\mathbb{C}(\boldsymbol{q}_k)$ approaches infinity as time evolves. Decisions are made in each slot to optimize

a performance metric F_k while keeping the Lyapunov drift function finite. We present a brief review of the fundamental concepts in Lyapunov learning.

2.3.1 Queue Dynamics

The value of each queue n is non-negative and evolves with the stochastic arrival and serving processes, and the initial backlog $q_{n,0}$ is bounded $n = 1, \ldots, N$. Note that the arrival rate $\nu_{n,k}$ and serving rate $r_{n,k}$ are non-negative random variables in slot k. Based on the arrival and serving processes, the queue dynamic function is defined as

$$q_{n,k+1} = \max[q_{n,k} - r_{n,k}, 0] + \nu_{n,k}. \tag{2.11}$$

The value of $q_{n,k}$ can represent an amount of remaining tasks. For example, the value of $q_{n,k}$ is the number of packets in Chapter 5 (or number of information bits in Chapter 6).

Definition 2.1 (Mean Rate Stable [30])**.** A queue n is mean rate stable when $\lim_{k\to\infty} \frac{1}{k}\mathbb{E}\big[q_{n,k}\big] = 0$, where $\mathbb{E}[\cdot]$ is the expectation over all random sources.

When a backlog queue is mean rate stable, the time-average expectation of arrival rate is less than or equal to the time-average expectation of serving rate. This statement can be justified as follows. Based on the dynamic function (2.11), we obtain

$$q_{n,k+1} - q_{n,k} = \max[\nu_{n,k} - r_{n,k}, \nu_{n,k} - q_{n,k}]. \tag{2.12}$$

Taking the expectation and the telescoping sums of (2.12) over $k = 0, 1, \ldots, K$, we have

$$\frac{\mathbb{E}\big[q_{n,K}\big]}{K} \geq \frac{\mathbb{E}\big[q_{n,K}\big]}{K} - \frac{\mathbb{E}\big[q_{n,0}\big]}{K} \geq \frac{\sum_{k=0}^{K}\mathbb{E}\big[\nu_{n,k} - r_{n,k}\big]}{k}. \tag{2.13}$$

Setting $K \to \infty$, we obtain $\lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K}\mathbb{E}\big[\nu_{n,k}\big] \leq \lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K}\mathbb{E}\big[r_{n,k}\big]$, where the terms $\lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K}\mathbb{E}\big[\nu_{n,k}\big]$ and $\lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K}\mathbb{E}\big[r_{n,k}\big]$ correspond to the time-average expectations of arrival rate and serving rate. A mean-rate-stable queue indicates the information bits in such a queue will depart this queue in finite time.

2.3.2 Lyapunov Drift Function

We define a scalar quadratic Lyapunov function as $\mathbb{C}(\boldsymbol{q}_k) := \frac{1}{2}\|\boldsymbol{q}_k\|^2$ where the operator $\|\cdot\|$ is the ℓ_2-norm. The quadratic Lyapunov function $\frac{1}{2}\|\boldsymbol{q}_k\|^2$ is non-negative, and it is equal to zero if and only if all entries of $\boldsymbol{q}_k$ are zero. Generally, there are some other forms of Lyapunov function according to different applications, such as $\mathbb{C}(\boldsymbol{q}_k) := \langle \boldsymbol{q}_k, \log(1+\boldsymbol{q}_k)\rangle$ and $\mathbb{C}(\boldsymbol{q}_k) := \sum_{n=1}^{N} \exp(-c_1(c_2 - q_{n,k}))$. Here, the parameters c_1 and c_2 are shape-forming constants.

2.3.3 One-slot Conditional Lyapunov Drift Function

With the defined Lyapunov function, we can calculate the one-slot conditional Lyapunov drift function as

$$D(\boldsymbol{q}_k) := \mathbb{E}_{\iota}\left[\mathbb{C}(\boldsymbol{q}_{k+1}) - \mathbb{C}(\boldsymbol{q}_k)\middle|\boldsymbol{q}_k\right] \tag{2.14}$$

where the expectation is taken over the random sources ι, and the drift is a difference of Lyapunov functions over slots $k+1$ and k given $\boldsymbol{q}_k$.

By introducing a control parameter $V > 0$, we can define the Lyapunov drift-plus-penalty as

$$D(\boldsymbol{q}_k) + V\mathbb{E}_{\iota}\left[F_k\middle|\boldsymbol{q}_k\right] \tag{2.15}$$

where the performance metric F_k can be used to trade for backlogs of queues.

Chapter 3

Communication-Efficient Robust Federated Learning Over Heterogeneous Datasets

A federated learning algorithm can be used for ahead-of-time energy planning upon the stochastic arrival of traffic demand and renewable energy of SGPCSs. Since the price of ahead-of-time energy is lower than that of real-time energy[10], performing ahead-of-time energy planning can reduce the energy bills of wireless operators. Suppose that a set of BSTs is deployed to cover a specific region. The BSTs are connected to the core network and smart grid through a gateway as shown in Fig. 3.1. Each BST scavenges energy via an energy harvester and collects log information of traffic demand, renewable energy arrival, and energy prices. The gateway aims at obtaining a global model parameter that can estimate the demand for ahead-of-time energy of all BSTs. The global model parameter can be obtained via a standard federated learning process as follows.

- Given the log information, each BST calculates a local penalty gradient.
- Each BST uploads the local penalty gradient to the gateway.
- The gateway updates and broadcasts global model parameter based on the local penalty gradients.

In practice, some of the received local penalty gradients at the gateway can be corrupted by communication failures or malicious actions of hijacked BSTs. As discussed in [43], the corrupted local penalty gradients can diverge vanilla SGD in the federated learning framework. Therefore,

[10] `https://www.aepenergy.com/2018/01/05/december-2017-edition/`

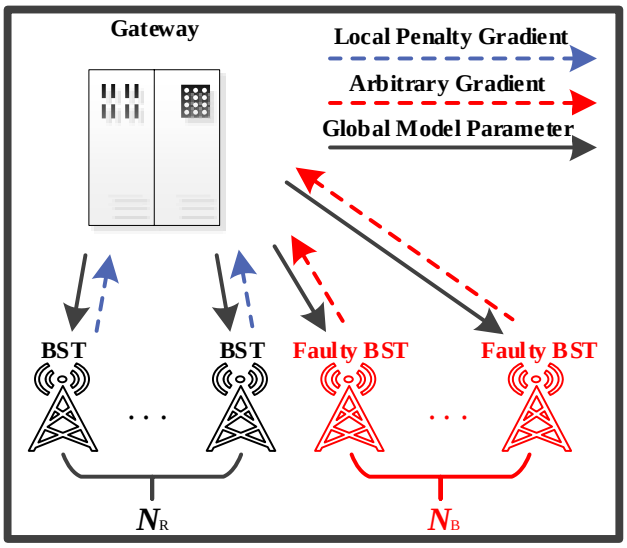


Figure 3.1: Illustration of a federated learning framework for solving the optimization problem (3.5).

secure federated learning algorithms are required to handle the faulty local penalty gradients. While some fault-resilient federated learning algorithms have been proposed to deal with faulty agents such as Krum [43], Bulyan [47], Byzantine SGD [48], Zeno [49], and DRACO [50], they are designed for homogeneous datasets. However, heterogeneous datasets should be considered while designing secure federated learning algorithms for the energy planning problem. More specifically, the log information is used as a dataset at each BST for energy planning of SGPCSs. The UEs' behaviors can be changed at different BSTs, and the arrival rates of renewable energy can be varied across different BSTs. The differences in renewable energy arrival rates and UEs' behaviors result in heterogeneous datasets at different BSTs. As mentioned in Chapter 1, the RSA framework [51] is a promising framework to handle heterogeneous datasets. However, the convergence rate $O(\log(k)/\sqrt{k})$ of the RSA algorithm in [51] can still be improved such that the communication overhead is reduced.

This chapter aims at designing secure federated learning algorithms that can reduce the communication overhead of the RSA framework. An FRPG algorithm is proposed to secure federated learning in the presence of faulty agents. By allowing the agents to exchange information periodically with the server, an LFRPG algorithm is also developed. Theoretical and numerical

results confirm that the LFRPG algorithm requires fewer communication rounds than the FRPG algorithm. Moreover, our numerical results also show that the FRPG and LFRPG algorithms converge faster than the state-of-the-art fault-resilient federated learning algorithms.

3.1 Problem Statement

Consider a federated learning framework comprising a parameter server and N agents. The overall loss given by [33]

$$\sum_{n=1}^{N} f_n(\boldsymbol{w}_n) + f_0(\boldsymbol{w}_0) \tag{3.1}$$

where $\boldsymbol{w}_0 \in \mathbb{R}^d$ denotes model parameter at the server; $\boldsymbol{w}_n \in \mathbb{R}^d$ are model parameter at agent n; and $f_0(\boldsymbol{w}_0)$ is a regularization function. The local loss at agent n is

$$f_n(\boldsymbol{w}_n) = \mathbb{E}_{X_n}[f(\boldsymbol{w}_n; X_n)] \tag{3.2}$$

where $\mathbb{E}_{X_n}[\cdot]$ is the expectation over random data X_n with possibly different distributions, and $f(\boldsymbol{w}_n; X_n)$ is the corresponding loss with respect to $\boldsymbol{w}_n$ and X_n.

The objective of fault-resilient federated learning is to minimize the loss function in (3.1) in a *distributed* fashion subject to the consensus constraints, expressed as

$$\boldsymbol{w}_0 = \boldsymbol{w}_n, n = 1, \ldots, N. \tag{3.3}$$

When there are multiple faulty agents, several researchers have demonstrated that obtaining the minimizer of (3.1) subject to (3.3) is less meaningful [42, 43, 51]. Therefore, the goal will be to minimize the loss function while avoiding consensus with faulty agents. The server cannot differentiate reliable agents from faulty ones and does not even know the number of faulty agents. The proposed algorithms should be robust to faulty agents under these challenging conditions. When analyzing the convergence rate in the presence of faulty agents, N_{R} agents are assumed to be reliable among N agents. For notational convenience, we will index the $N_{\mathrm{B}} = N - N_{\mathrm{R}}$ faulty agents by $n = N_{\mathrm{R}} + 1, \ldots, N$.

Dropping the losses of faulty agents in (3.1), the optimal solution needs to satisfy

$$\begin{aligned} \min_{\boldsymbol{w}} \ & \sum_{n=1}^{N_{\mathrm{R}}} f_n(\boldsymbol{w}_n) + f_0(\boldsymbol{w}_0) \\ \text{s.t. } & \boldsymbol{w}_0 = \boldsymbol{w}_n, n = 1, \ldots, N_{\mathrm{R}}. \end{aligned} \tag{3.4}$$

where $\boldsymbol{w} := \mathrm{vec}(\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_N)$.

Without information about faulty agents, it is ideal (and thus not meaningful) for the server to seek the solution to (3.4). Instead, we will adapt the robust stochastic aggregation approach [51] by adding a penalty term $p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)$ with weight $\gamma > 0$ for local loss $f_n(\boldsymbol{w}_n)$. We will then provide a solution to the penalized version of (3.4), namely

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) := \sum_{n=1}^{N_{\mathrm{R}}} (f_n(\boldsymbol{w}_n) + \gamma p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)) + f_0(\boldsymbol{w}_0). \tag{3.5}$$

Different from (3.4), the penalty terms in (3.5) allow the server parameter $\boldsymbol{w}_0$ to differ from those of faulty agents when heterogeneous datasets are considered. We will select convex and differentiable $\{p_n(\cdot)\}$, e.g., of the Huber type. Moreover, the gradients of $\{p_n(\cdot)\}$ for reliable and faulty agents must be similar, so that the undesirable influence of faulty agents is mitigated. As shown in Fig. 3.1, the parameter server broadcasts global model parameter $\boldsymbol{w}_0$ in each slot. Each reliable agent n uploads the gradient of penalty $\gamma \nabla p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)$ in each slot, and each faulty agent uploads an arbitrary gradient in each slot.

Our communication-efficient solvers to a non-ideal version of (3.5) will be developed in Sections 3.2 and 3.3, based on the following assumptions about f_0, f_n, and p_n, for $n = 1, \ldots, N$.

Assumption 3.1 (Lipschitz Continuity [118, (1.2.11)])**.** *Regularizer f_0 has an L_0-Lipschitz continuous gradient, and f_n has an L_n-Lipschitz continuous gradient for $n = 1, \ldots, N$.*

Assumption 3.2 (Strong Convexity [118, (2.1.20)])**.** *Regularizer f_0 is strongly convex with modulus δ_0, and loss f_n is strongly convex with modulus δ_n for $n = 1, \ldots, N$.*

Assumption 3.3 (Penalty)**.** *Penalty function $p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)$ is convex and differentiable, with bounded gradients $\|\nabla_{\boldsymbol{w}_0} p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)\|^2 \leq G$, and $\|\nabla_{\boldsymbol{w}_n} p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)\|^2 \leq G$ for $n = 1, \ldots, N$.*

Note that Assumption 3.1 is easily satisfied. Several functions have Lipschitz-continuous gra-

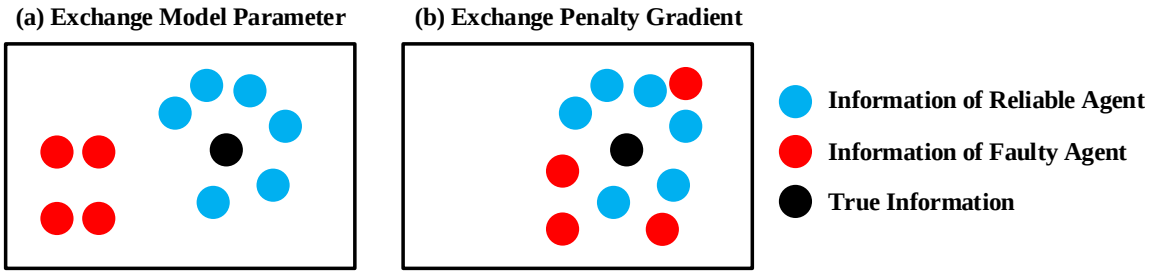


Figure 3.2: An illustration of the theoretical intuition of our proposed algorithms.

dients, such as the square of ℓ_2-norm, logistic regression function, and multinomial logistic regression function. Besides, some artificial neural networks also have Lipschitz-continuous gradients [119]. Assumption 3.2 can also be easily satisfied when the square of ℓ_2-norm is added to convex functions. Assumptions 3.1 and 3.2 are standard when the learning criterion entails smooth and strongly convex local loss functions. The negative effects of faulty agents can be bounded through Assumption 3.3, which is satisfied by, e.g., a Huber-type penalty.

Figure 3.2 shows the theoretical intuition for our proposed algorithms. When the model parameters are exchanged between the server and the agents in Fig. 3.2(a), red dots, blue dots, and black dot respectively represent the faulty model parameters, reliable model parameters, and true parameter at the server. When the penalty gradients are exchanged between the server and agents in Fig. 3.2(b), red dots, blue dots, and black dot represent the faulty penalty gradients, reliable penalty gradients, and true penalty gradients at the server.

Since the server cannot differentiate the model parameters of reliable agents from the model parameters of faulty agents, the average aggregation rule in [33] can result in a skewed estimation of true model parameter at server as shown in 3.2(a). The penalty functions $\{p_n(\cdot)\}_{n=1}^{\mathcal{Q}}$ allow the model parameter of server to be close but not equal to model parameters of agents. In this way, the model parameters of server can be different from the model parameters at the faulty agents. Besides, exchanging gradients of penalty functions can also unified the information from faulty agents and reliable agents as shown in Fig. 3.2(b). Comparing Fig. 3.2(a) and Fig. 3.2(b), the negative effects of faulty agents are mitigated by exchanging penalty gradients between the server and agents. As shown in Fig. 3.2(b), the penalty gradients from faulty agents and reliable agents are close to the true penalty gradient even when the model parameters of faulty agents can be

arbitrarily falsified. Therefore, our developed algorithms can still obtain a suboptimal solution to (3.5) by exchanging the penalty gradients between server and agents.

3.2 Fault-Resilient Proximal Gradient

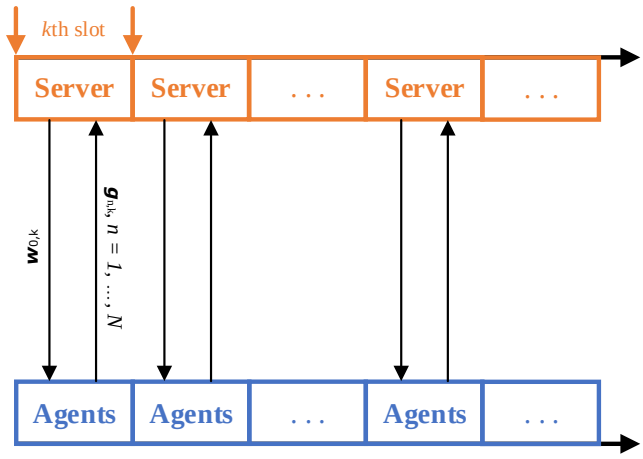


Figure 3.3: In each iteration of FRPG, the server broadcasts $\boldsymbol{w}_{0,k}$, and the agents upload $\boldsymbol{g}_{n,k}$, $n = 1, \ldots, N$.

In this section, we present our novel FRPG algorithm for the server to solve the *non-ideal* version of (3.5), with N replacing N_{R} when faulty agents exist. Figure 3.3 shows the communication protocol of FRPG algorithm, and we will analyze the convergence behavior of our iterative FRPG algorithm.

3.2.1 Algorithm

Along the lines of [120], the parameter server in our federated learning approach maintains three sequences in slot k, namely $\boldsymbol{u}_{0,k}$, $\boldsymbol{w}_{0,k}$ and $\boldsymbol{v}_{0,k}$. The resultant FRPG algorithm updates these three sequences using the iterations

$$\boldsymbol{u}_{0,k} = (1-\beta_k)\boldsymbol{w}_{0,k-1} + \beta_k \boldsymbol{v}_{0,k-1} \tag{3.6a}$$

$$\boldsymbol{w}_{0,k} = \boldsymbol{u}_{0,k} - \frac{1}{\alpha_{0,k}} \nabla f_0(\boldsymbol{u}_{0,k}) \tag{3.6b}$$

$$\boldsymbol{v}_{0,k} = \boldsymbol{v}_{0,k-1} - \frac{\delta_0\big(\boldsymbol{v}_{0,k-1} - \boldsymbol{u}_{0,k}\big) + \nabla f_0\big(\boldsymbol{u}_{0,k}\big) + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}}{\delta_0 + \alpha_{0,k}\beta_k} \tag{3.6c}$$

where $\boldsymbol{w}_{0,k-1}$ are the server parameter on slot $(k-1)$; and likewise for the auxiliary iterates $\boldsymbol{v}_{0,k-1}$; scalars $\alpha_{0,k}$ and β_k are stepsizes; and the sum over N in (3.6c) accounts for the non-ideal inclusion of faulty agents, where $\boldsymbol{g}_{n,k}$ is given by

$$\boldsymbol{g}_{n,k} := \gamma \nabla_{\boldsymbol{w}_0} p_n\big(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k}\big). \tag{3.7}$$

Each reliable agent also maintains sequences $\boldsymbol{u}_{n,k}$, $\boldsymbol{w}_{n,k}$ and $\boldsymbol{v}_{n,k}$ in slot k, that are locally updated as

$$\boldsymbol{u}_{n,k} = (1-\beta_k)\boldsymbol{w}_{n,k-1} + \beta_k \boldsymbol{v}_{n,k-1} \tag{3.8a}$$

$$\boldsymbol{w}_{n,k} = \boldsymbol{w}_{0,k} - \operatorname{prox}_{\frac{\gamma p_n}{\alpha_{n,k}}}\left\{\boldsymbol{w}_{0,k} - \boldsymbol{u}_{n,k} + \frac{\nabla f\big(\boldsymbol{u}_{n,k}; X_{n,k}\big)}{\alpha_{n,k}}\right\} \tag{3.8b}$$

$$\boldsymbol{v}_{n,k} = \boldsymbol{v}_{n,k-1} - \frac{\delta_n\big(\boldsymbol{v}_{n,k-1} - \boldsymbol{u}_{n,k}\big) + \nabla f\big(\boldsymbol{u}_{n,k}; X_{n,k}\big) - \boldsymbol{g}_{n,k}}{\delta_n + \alpha_{n,k}\beta_k} \tag{3.8c}$$

where subscript $k-1$ is the index of the previous slot; while $\alpha_{n,k}$ and β_k denote stepsizes as before; and $X_{n,k}$ is a random sample at slot k. Without adhering to (3.8a)–(3.8c), faulty agents generate model parameter $\{\boldsymbol{w}_{n,k}\}_{n=N_{\mathrm{R}}+1}^{N}$ using an unknown mechanism.

Based on (3.6) and (3.8), our novel FRPG solver of (3.5) is listed under Algorithm 1 with lines 5–12 showing that the agents generate their local model parameter in parallel. The motivations of several key steps in the FRPG algorithm are as follows.

- After receiving $\boldsymbol{g}_{n,k}$, the server performs summation over all penalty gradients $\{\boldsymbol{g}_{n,k}\}_{n=1}^{Q}$ as shown in (3.6c). Since the server has no information on faculty agents, the negative effects of faulty agents are mitigated by using bounded penalty gradients $\{\nabla_{\boldsymbol{w}_n} p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)\}_{n=1}^{Q}$. More specifically, we observe from the term $\sum_{n=1}^{Q} \boldsymbol{g}_{n,k}$ in (3.6c) that the impacts of a reliable agent and a faulty agent on $\boldsymbol{v}_{0,k}$ are similar. Recalling the bounded gradient property of penalty, we envision that the number of faulty agents (instead of the magnitudes of faulty parameters $\{\boldsymbol{w}_{n,k}\}_{n=N_{\mathrm{R}}+1}^{Q}$) will have influence on the update in (3.6c). In this case, the FRPG algorithm is robust to any type of faulty agents.

- After receiving $\boldsymbol{w}_{0,k}$, each reliable agent n performs the local calculation (3.8). In the presence of faculty agents, it is reasonable to allow a slight difference between the reliable parameters $\{\boldsymbol{w}_{n,k}\}_{n=1}^{N_\mathrm{R}}$ and the server parameter $\boldsymbol{w}_{0,k}$ in slot k. Therefore, the proximal step in (3.8b) is used to obtain the reliable parameter $\boldsymbol{w}_{n,k}$ while retaining a slight difference from the server parameter $\boldsymbol{w}_{0,k}$, $n = 1, \ldots, N_\mathrm{R}$. Besides, the local parameter $\boldsymbol{w}_{n,k}$ is updated based on $\nabla_{\boldsymbol{w}_n} p_n(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k})$ when the proximal step (3.8b) is used; otherwise, the local parameter $\boldsymbol{w}_{n,k}$ is updated based on outdated information $\nabla_{\boldsymbol{w}_n} p_n(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k-1})$ that slows down the convergence.

Note that while our algorithm is inspired by [120], the updates in (3.6) and (3.8) are distinct in three aspects. The update step in (3.6b) does not require a proximal operation since $\boldsymbol{w}_{0,k}$ and $\boldsymbol{w}_{n,k}$ must be recursively updated. Since FRPG is a distributed algorithm, the update step in (3.6c) contains penalty gradients $\{\boldsymbol{g}_{n,k}\}_{n=N_\mathrm{R}+1}^{N}$ from faulty agents. These three differences render the ensuing convergence analysis of FRPG challenging.

Algorithm 1 FRPG Algorithm

Initialize: $\boldsymbol{w}_{n,0}$ and $\boldsymbol{v}_{n,0}$ for $n = 0, \ldots, N$, and stepsizes as (3.18)
for $k = 1, \ldots, K$ **do**
  The server updates $\boldsymbol{u}_{0,k}$ and $\boldsymbol{w}_{0,k}$ via (3.6a) and (3.6b)
  The server broadcasts the model parameter $\boldsymbol{w}_{0,k}$
  parfor $n = 1, \ldots, N$ **do** ▷ Parallel Computation
    if $n = 1, \ldots, N_\mathrm{R}$ **then**
      Reliable agent n updates $\boldsymbol{w}_{n,k}$ via (3.8)
    end if
    if $n = N_\mathrm{R} + 1, \ldots, N$ **then**
      Faulty agent n generates faulty parameter
    end if
  end parfor
  All agents upload $\boldsymbol{g}_{n,k}$ to the server
  The server updates $\boldsymbol{v}_{0,k}$ via (3.6c)
end for

3.2.2 Convergence Analysis

Our analysis here is for a single realization of X_n in each slot, but can be directly extended to mini-batch realizations of X_n. Let us define the gradient error at agent n in slot k as

$$\boldsymbol{\zeta}_{n,k} := \nabla f\big(\boldsymbol{u}_{n,k}; X_{n,k}\big) - \nabla f_n\big(\boldsymbol{u}_{n,k}\big) \tag{3.9}$$

and adopt the following assumption on its moments [121].

Assumption 3.4 (Bounded Stochastic Noise)**.** *The gradient error (a.k.a. noise) is zero mean, that is* $\mathbb{E}_{X_{n,k}}[\boldsymbol{\zeta}_{n,k}] = 0$*, with bounded variance* $\mathbb{E}_{X_{n,k}}[\|\boldsymbol{\zeta}_{n,k}\|^2] \leq \sigma_n^2$*, for* $n = 1, \ldots, N$*.*

When the data samples are uniformly drawn from a local dataset, the gradient error in (3.9) is zero mean. In our analysis, we consider the worst-case effect of gradient error by choosing a large value of σ_n^2.

Lemma 3.1. *If Assumptions 3.1–3.3 hold, eq.* (3.6b) *implies that*

$$\begin{aligned} f_0(\boldsymbol{w}_{0,k}) - f_0(\boldsymbol{u}_0) \leq \left\langle \sum_{n=1}^{N_\mathrm{R}} \boldsymbol{g}_{n,k} + \boldsymbol{\zeta}_{0,k}, \boldsymbol{u}_0 - \boldsymbol{w}_{0,k} \right\rangle - \left(\alpha_{0,k} - \frac{L_0}{2}\right) \left\|\boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k}\right\|^2 \\ + \left\| \sum_{n=1}^{N} \boldsymbol{g}_{n,k} \right\| \left\|\boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k}\right\| - \frac{\delta_0}{2} \left\|\boldsymbol{u}_0 - \boldsymbol{u}_{0,k}\right\|^2 \\ - \left\langle \alpha_{0,k}\left(\boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k}\right) + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\rangle, \forall \boldsymbol{u}_0 \end{aligned} \tag{3.10}$$

where $\boldsymbol{\zeta}_{0,k} \coloneqq \sum_{n=N_\mathrm{R}+1}^{N} \boldsymbol{g}_{n,k}$*.*

Lemma 3.2. *If Assumptions 3.1–3.3 hold, eq.* (3.8b) *implies that*

$$\begin{aligned} f_n(\boldsymbol{w}_{n,k}) - f_n(\boldsymbol{u}_n) \leq \left\langle \boldsymbol{\zeta}_{n,k} - \boldsymbol{g}_{n,k}, \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle - \left(\alpha_{n,k} - \frac{L_n}{2}\right) \left\|\boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k}\right\|^2 \\ - \alpha_{n,k} \left\langle \boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k}, \boldsymbol{u}_n - \boldsymbol{u}_{n,k} \right\rangle - \frac{\delta_n}{2} \left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2, \forall \boldsymbol{u}_n. \end{aligned} \tag{3.11}$$

As $p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n)$ is a convex and differentiable function (cf. Assumption 3), eq. (3.7) implies that $\gamma \nabla_{\boldsymbol{w}_n} p_n(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k}) = -\boldsymbol{g}_{n,k}$, and thus

$$\gamma p_n(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k}) - \gamma p_n(\boldsymbol{u}_0 - \boldsymbol{u}_n) \leq \left\langle \boldsymbol{g}_{n,k}, \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle - \left\langle \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{w}_{0,k} \right\rangle. \tag{3.12}$$

Summing up (3.10)–(3.12) and using the definition of $F(\boldsymbol{w})$ in (3.5), we obtain

$$\begin{aligned} F(\boldsymbol{w}_k) - F(\boldsymbol{u}) \leq \sum_{n=0}^{N_\mathrm{R}} \left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle + \left\| \sum_{n=1}^{N} \boldsymbol{g}_{n,k} \right\| \left\|\boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k}\right\| \\ - \sum_{n=0}^{N_\mathrm{R}} \left(\alpha_{n,k} - \frac{L_n}{2}\right) \left\|\boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k}\right\|^2 - \sum_{n=0}^{N_\mathrm{R}} \frac{\delta_n}{2} \left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2 \end{aligned} \tag{3.13}$$

$$-\sum_{n=1}^{N_{\mathrm{R}}}\alpha_{n,k}\left\langle \boldsymbol{u}_{n,k}-\boldsymbol{w}_{n,k},\boldsymbol{u}_n-\boldsymbol{u}_{n,k}\right\rangle-\left\langle \alpha_{0,k}\left(\boldsymbol{u}_{0,k}-\boldsymbol{w}_{0,k}\right)+\sum_{n=1}^{N}\boldsymbol{g}_{n,k},\boldsymbol{u}_0-\boldsymbol{u}_{0,k}\right\rangle$$

where $\boldsymbol{w}_k := \mathrm{vec}(\boldsymbol{w}_{0,k},\ldots,\boldsymbol{w}_{N,k})$, and $\boldsymbol{u} := \mathrm{vec}(\boldsymbol{u}_0,\ldots,\boldsymbol{u}_N)$.

Based on the definition of $\boldsymbol{\zeta}_{0,k}$ and Assumption 3.3, it follows that $\|\boldsymbol{\zeta}_{0,k}\| \leq N_{\mathrm{B}}\|\boldsymbol{g}_{1,k}\|$ and $\|\sum_{n=1}^{N}\boldsymbol{g}_{n,k}\| \leq N\|\boldsymbol{g}_{1,k}\|$. Using also that $\|\boldsymbol{g}_{1,k}\|^2 \leq \gamma^2 G$, $\|\sum_{n=1}^{N}\boldsymbol{g}_{n,k}\| \leq N\|\boldsymbol{g}_{1,k}\|$ and $\|\boldsymbol{\zeta}_{0,k}\| \leq N_{\mathrm{B}}\|\boldsymbol{g}_{1,k}\|$, we deduce that

$$\left(\left\|\sum_{n=1}^{N}\boldsymbol{g}_{n,k}\right\|+\left\|\boldsymbol{\zeta}_{0,k}\right\|\right)^2 \leq \gamma^2(N+N_{\mathrm{B}})^2 G := \sigma_0^2. \tag{3.14}$$

Lemma 3.3. *Under Assumptions 3.1–3.4 and optimal solution $\boldsymbol{u}^*$, FRPG obtains model parameter $\boldsymbol{w}_k$ that satisfies*

$$F(\boldsymbol{w}_k)-F(\boldsymbol{u}^*) \leq (1-\beta_k)(F(\boldsymbol{w}_{k-1})-F(\boldsymbol{u}^*))+\sum_{n=0}^{N_{\mathrm{R}}}\left(\lambda_{5,n,k}+\lambda_{6,n,k}\right)+\frac{2\gamma^2N^2G}{\alpha_{0,k}}+\frac{\gamma^2N_{\mathrm{B}}^2G}{2\epsilon}\beta_k \tag{3.15}$$

where $\epsilon > 0$, while the scalars $\lambda_{5,n,k}$ and $\lambda_{6,n,k}$ are specified by

$$\lambda_{5,n,k} := \begin{cases} \frac{\epsilon\beta_k+\alpha_{0,k}\beta_k^2}{2}\left\|\boldsymbol{u}_0-\boldsymbol{v}_{0,k-1}\right\|^2-\frac{\delta_0\beta_k+\alpha_{0,k}\beta_k^2}{2}\left\|\boldsymbol{u}_0-\boldsymbol{v}_{0,k}\right\|^2, n=0 \\ \frac{\alpha_{n,k}\beta_k^2}{2}\left\|\boldsymbol{u}_n-\boldsymbol{v}_{n,k-1}\right\|^2-\frac{\delta_n\beta_k+\alpha_{n,k}\beta_k^2}{2}\left\|\boldsymbol{u}_n-\boldsymbol{v}_{n,k}\right\|^2, n=1,\ldots,N_{\mathrm{R}} \end{cases} \tag{3.16}$$

and

$$\lambda_{6,n,k} = \begin{cases} \frac{3\sigma_0^2}{2\left(2\alpha_{0,k}-3L_0\right)}, n=0 \\ \frac{\sigma_n^2}{2\left(\alpha_{n,k}-L_n\right)}, n=1,\ldots,N_{\mathrm{R}}. \end{cases} \tag{3.17}$$

Using Lemma 3.3, we now assert the convergence of the FRPG algorithm.

Theorem 3.4 (Convergence of FRPG)**.** *If under Assumptions 3.1–3.4, the stepsizes are updated as*

$$\alpha_{n,k} = \begin{cases} \frac{\delta_0}{14}(k+2)^2+\frac{3}{2}L_0, n=0 \\ \frac{3\delta_n}{14}(k+2)^2+L_n, n=1,\ldots,N \end{cases} \quad \text{and } \beta_k = \frac{2}{k+2} \tag{3.18}$$

FRPG converges as

$$F(\boldsymbol{w}_K)-F(\boldsymbol{u}^*) \leq \frac{4[F(\boldsymbol{w}_0)-F(\boldsymbol{u}^*)+\lambda_9]}{(K+2)^2}+\frac{4\lambda_{10}K}{(K+2)^2}+O\left(\frac{\gamma^2N_{\mathrm{B}}^2G}{\delta_0}\right) \tag{3.19}$$

where K is the number of communication rounds, while scalars λ_9 and λ_{10} are defined as

$$\lambda_9 := \left(\frac{3}{8}\delta_0 + \frac{1}{2}\alpha_{0,1}\right)\left\|\boldsymbol{u}_0^* - \boldsymbol{v}_{0,0}\right\|^2 + \sum_{n=1}^{N_\mathrm{R}} \frac{1}{2}\alpha_{n,1}\left\|\boldsymbol{u}_n^* - \boldsymbol{v}_{n,0}\right\|^2 \tag{3.20}$$

and

$$\lambda_{10} := \frac{8\gamma^2 N^2 G + 21\sigma_0^2}{8\delta_0} + \sum_{n=1}^{N_\mathrm{R}} \frac{7\sigma_n^2}{12\delta_n}. \tag{3.21}$$

As confirmed by the last term in (3.19), FRPG converges to a neighborhood of the optimum with radius on the same order as that of RSA [51]. Besides, the convergence rate of FRPG is $O(1/K^2 + 1/K)$, which is faster than $O(\log(K)/\sqrt{K})$ of RSA. This observation implies that FRPG is more communication-efficient than RSA. Base on the last term of (3.19), we observe that the radius increases with the number of faulty agents. Moreover, we can reduce the values of γ and G to improve convergence accuracy. Since each agent reports $\gamma\nabla_{\boldsymbol{w}_0} p_n\big(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k}\big)$ where $\|\nabla_{\boldsymbol{w}_0} p_n\big(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k}\big)\|^2 \leq G$, either γ or G cannot be zero. Otherwise, the server cannot obtain useful information from the agents. While achieving a faster convergence rate over RSA, FRPG still requires the agents to communicate with the parameter server on each slot. Our LFRPG algorithm developed in the next section reduces this overhead by skipping several communication rounds.

However, two questions remain: what is the convergence rate of LFRPG, and how does the convergence of LFRPG depend on the communication period between the agents and parameter server? We answer these two questions next.

3.3 Local Fault-Resilient Proximal Gradient

3.3.1 Algorithm

To reduce the communication overhead, the LFRPG algorithm allows the server to update periodically the model parameter $\boldsymbol{w}_0[i]$ and the stepsizes $\alpha_n[i]$ and $\beta[i]$ at the start of frame i, $n = 0, 1, \ldots, N_\mathrm{R}$ (as shown in Fig. 3.4). Here, each frame i consists of T slots. With $\boldsymbol{u}_0[i]$, $\boldsymbol{w}_0[i]$, and $\boldsymbol{v}_0[i]$ denoting the server sequences in frame i, the model parameter at the server are updated

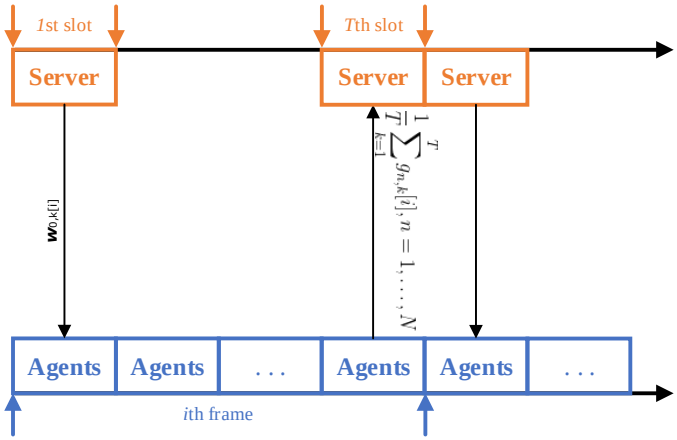


Figure 3.4: LFRPG iteration, where the server broadcasts $\boldsymbol{w}_0[i]$ at the beginning of frame i, and the agents upload $T^{-1}\sum_{k=1}^{T}\boldsymbol{g}_{n,k}[i]$ at the end of frame i, $n = 1, \ldots, N$.

as

$$\boldsymbol{u}_0[i] = (1-\beta[i])\boldsymbol{w}_0[i-1] + \beta[i]\boldsymbol{v}_0[i-1] \tag{3.22a}$$

$$\boldsymbol{w}_0[i] = \boldsymbol{u}_0[i] - \frac{1}{\alpha_0[i]}\nabla f_0(\boldsymbol{u}_0[i]) \tag{3.22b}$$

$$\boldsymbol{v}_0[i] = \boldsymbol{v}_0[i-1] - \frac{\delta_0(\boldsymbol{v}_0[i-1] - \boldsymbol{u}_0[i]) + \nabla f_0(\boldsymbol{u}_0[i]) + \frac{1}{T}\sum_{k=1}^{T}\sum_{n=1}^{N}\boldsymbol{g}_{n,k}[i]}{\delta_0 + \alpha_0[i]\beta[i]} \tag{3.22c}$$

where superscripts i and $i-1$ are indices of the corresponding frame in the sequences and stepsizes $\beta[i], \alpha_0[i]$; while $\boldsymbol{g}_{n,k}[i]$ is defined as

$$\boldsymbol{g}_{n,k}[i] := \gamma\nabla_{\boldsymbol{w}_0} p_n(\boldsymbol{w}_0[i] - \boldsymbol{w}_{n,k}[i]). \tag{3.23}$$

Accordingly, sequences at reliable agent n, slot k, and frame i are updated using stepsizes $\alpha_n[i], \beta[i]$, as

$$\boldsymbol{u}_{n,k}[i] = (1-\beta[i])\boldsymbol{w}_{n,k}[i-1] + \beta[i]\boldsymbol{v}_{n,k-1}[i] \tag{3.24a}$$

$$\boldsymbol{w}_{n,k}[i] = \boldsymbol{w}_0^i - \operatorname{prox}_{\frac{\gamma p_n}{\alpha_n[i]}}\left\{\boldsymbol{w}_0[i] - \boldsymbol{u}_{n,k}[i] + \frac{\nabla f\left(\boldsymbol{u}_{n,k}[i]; X_{n,k}[i]\right)}{\alpha_n[i]}\right\} \tag{3.24b}$$

$$\boldsymbol{v}_{n,k}[i] = \boldsymbol{v}_{n,k-1}[i] - \frac{\delta_n\left(\boldsymbol{v}_{n,k-1}[i] - \boldsymbol{u}_{n,k}[i]\right) + \nabla f\left(\boldsymbol{u}_{n,k}[i]; X_{n,k}[i]\right) - \boldsymbol{g}_{n,k}[i]}{\delta_m + \alpha_n[i]\beta[i]} \tag{3.24c}$$

while the resultant gradient noise is given by

$$\boldsymbol{\zeta}_{n,k}[i] := \nabla f\big(\boldsymbol{u}_{n,k}[i]; X_{n,k}[i]\big) - \nabla f_n\big(\boldsymbol{u}_{n,k}[i]\big). \tag{3.25}$$

Based on (3.22) and (3.24), our proposed LFRPG algorithm is listed in Algorithm 2, where lines 6–13 show that the agents update local model parameter in parallel. To proceed with convergence analysis of LFRPG, we need an assumption on the per-frame gradient noise too.

Algorithm 2 LFRPG Algorithm

1: **Initialize**: $\boldsymbol{w}_0[0]$, $\boldsymbol{v}_0[0]$, $\boldsymbol{w}_{n,1}[0]$ and $\boldsymbol{v}_{n,0}[1]$ for $n = 1, \ldots, N$, and stepsizes as (3.34).
2: **for** $i = 1, \ldots, I$ **do**
3: The server updates $\boldsymbol{u}_0[i]$ and $\boldsymbol{w}_0[i]$ via (3.22a) and (3.22b)
4: The server broadcasts the model parameter $\boldsymbol{w}_0^i$
5: **for** $k = 1, \ldots, T$ **do** ▷ Local Iterations
6: **parfor** $n = 1, \ldots, N$ **do** ▷ Parallel Computation
7: **if** $n = 1, \ldots, N_{\mathrm{R}}$ **then**
8: Reliable agent n updates $\boldsymbol{w}_{n,k}^i$ via (3.24)
9: **end if**
10: **if** $n = N_{\mathrm{R}} + 1, \ldots, N$ **then**
11: Faulty agent n generates faulty parameter
12: **end if**
13: **end parfor**
14: **end for**
15: All agents upload $\frac{1}{T}\sum_{k=1}^{T} \boldsymbol{g}_{n,k}[i]$ to the server
16: The server updates $\boldsymbol{v}_0^i$ via (3.22c)
17: **end for**

3.3.2 Convergence Analysis

Assumption 3.5 (Bounded Stochastic Noise)**.** *The gradient noise is zero mean; that is,* $\mathbb{E}_{X_n}[\boldsymbol{\zeta}_{n,k}[i]] = 0$*, with bounded mean-square error:* $\mathbb{E}_{X_n}[\|\boldsymbol{\zeta}_{n,k}[i]\|^2] \le \sigma_n^2$*, for* $n = 1, \ldots, N$*.*

Lemma 3.5. *Under Assumptions 3.1–3.3 and 3.5, eq.* (3.22b) *implies that*

$$f_0(\boldsymbol{w}_0^i) - f_0(\boldsymbol{u}_0) \le \left\langle \sum_{n=1}^{N_{\mathrm{R}}} \boldsymbol{g}_{n,k}[i] + \boldsymbol{\zeta}_{0,k}[i], \boldsymbol{u}_0 - \boldsymbol{w}_0^i \right\rangle - \left(\alpha_0[i] - \frac{L_0}{2}\right) \|\boldsymbol{u}_0[i] - \boldsymbol{w}_0[i]\|^2 \tag{3.26}$$

$$+ \left\| \sum_{n=1}^{N} \boldsymbol{g}_{n,k}[i] \right\| \|\boldsymbol{u}_0[i] - \boldsymbol{w}_0[i]\| - \frac{\delta_0}{2} \left\|\boldsymbol{u}_0 - \boldsymbol{u}_0^i\right\|^2 \tag{3.27}$$

$$- \left\langle \alpha_0[i](\boldsymbol{u}_0[i] - \boldsymbol{w}_0[i]) + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}[i], \boldsymbol{u}_0 - \boldsymbol{u}_0[i] \right\rangle, \forall \boldsymbol{u}_0$$

where $\boldsymbol{\zeta}_{0,k}[i] := \sum_{n=N_{\mathrm{R}}+1}^{N} \boldsymbol{g}_{n,k}[i]$*.*

Lemma 3.6. *Under Assumptions 3.1–3.3 and 3.5, eq.* (3.24b) *implies that*

$$f_n(\boldsymbol{w}_{n,k}[i]) - f_n(\boldsymbol{u}_n) \leq \left\langle \boldsymbol{\zeta}_{n,k}[i] - \boldsymbol{g}_{n,k}[i], \boldsymbol{u}_n - \boldsymbol{w}_{n,k}[i] \right\rangle - \left(\alpha_n[i] - \frac{L_n}{2}\right) \left\|\boldsymbol{u}_{n,k}[i] - \boldsymbol{w}_{n,k}[i]\right\|^2 \tag{3.28}$$

$$- \alpha_n[i] \left\langle \boldsymbol{u}_{n,k}[i] - \boldsymbol{w}_{n,k}[i], \boldsymbol{u}_n - \boldsymbol{u}_{n,k}[i] \right\rangle \tag{3.29}$$

$$-\frac{\delta_n}{2} \left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}[i]\right\|^2, \forall \boldsymbol{u}_n.$$

Since $\boldsymbol{u}_0[i]$ and $\boldsymbol{w}_0[i]$ are updated at the start of frame i, we set $\boldsymbol{u}_0[i] = \boldsymbol{u}_{0,k}[i]$ and $\boldsymbol{w}_0[i] = \boldsymbol{w}_{0,k}[i]$ with $k = 1, \ldots, T$. Summing (3.26) and (3.28), it follows after straightforward algebraic manipulations that the overall loss at $\boldsymbol{w}_k[i] := \mathrm{vec}(\boldsymbol{w}_{0,k}[i], \boldsymbol{w}_{1,k}[i], \ldots, \boldsymbol{w}_{N,k}[i])$ obeys

$$\begin{aligned} F(\boldsymbol{w}_k[i]) - F(\boldsymbol{u}) \leq & \sum_{n=0}^{N_\mathrm{R}} \left\langle \boldsymbol{\zeta}_{n,k}[i], \boldsymbol{u}_n - \boldsymbol{w}_{n,k}[i] \right\rangle + \left\| \sum_{n=1}^{N} \boldsymbol{g}_{n,k}[i] \right\| \|\boldsymbol{u}_0[i] - \boldsymbol{w}_0[i]\| \\ & - \sum_{n=0}^{N_\mathrm{R}} \frac{\delta_n}{2} \left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}[i]\right\|^2 - \sum_{n=1}^{N_\mathrm{R}} \alpha_{n,k}[i] \left\langle \boldsymbol{u}_{n,k}[i] - \boldsymbol{w}_{n,k}[i], \boldsymbol{u}_n - \boldsymbol{u}_{n,k}[i] \right\rangle \\ & - \sum_{n=0}^{N_\mathrm{R}} \left(\alpha_n[i] - \frac{L_n}{2}\right) \|\boldsymbol{u}_0[i] - \boldsymbol{w}_0[i]\|^2 \\ & - \left\langle \alpha_n[i](\boldsymbol{u}_0[i] - \boldsymbol{w}_0[i]) + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}[i], \boldsymbol{u}_0 - \boldsymbol{u}_{0,k}^i \right\rangle. \end{aligned} \tag{3.30}$$

Lemma 3.7. *Under Assumptions 3.1–3.3 and 3.5, LFRPG obtains model parameter* $\boldsymbol{w}_k[i]$ *that satisfies*

$$\begin{aligned} \frac{1}{T} \sum_{k=1}^{T} F(\boldsymbol{w}_k[i]) - F(\boldsymbol{u}^*) \leq & (1 - \beta[i]) \left(\frac{1}{T} \sum_{k=1}^{T} F(\boldsymbol{w}_k[i-1]) - F(\boldsymbol{u}^*) \right) + \frac{\gamma^2 N_\mathrm{B}^2 G}{2\epsilon} \beta[i] \\ & + \sum_{n=0}^{N_\mathrm{R}} \beta^2[i] \left(\lambda_{14,n}[i] + \lambda_{15,n}[i]\right) \end{aligned} \tag{3.31}$$

where $\lambda_{14,n}[i]$ *and* $\lambda_{15,n}[i]$ *are defined as*

$$\lambda_{14,n}[i] := \begin{cases} \frac{3\sigma_0^2 + 8\gamma^2 N^2 G}{2(2\alpha_0[i] - 3L_0)\beta^2[i]}, & n = 0 \\ \frac{\sigma_n^2}{2(\alpha_n[i] - L_n)\beta^2[i]}, & n = 1, \ldots, N_\mathrm{R} \end{cases} \tag{3.32}$$

and

$$\lambda_{15,n}[i] := \begin{cases} \frac{\epsilon+\alpha_0[i]\beta[i]}{2\beta[i]} \left\|\boldsymbol{u}_0^* - \boldsymbol{v}_0[i-1]\right\|^2 - \frac{\delta_0+\alpha_0[i]\beta[i]}{2\beta[i]} \left\|\boldsymbol{u}_0^* - \boldsymbol{v}_0[i]\right\|^2, & n = 0 \\ \frac{\alpha_n[i]}{2T} \left\|\boldsymbol{u}_n^* - \boldsymbol{v}_{n,T}[i-1]\right\|^2 - \frac{\delta_n+\alpha_n[i]\beta[i]}{2T\beta[i]} \left\|\boldsymbol{u}_n^* - \boldsymbol{v}_{n,T}[i]\right\|^2, & n = 1,\ldots,N_{\mathrm{R}}. \end{cases} \tag{3.33}$$

Lemma 3.7 leads to the convergence result for LFRPG.

Theorem 3.8 (Convergence of LFRPG)**.** *If Assumptions 3.1–3.3 and 3.5 hold, and stepsizes are respectively updated as*

$$\alpha_n[i] = \begin{cases} \frac{\delta_0}{14}(i+2)^2 + \frac{3}{2}L_0, & n = 0 \\ \frac{3\delta_n}{14}(i+2)^2 + L_n, & n = 1,\ldots,N \end{cases} \quad \text{and } \beta[i] = \frac{2}{i+2} \tag{3.34}$$

then LFRPG obtains model parameter $\bar{\boldsymbol{w}}[I] := T^{-1}\sum_{k=1}^{T} \boldsymbol{w}_k[I]$ *that converges*

$$F(\bar{\boldsymbol{w}}[I]) - F(\boldsymbol{u}^*) \leq \frac{2\lambda_{16}}{T(I+2)^2} + \frac{\lambda_{17} + I\lambda_{18}}{(I+2)^2} + O\left(\frac{\gamma^2 N_{\mathrm{B}}^2 G}{\delta_0}\right) \tag{3.35}$$

where λ_{16}*,* λ_{17} *and* λ_{18} *are respectively defined as*

$$\lambda_{16} := \sum_{n=1}^{N_{\mathrm{R}}} \alpha_n[1] \left\|\boldsymbol{u}_n^* - \boldsymbol{v}_n[0]\right\|^2 \tag{3.36}$$

$$\lambda_{17} := \left(\frac{3}{2}\delta_0 + 2\alpha_0[1]\right) \left\|\boldsymbol{u}_0^* - \boldsymbol{v}_0[0]\right\|^2 + \frac{4}{T}\sum_{k=1}^{T} F(\boldsymbol{w}_k[0]) - 4F(\boldsymbol{u}^*) \tag{3.37}$$

and

$$\lambda_{18} := \sum_{n=1}^{N_{\mathrm{R}}} \frac{7\sigma_n^2}{3\delta_n} + \frac{11\sigma_0^2 + 28\gamma^2 N^2 G}{\delta_0}. \tag{3.38}$$

We observe that the first term of (3.35) converges faster than that of (3.19), the second term of (3.35) converges with same rate as that of (3.19), and the third term of (3.35) confirms that the convergence accuracy of LFRPG is the same as FRPG. Therefore, eq. (3.35) reveals that LFRPG outperforms FRPG in communication efficiency. Based on (3.35), we observe that a long communication period of LFRPG can only reduce the first term of (3.35) and has no effect on the second and the third terms of (3.35). In other words, the communication reduction brought by the intermittent communication of LFRPG is diminishing with respect to the communication period.

3.4 Numerical Results

To validate our analytical results, we test the performance of FRPG and LFRPG numerically on real datasets (USPS[11] [122], MNIST[12] [123], and FMNIST[13] [124]). In the USPS set, we use $8,000$ data vectors of size 256×1 for training and $3,000$ for testing. In MNIST, we use $60,000$ data vectors of size 784×1 for training, and $10,000$ for testing. In FMNIST, we use $60,000$ data vectors of size 784×1 for training, and $10,000$ for testing. The top-1 accuracy is the ratio that predicted labels match the correct labels in the test set of data samples. The predicted labels are calculated by using the obtained model parameter at the server. The heterogeneity of datasets was manifested as follows. Each pair of agents is assigned data samples with the same labels, and 50% of the data samples were removed. For example, the data samples with labels 6, 7, 8, and 9 are removed in half of the tests. We consider the Label-Flipping attack [44] and the Gaussian attack [34] to verify the robustness of FRPG and LFRPG. For the Label-Flipping attack, the original label y is skewed to $9 - y$; while for the Gaussian attack the faulty gradient is set as $c \times \mathcal{N}(0, 1)$ with $c = 1 \times 10^4$. The numerical experiments are run on MATLAB R2018b with Intel i7-8700 CPU @ 3.20 GHz and 16 Gb RAM.

The multinomial logistic regression was employed as the loss with regularizer $(\delta_n/2)\|\boldsymbol{w}_n\|^2$. At the parameter server, we set $f_0(\boldsymbol{w}_0) = (\delta_0/2)\|\boldsymbol{w}_0\|^2$. Huber's cost with smoothing constant $\mu = 10^{-3}$ was adopted as the penalty function

$$p_n(\boldsymbol{w}_0 - \boldsymbol{w}_n) = \begin{cases} \frac{1}{2\mu}\|\boldsymbol{w}_0 - \boldsymbol{w}_n\|^2, \|\boldsymbol{w}_0 - \boldsymbol{w}_n\| \leq \mu \\ \|\boldsymbol{w}_0 - \boldsymbol{w}_n\| - \frac{\mu}{2}, \text{otherwise.} \end{cases} \tag{3.39}$$

We considered a setting with $N = 20$ agents, $N_{\mathrm{R}} = 16$ reliable ones, and weight $\gamma = 1.6$. The training data are evenly distributed across the agents. With faulty agents attacking by flipping labels, the mini-batch size is set to 15; while for those adopting a Gaussian attack, the mini-batch size is set to 10. To obtain a good top-1 accuracy convergence, we set the stepsizes for benchmark

[11]The USPS dataset contains grayscale images of postcodes that are automatically scanned from envelopes by the U.S. Postal Service.

[12]The MNIST dataset contains grayscale handwritten digits, which range from zero to nine. The digits are written by high school students and employees of the United States Census Bureau.

[13]The FMNIST dataset contains grayscale pictures for fashion products from ten categories, namely T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The pictures are collected by Zalando Research.

schemes to $3/\sqrt{k}$. A strongly convex modulus with $\delta_n = 0.003$ is chosen for $n = 0, 1, \ldots, N$; while the Lipschitz constants for the USPS, MNIST and FMNIST datasets are respectively set to 156, 295, and 524. The agents in LFRPG communicate with the server every ten slots.

We also test communication efficiency in comparison with Krum [43], GeoMed [42], and RSA [51] benchmarks. To demonstrate the negative effects of different attacks, we employ SGD by averaging the local gradients of agents heuristically. Figs. 3.5 and 3.6 show the convergence of FRPG, LFRPG and RSA under Label-Flipping, and Gaussian attacks, respectively. After $4,000$ communication rounds, FRPG and LFRPG converge faster than RSA, while LFRPG outperforms FRPG for the same number of rounds. To reach the same loss value with the FMNIST dataset, LFRPG takes about 400 communication rounds versus 800 required by FRPG under Label-Flipping attacks.

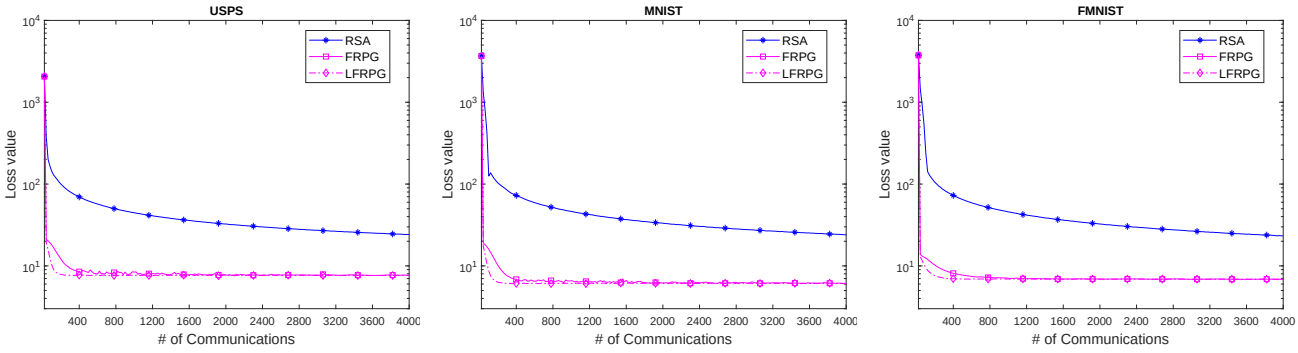


Figure 3.5: The loss values over the number of communication rounds under Label-Flipping attack and heterogeneous datasets.

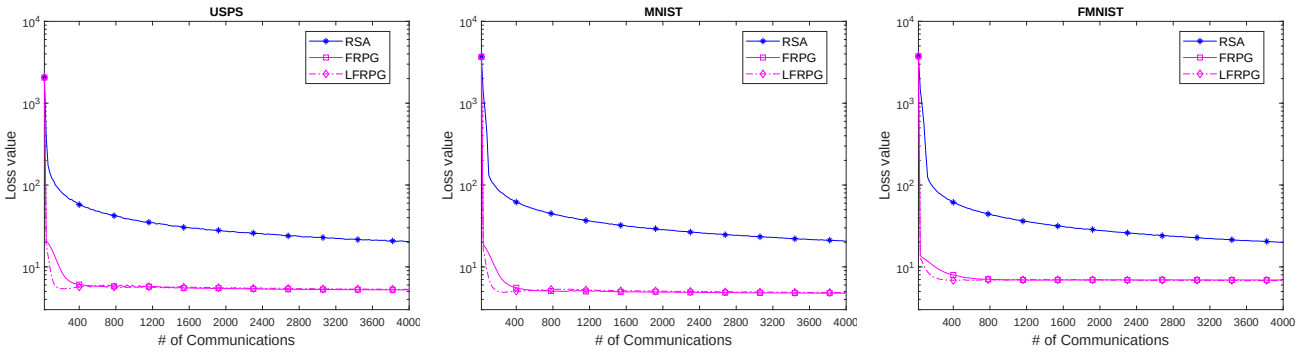


Figure 3.6: The loss values over the number of communication rounds under Gaussian attack and heterogeneous datasets.

Figure 3.7 compares the top-1 accuracy with Krum, GeoMed and RSA, under Label-Flipping attacks, respectively. With Label-Flipping, both FRPG and LFRPG converge faster than the

benchmarks. FRPG and LFRPG also achieve better top-1 accuracy for the USPS, MNIST, and FMNIST datasets, while Krum fails because it is designed for homogeneous datasets. When the USPS dataset is used, we observe that the top-1 accuracy of FRPG reaches about 71% after about 1200 communication rounds, and the top-1 accuracy of LFRPG reaches about 71% after 400 communication rounds. LFRPG allows agents to communicate with the parameter server every ten slots, during which each reliable agent updates the local model parameter based on the local dataset. Compared with FRPG, LFRPG has ten times the local computational cost. Therefore, we conclude that LFRPG can reduce the communication overhead at the expense of local computational cost. Besides, SGD requires 4000 communication rounds to reach the same top-1 accuracy as FRPG and LFRPG. GeoMed and RSA require more than 4000 communication rounds to achieve 71% accuracy. In other words, FRPG and LFRPG can reduce at-least 70% and 90% of communication overhead when the USPS dataset is used. Besides, the reduction of communication overhead can also be observed when the MNIST and FMNIST datasets are used.

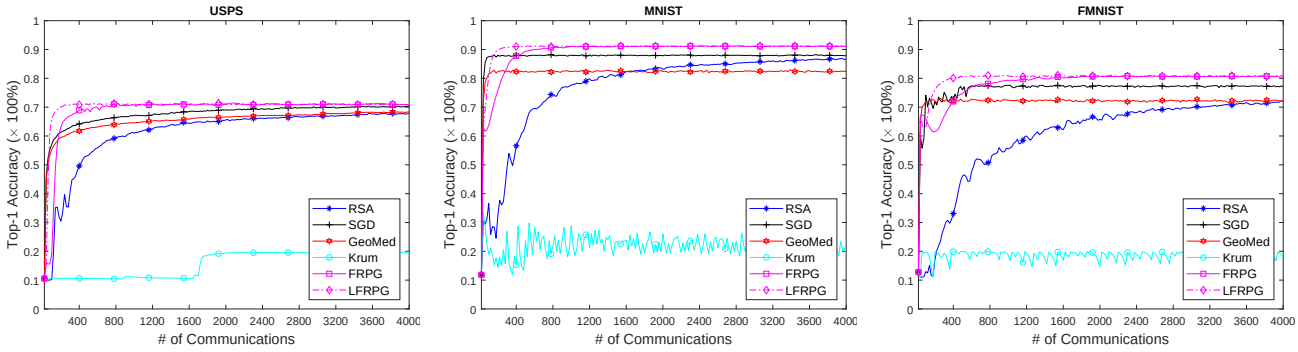


Figure 3.7: Top-1 accuracy over the number of communication rounds under Label-Flipping attack and heterogeneous datasets.

Since Label-Flipping attacks do not change the magnitude of local gradients, their negative effects on SGD are limited when heterogeneous datasets are used. For this reason, we considered the more severe Gaussian attack. Fig. 3.8 illustrates that SGD fails in the presence of Gaussian attacks. However, both FRPG and LFRPG converge faster and achieve better top-1 accuracy than Krum, GeoMed, and RSA. Using Gaussian attacks and the FMNIST dataset, the top-1 accuracy of FRPG and LFRPG is 4.13% better than that of GeoMed, and 9.89% better than that of RSA.

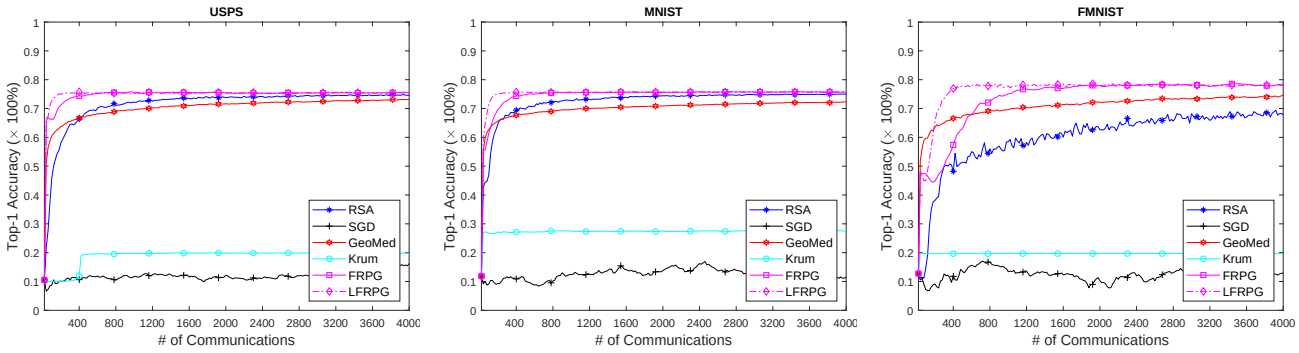


Figure 3.8: Top-1 accuracy over the number of communication rounds under Gaussian attack and heterogeneous datasets.

3.5 Summary

Adopting Nesterov's method, we have studied the fault resilience and communication efficiency issues in the federated learning framework. We have proposed two fault-resilient federated learning algorithms (FRPG and LFRPG algorithms) that can handle heterogeneous datasets. We have derived the convergence rates of the proposed FRPG and LFRPG algorithms. Using different practical datasets, we have performed numerical simulations to show the reduction in communication overhead. We have obtained the following engineering insights.

- Compared with the RSA algorithm, the obtained convergence rates show that the proposed FRPG and LFRPG algorithms can reduce the communication overhead from $\mathcal{O}(\log(k)/\sqrt{k})$ to $\mathcal{O}(1/k^2 + 1/k)$. Our numerical simulations also confirm the communication overhead over the benchmarks.

- Information exchanging between the parameter server and agents are necessary. Therefore, either γ or G cannot be set to zero. Besides, the convergence rate of LFRPG shows that the overhead reduction resulted from intermittent communication diminishes with respect to communication period.

Chapter 4

Linear-Approximate Decentralized Q-Learning

Motivated by the collaborative spectrum sensing and collaborative spectrum sharing problems in SGPCSs, decentralized Q-learning algorithms become a popular research topic. A potential application scenario of a decentralized Q-learning is discussed as follows. Several secondary BSTs collaboratively sense the spectrum bands of a primary BST, and exchange local sensing results with its neighboring secondary BSTs. Each secondary BST is powered by renewable energy and a smart grid, and consumes a certain amount of energy for spectrum sensing and exchanging local sensing results. Here, the environment state includes a map of spectrum occupancy and the renewable energy arrival at secondary BSTs. Each secondary BST decides whether to perform spectrum sensing or energy trading. A secondary BST needs to choose the set of spectrum bands for sensing if the secondary BST decides to perform spectrum sensing. A secondary BST needs to determine the number of energy units that will be sold to (or purchase from) a smart grid if the secondary BST decides to perform energy trading. The reward of each secondary BST is the number of identified vacant spectrum bands in each slot. The exchanged information between neighboring secondary BSTs is the local parameter that can characterize the statistics (i.e., whether the spectrum bands of primary BST are vacant or occupied) for the spectrum bands of primary BST.

Since the algorithmic and theoretical developments of decentralized Q-learning algorithms are limited, this chapter builds on this research front. In particular, a decentralized Q-learning is based on a multi-agent MDP, where agents have private rewards and collaboratively optimize a long-term global reward. Based on local behavior policy, each agent interacts with the environment and communicates with the other agents within its communication range (i.e., neighboring

agents). Then, the joint actions of agents determine the state transition of the environment. While current decentralized Q-learning algorithms mainly focus on tabular setups, this chapter proposes a decentralized Q-learning algorithm using LFA, called the linear-approximate decentralized Q-learning algorithm. Besides, a finite-sample analysis is performed to quantify the convergence rate of the linear-approximate decentralized Q-learning algorithm.

4.1 Preliminaries and Problem Statement

To facilitate the study of MARL, we begin by introducing some background on the multi-agent MDP, Q-function, and Bellman operators. We refer the readers to sources [29, 38, 115] for more details.

4.1.1 Collaborative Multi-Agent MDPs and the Q-Function

Consider a collaborative N-user MDP denoted by a hextuple $(\mathcal{S}, \{\mathcal{A}_n\}_{n=1}^N, \mathcal{P}, \{r_n\}_{n=1}^N, \gamma)$, where agents form an undirected network, $\mathcal{S}$ is the state space of size $|\mathcal{S}|$, and $\mathcal{A}_n$ is the action space of agent n. Moreover, we define the joint action space as $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. The set $\mathcal{P} = \{[p_{\boldsymbol{a}}^{s,s'}] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|} | s, s' \in \mathcal{S}, \boldsymbol{a} := [a_1, \cdots, a_N] \in \mathcal{A}\}$ collects all action-dependent transition probabilities, where $p_{\boldsymbol{a}}^{s,s'} = P(s'|s, \boldsymbol{a})$ is the probability of the environment transiting to state s' when taking joint action $\boldsymbol{a} \in \mathcal{A}$ at state $s \in \mathcal{S}$. Associated with a state transition, each agent n is given a real-valued reward $r_n(s, \boldsymbol{a})$ that is assumed upper-bounded by $r_{\max} > 0$, while $0 \leq \gamma < 1$ is the discounting factor. In this model, we assume that current state s of the environment, its successor state s', and the joint action $\boldsymbol{a}$ can be observed by all agents, whereas the rewards $\{r_n(s, \boldsymbol{a})\}_{n=1}^N$ corresponding to each transition are kept confidential for agent n. See Figure 4.1 for a depiction of such a model.

Let $\mu_n : \mathcal{S} \to \mathcal{A}_n$ denote the policy of agent n, which determines the probability of taking action $a_n \in \mathcal{A}_n$ at state $s \in \mathcal{S}$. The joint policy of all agents is denoted by $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_N]$, which along with $\mathcal{P}$ dictates the state transitions, the trajectory, and the stationary distribution $\boldsymbol{\pi}$ of the induced Markov chain $\{S_k\}_{k\geq 1}$. The quality of policy $\boldsymbol{\mu}$ is measured by the expectation of average discounted rewards of all agents given an initial state-action pair $(s, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$, while

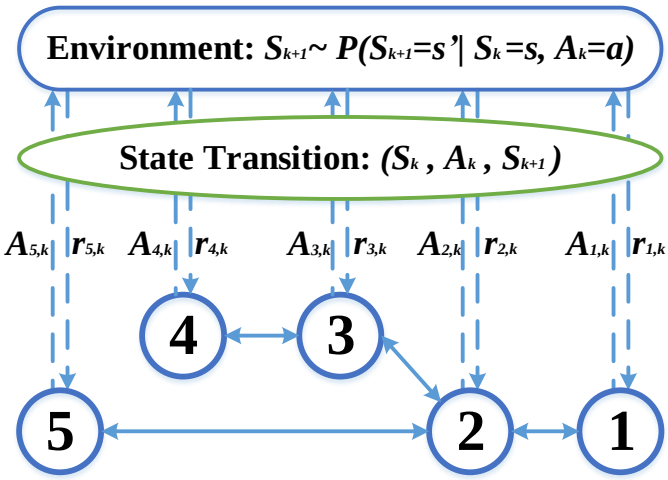


Figure 4.1: Illustration of a five-user MDP. In slot k, each agent n selects action $A_{n,k}$ based on state S_k following policy μ_n, the environment moves to state S_{k+1}, and agent n receives reward $r_{n,k}$.

following policy $\boldsymbol{\mu}$ to take future actions–that is, the so-called Q-function

$$Q(s,\boldsymbol{a}) = \mathbb{E}\left[\frac{1}{N}\sum_{k=1}^{\infty}\sum_{n=1}^{N}\gamma^k r_{n,k}\middle| S_1 = s, A_1 = \boldsymbol{a}\right] \tag{4.1}$$

where $r_{n,k} := r_n(S_k, A_k)$, and $A_k \sim \boldsymbol{\mu}(S_k)$ with $k \geq 1$.

4.1.2 Bellman Equation, Function Approximation, and Projected Bellman Equation

The goal is to obtain an optimal joint behavior policy $\boldsymbol{\mu}^*$ such that the Q-function in (4.1) is maximized for the state-action pair $(s,\boldsymbol{a}) \in \mathcal{S}\times\mathcal{A}$. For finite-state MDPs, it is well known [29, 74, 76] that there always exists an optimal deterministic policy $\boldsymbol{\mu}^*$, which gives rise to the optimal Q-function and satisfies the following Bellman's optimality equation [61, 81, 82, 87, 125]

$$Q^*(s,\boldsymbol{a}) = \mathbb{B}[Q^*(s,\boldsymbol{a})], \forall (s,\boldsymbol{a}) \in \mathcal{S}\times\mathcal{A} \tag{4.2}$$

where the Bellman operator $\mathbb{B}[\cdot]$ is defined as

$$\mathbb{B}[Q^*(s,\boldsymbol{a})] := \frac{1}{N}\sum_{n=1}^{N} r_n(s,\boldsymbol{a}) + \gamma \sum_{s'\in\mathcal{S}} p_{\boldsymbol{a}}^{s,s'}\left[\max_{\boldsymbol{a}'\in\mathcal{A}} Q^*(s',\boldsymbol{a}')\right]. \tag{4.3}$$

It has been shown [76, 80, 81] that the Bellman operator $\mathbb{B}[\cdot]$ is a contraction mapping with respect to the ℓ_∞-norm. In light of this fact, the optimal Q-function $Q^*(s, \boldsymbol{a})$ can be asymptotically found in a decentralized manner using stochastic approximation methods for solving (4.2) with suitable stepsizes, such as the decentralized tabular Q-learning algorithm [61]. However, when the state and action spaces become large, the explicit representation of Q-function for large state and action spaces becomes computationally burdensome or even intractable due to the issue of curse-of-dimensionality [38, 82]. To overcome this difficulty, a common approach is to combine the tabular Q-learning with parameterized function approximation, such as linear approximators [82] or a deep neural network [39]. Even though deep neural networks could offer more powerful approximations, the simplicity of reinforcement learning algorithms with LFA [83–87, 126] allows us to analyze them in detail.

We focus on LFA of the Q-function. Specifically, the approximate Q-function is defined as $Q(s, \boldsymbol{a}) \approx \langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w} \rangle$, where $\boldsymbol{\phi}(s, \boldsymbol{a}) = [\phi_1(s, \boldsymbol{a}), \ldots, \phi_d(s, \boldsymbol{a})]^\top \in \mathbb{R}^d$ is a basis vector, and $\boldsymbol{w} \in \mathbb{R}^d$ is the unknown parameter vector to be estimated, and $\|\boldsymbol{\phi}(s, \boldsymbol{a})\| \leq 1$ for all pairs $(s, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$ [82]. Oftentimes, we have the number of unknown parameters $d \ll |\mathcal{S}||\mathcal{A}|$ required by tabular Q-learning algorithms, thus offering scalability to deal with large state and/action spaces.

Upon fixing a canonical ordering on the elements $(s, \boldsymbol{a})$ of $\mathcal{S} \times \mathcal{A}$, we can define the feature matrix by stacking up the ordered feature vectors as its rows, namely $\boldsymbol{\Phi} = [\cdots \ \boldsymbol{\phi}(s, \boldsymbol{a}) \ \cdots]^\top \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d}$. The induced linear subspace of feature matrix $\boldsymbol{\Phi}$ is given by $\mathcal{Q} = \{\boldsymbol{\Phi}\boldsymbol{w} | \boldsymbol{w} \in \mathbb{R}^d\}$. In general, the optimal Q-function may not belong to $\mathcal{Q}$, so the exact solution of (4.2) cannot be obtainable. In the collaborative MARL setting, we are motivated to seek the best approximation of $Q^*(s, \boldsymbol{a})$ in the linear subspace $\mathcal{Q}$, which can be shown to satisfy the so-called projected Bellman equation

$$\tilde{Q}^*(s, \boldsymbol{a}) = \mathrm{proj}_{\mathcal{Q}} \left\{ \mathbb{B}\left[\tilde{Q}^*(s, \boldsymbol{a})\right] \right\}. \tag{4.4}$$

Due to the unknown transition probabilities and the unknown reward functions, the projected Bellman equation (4.4) cannot be directly evaluated. Nonetheless, any irreducible and aperiodic Markov chain converges geometrically fast to its unique stationary distribution (e.g., [127, Thm. 4.9]). Similar to the linear-approximate centralized Q-learning, an equivalent form of the projected Bellman equation (4.4) is first established that is amenable to deriving decentralized

Q-learning algorithms.

To that end, let $X := (s, \boldsymbol{a}, s')$ denote a state transition consisting of current state $s \in \mathcal{S}$, joint action $\boldsymbol{a} \in \mathcal{A}$, and the new state $s' \in \mathcal{S}$. Define the centralized TD error that is found with all local rewards

$$g(\boldsymbol{w}; X) = \frac{1}{N} \sum_{n=1}^{N} r_n(s, \boldsymbol{a}) + \gamma \max_{\boldsymbol{a}' \in \mathcal{A}} \langle \boldsymbol{\phi}(s', \boldsymbol{a}'), \boldsymbol{w} \rangle - \langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w} \rangle . \tag{4.5}$$

It can be shown that for any irreducible and aperiodic Markov chain, the Q-function $\tilde{Q}^*(s, \boldsymbol{a}) = \langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w}^* \rangle$ evaluated at the fixed point $\boldsymbol{w}^* \in \mathbb{R}^d$ of the next equation obeys (4.4) for all state-action pairs $(s, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$

$$\mathbb{E}_X [\nabla f(\boldsymbol{w}^*; X)] = 0 \tag{4.6}$$

where $\nabla f(w^*; X) := g(w^*; X)\boldsymbol{\phi}(s, \boldsymbol{a})$, π denotes the stationary distribution of the Markov chain. Please see Appendix B.1 for a proof and detailed discussions.

Algorithm 3 Linear-Approximate Decentralized Q-Learning

Input: Stepsize α_k, features $\boldsymbol{\Phi}$, mixing matrix $\boldsymbol{B}$
Initialize: Parameters $\{\boldsymbol{w}_{n,1}\}_{n=1}^N$.
for $k = 1, \ldots, \infty$ **do**
  for $n = 1, \ldots, N$ **do** ▷ Parallel computation
    Based on current state $S_{n,k}$, agent n takes action $A_{n,k}$ according to policy μ_n
    agent n observes the state transition (S_k, A_k, S_{k+1}) and a local reward $r_{n,k}$
    agent n receives parameters from its neighbors as $\{b_{n'n}\boldsymbol{w}_{n',k}\}_{n'=1}^N$
    agent n finds estimated action $\hat{A}_{n,k+1}$ via (4.9), and the TD error via (4.8)
    Each agent n updates $\boldsymbol{w}_{n,k+1}$ via (4.10)
  end for
end for

4.2 Linear-Approximate Decentralized Q-Learning

The goal is to develop a decentralized stochastic approximation scheme for finding the fixed-point solution of (4.6) with multiple agents acquiring data along a trajectory of a multi-agent MDP. Agents are allowed to exchange their local parameters with neighbors over a fixed communication network. The communication model is described by a mixing matrix $\boldsymbol{B} = \left[b_{n,n'}\right] \in \mathbb{R}^{N \times N}$, where $b_{n,n'}$ scales the information from agent n to n'. In particular, we extend the decentralized tabular Q-learning algorithm [61] to handle large state and action spaces by LFA. The resulting linear-approximate decentralized Q-learning algorithm is summarized in Algorithm 4.

In slot $k \geq 1$, each agent n takes an action $A_{n,k}$ based on the current state S_k by following its behavior policy μ_n, and the environment transits from current state S_k to a new state S_{k+1}. Meanwhile, agent n receives a private reward $r_n(S_k, A_k)$. Upon observing the state transition $X_k := (S_k, A_k, S_{k+1})$, each agent n computes the local TD error

$$\boldsymbol{g}_{n,k} := \boldsymbol{g}_n(\boldsymbol{w}_{n,k}; X_k) = r_n(S_k, A_k) + \gamma \max_{\boldsymbol{a}' \in \mathcal{A}} \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}), \boldsymbol{w}_{n,k} \right\rangle - \left\langle \boldsymbol{\phi}(S_k, A_k), \boldsymbol{w}_{n,k} \right\rangle \tag{4.7}$$

$$:= r_{n,k} + \gamma \left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_{n,k} \right\rangle - \left\langle \boldsymbol{\phi}_k, \boldsymbol{w}_{n,k} \right\rangle \tag{4.8}$$

where we define $r_{n,k} := r_n(S_k, A_k)$, $\boldsymbol{\phi}_k := \boldsymbol{\phi}(S_k, A_k)$, and $\hat{\boldsymbol{\phi}}_{n,k} := \boldsymbol{\phi}(S_{k+1}, \hat{A}_{n,k+1})$ for notational convenience, with the estimated joint action $\hat{A}_{n,k+1}$ by agent n in slot k given by

$$\hat{A}_{n,k+1} = \arg\max_{\boldsymbol{a}' \in \mathcal{A}} \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}'), \boldsymbol{w}_{n,k} \right\rangle. \tag{4.9}$$

Subsequently, agent n updates parameter $\boldsymbol{w}_{n,k}$ via the local "stochastic gradient" $\nabla f(\boldsymbol{w}_{n,k}; X_k) := \boldsymbol{g}_{n,k}\boldsymbol{\phi}_k$ with a stepsize $\alpha_k > 0$, and the parameters $\{b_{n',n}\boldsymbol{w}_{n',k}\}_{n'=1}^N$ received from its neighbors, by

$$\boldsymbol{w}_{n,k+1} = \sum_{n'=1}^{N} b_{n',n}\boldsymbol{w}_{n',k} + \alpha_k \boldsymbol{g}_{n,k}\boldsymbol{\phi}_k. \tag{4.10}$$

Similar to centralized Q-learning, Algorithm 4 does not require any statistical information on the multi-agent MDP, operates with fixed behavior policies of agents, and performs no projection steps unlike that analyzed in [89]. In other words, the proposed linear-approximate decentralized Q-learning is a model-free, and off-policy algorithm, whose benefits are that the learning and data sampling processes are decoupled; and learning process can be conveniently done through collected data trajectories. Hereinafter, we investigate the convergence properties, especially the statistical efficiency of Algorithm 4.

4.3 A Unifying Finite-Sample Convergence Analysis

The goal of this section is to analyze the finite-sample convergence performance of Algorithm 4 in a realistic setting where transitions are sampled along a trajectory of multi-agent MDP. For completeness, learning with both decaying and constant stepsizes will be studied. To proceed, we

will need the ensuing assumptions on the communication network and the behavior policies.

Assumption 4.1. *The communication network is undirected and connected. The mixing matrix* $\boldsymbol{B} = [b_{n,n'}]$ *is non-negative and doubly stochastic, namely* $\sum_{n=1}^{N} b_{n,n'} = 1$, $\sum_{n'=1}^{N} b_{n,n'} = 1$, *and* $b_{n,n'} \geq 0$.

Assumption 4.1 is standard and has been commonly adopted in decentralized optimization and learning; see e.g., [62, 63, 91, 92, 128]. On the other hand, it is well-known that Q-learning with function approximation can diverge in general [82, 129, 130]. This is mainly because Q-learning implements off-policy sampling to acquire data, which may render expectation of the Q-learning update (cf. (4.10)) diverge [131]. Under some regularity condition on the behavior policy, asymptotic convergence properties of the linear-approximate centralized Q-learning have been derived in [82]. Finite-sample performance guarantees have recently been provided in [86, 87, 89]. In light of those results, here we also introduce such a regularity condition for the collaborative multi-agent Q-learning on the joint policy $\boldsymbol{\mu}$, which is key to perform finite-sample analysis of the linear-approximate decentralized Q-learning algorithm in Algorithm 4.

Algorithm 4 Linear-Approximate Decentralized Q-Learning

Input: Stepsize α_k, features $\boldsymbol{\Phi}$, mixing matrix $\boldsymbol{B}$
Initialize: Parameters $\{\boldsymbol{w}_{n,1}\}_{n=1}^{N}$.
for $k = 1, \ldots, \infty$ **do**
  for $n = 1, \ldots, N$ **do** ▷ Parallel computation
    Based on current state $S_{n,k}$, agent n takes action $A_{n,k}$ according to policy μ_n
    agent n observes the state transition (S_k, A_k, S_{k+1}) and a local reward $r_{n,k}$
    agent n receives parameters from its neighbors as $\{b_{n'n}\boldsymbol{w}_{n',k}\}_{n'=1}^{N}$
    agent n finds estimated action $\hat{A}_{n,k+1}$ via (4.9), and the TD error via (4.8)
    Each agent n updates $\boldsymbol{w}_{n,k+1}$ via (4.10)
  end for
end for

Assumption 4.2. *Assume that the Markov chain* $\{S_k\}_{k\geq 1}$ *determined by the joint policy* $\boldsymbol{\mu}$ *and the transition probability matrices in* $\boldsymbol{P}$ *is irreducible and aperiodic. Denote its unique stationary distribution by* $\boldsymbol{\pi}$. *Furthermore, assume that the joint behavior policy* $\boldsymbol{\mu}$ *guarantees the following condition for all agents* n *with some constant* $0 < c_0 < 1$

$$\gamma^2 \mathbb{E}_X\left[\max_{\boldsymbol{a}'\in\mathcal{A}} \langle \boldsymbol{\phi}(s', \boldsymbol{a}'), \boldsymbol{w}_n\rangle^2\right] - \mathbb{E}_X\left[\langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w}_n\rangle^2\right] \leq -c_0\|\boldsymbol{w}_n\|_2^2. \tag{4.11}$$

Concerning Assumption 4.1, note that the first part (i.e., the irreducibility and aperiodicity) is a standard requirement for the theoretical analysis of reinforcement learning algorithms [81, 83–86, 132]. The second part (i.e., (4.11) for the joint policy) essentially guarantees the stability of our multi-agent, linear-approximate decentralized Q-learning algorithm, that resembles those imposed for their single-user counterparts in [82, 87].

When $d = |\mathcal{S}||\mathcal{A}|$ and $\gamma^2 \geq 1/|\mathcal{A}|$, there is no behavior policy satisfying Assumption 4.2 [87]. We consider another case that $d \ll |\mathcal{S}||\mathcal{A}|$. Setting the left-hand side of (4.11) to be less than zero, it holds that $\mathbb{E}\big[\sum_s \pi^s(\gamma^2 \max_{\boldsymbol{a}' \in \mathcal{A}} \langle \boldsymbol{\phi}(s, \boldsymbol{a}'), \boldsymbol{w}_n \rangle^2 - \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\mu}_{\boldsymbol{a}}^s \langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w}_n \rangle^2)\big] < 0$. One can employ the inequality $\gamma^2 < \sum_{\boldsymbol{a} \in \mathcal{A}'} \boldsymbol{\mu}_{\boldsymbol{a}}^s \langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w}_n \rangle^2 / \max_{\boldsymbol{a}' \in \mathcal{A}} \langle \boldsymbol{\phi}(s, \boldsymbol{a}'), \boldsymbol{w}_n \rangle^2$, where $\langle \boldsymbol{\phi}(s, \boldsymbol{a}), \boldsymbol{w}_n \rangle$ with $\boldsymbol{a} \in \mathcal{A}'$ and $s \in \mathcal{S}$. Since $w_n \in \mathbb{R}^d$ can be orthogonal to at most $d - 1$ basis functions, we have $|\mathcal{A}'| = |\mathcal{A}| - d + 1$. The inequality on γ^2 implies that the upper bound on γ increases with $|\mathcal{A}| - d$. Thus, we can find a behavior policy satisfying Assumption 4.2 when $d \ll |\mathcal{S}||\mathcal{A}|$ and $\gamma^2 > 1/|\mathcal{A}|$. Therefore, the proposed algorithm aims at solving a category of problems, where small dimensional feature vectors can represent the optimal Q-function (i.e., $d \ll |\mathcal{S}||\mathcal{A}|$).

Now, we are ready to analyze the convergence properties of Algorithm 4. Let us concatenate all local parameters $\{\boldsymbol{w}_{n,k}\}_{n=1}^N$ and stochastic gradients $\{\nabla f\big(\boldsymbol{w}_{n,k}; X_k\big)\}_{n=1}^N$ into vectors as follows

$$\nabla \boldsymbol{f}(\boldsymbol{w}; X) := \begin{bmatrix} \nabla f(\boldsymbol{w}_1; X) \\ \vdots \\ \nabla f(\boldsymbol{w}_N; X) \end{bmatrix} \text{ and } \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}; X) := \begin{bmatrix} \bar{f}(\boldsymbol{w}_1; X) \\ \vdots \\ \bar{f}(\boldsymbol{w}_N; X) \end{bmatrix} \tag{4.12}$$

where $\boldsymbol{w} = \text{vec}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N)$, $\nabla f(\boldsymbol{w}_n; X) := g_n(\boldsymbol{w}_n; X)\boldsymbol{\phi}(s, \boldsymbol{a})$ and $\bar{f}(\boldsymbol{w}_n; X) := g(\boldsymbol{w}_n; X)\boldsymbol{\phi}(s, \boldsymbol{a})$ based on the local and global TD errors in (4.8) and (4.5), respectively. Moreover, $\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}) = \mathbb{E}_X\big[\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}; X)\big]$, and it is evident from (4.6) that we have $\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}^*) = 0$ with $\boldsymbol{w}^* := \text{vec}(\boldsymbol{w}_1^*, \ldots, \boldsymbol{w}_N^*) \in \mathbb{R}^{Nd}$. With these definitions, the decentralized Q-learning update (4.10) can be compactly rewritten as follows

$$\boldsymbol{w}_{k+1} = (\boldsymbol{B} \otimes I_d)\, \boldsymbol{w}_k + \alpha_k \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k)\,. \tag{4.13}$$

Our goal now boils down to derive finite-sample bounds on the mean-square error of iterates $\boldsymbol{w}_k$ generated by (4.10) from the fixed point $\boldsymbol{w}^*$. Since dataset $\{X_k\}_{k \geq 1}$ constitutes a trajectory of the Markov chain induced by $\boldsymbol{\mu}$, the resultant stochastic gradients $\{\nabla \boldsymbol{f}(\boldsymbol{w}; X_k)\}$ asso-

ciated with $\{X_k\}_{k\geq 1}$ are biased estimates of the expected gradient in the limit, i.e., $\nabla\bar{\boldsymbol{f}}(\boldsymbol{w}) := \lim_{k\to\infty}\mathbb{E}_{X_k}\left[\nabla\bar{\boldsymbol{f}}(\boldsymbol{w};X_k)\right]$, where expectation is taken over samples obtained from the stationary distribution. This is known as the gradient bias in reinforcement learning algorithms that deal with Markovian data, which has been the major hurdle challenging their non-asymptotic performance analyses.

Let us introduce the average operator $\boldsymbol{\Lambda} = N^{-1}\mathbf{1}_{N\times N}\otimes I_d \in \mathbb{R}^{Nd}$ that first computes the average of all local quantities and replicates it N times; and also the difference operator $\boldsymbol{\Delta} := (I_N - N^{-1}\mathbf{1}_{N\times N})\otimes I_d$, which subtracts the averaged quantity (e.g., parameter vector) over all agents from each local one. Then, we can define the averaged parameter vector $\bar{\boldsymbol{w}} := \boldsymbol{\Lambda}\boldsymbol{w}$, and the difference vector $\boldsymbol{\Delta}\boldsymbol{w} := \boldsymbol{w} - \bar{\boldsymbol{w}}$. Since the mixing matrix $\boldsymbol{B}$ is doubly stochastic, it can be verified that $\boldsymbol{\Lambda}(\boldsymbol{B}\otimes I_d) = I_{Nd} = (\boldsymbol{B}\otimes I_d)\boldsymbol{\Lambda}$. Then, pre-multiplying both sides of (4.13) by $\boldsymbol{\Lambda}$ yields

$$\bar{\boldsymbol{w}}_{k+1} = \boldsymbol{\Lambda}(\boldsymbol{B}\otimes I_d)\,\boldsymbol{w}_k + \alpha_k\boldsymbol{\Lambda}\nabla\boldsymbol{f}(\boldsymbol{w}_k;X_k) = \bar{\boldsymbol{w}}_k + \alpha_k\boldsymbol{\Lambda}\nabla\boldsymbol{f}(\boldsymbol{w}_k;X_k)\,. \tag{4.14}$$

Subtracting (4.14) from (4.13), the difference vector $\boldsymbol{\Delta}\boldsymbol{w}_{k+1} = \boldsymbol{w}_{k+1} - \bar{\boldsymbol{w}}_{k+1}$ can be found as follows

$$\boldsymbol{\Delta}\boldsymbol{w}_{k+1} = (\boldsymbol{B}\otimes I_d)\,\boldsymbol{\Delta}\boldsymbol{w}_k + \alpha_k\boldsymbol{\Delta}\nabla\boldsymbol{f}(\boldsymbol{w}_k;X_k)\,. \tag{4.15}$$

To gain control over the gradient bias, we consider the T-step iteration of (4.14) as

$$\bar{\boldsymbol{w}}_{k+T} = \bar{\boldsymbol{w}}_k + \boldsymbol{\Lambda}\sum_{t=k}^{T+k-1}\alpha_t\nabla\bar{\boldsymbol{f}}(\boldsymbol{w}_k) + \boldsymbol{\zeta}_T(\boldsymbol{w}_k;X_{k:k+T-1}) \tag{4.16}$$

where the residual $\boldsymbol{\zeta}_T(\boldsymbol{w}_k;X_{k:k+T-1})$ is the T-step accumulated gradient bias.

Before characterizing the convergence properties of Algorithm 4, we begin by establishing several lemmas.

Lemma 4.1. *Fix any $\boldsymbol{w},\boldsymbol{w}'\in\mathbb{R}^{Nd}$. For each transition X, then $\nabla\boldsymbol{f}(\boldsymbol{w};X)$ satisfies that*

$$\|\nabla\boldsymbol{f}(\boldsymbol{w};X) - \nabla\boldsymbol{f}(\boldsymbol{w}';X)\| \leq L\|\boldsymbol{w}-\boldsymbol{w}'\| \tag{4.17}$$

and

$$\|\nabla\boldsymbol{f}(\boldsymbol{w};X)\| \leq L(\|\boldsymbol{w}-\boldsymbol{w}^*\| + G) \tag{4.18}$$

where $L = 1 + \gamma$, *and* $G = \|\boldsymbol{w}^*\| + \sqrt{N} r_{\max} L^{-1}$.

Proof of Lemma 4.1 is deferred to Appendix B.2 for readability. Lemma 4.1 shows that the stochastic gradient $\nabla \boldsymbol{f}(\boldsymbol{w}; X_k)$ is Lipschitz-continuous in $\boldsymbol{w}$, which will be useful for controlling (i.e., upper-bounding) the accumulated gradient bias $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ in (4.16).

Lemma 4.2. *For any* $\boldsymbol{w}, \boldsymbol{w}' \in \mathbb{R}^{Nd}$, *the following holds for* $\bar{\boldsymbol{f}}(\boldsymbol{w})$ *with constant* $0 < \delta \leq c_0 L^{-1}/2$

$$\left\langle \boldsymbol{w} - \boldsymbol{w}', \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}) - \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}') \right\rangle \leq -\delta L \|\boldsymbol{w} - \boldsymbol{w}'\|^2. \tag{4.19}$$

Proof of Lemma 4.2 is provided in Appendix B.3. This condition can be viewed as a strongly monotone property of the nonlinear mapping $\nabla \bar{\boldsymbol{f}}(\boldsymbol{w})$. Using Lemma 4.2, we develop upper bounds on $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ conditioning on $\boldsymbol{w}_k$.

Lemma 4.3. *Let Assumptions 4.1 and 4.2 hold. For any non-increasing sequence of stepsizes* $\{\alpha_k \geq 0\}_{k \geq 1}$, *the* T*-step accumulated gradient bias* $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ *satisfies*

$$\left\| \mathbb{E}\left[\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1}) \middle| \boldsymbol{w}_k \right] \right\| \leq \alpha_k LT [\lambda_1(T,k) + \alpha_k LT \lambda_2(T)] (\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G) \tag{4.20}$$

where the expectation is taken over $X_{k:k+T-1}$, *and*

$$\|\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})\|^2 \leq 3\alpha_k^2 L^2 T^2 \left[3 + 2\alpha_k^2 L^2 T^2 \lambda_2^2(T)\right] \|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2 + 6\alpha_k^2 L^2 T^2 G^2 \left[1 + \alpha_k^2 L^2 T^2 \lambda_2^2(T)\right] \tag{4.21}$$

where $\lambda_1(T,k) = T^{-1} \sum_{t=k}^{k+T-1} 2c_1 \rho^t$ *with constants* $c_1 > 0$ *and* $0 < \rho < 1$ *depending only on the induced Markov chain, and* $\lambda_2(T) = T^{-2} \sum_{t=1}^{T-1} t(1 + \alpha_1 L)^{T-1-t}$.

Proof of Lemma 4.3 is relegated to Appendix B.4. Lemma 4.3 essentially implies that the accumulated gradient bias $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ does not grow rapidly in $\|\boldsymbol{w}_k - \boldsymbol{w}^*\|$, which is indeed the key in developing our subsequent convergence rate. Noting that $\|\boldsymbol{w}_k - \boldsymbol{w}^*\| \leq \|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\| + \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|$, we will divide our analysis into establishing the convergence rate of each of the two terms on the right-hand-side (RHS); that is, the multi-agent consensus error $\|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\|$, and the parameter average estimation error $\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|$.

To start, we formally present the convergence results of $\|\boldsymbol{\Delta w}_k\| = \|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\|$ in the cases of decaying and constant stepsizes in Lemma 4.4, whose proof is provided in Appendix B.5.

Lemma 4.4. *Consider the iteration* (4.13) *with non-increasing stepsizes* $\{\alpha_k \geq 0\}_{k\geq 1}$*, and let* $c_2 \in [0,1)$ *denote the second largest singular value of the mixing matrix* $\boldsymbol{B}$*. Under Assumptions 4.1 and 4.2, the following results hold true for any initialization* $\boldsymbol{w}_1$*.*

i) When decaying stepsize $\alpha_k = \bar{\alpha}L^{-1}/k$ *is used with* $\bar{\alpha} < (1-c_2)/2$*, we have that*

$$\|\boldsymbol{\Delta w}_k\| \leq (c_2 + 2\bar{\alpha})^k \|\boldsymbol{\Delta w}_1\| + \frac{c_3 \bar{\alpha}\sqrt{N} r_{\max}}{k} \tag{4.22}$$

where $c_3 > 0$ *is some constant defined in* (B.42)*.*

ii) When constant stepsize $\alpha_k = \bar{\alpha}L^{-1}$ *is used with* $\bar{\alpha} < (1-c_2)/4$*, we have that*

$$\|\boldsymbol{\Delta w}_k\| \leq (c_2 + 2\bar{\alpha})^k \|\boldsymbol{\Delta w}_1\| + \frac{2\bar{\alpha}\sqrt{N} r_{\max}}{L(1-c_2)}. \tag{4.23}$$

To establish the convergence rate of $\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|$, we note from Lemma 4.4 that the T-step gradient bias $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ in (4.20) is bounded. This motivates us to consider a T-step Lyapunov function $\mathbb{C}_{T,k} = \frac{1}{2}\sum_{t=k}^{k+T-1} \|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\|^2$, where T is a parameter chosen such that convergence can be ensured. Before presenting the main results, we derive an upper bound on the drift of the T-step Lyapunov function in Lemma 4.5.

Lemma 4.5. *Let Assumptions 4.1 and 4.2 hold. For any non-increasing stepsizes* $\{\alpha_k \geq 0\}_{k\geq 1}$ *with* $\alpha_1 \leq \alpha_\epsilon$*, there exist* $\epsilon > 0$ *and* $T_\epsilon \geq 1$ *such that: 1)* $\lambda_1(T_\epsilon, k) < T_\epsilon^{-1}\log(1+T_\epsilon)\delta/2 - \epsilon$ *for decaying stepsize* $\alpha_k = \bar{\alpha}L^{-1}/k$ *with* $\bar{\alpha} < (1-c_2)/2$*, or 2)* $\lambda_1(T_\epsilon, k) < \delta/2 - \epsilon$ *for constant stepsize* $\alpha_k = \bar{\alpha}L^{-1}$ *with* $\bar{\alpha} < (1-c_2)/4$*. Then, the drift of the* T_ϵ*-step Lyapunov function is upper bounded by*

$$\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k+1} - \mathbb{C}_{T_\epsilon,k}\big] \leq -\epsilon\alpha_k L T_\epsilon \mathbb{E}\big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\big] + \lambda_3(T_\epsilon, \alpha_k)\mathbb{E}\big[\|\boldsymbol{\Delta w}_k\|^2\big] + 2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon, \alpha_k) \tag{4.24}$$

where

$$\lambda_3(T_\epsilon, \alpha_k) = 18\alpha_k^2 L^2 T_\epsilon^2 + 12\alpha_k^4 L^4 T_\epsilon^4 \lambda_2^2(T_\epsilon) + 4^{-1} T_\epsilon^{-2}\big[1 + \lambda_1^2(T_\epsilon, k) + \alpha_k^2 L^2 T_\epsilon^2 \lambda_2^2(T_\epsilon)\big] \tag{4.25}$$

$$\lambda_4(T_\epsilon, \alpha_k) = \lambda_1(T_\epsilon, k) + 3\alpha_k L T_\epsilon + \alpha_k L T_\epsilon \lambda_2(T_\epsilon) + 3\alpha_k^3 L^3 T_\epsilon^3 \lambda_2^2(T_\epsilon). \tag{4.26}$$

Proof of Lemma 4.5 is presented in Appendix B.6. At this point, two remarks come in order. By means of carefully considering a multistep Lyapunov function, a unifying upper bound on the drift of the Lyapunov function is obtained for both decaying and constant stepsizes. Moreover, the bound in (4.24) couples drift, consensus error, and average parameter estimation error altogether. The following two subsections will analyze the average parameter estimation error under decaying and constant stepsizes separately. The convergence rates of linear-approximate decentralized Q-learning will then be established for decaying and constant stepsizes.

4.3.1 Decaying Stepsize

When the decaying stepsize in Lemma 4.4 is used, we formally establish the convergence rate of $\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|$ in the following theorem, which relies critically on the results in Lemmas 4.1–4.5. Its proof is provided in Appendix B.7.

Theorem 4.6. *Let Assumptions 4.1 and 4.2 hold. Let $\epsilon > 0$ and $T_\epsilon \geq 1$ satisfy $\lambda_1(T_\epsilon, 1) < T_\epsilon^{-1}\log(1+T_\epsilon)\delta/2 - \epsilon$. Choosing stepsize $\alpha_k = \bar{\alpha}L^{-1}/k$ with $\bar{\alpha} < \min\{L\alpha_\epsilon, (1-c_2)/2\}$. Then for all $k \geq 1$, we have that*

$$\frac{1}{N}\mathbb{E}\left[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\right] \leq \frac{2c_5}{Nk} \tag{4.27}$$

where $c_5 := \max\{\mathbb{E}\left[\mathbb{C}_{T_\epsilon,1}\right], c_4\epsilon^{-1}\bar{\alpha}^{-2}T_\epsilon^{-1}\exp(2\bar{\alpha}T_\epsilon)\}$, and c_4 is a constant defined in (B.75).

Note that c_2 reflects the connectivity of the communication graph [133]. When $c_2 = 0$, each agent n can directly exchange local weight $\boldsymbol{w}_n$ with the other agents in the system. When c_2 increases from zero to one, each agent n can directly exchange local weight $\boldsymbol{w}_n$ with less and less agents in the system. When $c_2 = 1$, each agent cannot talk with other agents. Based on Lemma 4.4 and Theorem 4.6, we have the following observations.

- Based on (4.4), we observe that smaller c_2 leads to smaller $c_2 + 2\bar{\alpha}$ given $\bar{\alpha}$, and thus the first term on the RHS of (4.4) converges faster.
- As shown in (B.42), we observe that c_3 is a monotonically increasing function of c_2. Given $\bar{\alpha}$, we observe that a smaller value of c_2 can make the second term on the RHS of (4.4) converge faster.

- From (B.75), we observe that c_4 is a monotonically increasing function of c_2. Hence, the term $\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2$ converges faster with a smaller value of c_2 based on (4.27).

Based on the above three observations, we conclude that the linear-approximate decentralized Q-learning converges faster when each agent n has more agents to exchange directly the local weight (i.e., smaller value of c_2).

Since $T_{\boldsymbol{\epsilon}}$ reflects the mixing time of the N-agent MDP, we observe that the terms $\mathbb{E}\big[\mathbb{C}_{T_{\boldsymbol{\epsilon}},1}\big]$ and $c_4\boldsymbol{\epsilon}^{-1}\bar{\alpha}^{-2}T_{\boldsymbol{\epsilon}}^{-1}\exp(2\bar{\alpha}T_{\boldsymbol{\epsilon}})$ are monotonically increasing with $T_{\boldsymbol{\epsilon}}$. If an initial state distribution is close to the stationary distribution induced by a behavior policy $\boldsymbol{\mu}$, the mixing time of the N-agent MDP decreases. Thus, the convergence rate of the linear-approximate decentralized Q-learning is improved.

Given the fact that all stochastic gradients in (4.10) and the initialization $\boldsymbol{w}_1$ are bounded, it is evident that $\mathbb{E}\big[\mathbb{C}_{T_{\boldsymbol{\epsilon}},1}\big] = \frac{1}{2}\sum_{t=1}^{T}\|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\|^2$ is bounded. Using $\mathbb{E}\big[\|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2\big] \leq 2\mathbb{E}\big[\|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\|^2\big] + 2\mathbb{E}\big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\big]$, and putting the results of Theorem 4.6 and Lemma 4.4 together, we can derive the convergence rate for the linear-approximate decentralized Q-learning in Algorithm 4.

Corollary 4.7. *Under the assumptions and conditions in Theorem 4.6, the following inequality holds for all $k \geq 1$*

$$\frac{1}{N}\mathbb{E}\big[\|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2\big] \leq \frac{2}{N}(c_2 + 2\bar{\alpha})^{2k}\mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] + \frac{2c_5}{Nk} + \frac{2\bar{\alpha}^2 c_3^2 r_{\max}^2}{k^2}. \tag{4.28}$$

Corollary 4.7 characterizes the relationship between convergence rate of $\boldsymbol{w}_k$ and iteration index k. The first and the third terms on the RHS of (4.28) correspond to the convergence of $\boldsymbol{w}_k$ to $\bar{\boldsymbol{w}}_k$ at rate $1/k^2$; while the second term incorporates a gradient bias that vanishes at rate $1/k$. Corollary 4.7 shows that Algorithm 4 with suitable decaying stepsizes enjoys a sub-linear convergence rate of $\mathcal{O}(1/k)$. It is known that tabular Q-learning converges at rate $\mathcal{O}(1/k)$ too [81]. We conclude that the rate of our linear-approximate decentralized Q-learning matches that of tabular Q-learning. Moreover, when compared with the finite-sample results of the centralized linear-approximate Q-learning in [87], our error bounds in (4.28) hold for all $k \geq 1$, whereas those of [87] become available only after a mixing-time of updates. Moreover, our convergence rate $\mathcal{O}(1/k)$ is tighter than the rate $\mathcal{O}(\log(k)/k)$ provided in [87].

4.3.2 Constant Stepsize

Leveraging Lemmas 4.1–4.5, we establish the convergence rate of $\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|$ with constant stepsizes in Theorem 4.8. Its proof is provided in Appendix B.8.

Theorem 4.8. *Let Assumptions 4.1 and 4.2 hold. Choose $\epsilon > 0$ and $T_\epsilon \geq 1$ such that $\lambda_1(T_\epsilon, 1) < \delta/2 - \epsilon$. Fix any constant stepsize $\alpha = \bar{\alpha} L^{-1}$ with $\bar{\alpha} < \min\{L\alpha_\epsilon, (1-c_2)/4\}$. Then, it holds for all $k \geq 1$ that*

$$\frac{1}{N}\mathbb{E}\big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\big] \leq \frac{2\mathbb{E}\big[\mathbb{C}_{T_\epsilon,1}\big]}{N} c_6^{k-1} + \lambda_5(k-1) + \frac{2\bar{\alpha}}{\epsilon T_\epsilon} \sum_{\tau=0}^{T_\epsilon - 1} (1+\bar{\alpha})^{2\tau} c_7 \tag{4.29}$$

where $c_6 = 1 - \bar{\alpha}\epsilon T_\epsilon / \sum_{\tau=0}^{T_\epsilon-1}(1+\bar{\alpha})^{2\tau}$, $c_7 = \frac{8r_{\max}^2}{L^2(1-c_2)}\lambda_3(T_\epsilon, \bar{\alpha}L^{-1}) + \frac{T_\epsilon^2 G^2}{N}[6 + 2\lambda_2(T_\epsilon) + \epsilon\bar{\alpha}T_\epsilon + 6\bar{\alpha}T_\epsilon\lambda_2^2(T_\epsilon)]$ *and*

$$\lambda_5(k) = \frac{4\lambda_3\big(T_\epsilon, \bar{\alpha}L^{-1}\big)[c_6^k - (c_2+2\bar{\alpha})^{2k}]\mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big]}{N\big[c_6 - (c_2+2\bar{\alpha})^2\big]} + \frac{8\bar{\alpha}c_1 G^2(c_6^k - \rho^k)}{N(c_6-\rho)(1-\rho)}. \tag{4.30}$$

Again combining Theorem 4.8 and Lemma 4.4, the convergence rate of Algorithm 4 with constant stepsizes can be summarized in the following result.

Corollary 4.9. *Under the same assumptions and conditions in Theorem 4.8, it holds for all $k \geq 1$ that*

$$\frac{1}{N}\mathbb{E}\big[\|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2\big] \leq \frac{2\mathbb{E}\big[\mathbb{C}_{T_\epsilon,1}\big]}{N} c_6^{k-1} + \lambda_5(k-1) + \frac{2\mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big]}{N}(c_2+2\bar{\alpha})^{2(k-1)} + 2\bar{\alpha}c_8 \tag{4.31}$$

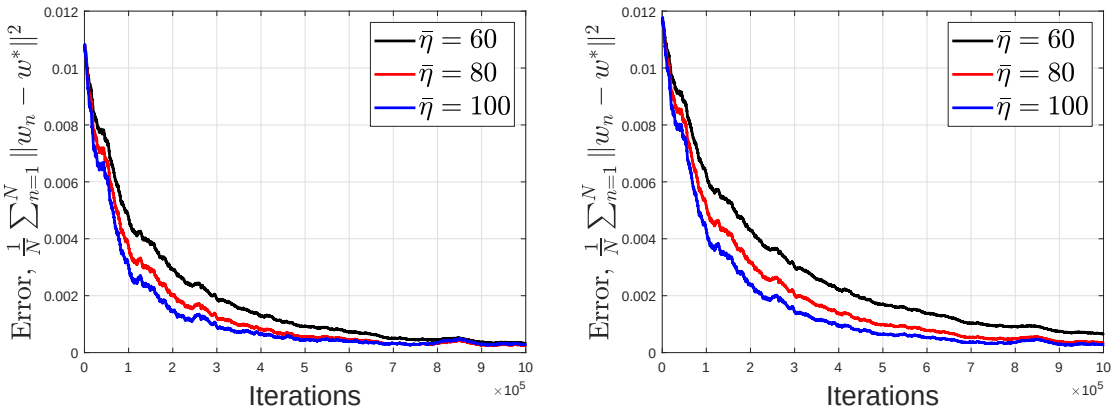
where $c_8 = \frac{c_7}{\epsilon T_\epsilon}\sum_{\tau=0}^{T_\epsilon-1}(1+\bar{\alpha})^{2\tau} + \frac{4\bar{\alpha}r_{\max}^2}{L^2(1-c_2)}$.

Corollary 4.9 asserts that Algorithm 4 converges exponentially fast to a neighborhood of the optimal vector $\boldsymbol{w}^*$, whose size can be made arbitrarily small by taking small enough α. Specifically, the first two terms in (4.31) are associated with the linear convergence of $\mathbb{E}\big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\big]$, the third term captures the convergence of $\mathbb{E}\big[\|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\|^2\big]$, while the last term defines the neighborhood. We further observe that a smaller stepsize can increase the convergence rate of $\mathbb{E}\big[\|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\|^2\big]$ at the expense of sacrificing that of $\mathbb{E}\big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\big]$. Different from the two-phase convergence behaviors [84, 86, 87, 92], our results feature a single-phase convergence to an $\bar{\alpha}$-neighborhood of the optimal vector $\boldsymbol{w}^*$.

4.4 Numerical Results

In this section, we present simulation results to demonstrate the performance of linear-approximate decentralized Q-learning. We consider a six-agent MDP, where the six agents can talk through a predefined topology. The six-agent MDP has 20 states, and each agent can choose three actions. For each state-action pair, the reward is generated by following the standard uniform distribution, and the feature vector is generated by cosine functions. Dimension of features d, value of γ and stepsize α_k are respectively set as 30, 0.7, and $\bar{\alpha} = \bar{\eta}/L(k+3\times 10^5)$ where $L = 1+\gamma$. For each action, the state transition matrix is set as $\frac{1}{20}\mathbf{1}_{20\times 20}$. The mixing matrix $\boldsymbol{B}$ of the predefined topology is set as

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}. \tag{4.32}$$



(a) $\gamma = 0.5$ (b) $\gamma = 0.9$

Figure 4.2: Convergence behavior of the proposed linear-approximate decentralized Q-learning.

Figure 4.2 shows the convergence of the linear-approximate decentralized Q-learning under different values of γ. Given discounting factor γ, we observe that a larger $\bar{\eta}$ leads to a faster convergence. For example, Fig. 4.2(a) shows that it takes 6×10^5 iterations to convergence for the

case $\bar{\eta} = 100$, while extra 3×10^5 iterations are required by the case $\bar{\eta} = 60$ to converge. Given $\bar{\eta}$, we observe that a larger discounting factor γ requires more iterations to converge. When comparing Fig. 4.2(a) and Fig. 4.2(b), we observe that the case with $\bar{\eta} = 100$ and $\gamma = 0.5$ takes around 6×10^5 iterations to converge. When γ increases to 0.9 given $\bar{\eta} = 100$, it requires 8×10^5 iterations to converge. This is because a larger discounting factor γ requires the Q-function to store more historical information, which slows down the linear-approximate decentralized Q-learning.

4.5 Summary

In this chapter, we have addressed policy optimization in a collaborative MARL setting, where agents cooperate to learn an optimal Q-function in a fully decentralized manner allowing only simple neighboring communications. We have proposed a decentralized Q-learning algorithm. Considering a single state-action trajectory of a multi-agent MDP, we have established the convergence rates of the proposed algorithm for decaying and constant stepsizes. Our rate results match the rates of tabular Q-learning, but improve upon those of existing linear-approximate centralized Q-learning.

Chapter 5

Grid-Energy Expenditure Minimization in SGPCSs

Applying renewable energy to wireless communication systems is an effective way to reduce the energy bills of wireless operators. When a BST is solely powered by renewable energy, the unpredictable and intermittent arrival of renewable energy can result in power outages of BST. Therefore, the BST needs to obtain electricity from the power grid and renewable resources. While the harvest-store-use and harvest-use-store strategies suffer from the imperfections of storage media, the harvest-use-trade strategy can avoid such imperfections (See Chapter 1). Besides, the harvest-use-trade strategy can also generate revenue for wireless operators by selling surplus harvested energy to the power grid by using the two-way energy-trading capability in SGPCS. Therefore, this chapter aims at solving the TAEGEE minimization problem in a single-cell SGPCS by considering the random system states (e.g., CSI, ESI, and packet arrival) and long-term communication QoS constraints. The decisions are dependent across different slots when optimizing a time-average expectation under long-term constraints. The analysis method of Lyapunov learning is adopted to reformulate the TAEGEE minimization problem as per-slot subproblems. Solving each subproblem, the BST can directly obtain beamforming vectors for each slot. Since each subproblem is non-convex, two algorithms (i.e., SABF and ZFBF algorithms) are proposed to obtain the suboptimal solution to each subproblem. The convergence behaviors of SABF and ZFBF algorithms are established, and the corresponding computational complexities are analyzed. Besides, the properties of the obtained beamforming vectors are investigated.

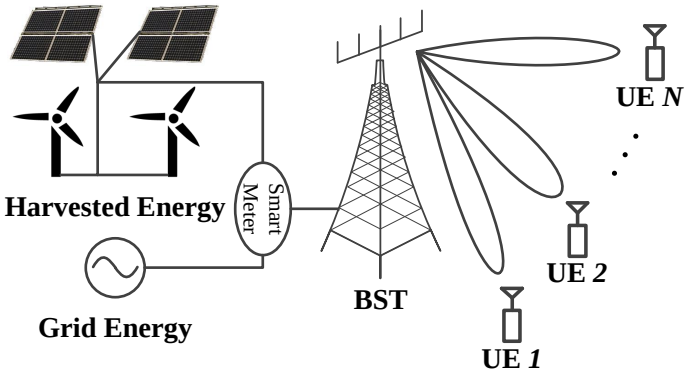


Figure 5.1: An illustration of SGPCS with N UEs.

5.1 System Model and Problem Statement

Consider the downlink transmissions of an SGPCS with a BST and N UEs. The BST is equipped with d antennas, and each UE is equipped with a single antenna. Fig. 5.1 shows that the BST is powered by the smart grid and energy harvesters, such as photovoltaic panels and wind turbines. The BST can obtain instantaneous CSI by exploiting channel reciprocity and handshaking signals. The UEs receive information from the BST via beamforming in the downlink period. The SGPCS operates in discrete-time mode with index k, $k = 1, 2, \ldots$, denoting a unit duration; therefore, the term "power" and "energy" can be used interchangeably. We assume that the channel fading and renewable energy arrivals follow the block-based model, where the channel coefficient vectors and renewable energy arrival rate remain constant during each slot and vary over different time scales. Specifically, the CSI changes over different slots, and the ESI varies over several slots. This assumption is reasonable because the coherence time of the renewable energy arrival is generally longer than the channel coherence time [134]. In this chapter, we do not consider the storage of renewable energy; therefore, the BST can sell the surplus renewable energy to (or purchase energy from) the power grid via a smart meter. Moreover, such an operation can reduce the GEE of SGPCS.

5.1.1 Signal Model

Allocating each UE n with a beamforming vector $\boldsymbol{w}_n$, the received signal at the UE n at slot k is obtained as

$$y_{n,k} = \boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{n,k} + \sum_{l \neq n}^{N} \boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{l,k} + z_{n,k} \tag{5.1}$$

where $z_{n,k} \sim \mathcal{CN}(0, \sigma_n^2)$ is the additive white Gaussian noise (AWGN) at UE n; $w_{n,k}$ is the single-stream beamforming vector for the UE n in slot k; and $\boldsymbol{h}_{n,k} \in \mathbb{C}^d$ is the channel coefficient vector of UE n in slot k, and each entry of $\boldsymbol{h}_{n,k}$ is a circularly-symmetric complex Gaussian (CSCG) random variable with mean zero and variance ω_n^{-1}, where ω_n is the pathloss of UE n. Here, we note that the vector $\boldsymbol{h}_{n,k}$ captures the composite effects of multipath fading and pathloss. We consider that the SGPCS operates in time-division-duplex mode. At the beginning of each slot, each UE sends a pilot signal to the BST. After receiving the pilot signals, the BST estimates the channel associated with each UE that sends the pilot signal. Since each UE facilitates each uplink transmission with a pilot signal, the BST can exploit the uplink reciprocity to update periodically the downlink CSI for BST-UE links.

The received signal-to-interference-plus-noise ratio (SINR) of UE n in slot k is given as

$$\mathrm{SINR}_{n,k} = \frac{|\boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{n,k}|^2}{\sum_{l \neq n}^{N} |\boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{l,k}|^2 + \sigma_n^2}. \tag{5.2}$$

A certain amount of power is consumed to enable the information transmission [135]. In particular, the power consumption of SGPCS is denoted by

$$P_k^{\mathrm{BST}} = \frac{1}{\eta} \sum_{n=1}^{N} \left\| \boldsymbol{w}_{n,k} \right\|^2 + P^{\mathrm{CIR}} \tag{5.3}$$

where η is the efficiency of the power amplifier; the term P^{CIR} is the circuit power which is calculated as [135]

$$P^{\mathrm{CIR}} = 0.87 P^{\mathrm{SP}} + 0.1 d P^{\mathrm{SP}} + 0.03 d^2 P^{\mathrm{SP}} \tag{5.4}$$

where the linear term of d in (5.4) represents power overhead of MISO pilots, and the quadratic term of d in (5.4) stands for power overhead of MISO signal processing. Note that since our objective is to minimize the energy expenditure for access links, we ignore the power consumed

for backhaul transmission in this work.

5.1.2 Grid Energy Expenditure Model

Since the renewable energy arrival varies over several slots, we denote the renewable energy arrival rate in slot k by E_k^{HAV}. Setting the price of purchasing (selling) a unit energy from (to) the power grid as $\gamma_b > 0$ ($\gamma_s > 0$), the GEE in slot k is calculated as

$$F(P_k^{\mathrm{BST}}) = \gamma_b \left[P_k^{\mathrm{BST}} - E_k^{\mathrm{HAV}}\right]^+ - \gamma_s \left[E_k^{\mathrm{HAV}} - P_k^{\mathrm{BST}}\right]^+ \tag{5.5}$$

where $[x]^+ = \max[x, 0]$. To avoid the operator of SGPCS making non-justifiable profit, the operator of smart grid should set $\gamma_b \geq \gamma_s > 0$.

5.1.3 Packet Rate and Traffic Queues

For the wireless data traffic, multiple modulation and coding schemes (MCSs) can be used to achieve tradeoff between the data rate and the transmission reliability [136]. High order modulation schemes, which allow more information bits to be transmitted per symbol, shorten the Euclidean distance of the signal constellation points. Therefore, more errors occur in the decoding process. Various coding schemes, accompanied with the modulation schemes, are used to adapt to the channel variations. Decreasing the coding rate leads to a decrease in the effective packet rate. Suppose that a packet has C information bits, and an error occurs when one of the C information bits is erroneously decoded. Hence, packet rate of UE n in slot k is obtained as

$$c_{0,n} \prod_{\tau=1}^{C} \left[1 - P_\tau(\mathrm{SINR}_{n,k})\right] \tag{5.6}$$

where $c_{0,n}$ is packet transmission rate of UE n in slot k, and $P_\tau(\mathrm{SINR}_{n,k})$ is the error probability of τth bits for UE n in slot k. Using the approximation method developed in [137], we obtain the correct packet reception rate as

$$\frac{c_{0,n}}{1 + \exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k}) - c_{2,n}])} \tag{5.7}$$

where $c_{1,n}$ and $c_{2,n}$ are MCS specific parameters.

Dividing the packet departure process and packet arrival process at UE n by $c_{0,n}$, we obtain the normalized packet rate as [20]

$$r_{n,k} = \frac{1}{1 + \exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k}) - c_{2,n}])}. \tag{5.8}$$

Given the fixed MCS and noise-free channel model, the BST-UE rate is determined by $c_{0,n}$ as shown in (5.6) and (5.7). When noise and interference are considered, decoding errors may occur for each bit of a packet. In our formulation, we consider that a packet can be correctly decoded when each bit of the packet is correctly decoded. Therefore, the packet rate, its approximate version, and its normalized approximate version are respectively shown in (5.6), (5.7), and (5.8). In other words, the normalized packet rate is a sigmoid function of SINR as shown in (5.8) when the joint impacts of data rate and decoding error are considered. Besides, the packet rate of UEs are controlled by tuning adaptively the beamforming vectors of UEs as shown in (5.8).

The BST maintains N access queues to buffer the random arrival packets for the corresponding to the N UEs. Denote the backlog of each access queue n in slot k by $q_{n,k}$. Thus, the backlog of access queue n evolves as

$$q^{\mathrm{A}}_{n,k+1} = \left[q^{\mathrm{A}}_{n,k} - r_{n,k}\right]^{+} + \nu_{n,k} \tag{5.9}$$

where $\nu_{n,k} \in (0, 1)$ is normalized packet arrival rate, which is assumed to be an independent and identically distributed over different slots with mean value $\bar{\nu}_n = \mathbb{E}\left[\nu_{n,k}\right]$.

5.1.4 Problem Statement

After obtaining the instantaneous CSI and ESI, the BST performs dynamic beamforming to minimize the time-average expectation of GEE in the SGPCS with harvest-use-trade protocol. Based on the aforementioned description, the time-average GEE minimization problem is mathematically formulated as follows.

- Access queue constraints:

$$\text{Traffic queue } q^{\mathrm{A}}_{n,k},\ \mathrm{n} = 1, \ldots, \mathrm{N}, \text{ is mean rate stable.} \tag{5.10}$$

- UEs' SINR constraints:

$$\mathrm{SINR}_{n,k} \geq \gamma_n^{\mathrm{REQ}}, n = 1, \ldots, N \tag{5.11}$$

  where γ_n^{REQ} is required SINR of UE n. This set of constraints is used to guarantee that the packet error probability of each WN is at an acceptable level in slot k.

- Transmission power constraint: Due to the circuit limitation, the transmission power is limited by

$$\sum_{n=1}^{N} \left\| \boldsymbol{w}_{n,k} \right\|^2 \leq P^{\max} \tag{5.12}$$

  where $P^{\max}$ is the maximum transmit power of BST.

Our objective is to minimize the time-average expectation of GEE under the constraints on access queue (5.10), instantaneous SINR requirement (5.11), and transmit power (5.12). To incorporate the uncertainty of ESI, the objective function is defined as the time-average expectation of GEE. Therefore, we can obtain the corresponding problem via a stochastic optimization model as

$$\min_{\{w_{n,k}\}_{n,k}} \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[F(P_k^{\mathrm{BST}})\right] \tag{5.13a}$$

$$\text{s.t. } (5.10) - (5.12). \tag{5.13b}$$

The optimization problem in (5.13) is a cross-layer optimization because it considers the effects of CSI, ESI, and packet failure while designing the beamforming vectors.

5.2 Dynamic Cross-Layer Beamforming via Lyapunov Learning

Since the problem (5.13) includes the time-average objective function and time-average constraints in (5.10), it is challenging to solve the problem (5.13) via standard convex optimization. Therefore, we leverage the Lyapunov learning technique to obtain a set of slot-by-slot subproblems, whose solutions satisfy the time-average constraints in (5.10). Moreover, we also reveal a tradeoff relation between the GEE and access delay of UEs.

We first construct a Lyapunov drift-plus-penalty function with respect to the time-average

objective (5.13a) and constraints in (5.10) as

$$D(\boldsymbol{q}_k^{\mathrm{A}}) + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})|\boldsymbol{q}_k^{\mathrm{A}}\right] \tag{5.14}$$

where $\iota_{1,k} := \{\boldsymbol{h}_{n,k}, E_k^{\mathrm{HAV}}, \nu_{n,k}\}_{n=1}^N$ is the set of random sources; $\boldsymbol{q}_k^{\mathrm{A}} := [q_{1,k}^{\mathrm{A}}, \ldots, q_{N,k}^{\mathrm{A}}]$; and $V > 0$ is an introduced control parameter. The one-slot conditional drift function $D(\boldsymbol{q}_k^{\mathrm{A}})$ is defined as

$$D(\boldsymbol{q}_k^{\mathrm{A}}) := \frac{1}{2}\mathbb{E}_{\iota_{1,k}}\left[\left\|\boldsymbol{q}_{k+1}^{\mathrm{A}}\right\|^2 - \left\|\boldsymbol{q}_k^{\mathrm{A}}\right\|^2 |\boldsymbol{q}_k^{\mathrm{A}}\right]. \tag{5.15}$$

Lemma 5.1. *When the random sources in $\iota_{1,k}$ are independent and identically distributed over slots, the upper bound of Lyapunov drift-plus-penalty function in* (5.14) *is derived as [30]*

$$D(\boldsymbol{q}_k^{\mathrm{A}}) + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})|\boldsymbol{q}_k^{\mathrm{A}}\right] \le N + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})|\boldsymbol{q}_k^{\mathrm{A}}\right] + \sum_{n=1}^{N} q_{n,k}^{\mathrm{A}}\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k} - r_{n,k}|\boldsymbol{q}_k^{\mathrm{A}}\right]. \tag{5.16}$$

It can be shown that the minimizer to the right-hand side of (5.16) given the random set $\iota_{1,k}$ and backlog of access queues $\boldsymbol{q}_k^{\mathrm{A}}$ is also a feasible solution to the problem (5.13). Moreover, a tradeoff relation between the GEE and the average backlog of the traffic queues will be revealed in Theorem 5.2. The introduced control parameter V is used to control the tradeoff between the GEE and the average backlog of the traffic queues. Thus, the RHS of (5.16) is useful in developing the dynamic cross-layer beamforming algorithm.

We observe that the constant N and the term $\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k}|\boldsymbol{q}_k^{\mathrm{A}}\right]$ have no effect on the beamforming vectors. Minimizing the conditional expectation $\mathbb{E}_{\iota_{1,k}}\left[VF(P_k^{\mathrm{BST}}) - \sum_{n=1}^N q_{n,k}^{\mathrm{A}}\boldsymbol{r}_{n,k}|\boldsymbol{q}_k^{\mathrm{A}}\right]$ can be manipulated as minimizing $VF(P_k^{\mathrm{BST}}) - \sum_{n=1}^N q_{n,k}^{\mathrm{A}} r_{n,k}$ after obtaining the random set $\iota_{1,k}$ and backlog of access queues $\boldsymbol{q}_k^{\mathrm{A}}$ according to the principle of opportunistically minimizing an expectation [30, Section 1.8]. Hence, we can simplify the original optimization problem (5.13) according to (5.16) as

$$\min_{\{\boldsymbol{w}_{n,k}\}_{n,k}} VF(P_k^{\mathrm{BST}}) - \sum_{n=1}^{N} \boldsymbol{q}_{n,k}^{\mathrm{A}} r_{n,k} \tag{5.17a}$$

$$\text{s.t. } \mathrm{SINR}_{n,k} \ge \gamma_n^{\mathrm{REQ}}, n = 1, \ldots, N \tag{5.17b}$$

$$\sum_{n=1}^{N} \left\|\boldsymbol{w}_{n,k}\right\|^2 \le P^{\max}. \tag{5.17c}$$

Note that the objective function of problem (5.17) is non-convex, which is still challenging to be handled by standard convex optimization methods. Hence, we are motivated to obtain the suboptimal solutions to the problem (5.17). Besides, different beamforming techniques induce different feasible regions of problem (5.17). We will introduce the feasibility check methods for the developed beamforming techniques.

Algorithm 5 Dynamic Cross-Layer Beamforming Framework

1: In slot k, the BST observes backlogs of access queues $\boldsymbol{q}_k^{\mathrm{A}}$, channel coefficient vectors $\{\boldsymbol{h}_{n,k}\}_{n=1}^{N}$ and the amount of harvested energy E_k^{HAV}
2: **if** the optimization problem (5.17) is feasible **then**
3: The BST obtains GEE of slot k via solving suboptimally the optimization problem (5.17)
4: **else**
5: The BST obtains GEE of slot k via choosing randomly the beamforming vectors $\{\boldsymbol{w}_{n,k}\}_{n=1}^{N}$ under the transmit power constraint (5.17c)
6: **end if**
7: The BS updates backlogs of access queues $\boldsymbol{q}_k^{\mathrm{A}}$ according to (5.9)

When the optimization problem (5.17) is feasible, we can apply the respective algorithm to solve (5.17) as shown in line 3 of Algorithm 5. When the optimization problem (5.17) is infeasible, the BST will equally allocate the transmit power to each UE and randomly generate a beamforming vector as shown in line 5 of Algorithm 5. Following the procedures in Algorithm 5, the beamforming vectors $\{\boldsymbol{w}_{n,k}\}_n$ are updated per slot. Algorithm 5 achieves a tradeoff between the GEE and access delay of UEs, and the quantified tradeoff relation is revealed in Theorem 5.2.

Theorem 5.2. *Suppose the random sources in $\{\iota_{1,k}\}_k$ are independent and identically distributed over slots, and the initial expected backlog of access queues $\mathbb{E}\big[\|\boldsymbol{q}_1^{\mathrm{A}}\|^2\big] < \infty$. When the expected traffic arrival rate $\bar{\nu}_n$ is in the stable region of system for $n = 1, \ldots, N$, the proposed Algorithm 5 achieves the following properties.*

1. *Each access queue $q_{n,k}^{\mathrm{A}}$ is mean rate stable [30] for $n = 1, \ldots, N$.*

2. *The time-average expected GEE is upper bounded by*

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\big[F(P_k^{\mathrm{BST}})\big] \leq \frac{N}{V} + F^{\mathrm{SOPT}} \tag{5.18}$$

where F^{SOPT} is the maximum suboptimal value of GEE in (5.17).

3. The average backlog of access queues is upper-bounded by

$$\lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}\left[q_{n,k}^{\mathrm{A}}\right]\leq\frac{N+V\left(F^{\mathrm{SOPT}}-F^{\min}\right)}{\epsilon} \tag{5.19}$$

where $F^{\min}$ is the minimum GEE.

Following the Little's law, the backlog of access queues is proportional to the access delay of UEs (i.e., $\bar{\nu}_n \times$delay = average queue length). Therefore, we obtain two conclusions from Theorem 5.2: 1) the output of Algorithm 5 is a feasible solution to the problem (5.13), and 2) the tradeoff between the GEE and access delay of UEs is quantitatively revealed. More specifically, the time-average expected GEE of SGPCS is reduced by increasing the control parameter C at the expense of increasing access delay of UEs, and vice versa. We also observe that the proposed Algorithm 5 does not have a limitation on the number of antennas for UEs; therefore, the proposed Algorithm 5 can be used to the multiple-input and multiple-output channels.

5.3 Design of Beamforming in Each Slot

We observe that the BST needs to consume a certain amount of power to guarantee the instantaneous SINR constraints in (5.17c). Since the transmit power of BST is upper-bounded by $P^{\max}$, the problem (5.17) can be infeasible under certain CSI and network scenarios when $\{\gamma_n^{\mathrm{REQ}}\}_{n=1}^{N}$ is too high. Hence, the feasibility of (5.17) should be checked via solving the following optimization problem

$$\begin{aligned}&\text{find } \{\boldsymbol{w}_{n,k}\}_{n=1}^{N}\\ &\text{s.t. } \mathrm{SINR}_{n,k}\geq\gamma_n^{\mathrm{REQ}}, n=1,\ldots,N\\ &\quad\sum_{n=1}^{N}\left\|\boldsymbol{w}_{n,k}\right\|^2\leq P^{\max}.\end{aligned} \tag{5.20}$$

The optimization problem (5.20) can be solved by standard optimization techniques, such as a second-order cone programming [138] and uplink-downlink duality based algorithm [139].

To deal with the non-convex terms $\{\mathrm{SINR}_{n,k}\}_{n=1}^{N}$ in the objective function (5.17), we introduce a set of auxiliary variables $\{\lambda_{1,n,k}\}_{n=1}^{N}$. Performing several mathematical manipulations, the

optimization problem (5.17) is recast as

$$\min_{\{\boldsymbol{w}_{n,k},\lambda_{1,n,k}\}_{n=1}^{N}} \; VF(P_k^{\mathrm{BST}}) - \sum_{n=1}^{N} \frac{q_{n,k}^{\mathrm{A}}}{1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])} \tag{5.21a}$$

$$\text{s.t. } \mathrm{SINR}_{n,k} \geq \lambda_{1,n,k}, n=1,\ldots,N \tag{5.21b}$$

$$\mathrm{SINR}_{n,k} \geq \gamma_n^{\mathrm{REQ}}, n=1,\ldots,N \tag{5.21c}$$

$$\sum_{n=1}^{N} \left\|\boldsymbol{w}_{n,k}\right\|^2 \leq P^{\max}. \tag{5.21d}$$

Note that the constraints in (5.21c) are active at each suboptimal point; otherwise, the objective function (5.21a) can take a strictly smaller value by increasing $\{\lambda_{1,n,k}\}_{n=1}^{N}$. Based on this observation, we can also conclude that each local optimal value of $\lambda_{1,n,k}$ satisfies the following inequality

$$\lambda_{1,n,k} \geq \gamma_n^{\mathrm{REQ}}, n=1,\ldots,N. \tag{5.22}$$

The major challenges in solving (5.21d) are two folds: 1) the sum-of-ratios component in the objective function (5.21a), and 2) the non-convex constraints (5.21b) and (5.21c). We leverage the Lagrangian duality theorem to deal with the sum-of-ratios component in (5.21a) and obtain an equivalent form of problem (5.21), which is established in Theorem 5.3.

Theorem 5.3. *If* $\{\boldsymbol{w}_{n,k},\lambda_{1,n,k}\}_{n=1}^{N}$ *satisfy the Karush-Kuhn-Tucker (KKT) conditions of the optimization problem* (5.21)*, we can conclude that there exist parameters*

$$c_{3,n,k} = \frac{1}{1+\exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k})-c_{2,n}])} \tag{5.23a}$$

$$c_{4,n,k} = \frac{q_{n,k}^{\mathrm{A}}}{1+\exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k})-c_{2,n}])} \tag{5.23b}$$

such that an optimization problem has the same KKT conditions as in (5.21)*, which is shown as*

$$\min_{\{\boldsymbol{w}_{n,k},\lambda_{1,n,k}\}_{n=1}^{N}} \; VF(P_k^{\mathrm{BST}}) + \sum_{n=1}^{N} c_{3,n,k}c_{4,n,k}\exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k})-c_{2,n}]) \tag{5.24a}$$

$$\text{s.t. } \mathrm{SINR}_{n,k} \geq \lambda_{1,n,k}, n=1,\ldots,N \tag{5.24b}$$

$$\mathrm{SINR}_{n,k} \geq \gamma_n^{\mathrm{REQ}}, n=1,\ldots,N \tag{5.24c}$$

$$\sum_{n=1}^{N} \left\| \boldsymbol{w}_{n,k} \right\|^2 \leq P^{\max}. \tag{5.24d}$$

We note that the GEE $F(P_k^{\mathrm{BST}})$ in (5.24a), can be reformulated as

$$F(P_k^{\mathrm{BST}}) = (\gamma_b - \gamma_s)\left[P_k^{\mathrm{BST}} - E_k^{\mathrm{HAV}}\right]^+ + \gamma_s[P_k^{\mathrm{BST}} - E_k^{\mathrm{HAV}}] \tag{5.25}$$

which is a convex function of $\{\boldsymbol{w}_{n,k}\}_{n=1}^N$. The second term of (5.24a) is a convex function of $\{\lambda_{1,n,k}\}_{n=1}^N$. Hence, the objective function (5.24a) is a convex function of $\{\boldsymbol{w}_{n,k}, \lambda_{1,n,k}\}_{n=1}^N$. Introducing two constants $c_{3,n,k}$ and $c_{4,n,k}$, Theorem 5.3 shows the equivalence between the problems (5.21) and (5.24). Moreover, the objective function in (5.24) is strictly convex such that the complexity of obtaining the solution to (5.24) is lower than that of (5.21). In the remaining part of this chapter, the design of algorithm is based on the optimization problem (5.24).

5.3.1 Successive Convex Approximation Based Beamforming

To handle the non-convex constraints in (5.24b), we use the successive approximation technique. Introducing another set of auxiliary variables $\{\lambda_{2,n,k}\}_{n=1}^N$, we can reformulate the optimization problem (5.24) as

$$\min_{\{\mathcal{Y}_{1,n,k}\}_{n=1}^N} \; VF(P_k^{\mathrm{BST}}) + \sum_{n=1}^{N} c_{3,n,k} c_{4,n,k} \exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k}) - c_{2,n}]) \tag{5.26a}$$

$$\text{s.t.} \; \sum_{n=1}^{N} \left\| \boldsymbol{w}_{n,k} \right\|^2 \leq P^{\max} \tag{5.26b}$$

$$\boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{n,k} \geq \lambda_{2,n,k} \sqrt{\gamma_n^{\mathrm{REQ}}}, n = 1, \ldots, N \tag{5.26c}$$

$$\boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{n,k} \geq \lambda_{2,n,k} \sqrt{\lambda_{1,n,k}}, n = 1, \ldots, N \tag{5.26d}$$

$$\sqrt{\sigma_n^2 + \sum_{l \neq n}^{N} \left| \boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{l,k} \right|^2} \leq \lambda_{2,n,k}, n = 1, \ldots, N \tag{5.26e}$$

$$\mathrm{Im}\left(\boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{n,k} \right) = 0, n = 1, \ldots, N \tag{5.26f}$$

where $\mathcal{Y}_{1,n,k} := \{\boldsymbol{w}_{n,k}, \lambda_{1,n,k}, \lambda_{2,n,k}\}$ is the set of optimization variables, and the operator $\mathrm{Im}(x)$ takes the imaginary part of a complex value x.

We justify the equivalence between the problems (5.24) and (5.26) based on the following

two arguments: 1) forcing the phase of $\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}$ to zero does not change the value of the objective function since rotating the phase of $\boldsymbol{w}_{n,k}$ leads to the same value of $|\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}|$, and 2) the constraints in (5.26e) are active. See Appendix C.4 for detailed proof.

The remaining non-convexity of problem (5.26) comes from the constraints in (5.26d). We recast (5.26d) into a quadratic-over-linear term as

$$\frac{|\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}|^2}{\lambda_{1,n,k}} \geq \lambda_{2,n,k}^2 \tag{5.27}$$

which is a joint convex function of $\boldsymbol{w}_{n,k}$ and $\lambda_{1,n,k}$, and has a liner lower bound as

$$\frac{|\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}|^2}{\lambda_{1,n,k}} \geq \frac{2\mathrm{Re}\Big(\boldsymbol{w}_{n,k}^{\tau,\mathrm{H}}\boldsymbol{h}_{n,k}\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}\Big)}{\lambda_{1,n,k}^{\tau}} - \left(\frac{|\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}^{\tau}|^2}{\lambda_{1,n,k}^{\tau}}\right)\lambda_{1,n,k}, n = 1,\ldots,N. \tag{5.28}$$

where τ represents the iteration index.

Using (5.28), we obtain the convex approximation of the non-convex constraints in (5.26d) successively in each iteration τ as

$$\frac{2\mathrm{Re}\Big(\boldsymbol{w}_{n,k}^{\tau,\mathrm{H}}\boldsymbol{h}_{n,k}\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}\Big)}{\lambda_{1,n,k}^{\tau}} - \left(\frac{|\boldsymbol{h}_{n,k}^{\mathrm{H}}\boldsymbol{w}_{n,k}^{\tau}|^2}{\lambda_{1,n,k}^{\tau}}\right)\lambda_{1,n,k} \geq \lambda_{2,n,k}^2, n = 1,\ldots,N. \tag{5.29}$$

Finally, we obtain a convex approximation of problem (5.24) as

$$\begin{aligned}\min_{\{\mathcal{V}_{1,n,k}\}_{n=1}^{N}} \; & VF(P_k^{\mathrm{BST}}) + \sum_{n=1}^{N} c_{3,n,k}c_{4,n,k}\exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k}) - c_{2,n}]) \\ \text{s.t. } & (5.26\text{b}), (5.26\text{c}), (5.26\text{e}), (5.26\text{f}) \text{ and } (5.29).\end{aligned} \tag{5.30}$$

Using (5.23a), (5.23b) and (5.30), we summarize the procedures of the SABF algorithm in Algorithm 6. The proposed SABF algorithm converges to a solution that satisfies the KKT conditions of (5.21), and the proof is relegated to Appendix C.5.

5.3.2 Zero-Forcing Beamforming

The SABF algorithm incurs a high computational complexity since multiple auxiliary variables are introduced. Therefore, we are motivated to investigate the low-complexity algorithm by using

Algorithm 6 SABF Algorithm

1: **Initialize**: iteration index $\tau = 0$, stop threshold ϵ, maximum number of iterations $T_1^{\max}$, and backlogs of access queues $\boldsymbol{q}_k^{\mathrm{A}}$
2: BST obtains a feasible point $\{\boldsymbol{w}_{n,k}^0\}_{n=1}^N$ via (5.20)
3: BST respectively updates $\{\lambda_{1,n,k}^0\}_{n=1}^N$, $\{c_{3,n,k}^0\}_{n=1}^N$, and $\{c_{4,n,k}^0\}_{n=1}^N$ via (5.22), (5.23a), and (5.23b)
4: **repeat**
5: $\quad\tau \leftarrow \tau + 1$
6: $\quad$Via $\{\boldsymbol{w}_{n,k}^{\tau-1}, \lambda_{1,n,k}^{\tau-1}, c_{3,n,k}^{\tau-1}, c_{4,n,k}^{\tau-1}\}_{n=1}^N$, BST solves the problem (5.30), and obtains $\{\boldsymbol{w}_{n,k}^{\tau}, \lambda_{1,n,k}^{\tau}\}_{n=1}^N$
7: $\quad$BST updates the following parameters:

$$c_{3,n,k}^{\tau} = \frac{1}{1+\exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k}^{\tau}) - c_{2,n}])}$$

$$c_{4,n,k}^{\tau} = \frac{q_{n,k}^{\mathrm{A}}}{1+\exp(-c_{1,n}[10\log_{10}(\lambda_{1,n,k}^{\tau}) - c_{2,n}])}$$

8: $\quad$BST stacks $\boldsymbol{c}_{3,k}^{\tau} := [c_{3,1,k}^{\tau}, \ldots, c_{3,N,k}^{\tau}]$ and $\boldsymbol{c}_{4,k}^{\tau} := [c_{4,1,k}^{\tau}, \ldots, c_{4,N,k}^{\tau}]$
9: **until** $\frac{\left\|\boldsymbol{c}_{3,k}^{\tau} - \boldsymbol{c}_{3,k}^{\tau-1}\right\|}{\left\|\boldsymbol{c}_{3,k}^{\tau-1}\right\|} \le \epsilon$ and $\frac{\left\|\boldsymbol{c}_{4,k}^{\tau} - \boldsymbol{c}_{4,k}^{\tau-1}\right\|}{\left\|\boldsymbol{c}_{4,k}^{\tau-1}\right\|} \le \epsilon$ or $\tau > T^{\max}$

the ZFBF technique, where the beamforming vectors are designed to null the interference among UEs. However, to perform ZFBF, the number of transmission antennas should be equal to or larger than the number of UEs ($d \ge N$).

To derive the ZFBF vector, we first decouple the power $p_{n,k}$ from the corresponding beamforming vector $\boldsymbol{w}_{n,k}$, $n = 1, \ldots, N$. Then, we define the channel coefficient matrix and beamforming matrix as $\boldsymbol{H}_k := [\boldsymbol{h}_{1,k}, \ldots, \boldsymbol{h}_{N,k}]$ and $\boldsymbol{W}_k := [\boldsymbol{w}_{1,k}, \ldots, \boldsymbol{w}_{N,k}]$. One each choice of $\boldsymbol{W}_k$ that has zero-interference is the pseudo-inverse of $\boldsymbol{H}_k$ as

$$\boldsymbol{W}_k = \boldsymbol{H}_k\left(\boldsymbol{H}_k^{\mathrm{H}}\boldsymbol{H}_k\right)^{-1}. \tag{5.31}$$

Since $\boldsymbol{H}_k^{\mathrm{H}}\boldsymbol{W}_k = \boldsymbol{I}_N$, we respectively obtain the ZFBF vector and receive signal-to-noise ratio for UE n

$$\boldsymbol{w}_{n,k} = \boldsymbol{W}_k(:, n) \text{ and } \mathrm{SNR}_{n,k} = \frac{p_{n,k}}{\sigma_n^2}. \tag{5.32}$$

Using the beamforming vector $\boldsymbol{w}_{n,k}$ in (5.32), the effective channel gain of UE n is $\|\boldsymbol{w}_{n,k}\|^{-2}$

[140]. Based on (5.32) and effective channel gains $\{\|\boldsymbol{w}_{n,k}\|^{-2}\}_{n=1}^{N}$, we obtain

$$\begin{aligned}\min_{\{p_{n,k}\}_{n=1}^{N}} \; & VF(P_k^{\mathrm{BST}}) + \sum_{n=1}^{N} c_{3,n,k} c_{4,n,k} \exp(-c_{1,n}[10\log_{10}(p_{n,k}\sigma_n^{-2}) - c_{2,n}]) \\ \text{s.t.} \; & \sum_{n=1}^{N} p_{n,k} \left\|\boldsymbol{w}_{n,k}\right\|^2 \le P^{\max} \\ & p_{n,k} \ge \gamma_n^{\mathrm{REQ}} \sigma_n^2, n = 1, \ldots, N.\end{aligned} \tag{5.33}$$

Note that the feasibility check of (5.33) as a linear problem is

$$\begin{aligned}\text{find} \; & \{p_{n,k}\}_{n=1}^{N} \\ \text{s.t.} \; & \sum_{n=1}^{N} p_{n,k} \left\|\boldsymbol{w}_{n,k}\right\| \le P^{\max} \\ & p_{n,k} \ge \gamma_n^{\mathrm{REQ}} \sigma_n^2, n = 1, \ldots, N.\end{aligned} \tag{5.34}$$

The corresponding $c_{3,n,k}$ and $c_{4,n,k}$ reduce to

$$c_{3,n,k} = \frac{1}{1 + \exp(-c_{1,n}[10\log_{10}(p_{n,k}\sigma_n^{-2}) - c_{2,n}])} \tag{5.35a}$$

$$c_{4,n,k} = \frac{q_{n,k}^{\mathrm{A}}}{1 + \exp(-c_{1,n}[10\log_{10}(p_{n,k}\sigma_n^{-2}) - c_{2,n}])}. \tag{5.35b}$$

Based on the above analysis, we propose the ZFBF algorithm, the detailed procedures of which are summarized in Algorithm 7, and the proposed ZFBF algorithm converges to a KKT point of (5.24) with $d \ge N$.

5.3.3 Complexity Analysis

The major complexity of the SABF algorithm in slot lies in the iteration loop in lines 6–8. Therefore, we focus on analyzing the computational complexity of the iteration loop. Moreover, the major computational complexity in the iteration loop comes from solving problem (5.30) via the interior-point method. Hence, we evaluate the worst-case computational complexity of solving (5.30) via the interior-point method and multiply it by the number of iterations $T_1^{\max}$. Problem (5.30) belongs to the class of second-order conic optimization. The number of second-order cone with dimension dN is $N + 1$, and the number of linear constraints is $3N$. Thus,

Algorithm 7 ZFBF Algorithm

1: **Initialize**: iteration index $\tau = 0$, stop threshold ϵ, maximum number of iterations $T_1^{\max}$, and backlogs of access queues $\boldsymbol{q}_k^{\mathrm{A}}$
2: BST obtains a feasible point $\{p_{n,k}^0\}_{n=1}^N$ via (5.34).
3: BST updates $\{c_{3,n,k}^0\}_{n=1}^N$ and $\{c_{4,n,k}^0\}_{n=1}^N$ via (5.35a) and (5.35b)
4: **repeat**
5: $\tau \leftarrow \tau + 1$
6: Via $\{c_{3,n,k}^{\tau-1}, c_{4,n,k}^{\tau-1}\}_{n=1}^N$, BST solves the optimization problem (5.33), and obtains $\{p_{n,k}^{\tau}\}_{n=1}^N$
7: BST updates the following parameters:

$$c_{3,n,k}^{\tau} = \frac{1}{1+\exp(-c_{1,n}[10\log_{10}(p_{n,k}^{\tau}\sigma_n^{-2}) - c_{2,n}])}$$

$$c_{4,n,k}^{\tau} = \frac{q_{n,k}^{\mathrm{A}}}{1+\exp(-c_{1,n}[10\log_{10}(p_{n,k}^{\tau}\sigma_n^{-2}) - c_{2,n}])}$$

8: BST stacks $\boldsymbol{c}_{3,k}^{\tau} := [c_{3,1,k}^{\tau}, \ldots, c_{3,N,k}^{\tau}]$ and $\boldsymbol{c}_{4,k}^{\tau} := [c_{4,1,k}^{\tau}, \ldots, c_{4,N,k}^{\tau}]$
9: **until** $\frac{\left\|\boldsymbol{c}_{3,k}^{\tau} - \boldsymbol{c}_{3,k}^{\tau-1}\right\|}{\left\|\boldsymbol{c}_{3,k}^{\tau-1}\right\|} \leq \epsilon$ and $\frac{\left\|\boldsymbol{c}_{4,k}^{\tau} - \boldsymbol{c}_{4,k}^{\tau-1}\right\|}{\left\|\boldsymbol{c}_{4,k}^{\tau-1}\right\|} \leq \epsilon$ or $\tau > T^{\max}$

the number of iterations to solve (5.30) is $\mathcal{O}(\log \epsilon^{-1}\sqrt{4N+1})$, where ϵ is the required accuracy [141, 142]. Since the number of variables is $(d+2)N$, the computational complexity in each iteration is $\mathcal{O}((d+2)N^2[3 + d^2(N^2+N) + (d+2)^2N])$. The computational complexity of the SABF algorithm is $\mathcal{O}(\log \epsilon^{-1}T_1^{\max}\sqrt{4N+1}(d+2)N^2[3+d^2(N^2+N)+(d+2)^2N])$. Since the ZFBF algorithm has $N+1$ linear constraints and N variables, following similar arguments, we obtain the number of iterations to solve (5.33) as $\mathcal{O}(\log \epsilon^{-1}\sqrt{N+1})$, and the computational complexity in each iteration is $\mathcal{O}(N(N^2+N+d^2))$. Therefore, the computation complexity of the ZFBF algorithm is $\mathcal{O}(\log \epsilon^{-1}T_1^{\max}\sqrt{N+1}N(N^2+N+d^2))$.

We observe that the ZFBF algorithm has a lower computational complexity than the SABF algorithm. The next section will show that the SABF algorithm has a lower GEE than the ZFBF algorithm. Hence, the proposed SABF and ZFBF algorithms provide the operators with flexibility in balancing the performance and convergence.

5.4 Numerical Results

In this section, we present simulation results to evaluate the proposed dynamic cross-layer beamforming framework with the SABF and ZFBF algorithms. The pathloss of UE n is calculated

as

$$\omega_n = 17.3 + 38.3 \log_{10} \kappa_n + 24.9 \log_{10} f_c \text{ dB} \tag{5.36}$$

where κ_n is the link distance of UE n, and carrier frequency $f_c = 2.1$ GHz.

The BST is associated with three UEs and is equipped with six antennas. The distance between BST and UE is set as 200 m, and it is a representative value for cell-edge UEs. The AWGN power is set as $1 \times 10^{-10.7}$ mW. The power amplifier efficiency, maximum transmit power, baseband processing power of BSTs are, respectively, set as $\eta = 0.8$, $P^{\max} = 27$ dBm and $P^{\text{SP}} = 20$ dBm. The minimum SINR requirement $\gamma_n^{\text{REQ}} = 2$ dB. The MCS related factors are set as $c_{1,n} = 0.451$ and $c_{2,n} = 20$. Unless otherwise specified, the purchasing and selling prices of a unit energy is set as $\gamma_b = 1.6 \times 10^{-9}$ cents/slot/mW and $\gamma_s = 0.6 \times 10^{-9}$ cents/slot/mW. The average traffic arrival rate is set as $\bar{\nu}_n = 0.3$ packets/slot, the arrival rate of renewable energy is 150 mW[14], and the duration of slot is set as 1 ms. The value of control parameter V is empirically tuned to demonstrate tradeoff between the GEE and access delay.

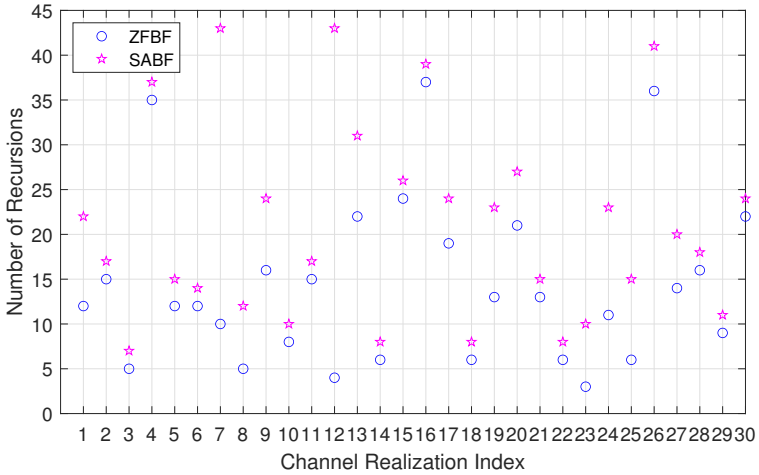


Figure 5.2: Number of iterations for the SABF and ZFBF algorithms, obtained for 30 different channel realizations with control parameter $V = 0.001$ and initial backlog of access queue as $q_{n,0} = 5$, $n = 1, \ldots, N$.

Figure 5.2 illustrates the number of iterations for the SABF and ZFBF algorithms obtained for 30 different channel realizations. We select the initial point of the SABF algorithm as the output of the ZFBF algorithm when the ZFBF algorithm is feasible; otherwise, we randomly select a feasible point of SABF algorithm in the feasible region. Therefore, the number of iterations for the SABF algorithm is counted as the summation of iterations for finding a solution to the

[14]For notational convenience, milliwatt is used to measure the arrival rate of renewable energy. In our system setup, 1 mW $= 1 \times 10^{-6}$ joules/slot.

ZFBF algorithm and the local optimal solution. We observe that the SABF and ZFBF algorithms can converge to the local optimal value within 30 iterations in most of the considered channel realizations.

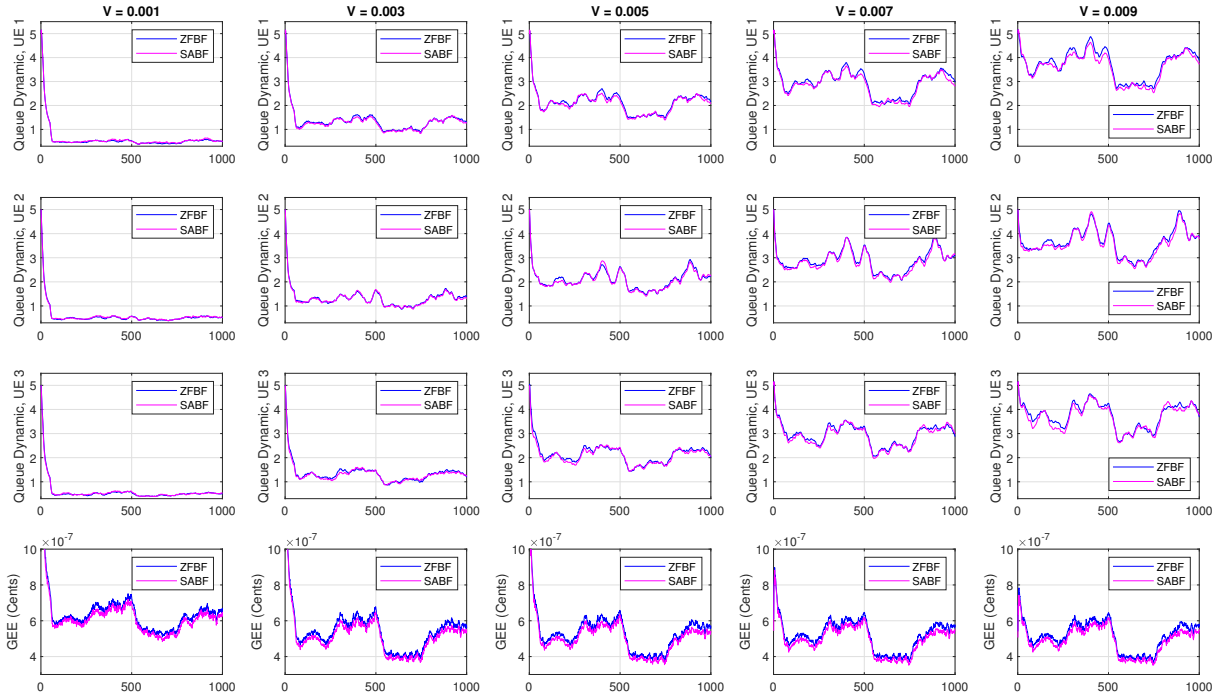


Figure 5.3: An illustration of moving-average queue dynamics and GEE with window size 50.

Figure 5.3 illustrates the dynamics of access queues of each UE and the GEE over 1000 slots. The prices of purchasing a unit of grid energy varies per 250 slots, and the prices are set as 1.6×10^{-9} cents/slot/mW, 1.9×10^{-9} cents/slot/mW, 1.3×10^{-9} cents/slot/mW and 1.8×10^{-9} cents/slot/mW. The dynamics of access queues demonstrates that combining the proposed dynamic cross-layer beamforming framework with the proposed SABF algorithm and ZFBF algorithm can stabilize the access queues of SGPCS. We also observe that a larger control parameter V leads to a larger backlog of access queues and a smaller GEE. We conclude that the access delay of UEs increases with the control parameter V based on Little's law. These observations confirm that the proposed dynamic cross-layer beamforming framework can effectively control the average transmission power via tuning V. We also observe that the GEE fluctuates at around the 250th, 500th, 750th, and 1000th slot, and the fluctuations come from the variation of the purchasing price of grid energy.

Figures 5.4 and 5.5 show the tradeoff between the annualized average GEE and access delay

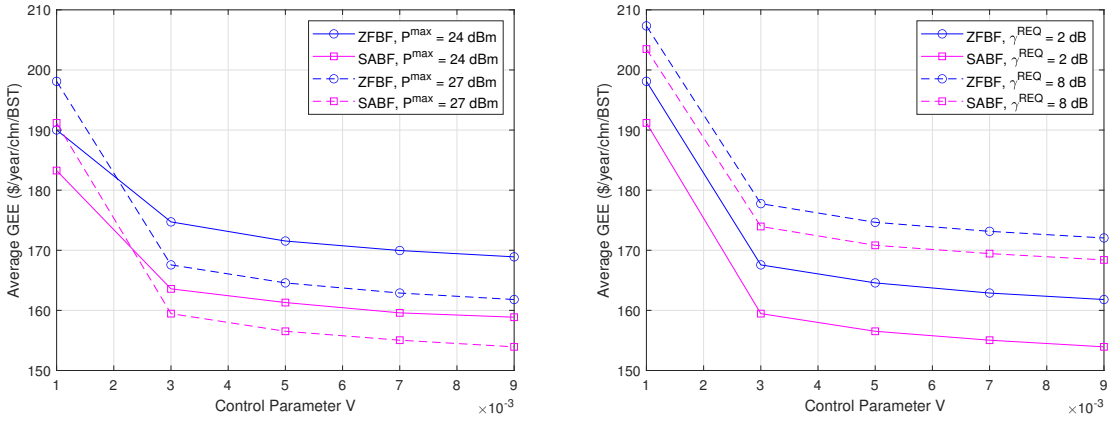


Figure 5.4: The average annualized GEE under different power budgets $P^{\max}$ and required SINR γ_n^{REQ}.

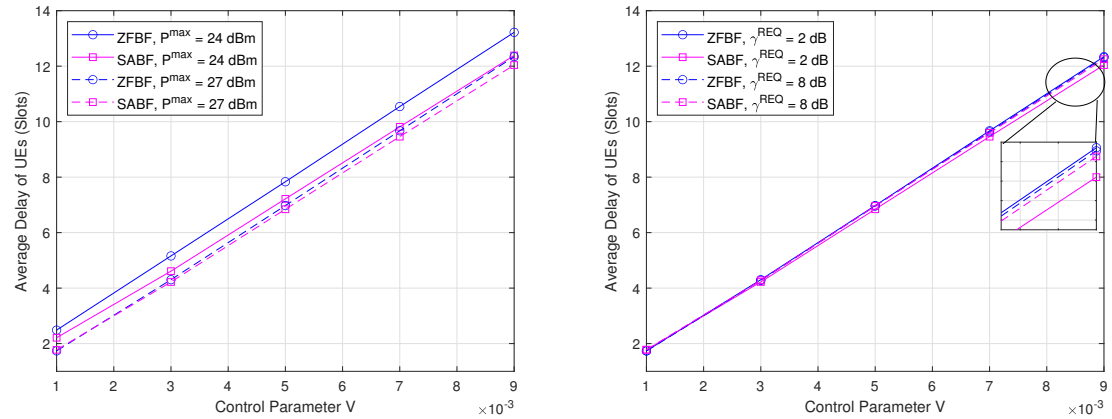


Figure 5.5: The access delay of UEs under different power budgets $P^{\max}$ and required SINR γ_n^{REQ}.

of UEs under different transmit power budgets of BST $P^{\max}$ and required SINR γ_n^{REQ}. Here, the GEE is annualized by considering the duration of a slot as 1 ms. We observe from Fig. 5.4 that the annualized average GEE decreases monotonically with the control parameter V, and the average access delay of UEs increases with the control parameter V. Therefore, the operator of SGPCS can tune the control parameter to achieve the target GEE at the expense of access delay of UEs. The SABF algorithm performs better than the ZFBF algorithm in the annualized average GEE and access delay under different parameter settings. We also observe that the performance gap in access delay increases with the control parameter V. For example, the gap of access delay increases from 0.27 slot to 0.83 slot when the power budget of BST is 24 dBm. This is due to the fact that a large control parameter V means a stringent demand for GEE and a loose requirement for access delay. Moreover, a large power budget of BST $P^{\max}$ gives the BST more flexibility to allocate the transmit power over different slots such that the GEE decreases when the power

budget of BST increases from $P^{\max} = 24$ dBm to $P^{\max} = 27$ dBm.

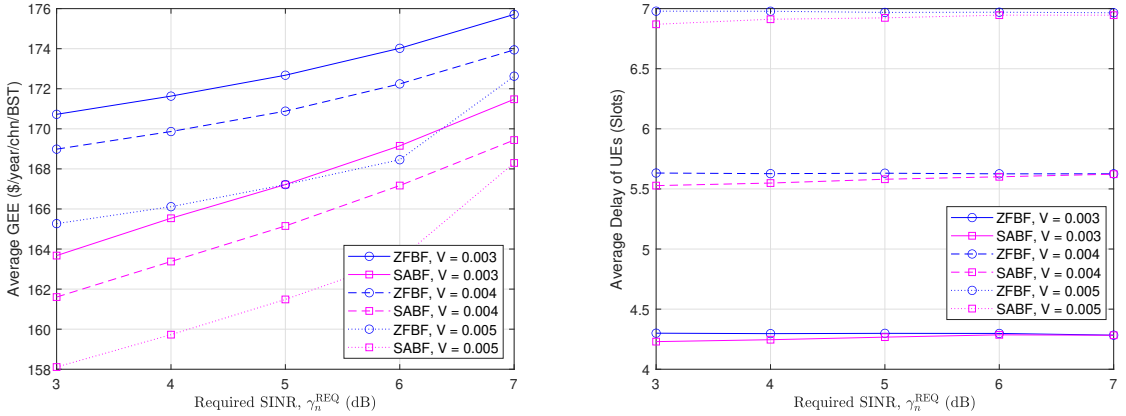


Figure 5.6: The impact of required SINR on the annualized average GEE and access delay of UEs.

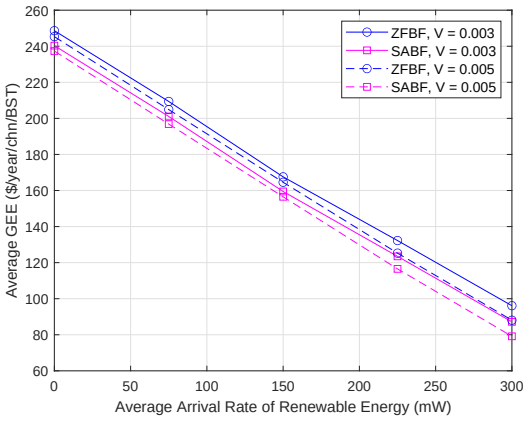


Figure 5.7: The impact of energy arrival on annualized average GEE.

Figure 5.6 illustrates the impact of required SINR on the annualized average GEE and access delay of UEs. We observe that a larger value of required SINR results in a larger GEE between the SABF and ZFBF algorithms. This observation can be explained by the fact that more energy needs to be consumed to satisfy the instantaneous SINR requirements. Increasing the value of required SINR also diminishes the gap of access delay between the SABF and ZFBF algorithms. This is due to the fact that the ZFBF algorithm can obtain a near-optimal solution in the high SINR region, for example, $\gamma_n^{\mathrm{REQ}} \geq 6$ dB in our setting. Fig. 5.7 shows that the annualized average GEE monotonically decreases with the arrival rate of renewable energy. This observation confirms that the application of renewable energy can reduce the energy bills of wireless operators. For

example, the GEE can be reduced by 33.65% and 63.73% when arrival rates of renewable energy are 150 mW and 300 mW with the control parameter $V = 0.003$. Besides, Figs. 5.6 and 5.7 also confirm that the proposed SABF algorithm performs better than the ZFBF algorithm in GEE and access delay of UEs.

5.5 Summary

We have developed cross-layer beamforming algorithms in the SGPCS using harvest-use-trade strategy, where the BST can purchase electricity from the smart grid if the harvested energy is insufficient and sell the surplus harvested energy to generate revenue. We have leveraged a Lyapunov learning model to formulate the TAEGEE minimization problem. Reformulating the TAEGEE minimization problem into a set of per-slot Lyapunov drift-plus-penalty minimization problems, we have revealed a tradeoff between the time-average expected GEE and access delay of UEs. For example, setting a large value of control parameter V, the GEE can be reduced at the expense of large access delay of UEs. Therefore, the operators can set the control parameter according to the arrival rate of renewable energy. Due to the non-convexity of per-slot Lyapunov drift-plus-penalty minimization problem, two suboptimal algorithms, namely SABF and ZFBF algorithms, have been proposed. The SABF algorithm outperforms the ZFBF algorithm in both GEE and access delay of UEs at the expense of a higher computational complexity.

Chapter 6

Joint Scheduling and Beamforming in Multi-Cell SGPCS With Energy Coordination

Chapter 5 investigates the application of renewable energy in a single-cell SGPCS and demonstrates that using renewable energy can reduce long-term GEE. The present chapter develops algorithms for integrating renewable energy into a multi-cell SGPCS. Besides, the present chapter considers the local energy exchanging among BSTs. More specifically, user scheduling, beamforming, and energy coordination are investigated in the multi-cell SGPCS, where the BSTs are powered by a smart grid and renewable energy resources. Heterogeneous energy coordination (i.e., energy merchandising with grid and energy exchanging among BSTs) is considered in the multi-cell SGPCS. On the one hand, users need to be rescheduled over several slots to avoid draining users' battery quickly (see discontinuous reception mode[15] in LTE). On the other hand, beamforming and energy exchanging need to be performed to adapt the channel variations in each slot. Therefore, a two time-scale resource allocation algorithm is required to minimize the long-term GEE in the multi-cell SGPCS. While the proposed algorithms in Chapter 5 are designed for one time-scale resource allocation, a practical two time-scale algorithm is required to schedule users in each frame[16], and obtain the beamforming vectors and amount of exchanged renewable energy in each slot. Base on the finite-sample analysis method, we prove that the proposed two time-scale algorithm can asymptotically achieve the optimal solutions via tuning a control parameter. The computational complexity of the proposed two time-scale algorithm is analyzed.

[15] `https://www.sharetechnote.com/html/Handbook_LTE_DRX.html`

[16] In our system setup, each frame consists of several slots.

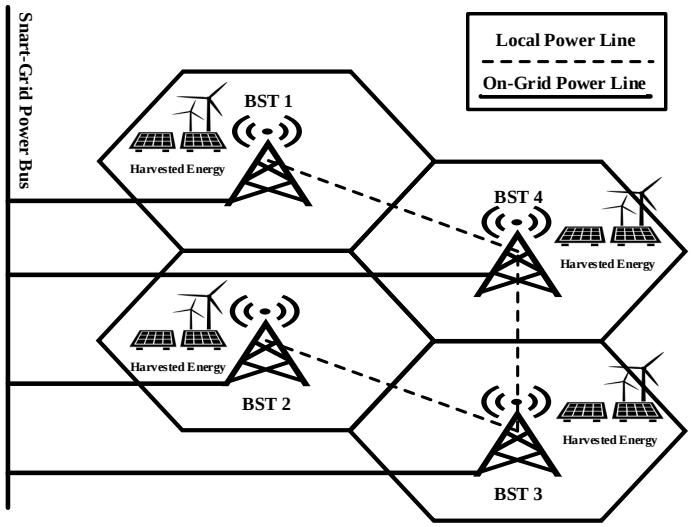


Figure 6.1: An illustration of multi-cell SGPCS.

6.1 System Model and Problem Statement

As shown in Fig. 6.1, we consider the downlink transmission of a multi-cell SGPCS with M BSTs. Each BST is equipped with d antennas. BST m is associated with N_m single-antenna UEs. Therefore, the total number of UEs is $N = \sum_{m=1}^{M} N_m$. Each BST connects to the core network and UEs via an optical-fiber link and wireless links, respectively. Moreover, each BST is powered by renewable resources (e.g., solar and/or wind) and a smart grid. A two time-scale framework is considered for scheduling UEs, design beamforming vectors, and exchange renewable energy. Since the arrival rates of renewable energy and channel-coefficient vectors vary at different time scales in practice [134], the arrival rates of renewable energy and channel-coefficient vectors are respectively updated over frames and slots. Here, each frame consists of T discrete slots. We respectively denote the indices for the frame and slot as frame i and slot k with $k = 1, \ldots, T$ and $i = 0, 1, \ldots, \infty$. Following Chapter 5, each slot has unit duration; therefore, the terms "energy" and "power" are used interchangeably.

6.1.1 Signal Model

Let $\boldsymbol{h}^i_{m,n,k} \in \mathbb{C}^d$ denote the channel-coefficient vector of link between UE n and BST m (or the (m,n)th access link) in slot k of frame i. Each entry of $\boldsymbol{h}^i_{m,n,k}$ follows CSCG with mean zero

and variance $\omega_{m,n}^{-1}$, where $\omega_{m,n}$ is the pathloss of the (m,n)th access link. We define the scheduled UE indicator as $a_{m,n}[i]$ which equals to one when the (m,n)th access link is scheduled at frame i; otherwise, it equals to zero. Hence, the received signal and SINR of the (m,n)th access link in slot k of frame i are, respectively, denoted as

$$y_{m,n,k}[i] = \sqrt{a_{m,n}[i]}\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,n,k}[i] + \sum_{j\neq n}\sqrt{a_{m,j}[i]}\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,l,k}[i] + \sum_{l\neq m}\sum_{j=1}^{N_j}\sqrt{a_{l,j}[i]}\boldsymbol{h}_{l,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{l,j,k}[i] + z_{m,n,k}[i] \quad (6.1)$$

and

$$\mathrm{SINR}_{m,n,k}[i] = \frac{a_{m,n}[i]|\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,n,k}[i]|^2}{I_{m,n,k}^{\mathrm{INTRA}}[i] + I_{m,n,k}^{\mathrm{INTER}}[i] + \sigma_{m,n,k}^2[i]} \quad (6.2)$$

where the term $z_{m,n,k}[i] \sim \mathcal{CN}(0,\sigma_{m,n}^2)$ is the AWGN of the (m,n)th access link in slot k of frame i; $\boldsymbol{w}_{m,n,k}^i$ is the single-stream beamforming vector for the (m,n)th access link in slot k of frame i; and the intra-cell interference and inter-cell interference are, respectively, given as

$$I_{m,n,k}^{\mathrm{INTRA}}[i] = \sum_{j\neq n} a_{m,j}[i]|\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,j,k}[i]|^2 \quad (6.3)$$

and

$$I_{m,n,k}^{\mathrm{INTER}}[i] = \sum_{l\neq m}\sum_{j=1}^{N_j} a_{l,j}[i]|\boldsymbol{h}_{l,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{l,j,k}[i]|^2. \quad (6.4)$$

Hence, the data rate of the (m,n)th access link in slot k of frame i is given as $r_{m,n,k}[i] = \log\big(1+\mathrm{SINR}_{m,n,k}[i]\big)$.

Based on (6.1), the consumed power of BST m in slot k of frame i is denoted by

$$P_{m,k}^{\mathrm{BST}}[i] = \frac{1}{\eta}\sum_{n\in\mathcal{N}_m^{\mathrm{ACT}}[i]}\|\boldsymbol{w}_{m,n,k}^i\|^2 + P_m^{\mathrm{CIR}} \quad (6.5)$$

where the circuit power consumption is defined as $P_m^{\mathrm{CIR}} := P_m^{\mathrm{SP}}\big(0.87+0.1d+0.03d^2\big)$ with P_m^{SP} as the consumed power on baseband processing of BST m [20]; and η is the power amplifier efficiency of BSTs. Here, $\mathcal{N}_m^{\mathrm{ACT}}[i]$ denotes the set of scheduled UEs of BST m in frame i.

6.1.2 Energy-Coordination Model

As shown in Fig. 6.1, the BSTs have two options to perform the heterogeneous energy coordination: 1) energy merchandizing via the on-grid power lines; and 2) energy exchanging via the local power lines.

Energy merchandizing. Since smart meters enable BSTs to trade energy bi-directionally with the smart grid, BSTs can purchase/sell energy when the renewable energy of the BSTs is insufficient/surplus. Let γ_b, and γ_s respectively denote the price of unit energy when the BSTs purchase from and sell to the smart grid. Following Chapter 5, we set $\gamma_b > \gamma_s \geq 0$ to avoid the redundant energy merchandizing.

Energy exchanging. Another way to share energy is leveraging the local power lines. Due to the issues of regulation and resistive loss, the BSTs are partially connected as shown in Fig. 6.1. Let $\varpi_{m\to l,k}[i]$ and $\varpi_{l\to m,k}[i]$ respectively denote the amount of power delivered from BST m to BST l and vice versa in slot k of frame i. Two-way energy flow needs to be avoided in a specific slot. Hence, we include the energy-flow constraints as

$$\varpi_{m\to l,k}[i] + \varpi_{l\to m,k}[i] = 0. \tag{6.6}$$

The case $\varpi_{m\to l,k}[i] \geq 0$ indicates that the renewable energy is delivered from BST m to BST l, and vice versa. Moreover, $\varpi_{m\to m,k}[i] = 0$, $m = 1, 2, \ldots, M$. We formulate the attenuation of local power lines by considering the efficiency of local power line between BST m to BST l as $\chi_{m\to l} \in (0, 1)$ with $\chi_{m\to l} = \chi_{l\to m}$. Let the set $\mathcal{N}_m$ be the neighbor BSTs who have one-hop local power lines to BST m. The amount of net exchanged energy via the local power lines for BST m in slot k of frame i is denoted as

$$E^{\mathrm{LPE}}_{m,k}[i] = \sum_{l\in\mathcal{N}_m} \max\left\{\varpi_{m\to l,k}[i], \chi_{m\to l}\varpi_{m\to l,k}[i]\right\}. \tag{6.7}$$

The amount of net exchanged energy in (6.7) is calculated as follows.

- When the value of $\varpi_{m\to l,k}[i]$ is positive, the energy is flowing from BST m to BST l in slot k of frame i. The net output energy of BST m and net input energy of BST l are, respectively, $\varpi_{m\to l,k}[i]$ and $\chi_{m\to l}\varpi_{m\to l,k}[i]$. Therefore, the net output energy of BST l is

$-\chi_{m\to l}\varpi_{m\to l,k}[i]$. Recalling the energy-flow constraints in (6.6) and the fact $\chi_{m\to l} = \chi_{l\to m}$, the net output energy of BST l is $\chi_{l\to m}\varpi_{l\to m,k}[i]$.

- When the value of $\varpi_{m\to l,k}[i]$ is negative, the energy is flowing from BST l to BST m in slot k of frame i. Following similar arguments, we respectively obtain the net energy outputs of BST m and BST l as $\chi_{m\to l}\varpi_{m\to l,k}[i]$ and $\varpi_{l\to m,k}[i]$.

The wireline delivers electricity using electrons, and the electrons are uniformly distributed in the wireline. When voltage increases between two ends of a wireline, the electricity can be exchanged between two BSTs in real time. In practical system, changing the direction of electric current can take some time (i.e., turnaround time). However, we assume the turnaround time to be zero to exploit the upper bound of grid-energy saving when the renewable energy is used in the multi-cell SGPCS.

Using energy merchandizing and exchanging, the GEE of BST m in slot k of frame i is calculated as [20]

$$F(P_{m,k}^{\mathrm{BST}}[i]) = (\gamma_b - \gamma_s)\left[P_{m,k}^{\mathrm{BST}}[i] + E_{m,k}^{\mathrm{LPE}}[i] - \frac{1}{T}E_m^{\mathrm{HAV}}[i]\right]^+ + \gamma_s\left[P_{m,k}^{\mathrm{BST}}[i] + E_{m,k}^{\mathrm{LPE}}[i] - \frac{1}{T}E_m^{\mathrm{HAV}}[i]\right] \quad (6.8)$$

where $E_m^{\mathrm{HAV}}[i]$ denotes the amount of harvested renewable energy of BST m in frame i. Since arrival of renewable energy remains stable in a frame, the amount of harvested renewable energy by BST m is $\frac{1}{T}E_m^{\mathrm{HAV}}[i]$.

When the (m,n)th link has a larger inter-cell interference term, BST m will consume more energy to guarantee a certain SINR requirement based on (6.2). Based on (6.5) and (6.8), the GEE of BSTs is a function of BSTs' transmit power and exchanged energy. To reveal the interaction between exchanged energy and inter-cell interference, we assume the same renewable energy arrival rates at the two BSTs. When inter-cell interference is zero, the renewable energy will flow from the low-load BST to the high-load BST[17] for the compensation of energy usage. When inter-cell interference increases, the energy exchanging direction will depend on the level of inter-cell interference and traffic volume at the two BSTs. When the associated UEs of high-load BST suffer more interference than UEs of low-load BST, the high-load BST will consume more energy.

[17]By comparing the traffic volume (i.e., backlogs in access queues), the BST with lower traffic volume than another BST is termed as low-load BST; otherwise, the BST is termed as high-load BST.

Therefore, the renewable energy flows from low-load BST to high-load BST.

6.1.3 Traffic Model

Access queue. At each BST, we consider that BST m maintains N_m access queues for the associated UEs. The dynamic equation for the nth access queue of BST m (or the (m,n)th access queue) is given as

$$q^{\mathrm{A}}_{m,n,k+1}[i] = q^{\mathrm{A}}_{m,n,k}[i] - r_{m,n,k}[i] + \nu_{m,n,k}[i] \tag{6.9}$$

where $q^{\mathrm{A}}_{m,n,k+1}[i]$ and $q^{\mathrm{A}}_{m,n,k}[i]$ are the backlogs of the (m,n)th access queue in slots $k+1$ and k of frame i with $q^{\mathrm{A}}_{m,n,T+1}[i] = q^{\mathrm{A}}_{m,n,1}[i+1]$; $\nu_{m,n,k}[i]$ and $r_{m,n,k}[i]$ are, respectively, the arrival rate and data rate of the (m,n)th access queue in slot k of frame i.

Processing queue. At each UE, we consider that UE n of BST m (or the (m,n)th UE) maintains a processing queue (or the (m,n)th processing queue) for upper layer processing corresponding to the (m,n)th access queue. The dynamic equation for the (m,n)th processing queue is given as

$$q^{\mathrm{U}}_{m,n,k+1}(i) = q^{\mathrm{U}}_{m,n,k}[i] - s_{m,n,k}[i] + r_{m,n,k}[i] \tag{6.10}$$

where $q^{\mathrm{U}}_{m,n,k+1}(i)$ and $q^{\mathrm{U}}_{m,n,k}[i]$ are the backlogs at the beginning of slots $k+1$ and k of frame i with $q^{\mathrm{U}}_{m,n,T+1}[i] = q^{\mathrm{U}}_{m,n,1}[i+1]$. We consider the constant processing rate of the (m,n)th processing queue. Therefore, the processing rate $s_{m,n,k}[i] := \min\{\tilde{s}_{m,n}, q^{\mathrm{U}}_{m,n,k}[i]\}$ where $\tilde{s}_{m,n}$ denotes the constant processing rate of the (m,n)th processing queue.

Here, we consider that the (m,n)th UE is associated with an access queue $q^{\mathrm{A}}_{m,n,k}[i]$ at the BST m and a processing queue at the (m,n)th UE. The motivations can be justified as follows. Since the transmit power of BSTs is finite, the data rate of the (m,n)th UE $r_{m,n,k}[i]$ is limited. Therefore, a buffer (i.e., access queue) is required at BST m to store the data that has not been transmitted to the (m,n)th UE. When the (m,n)th UE has limited computational capability, a buffer (i.e., processing queue) is required to store the unprocessed information bits that have been received at the (m,n)th UE. Note that the system setup of access and processing queues can be adopted to delay insensitive traffics (e.g., file transfer and hypertext transfer traffics).

In practical systems, the values of arrival rate $\nu_{m,n,k}[i]$, data rate $r_{m,n,k}[i]$ and processing rate

$s_{m,n,k}[i]$ are bounded as

$$\begin{aligned} \nu_{m,n,k}[i] &\in [0, \nu^{\max}] \\ r_{m,n,k}[i] &\in [0, r^{\max}] \\ s_{m,n,k}[i] &\in [0, s^{\max}] \end{aligned} \tag{6.11}$$

where $\nu^{\max}$, $r^{\max}$ and $s^{\max}$ are, respectively, the maximum arrival rate, maximum data rate and maximum processing rate.

The average arrival rate of the (m,n)th access queue and the average processing rate of the (m,n)th processing queue are, respectively, given as $\bar{\nu}_{m,n} := \mathbb{E}\big[\nu_{m,n,k}[i]\big]$ and $\bar{s}_{m,n} := \mathbb{E}\big[s_{m,n,k}[i]\big]$. Moreover, the average arrival rate vector is given as $\bar{\boldsymbol{\nu}} = \mathrm{vec}(\bar{\nu}_{1,1}, \ldots, \bar{\nu}_{M,N_M})$, and the average processing rate vector is given as $\bar{\boldsymbol{s}} = \mathrm{vec}(\bar{s}_{1,1}, \ldots, \bar{s}_{M,N_M})$.

6.1.4 Problem Statement

Our objective is to minimize the time-average GEE via designing jointly the scheduled UE indicators $\{a_{m,n}[i]\}_{m,n,i}$ in each frame and the beamforming vectors $\{\boldsymbol{w}_{m,n,k}[i]\}_{m,n,k,i}$ and exchanged renewable energy variables $\{\varpi_{m\to l,k}[i]\}_{m,l,k,i}$ in each slot. Due to the lack of knowledge on stochastic arrival of renewable energy and variations of channel states, we consider the following constraints in the time-average GEE minimization problem:

- Rate-limit constraints:

$$r_{m,n,k}[i] \le q^{\mathrm{A}}_{m,n,k}[i], n = 1, \ldots, N_m \text{ and } m = 1, \ldots, M \tag{6.12}$$

  which guarantee that each BST does not transmit blank information in slot k of frame i.

- Dynamic proportional-rate constraints:

$$\frac{r_{m,n,k}[i]}{r_{l,j,k}[i]} = \frac{q^{\mathrm{A}}_{m,n,k}[i]}{q^{\mathrm{A}}_{l,j,k}[i]}, n \in \mathcal{N}^{\mathrm{ACT}}_m[i], j \in \mathcal{N}^{\mathrm{ACT}}_l[i], m, l = 1, \ldots, M \tag{6.13}$$

  which guarantees that the UE with larger backlog obtains a better service rate in the respective slot.

- Slot-level power constraints:

$$\sum_{n \in N_m^{\mathrm{ACT}}[i]} \left\| \boldsymbol{w}_{m,n,k}[i] \right\|^2 \leq P_m^{\max}, m = 1, \ldots, M \tag{6.14}$$

  where $P_m^{\max}$ is the maximum transmit power of BST m.

- Queue-stable constraints:

$$\limsup_{I \to \infty} \frac{1}{I} \sum_{i=1}^{I} \mathbb{E}\left[q_{m,n,1}^{\mathrm{A}}[i] + q_{m,n,1}^{\mathrm{U}}[i] \right] < \infty, \forall m, n \tag{6.15}$$

  which guarantee that the data of UEs will be served in finite time.

As a result, the TAEGEE minimization problem is formulated as

$$\min_{\{\boldsymbol{w}_{m,n,k}[i], a_{m,n}[i], \varpi_{m \to l,k}[i]\}_{m,l,n,k,i}} \lim_{I \to \infty} \frac{1}{IT} \sum_{i=1}^{I} \sum_{k=1}^{T} \sum_{m=1}^{M} \mathbb{E}\left[F(P_{m,k}^{\mathrm{BST}}[i]) \right] \tag{6.16a}$$

$$\text{s.t. (6.6) and (6.12)} - \text{(6.15)}. \tag{6.16b}$$

When the proportional ratios are not set according to the backlog at access queues as shown in (6.13), certain BSTs will waste some time slots. This observation can be justified by the following three-UE case. Suppose that we have $q_{m,1,k}^{\mathrm{A}}[i] = 2$ nats/slot/Hz, $q_{m,2,k}^{\mathrm{A}}[i] = 3$ nats/slot/Hz, and $q_{m,3,k}^{\mathrm{A}}[i] = 5$ nats/slot/Hz. Setting the rate ratios of the three UEs as $r_{m,1,k}[i] : r_{m,2,k}[i] : r_{m,3,k}[i] = 1 : 1 : 1$. When the optimal rates are $r_{m,1,k}[i] = r_{m,2,k}[i] = r_{m,3,k}[i] = 2$ nats/slot/Hz in slot k, the BST cannot send information to the second and third UEs in slots $k+1$ to T of frame i based on the predefined ratio $r_{m,1,k}[i] : r_{m,2,k}[i] : r_{m,3,k}[i] = 1 : 1 : 1$. Therefore, we use constraints in (6.13) to allocate adaptively the transmission rate of scheduled UEs without wasting time slots. Since the scheduled UE indicators are coupled with the beamforming vectors, the TAEGEE minimization problem (6.16) is challenging to handle via classical convex optimization methods. Therefore, we are motivated to use the Lyapunov learning technique to obtain a feasible solution to the TAEGEE minimization problem (6.16). The optimality of a feasible solution is analyzed. Moreover, we also investigate the tradeoff between the GEE and the end-to-end delay of UEs.

6.2 Tradeoff Between Grid Energy Expenditure and End-to-End Delay

We construct a Lyapunov drift-plus-penalty function as

$$D\big(\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i]\big)+V\sum_{m=1}^{M}\mathbb{E}_{\iota_{2,k}[i]}\big[F(P_{m,k}^{\mathrm{BST}}[i])\big] \tag{6.17}$$

where V is an introduced control parameter; the operator $\mathbb{E}_{\iota_{2,k}[i]}[\cdot]$ is expectation over random sources $\iota_{2,k}[i]=\{\boldsymbol{h}_{m,n,k}[i],E_{m,k}^{\mathrm{HAV}}[i],\nu_{m,n,k}[i],s_{m,n,k}[i]\}_{m,n}$; and $\boldsymbol{q}^{\mathrm{A}}[i]$ and $\boldsymbol{q}^{\mathrm{U}}[i]$ are respectively obtained by stacking the backlogs of access queues and processing queues as

$$\boldsymbol{q}^{\mathrm{A}}[i]:=\mathrm{vec}(q_{1,1,1}^{\mathrm{A}}[i],\ldots,q_{M,N_M,1}^{\mathrm{A}}[i])$$
$$\boldsymbol{q}^{\mathrm{U}}[i]:=\mathrm{vec}(q_{1,1,1}^{\mathrm{U}}[i],\ldots,q_{M,N_M,1}^{\mathrm{U}}[i]).$$

The one-frame drift function $D(\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i])$ is obtained as

$$D\big(\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i]\big)=\frac{1}{2}\mathbb{E}_{\iota_{2,k}[i]}\big[\|\boldsymbol{q}^{\mathrm{A}}[i+1]\|^2-\|\boldsymbol{q}^{\mathrm{A}}[i]\|^2+\|\boldsymbol{q}^{\mathrm{U}}[i+1]\|^2-\|\boldsymbol{q}^{\mathrm{U}}[i]\|^2\big|\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i]\big]. \tag{6.18}$$

When random sources in $\iota_{2,k}[i]$ are independent and identically distributed over different slots, we obtain the upper bound of the Lyapunov drift-plus-penalty function in (6.17) as

$$\begin{aligned}&D\big(\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i]\big)+V\sum_{m=1}^{M}\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}\big[F(P_{m,k}^{\mathrm{BST}}[i])|\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i]\big]\\ \le& Tc_1+V\sum_{m=1}^{M}\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}\big[F(P_{m,k}^{\mathrm{BST}}[i])|\boldsymbol{q}^{\mathrm{A}}[i],\boldsymbol{q}^{\mathrm{U}}[i]\big]\\ &+\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}^{\top}\big[\boldsymbol{\nu}_k[i]-\boldsymbol{r}_k[i]|\boldsymbol{q}^{\mathrm{A}}[i]\big]\boldsymbol{q}^{\mathrm{A}}[i]+\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}^{\top}\big[\boldsymbol{r}_k[i]-\boldsymbol{s}_k[i]|\boldsymbol{q}^{\mathrm{U}}[i]\big]\boldsymbol{q}^{\mathrm{U}}[i]\end{aligned} \tag{6.19}$$

where $c_1=\frac{1}{2}[(s^{\max})^2+2(r^{\max})^2+(\nu^{\max})^2]\sum_{m=1}^{M}N_m$. The traffic arrival, data rate, and service rate vectors are obtained as $\boldsymbol{\nu}_k[i]:=\mathrm{vec}(\nu_{1,1,k}[i],\ldots,\nu_{M,N_M,k}[i])$, $\boldsymbol{r}_k[i]:=\mathrm{vec}(r_{1,1,k}[i],\ldots,r_{M,N_M,k}[i])$, and $\boldsymbol{s}_k[i]:=\mathrm{vec}(s_{1,1,k}[i],\ldots,s_{M,N_M,k}[i])$, respectively. The proof of (6.19) is relegated to Appendix D.1.

Minimizing the RHS of (6.19) under the constraints in (6.6) and (6.12)–(6.14) gives us a

feasible solution to the time-average GEE minimization problem (6.16). Due to the constraints in (6.14), the GEE is bounded by

$$\left|\sum_{m=1}^{M}\mathbb{E}\left[F(P_{m,k}^{\mathrm{BST}}[i])\right]\right| \leq F. \tag{6.20}$$

The properties of the obtained feasible solution is discussed in Theorem 6.1.

Theorem 6.1. *Suppose the random sources in* $\{\iota_{2,k}[i]\}_{k,i}$ *are independent and identically distributed over slots, and the initial expected backlogs of access queues and processing queues satisfy* $\mathbb{E}\left[\|\boldsymbol{q}^{\mathrm{A}}[1]\|^2\right] < \infty$ *and* $\mathbb{E}\left[\|\boldsymbol{q}^{\mathrm{U}}[1]\|^2\right] < \infty$. *Suppose the resource allocation variables* $\{\boldsymbol{w}_{m,n,k}[i], a_{m,n}[i], \varpi_{m\to l,k}[i]\}_n$ *satisfy*

$$\bar{\boldsymbol{\nu}} + \epsilon\mathbf{1}_{N\times 1} \leq \mathbb{E}[\boldsymbol{r}_k[i]] \leq \bar{\boldsymbol{s}} - \epsilon\mathbf{1}_{N\times 1} \tag{6.21}$$

where ϵ *is a small positive constant.*

When the above assumptions are satisfied, the minimizer to RHS of (6.19) *under the constraints in* (6.6) *and* (6.12)–(6.14) *asymptotically achieves the optimal GEE* F^* *as*

$$F^* \leq \lim_{I\to\infty}\frac{1}{IT}\sum_{i=1}^{I}\sum_{k=1}^{T}\sum_{m=1}^{M}\mathbb{E}\left[F(P_{m,k}^{\mathrm{BST}}[i])\right] \leq F^* + \frac{c_1}{V} \tag{6.22}$$

when the control parameter V *approaches infinity.*

Moreover, the queue backlogs satisfy

$$\limsup_{I\to\infty}\frac{1}{I}\sum_{i=1}^{I}\mathbb{E}\left[q_{m,n,1}^{\mathrm{A}}[i] + q_{m,n,1}^{\mathrm{U}}[i]\right] \leq \frac{c_1 + 2VF}{\epsilon} \tag{6.23}$$

such that the constraints in (6.15) *are satisfied.*

Based on Theorem 6.1, we conclude that the set of minimizers to the RHS of (6.19) under the constraints in (6.6) and (6.12)–(6.14) is a feasible solution to the TAEGEE minimization problem (6.16). Based on (6.22), we observe that gap between the optimal GEE and obtained GEE decreases with the control parameter as $O(1/V)$. Here, $O(1/V)$ is a polynomial of $1/V$. By Little's law and (6.23), we observe that the end-to-end delay of UEs is a linearly increasing function of the control parameter V as $O(V)$. When the control parameter V approaches infinity, the end-to-end delay of UEs increases to infinity. Hence, we conclude that the time-average expected GEE can

be traded for the end-to-end delay of UEs by tuning the control parameter. Besides, the proposed analysis method in Appendix D.2 also explicitly defines the stable region of multi-cell SGPCS as shown in (6.21).

6.3 Two Time-Scale UE Scheduling, Beamforming and Energy Exchanging

Based on Theorem 6.1, we observe the elegance of a minimizer to the RHS of (6.19) under the constraints in (6.6) and (6.12)–(6.14). In this section, we propose a practical two time-scale algorithm that jointly designs the scheduled UE indicators $\{a_{m,n}[i]\}_{m,n}$ in frame i and the beamforming vectors and exchanged renewable energy variables $\{\boldsymbol{w}_{m,n,k}[i], \varpi_{m\to l,k}[i]\}_{m,l,n}$ in slot k of frame i.

6.3.1 Optimal Scheduled UE Indicator in Each Frame

After some algebraic manipulations on the RHS of (6.19), we obtain the term related to $\{r_{m,n,k}[i]\}_{m,n}$ as

$$\sum_{m=1}^{M}\sum_{n=1}^{N_m}\Big(q^{\mathrm{U}}_{m,n,1}[i]-q^{\mathrm{A}}_{m,n,1}[i]\Big)\mathbb{E}_{\iota_{2,k}[i]}\left[\sum_{k=1}^{T} r_{m,n,k}[i]\right]. \tag{6.24}$$

The term related to GEE in the RHS of (6.19) is denoted by

$$V\sum_{m=1}^{M}\mathbb{E}_{\iota_{2,k}[i]}\left[\sum_{k=1}^{T} F\Big(P^{\mathrm{BST}}_{m,k}[i]\Big)\right]. \tag{6.25}$$

We observe that the terms $\sum_{k=1}^{T} r_{m,n,k}[i]$ and $\sum_{k=1}^{T} F(P^{\mathrm{BST}}_{m,k}[i])$ are coupled via $a_{m,n}[i]$. In order to minimize the RHS of (6.19), we obtain the *optimal* scheduled UE indicator as

$$a^{*}_{m,n}[i]=\begin{cases} 0, q^{\mathrm{U}}_{m,n}[i]-q^{\mathrm{A}}_{m,n,1}[i]\geq 0 \text{ or } q^{\mathrm{A}}_{m,n,1}[i]=0 \\ 1, \text{otherwise.}\end{cases} \tag{6.26}$$

The optimality of (6.26) is justified via contradiction. Based on the principle of opportunis-

tically minimizing an expectation [30], the optimal scheduled UE indicators minimize

$$\underbrace{V\sum_{m=1}^{M}\sum_{k=1}^{T}F\Big(P_{m,k}^{\mathrm{BST}}[i]\Big)}_{\text{GEE}}+\underbrace{\sum_{m=1}^{M}\sum_{k=1}^{T}\sum_{n=1}^{N}\Big(q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\Big)r_{m,n,k}[i]}_{\text{Data Rates of UEs}}. \tag{6.27}$$

Since $\frac{\partial F\left(P_{m,k}^{\mathrm{BST}}[i]\right)}{\partial a_{m,n}[i]}\geq 0$ is based on the definition of $F\left(P_{m,k}^{\mathrm{BST}}[i]\right)$ in (6.8), the GEE monotonically increases with $a_{m,n}[i]$. Therefore, we have the following reasoning.

- Suppose that the (m,n)th UE is scheduled (namely $a_{m,n}[i]=1$) when $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\geq 0$ or $q_{m,n,1}^{\mathrm{A}}[i]=0$. In the case with $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\geq 0$, we observe that data rate of the (m,n)th UE increases the value of (6.27). To minimize (6.27), the data rate of the (m,n)th UE needs to be set to zero in frame i. The data rate of the (m,n)th UE with $q_{m,n,1}^{\mathrm{A}}[i]=0$ needs to be set to zero following the similar arguments. The scheduled UE indicator of the (m,n)th UE needs to be set as $a_{m,n}[i]=0$ which contradicts the assumption. Therefore, we conclude that $a_{m,n}[i]=0$ when $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\geq 0$ or $q_{m,n,1}^{\mathrm{A}}[i]=0$.

- Suppose that the (m,n)th UE is not scheduled (namely $a_{m,n}[i]=0$) when $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]<0$ and $q_{m,n,1}^{\mathrm{A}}[i]\neq 0$. Based on the previous reasoning, we obtain that the (m,n)th UE is not scheduled when $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\geq 0$ or $q_{m,n,1}^{\mathrm{A}}[i]=0$. When the (m,n)th UE will not be scheduled in the case $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]<0$ and $q_{m,n,1}^{\mathrm{A}}[i]\neq 0$, the backlog of access queue $q_{m,n,1}^{\mathrm{A}}[i]$ will become infinite which contradicts the (m,n)th queue-stable constraint in (6.15). Therefore, we conclude that $a_{m,n}[i]=1$ when $q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]<0$ and $q_{m,n,1}^{\mathrm{A}}[i]\neq 0$.

In our setting, the unscheduled UE is set into deep sleep mode[18] such that the unscheduled UE can save more energy than the traditional idle mode. It usually takes time to recover from the deep sleep mode to the operation mode in practice. When the per-slot optimization is used, the recovery time from deep sleep mode cannot be negligible compared the the duration of slot. Hence, the obtained end-to-end delay by the per-slot optimization is inaccurate. Using our proposed two-scale optimization framework, one can choose a frame duration such that the recovery time from

[18]Here, the deep sleep mode is similar to discontinuous reception model [143, Section 5.7].

deep sleep mode is negligible. Therefore, the two-scale optimization framework obtains a more accurate characterization of the end-to-end delay compared with the per-slot optimization.

6.3.2 Optimal Beamforming and Renewable Energy Exchanging in Each Slot

The arrival rates of renewable energy $\{E_m^{\mathrm{HAV}}[i]\}_m$ remain constant in frame i, and channel states in $\{\boldsymbol{h}_{m,n,k}[i]\}_{m,n}$ are independent and identically distributed over slots. Based on the principle of opportunistically minimizing an expectation [30], the optimal beamforming vectors and exchanged renewable energy variables to RHS of (6.19) with the constraints in (6.6) and (6.12)–(6.14) can be obtained by solving a per-slot optimization problem as

$$\min_{\mathcal{Y}_{2,k,i}} \sum_{m=1}^{M}\sum_{n=1}^{N_m}\Big(q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\Big)r_{m,n,k}[i]+V\sum_{m=1}^{M}F\Big(P_{m,k}^{\mathrm{BST}}[i]\Big) \tag{6.28a}$$

$$\text{s.t.} (6.6) \text{ and } (6.12)-(6.14). \tag{6.28b}$$

where $\mathcal{Y}_{2,k,i}=\{\boldsymbol{w}_{m,n,k}[i],\varpi_{m\to l,k}[i]\}_{m,l,n}$ is the set of optimization variables. Note that different values $\{q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i]\}_{m,n}$ can lead to different relations between GEE $\sum_{m=1}^{M}F(P_{m,k}^{\mathrm{BST}}[i])$ and data rates $\sum_{m=1}^{M}\sum_{n=1}^{N_m}(q_{m,n,1}^{\mathrm{U}}[i]-q_{m,n,1}^{\mathrm{A}}[i])r_{m,n,k}[i]$.

Solving the per-slot optimization problem (6.28) is challenging due to the non-convexity of rate-limit constraints in (6.12) and the proportional-rate constraints in (6.13). To handle the non-convex proportional-rate constraints in (6.13), we introduce an auxiliary variable $\theta_{m,n,k}[i]$ such that $r_{m,n,k}[i]=q_{m,n,k}^{\mathrm{A}}[i]\theta_k[i]$. Hence, we obtain the proportional-rate constraints in (6.13) as

$$\frac{\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,n,k}[i]}{\sqrt{\exp(q_{m,n,k}^{\mathrm{A}}[i]\theta_k[i])-1}}=\sqrt{I_{m,n,k}^{\mathrm{INTRA}}[i]+I_{m,n,k}^{\mathrm{INTER}}[i]+\sigma_{m,n}^2},n\in\mathcal{N}_m^{\mathrm{ACT}}[i],m=1,\ldots,M \tag{6.29}$$

$$\mathrm{Im}\Big(\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,n,k}[i]\Big)=0,n\in\mathcal{N}_m^{\mathrm{ACT}}[i],m=1,\ldots,M. \tag{6.30}$$

where the range of $\theta_{m,n,k}[i]$ is set as $[0,1]$ to guarantee the constraints in (6.12).

Replacing $r_{m,n,k}[i]$ by $q^{\mathrm{A}}_{m,n,k}[i]\theta_{m,n,k}[i]$ in the objective function (6.28a), we obtain

$$\begin{aligned}\mathcal{OBJ}_{m,k}[i] =& V(\gamma_b-\gamma_s)\left[P^{\mathrm{BST}}_{m,k}[i]+E^{\mathrm{LPE}}_{m,k}[i]-\frac{1}{T}E^{\mathrm{HAV}}_{m}[i]\right]^{+}\\ &+V\gamma_s\left[P^{\mathrm{BST}}_{m,k}[i]+E^{\mathrm{LPE}}_{m,k}[i]-\frac{1}{T}E^{\mathrm{HAV}}_{m}[i]\right]\\ &+\sum_{n\in\mathcal{N}^{\mathrm{ACT}}_m[i]}\left(q^{\mathrm{U}}_{m,n,1}[i]-q^{\mathrm{A}}_{m,n,1}[i]\right)q^{\mathrm{A}}_{m,n,k}[i]\theta_{m,n,k}[i]\end{aligned} \tag{6.31}$$

where $P^{\mathrm{BST}}_{m,k}[i]$ is defined in (6.5), and $E^{\mathrm{LPE}}_{m,k}[i]$ is defined in (6.7).

Relaxing the constraints in (6.29), we obtain a convex optimization problem as

$$\min_{\mathcal{Y}_{2,k,i}} \sum_{m=1}^{M}\mathcal{OBJ}_{m,k}[i] \tag{6.32a}$$

$$\text{s.t. } \varpi_{m\to l,k}[i]+\varpi_{l\to m,k}[i]=0, l\in\mathcal{N}_m, m=1,\ldots,M \tag{6.32b}$$

$$\frac{\boldsymbol{h}^{\mathrm{H}}_{m,n,k}[i]\boldsymbol{w}_{m,n,k}[i]}{\sqrt{\exp(q^{\mathrm{A}}_{m,n,k}[i]\theta_k[i])-1}}\geq\sqrt{I^{\mathrm{INTRA}}_{m,n,k}[i]+I^{\mathrm{INTER}}_{m,n,k}[i]+\sigma^2_{m,n}}, n\in\mathcal{N}^{\mathrm{ACT}}_m[i], m=1,\ldots,M \tag{6.32c}$$

$$\mathrm{Im}\left(\boldsymbol{h}^{\mathrm{H}}_{m,n,k}[i]\boldsymbol{w}_{m,n,k}[i]\right)=0, n\in\mathcal{N}^{\mathrm{ACT}}_m[i], m=1,\ldots,M \tag{6.32d}$$

$$\sum_{n\in\mathcal{N}^{\mathrm{ACT}}_m[i]}\|\boldsymbol{w}_{m,n,k}[i]\|^2\leq P^{\max}_m, m=1,\ldots,M. \tag{6.32e}$$

In slot k, the constraints in (6.32b)–(6.32e) constitute a convex hull of the constraints in (6.6), (6.14), (6.29) and (6.30), when the values of $\theta_k[i]$ and $\{a_{m,n}[i]\}_{m,n}$ are fixed. Therefore, the objective value of (6.32) is lower than that of (6.28). Since the constraints in (6.32c) are active (see Appendix D.3 for a detailed proof), we conclude that the objective value of (6.32) is equal to that of (6.28). Motivated by Proposition 2 of [144], the optimal $\theta^*_k[i]$ can be obtained via a one-dimensional search method. Therefore, the optimization problem (6.28) can be optimally solved.

Based on (6.26), the scheduled UE indicators are updated in each frame. Performing a one-dimensional search and solving the optimization problem (6.32), the beamforming vectors and exchanged renewable energy variables are updated in each slot. Therefore, we summarize the two time-scale UE scheduling, beamforming, and energy trading (TSUBE) algorithm in Algorithm 8.

Algorithm 8 TSUBE Algorithm

1: In frame i, the core network estimates the harvested renewable energy as $\{E_m^{\mathrm{HAV}}[i]\}_m$
2: In slot k of frame i, the core network obtains backlogs of access queues $\{q_{m,n,k}^{\mathrm{A}}[i]\}_{m,n}$ and processing queues $\{q_{m,n,k}^{\mathrm{U}}[i]\}_{m,n}$
3: At the start of frame i, the core network updates the scheduled UE indicators $\{a_{m,n}[i]\}_{m,n}$ via (6.26)
4: **repeat**
5: In slot k of frame i, the core network estimates the channel-coefficient vectors $\{\boldsymbol{h}_{m,n,k}[i]\}_{m,n}$
6: In slot k of frame i, the core network solves the optimization problem (6.32) based on $\{\boldsymbol{h}_{m,n,k}[i]\}_{m,n}$ and $\{a_{m,n}[i]\}_{m,n}$
7: In slot k of frame i, the core network performs one dimensional search for the optimal $\theta_k[i]$
8: **until** The optimal $\theta_k^*[i]$ is obtained
9: At the end of slot k, the core network updates the access queues and processing queues according to (6.9) and (6.10)

6.3.3 Complexity Analysis

For notational brevity, we assume that $N = N_m$, $m = 1, \ldots, M$. The number of UEs in the multi-cell SGPCS is MN. The complexity of scheduling UEs via (6.26) is calculated as MN at the start of each frame.

The major complexity of the TSUBE Algorithm in each slot lies in the iteration loop in lines 4–8. Hereinafter, we focus on analyzing the computational complexity of the iteration loop. Moreover, the computational complexity of the iteration loop in lines 4–8 comes from solving (6.32) via the interior-point method and the one-dimensional search. Hence, we evaluate the worst-case computational complexity of solving (6.32) via the interior-point method and multiply it by the number of points in the one-dimensional search to obtain the computational complexity of the iteration loop in lines 4–8. We observe that the optimization problem (6.32) is second-order conic programming. In the optimization problem (6.32), the number of second-order cones with dimension dMN is M, and the number of second-order cones having dimension dN is M. The number of linear constraints is $c_2 = MN + \frac{1}{2}\sum_{m=1}^{M} |\mathcal{N}_m|$. The number of variables is $c_3 = dMN + \frac{1}{2}\sum_{m=1}^{M} |\mathcal{N}_m|$ in the optimization problem (6.32). According to [141, 142], an ϵ-accurate solution to (6.32) requires $c_4 = O(\log \epsilon^{-1}\sqrt{MN + 2M + \frac{1}{2}\sum_{m=1}^{M} |\mathcal{N}_m|})$ iterations, and the computational complexity in each iteration is $O((c_3+1)c_3c_2 + c_3c_5 + c_3^3)$ where $c_5 = MN(M+1)$. Therefore, the computational complexity of the iteration loop in lines 4–8 is $O(T_2^{\max}c_4c_3\big((c_3+1)c_2 + c_5 + c_3^2\big))$ where $T_2^{\max}$ is the number of points for a one-dimensional search.

6.4 Numerical Results

In this section, we present simulation results to evaluate the proposed TSUBE algorithm. The pathloss of the (m,n)th access link is calculated as

$$\omega_{m,n} = 17.3 + 38.3 \log_{10} \kappa_{m,n} + 24.9 \log_{10} f_c \text{ dB} \tag{6.33}$$

where $\kappa_{m,n}$ is the link distance of the (m,n)th access link, and carrier frequency $f_c = 2.1$ GHz.

We consider a two-BST SGPCS, where each BST is associated with three UEs and is equipped with six antennas. The multi-cell SGPCS operates in two time scales, where each frame consists of five slots. The inter-BST distance is set as 400 meters. The UEs are deployed at the middle point between the two BSTs such that the worst-case interference is considered. The AWGN power is set as $1 \times 10^{-10.7}$ mW. The power amplifier efficiency, maximum transmit power, baseband processing power of BSTs are, respectively, set as $\eta = 0.8$, $P_m^{\max} = 400$ mW and $P_m^{\text{SP}} = 100$ mW. The efficiency of local power lines is set as $\chi_{m\to l} = 0.8$. Unless otherwise specified, the purchasing and selling prices of a unit energy are respectively set as $\gamma_b = 1.6 \times 10^{-9}$ cents/slot/mW and $\gamma_s = 0.6 \times 10^{-9}$ cents/slot/mW. The average arrival rate $\bar{\nu}_{m,n}$ and constant processing rate $\tilde{s}_{m,n}$ are, respectively, set as 2.1 nats/slot/Hz and 8 nats/slot/Hz. The arrival rates of renewable energy for the first BST and the second BST are respectively set as 300 mW and 200 mW. The duration of a slot is set as 1 ms. The number of slots in each frame is set as 5. The value of the control parameter V is empirically tuned to demonstrate the tradeoff between the end-to-end delay and GEE. We consider two benchmarks, namely without local power exchanging (WOLPE) algorithm and ZFBF algorithm.

Figures 6.2 and 6.3 show the moving-average annualized GEE and moving-average end-to-end delay of UEs when the moving-average window is set as 10. We observe that the moving-average GEE of the proposed TSUBE algorithm, the WOLPE algorithm, and the ZFBF algorithm converge within $1,000$ slots. The moving-average end-to-end delay of UEs becomes stable after 400 slots. Note that the end-to-end delay is calculated according to Little's law for the two cascading queues. When the control parameter V is set as 0.01, 0.1 and 1, the GEE of the proposed TSUBE algorithm are, respectively, 3.15%, 7.85% and 8.85% lower than that of the WOLPE algorithm,

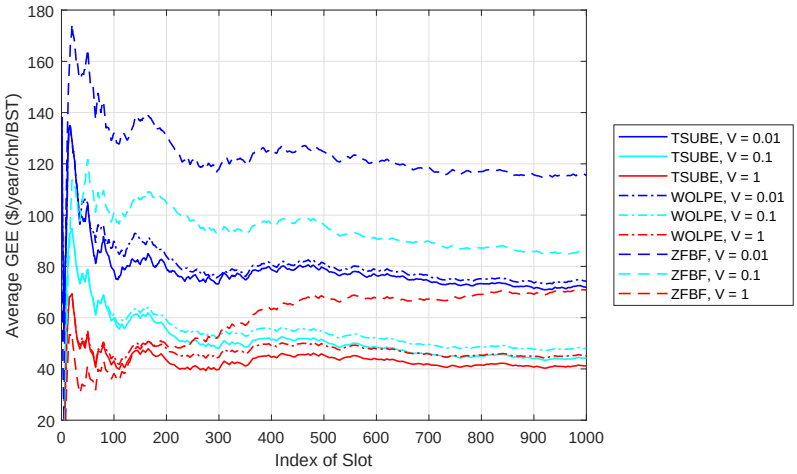


Figure 6.2: The moving-average GEE with window size = 10.

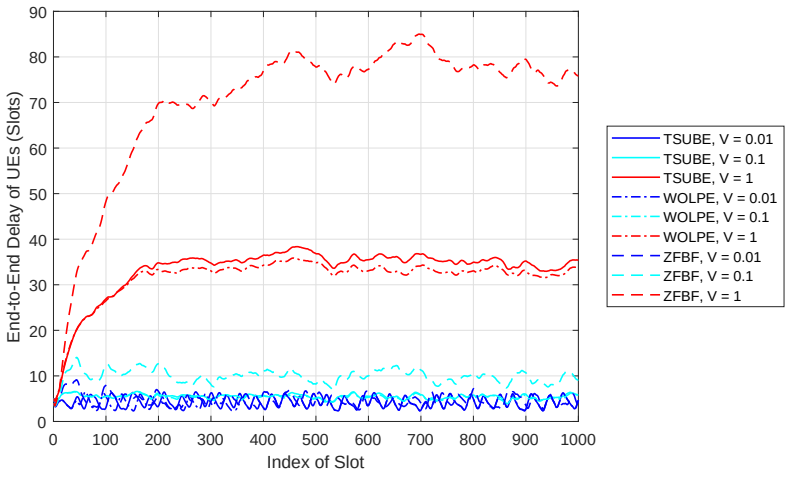
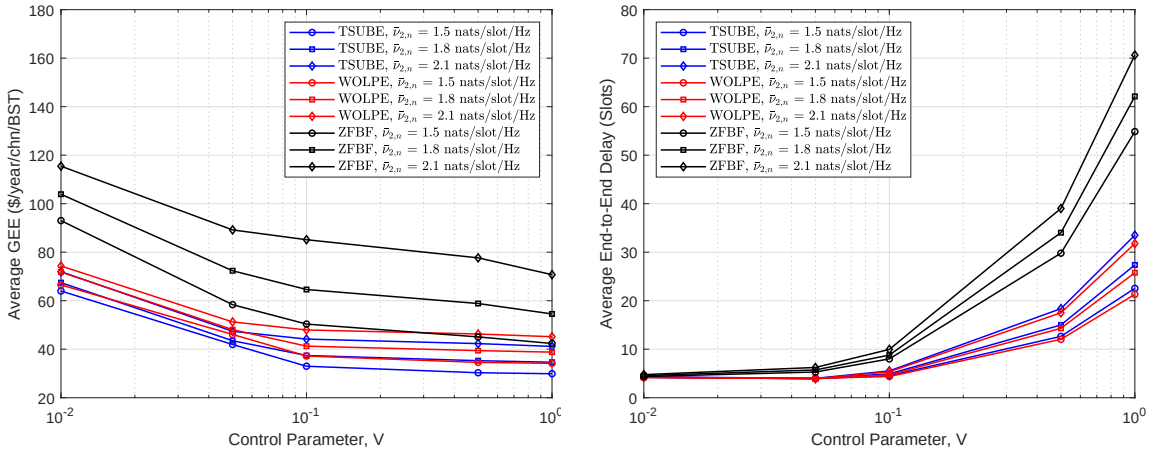


Figure 6.3: The moving-average end-to-end delay with window size = 10.

and 37.67%, 48.12% and 41.82% lower than that of the ZFBF algorithm. This observation is due to the facts that 1) the proposed TSUBE algorithm intelligently makes decisions on whether to purchase grid energy or exchange renewable energy to avoid redundant grid energy transactions, and 2) the WOLPE algorithm introduces redundant purchasing/selling of grid energy when the multi-cell SGPCS has insufficient/surplus renewable energy; 3) the ZFBF algorithm prefers to mitigate interference.

Figure 6.4 reveals the tradeoff between the average GEE and the end-to-end delay of UEs under different average arrival rates of UEs in the second BST (i.e., $\bar{\nu}_{2,n}$). We observe that increasing the control parameter induces a decrease in the GEE (as shown in Fig. 6.4(a)) and an increase in the



(a) GEE v.s. control parameter

(b) End-to-end delay v.s. control parameter

Figure 6.4: The tradeoff between the average GEE and average end-to-end delay of UEs.

end-to-end delay of UEs (as shown in Fig. 6.4(b)). Therefore, the proposed TSUBE algorithm, the WOLPE algorithm, and the ZFBF algorithm provide the operator with flexibility in controlling the GEE while maintaining a satisfactory level of communication QoS. Moreover, we also observe that the proposed TSUBE algorithm outperforms the WOLPE algorithm and ZFBF algorithm in terms of the GEE. For example, when $V = 0.1$ and $\bar{\nu}_{2,n} = 1.5$ nats/slot/Hz, the TSUBE algorithm achieves 11.32% lower GEE than the WOLPE algorithm by sacrificing 3.86% the end-to-end delay of UEs. When $V = 1$ and $\bar{\nu}_{2,n} = 1.5$ nats/slot/Hz, the TSUBE algorithm achieves 12.51% lower GEE than the WOLPE algorithm by sacrificing 5.45% the end-to-end delay of UEs. Moreover, the TSUBE algorithm outperforms the ZFBF algorithm in terms of GEE and end-to-end delay. For example, when $V = 0.1$ and $\bar{\nu}_{2,n} = 1.8$ nats/slot/Hz, the TSUBE algorithm achieves 35.08% lower GEE and 7.41% lower end-to-end delay than the ZFBF algorithm. This observation is because the local power exchanging introduces a new dimension of freedom to reduce the GEE when the multi-cell SGPCS has a more stringent energy demand. When more renewable energy is traded to reduce the GEE, the end-to-end delay of UEs increases.

Figure 6.5 shows that the GEE increases with the purchasing price of unit energy under various control parameters. More specifically, the gap of GEE between the proposed TSUBE algorithm and the WOLPE algorithm increases with the purchasing price α_b. The reason is as follows.

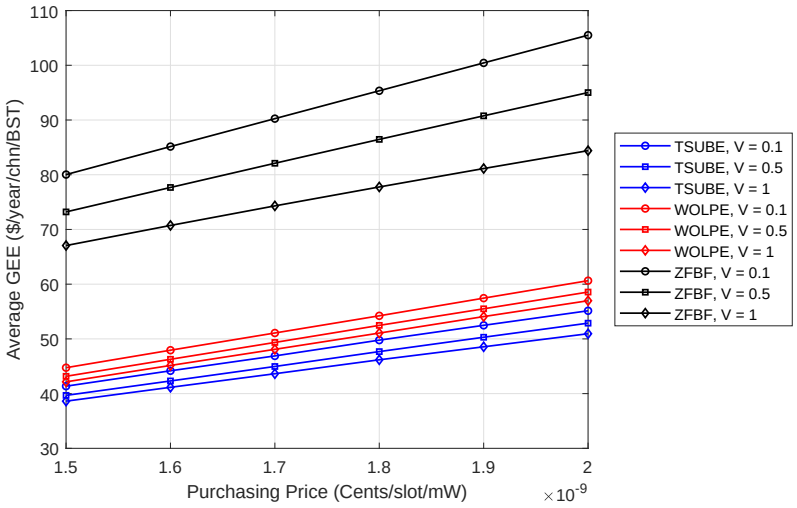


Figure 6.5: The average GEE versus the purchasing price α_b.

A higher purchasing price α_b motivates the BSTs to exchange renewable energy via the local power line such that the GEE of the TSUBE algorithm increases slower than that of the WOLPE algorithm. Compared with the WOLPE algorithm, the proposed TSUBE algorithm can reduce the GEE by 9.07%, 9.71%, and 10.58% when the control parameters are respectively set as 0.1, 0.5 and 1. In other words, a higher control parameter induces a more effective GEE reduction of the TSUBE algorithm than the WOLPE algorithm. Besides, we also observe that the GEE of the TSUBE algorithm is lower than that of the ZFBF algorithm. This observation is because the ZFBF algorithm requires the BSTs to consume more grid energy than the TSUBE algorithm to guarantee the stability of multi-cell SGPCS.

Figure 6.6 illustrates the GEE as a function of the arrival rate of renewable energy for the first BST under different control parameters. Increasing the arrival rate of renewable energy of the first BST from 250 mW to 500 mW, we observe that the GEE of the proposed TSUBE algorithm, WOLPE algorithm, and ZFBF algorithm decrease. Moreover, by increasing the arrival rate of renewable energy, we also observe that the gaps of GEE between the proposed TSUBE algorithm and WOLPE algorithm increase from 2.07 $/year/chn/BST, 2.18 $/year/chn/BST and 2.22 $/year/chn/BST to 8.25 $/year/chn/BST, 8.45 $/year/chn/BST and 8.40 $/year/chn/BST when the control parameters are respectively set as $V = 0.1$, $V = 0.5$ and $V = 1$. These observations demonstrate that the proposed TSUBE algorithm outperforms the WOLPE algorithm.

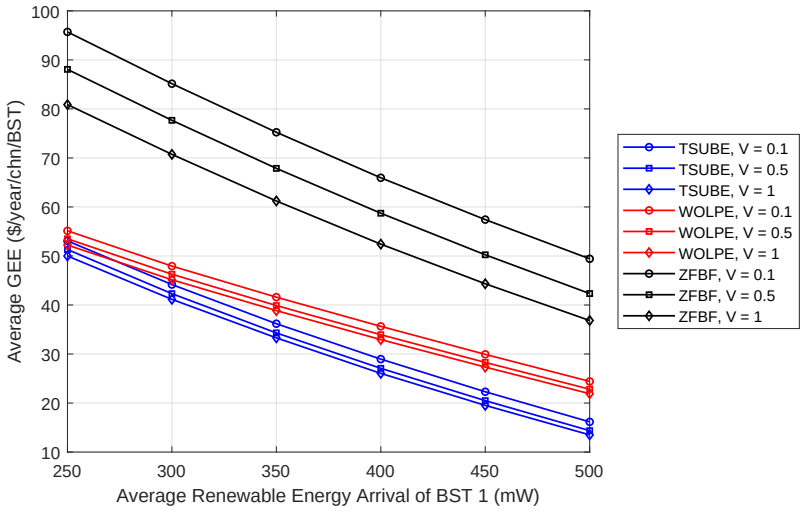


Figure 6.6: The average GEE versus the average renewable energy arrival rate of the first BST.

Moreover, the proposed TSUBE algorithm can reduce the GEE by 72.99% when the control parameter and the arrival rate of renewable energy are $V = 1$ and 500 mW. Since the arrival rate of renewable energy at the second BST is 200 mW, we conclude that a more asymmetric arrival rate of renewable energy induces a more frequent local energy exchange under symmetric data rate. Therefore, the gaps in GEE between the proposed TSUBE algorithm and the WOLPE algorithm can increase with the arrival rate of renewable energy of the first BST. Figure 6.6 also shows that the proposed TSUBE algorithm outperforms the ZFBF algorithm when the arrival rate of renewable energy increases. The gaps of GEE are as large as 42.64 $/year/chn/BST, 36.75 $/year/chn/BST and 30.89 $/year/chn/BST when the control parameters are respectively 0.1, 0.5 and 1. This observation indicates that the wireless operator can choose the ZFBF algorithm for low computational complexity at the expense of GEE.

Figure 6.7 shows that the GEE increases with the average arrival rate of UEs in the second BST under different control parameters. Fig. 6.7 also confirms that the proposed TSUBE algorithm outperforms the WOLPE and ZFBF algorithms under different control parameters.

6.5 Summary

We have investigated the TAEGEE minimization problem with proportional-rate constraints in multi-cell SGPCSs and proposed a TSUBE algorithm for multi-cell SGPCSs to allocate the

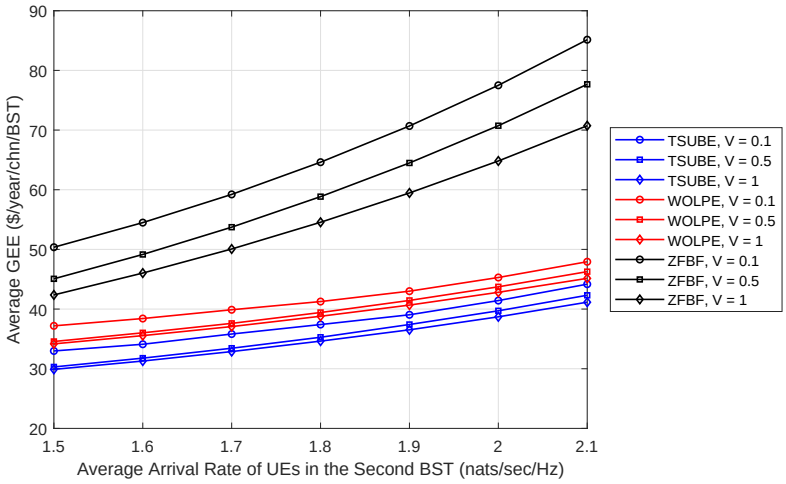


Figure 6.7: The average GEE versus the average arrival rate $\bar{\nu}_{2,n}$.

scheduled UE indicators, beamforming vectors jointly and exchanged renewable energy variables. We have leveraged the Lyapunov learning method to decouple the beamforming vectors and scheduled UE indicators allocation. The scheduled UE indicators are optimally allocated at each frame in order to avoid redundant scheduling/unscheduling UEs based on the proposed TSUBE algorithm. The beamforming vectors and exchanged renewable energy variables are optimally allocated to minimize the per-slot subproblems. When the control parameter approaches infinity, the proposed TSUBE algorithm asymptotically achieves the optimal GEE. The tradeoff between the GEE and end-to-end delay of UEs has been theoretically established when three sets of resources (i.e., scheduled UE indicators, beamforming vectors, and exchanged renewable energy variables) are jointly allocated. Numerical results have been presented to demonstrate that the TSUBE algorithm outperforms the WOLPE and ZFBF algorithms in terms of GEE. Therefore, the joint allocation of three-dimensional resources (scheduled UE indicators, beamforming vectors, and exchanged renewable energy variables) helps to reduce GEE and yields a better tradeoff between GEE and end-to-end delay of UEs compared with the joint allocation of two-dimensional resources (scheduled UE indicators and beamforming vectors).

Chapter 7

Conclusions and Future Works

This chapter concludes the thesis with some comments and discusses several extensions in the future.

7.1 Concluding Remarks

In this thesis, we have investigated the convergence behaviors and applications of machine learning algorithms in SGPCSs. In particular, the contributions are summarized as follows.

- Motivated by the applications in energy planning of SGPCSs, we have investigated the issues in federated learning algorithms. In the presence of faulty UEs, the classical federated learning algorithms (such as gradient descent and stochastic gradient descent algorithms) may diverge. As a remedy, we have developed an FRPG algorithm by adapting Nesterov's acceleration [118, 120] and stochastic approximation for fault-resilient federated learning in Chapter 3. To further reduce communication overhead, we have also developed an LFRPG algorithm where the parameter server periodically communicates with UEs. We have proved that LFRPG has a lower communication overhead than FRPG. We have established the convergence rates for the proposed FRPG and LFRPG algorithms, which are challenging to analyze when faulty UEs exist. Our theoretical results demonstrate that the FRPG and LFRPG algorithms require lower communication overheads than existing fault-resilient federated learning algorithms.

- When the agents are spatially dispersed, decentralized machine learning algorithms are required to finish some tasks in SGPCSs, such as the collaborative spectrum sensing and collaborative spectrum sharing tasks. Therefore, we have investigated the issues that are related to decentralized Q-learning algorithms. We have derived an equivalent form of the

multi-user Bellman equation in Chapter 4, based on which the local vectors are updated. To gain control over the gradient bias and variance present in each agent's local updates, we have performed a unifying finite-sample analysis of collaborative multi-agent Q-learning with LFA in a fully decentralized setting, by studying a multi-step Lyapunov function carefully. When a decaying stepsize c/k is used, we have shown that the LFA based decentralized Q-learning algorithm converges to the fixed point of Bellman's optimality equation at rate $\mathcal{O}(1/k)$ under an appropriate condition on the joint behavior policy. While gaining scalability, privacy, and parallel computation to deal with large state and action spaces, the linear-approximate decentralized Q-learning converges as fast as the tabular Q-learning [81]. Besides, our obtained convergence rate improves upon that of centralized Q-learning with LFA reported in [87].

- Considering the uncertainties of ESI and CSI, we have formulated the TAEGEE minimization problem via Lyapunov learning in Chapter 5. Different from using a log-concave data rate, we have considered the joint effects of packet failure and data rate of each UE while designing beamforming algorithms. Therefore, the investigated optimization problem is a cross-layer one. Using Lyapunov learning, we have reformulated the TAEGEE minimization problem to per-slot subproblems. Moreover, each per-slot subproblem is non-convex and challenging to handle. Two suboptimal beamforming algorithms have been proposed based on SABF and ZFBF techniques. The convergence properties have been established, and the corresponding computational complexities have been analyzed. By tuning the introduced control parameter, the proposed algorithms allow the wireless operator to trade the GEE for the access delay of UEs.

- We have investigated the TAEGEE minimization problem in multi-cell SGPCSs via the joint design of scheduled UE indicators, beamforming vectors, and exchanged renewable-energy variables in Chapter 6. Beamforming and energy exchanging are physical-layer functions, and user scheduling is a link-layer function. Hence, the investigated TAEGEE minimization problem is a cross-layer problem. After transforming the TAEGEE minimization problem into minimizing the upper bound of drift-plus-penalty function, we have decoupled beamforming and energy exchanging from user scheduling. Hence, the TSUBE algorithm has

been proposed for beamforming and energy exchanging per slot, update the scheduled UE indicators per frame. We have theoretically proved that the minimizer to the upper bound of drift-plus-penalty function can be obtained via the proposed TSUBE algorithm. Based on Lyapunov learning, we have revealed that the obtained minimizer can achieve the optimal grid-energy expenditure via tuning a control parameter at the expense of end-to-end delay.

7.2 Future Works

The design of machine learning algorithms and their applications in wireless communications are still hotspots of research communities. We briefly review several open problems, which are extensions to the thesis.

- **Robust decentralized learning algorithms with Byzantine adversaries:** The proposed FRPG and LFRPG algorithms in Chapter 3 still require a parameter server to collect the local parameters of UEs. When the parameter server stops working, the FRPG and LFRPG algorithms will fail. Therefore, decentralized robust learning algorithms are preferred. However, faulty UEs will have more severe effects by injecting multiple falsified local parameters to their neighbors such that the introduced penalty functions in FRPG and LFRPG algorithms cannot mitigate the negative effects of faulty UEs. Based on the recent development of the adversary detection method in [48], we are motivated to investigate decentralized methods to detect the Byzantine adversaries. The statistical characteristics of byzantine local gradients need to be revealed. Moreover, the convergence rates for (smooth) convex and non-convex loss functions need to be investigated for the decentralized fault-resilient algorithms.

- **Finite-sample analysis for linear-approximate deep reinforcement learning:** By approximating the state-value function, action-value function, and policy gradient via deep neural networks, deep reinforcement learning has many successful industrial applications (e.g., AlphaGo and Atari 2600 games [39]). While the proposed finite-sample analysis in Chapter 4 is for linear-approximate decentralized Q-learning, the finite-sample analysis for (decentralized) deep reinforcement learning is unknown. Moreover, the finite-sample

analysis can also estimate the required data samples of an algorithm to achieve a certain accuracy. Therefore, a finite-sample analysis of (decentralized) deep reinforcement learning deserves a future investigation.

- **Lyapunov learning with time-correlated data samples:** Using Lyapunov learning, the proposed algorithms in Chapters 5 and 6 are based on the assumption that the random sources are independent and identically distributed over different slots. When the random sources are non-independent or non-identically distributed, the tradeoff between the grid-energy expenditure and delay of UEs needs to be quantified. When the tradeoff does not exist, algorithms are required to stabilize queues in the systems. Motivated by these facts, the investigation of Lyapunov learning with non-independent or non-identically distributed random sources is another interesting research direction.

Bibliography

[1] Ericsson Inc., *Mobile data traffic outlook*, June 2019. → pages 1

[2] J. Liu, M. Sheng, and J. Li, "Improving network capacity scaling law in ultra-dense small cell networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6218–6230, Sept. 2018. → pages 1

[3] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 524–540, Fourth quarter 2011. → pages 1

[4] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, "The global footprint of mobile communications: The ecological and economic perspective," *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 55–62, Aug. 2011. → pages 1

[5] Q. Wu, G. Y. Li, W. Chen, D. W. K. Ng, and R. Schober, "An overview of sustainable green 5G networks," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 72–80, Aug. 2017. → pages

[6] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, June 2019. → pages 1

[7] J. Xu, L. Duan, and R. Zhang, "Cost-aware green cellular networks with energy and communication cooperation," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 257–263, May 2015. → pages 1

[8] Ericsson Inc., *Sustainable Energy Use in Mobile Communications*, Aug. 2007. → pages

[9] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implica-

tions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, Third quarter 2011. → pages 1, 2

[10] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015. → pages 1

[11] Huawei, "Sustainability report 2017." [Online]. Available: https://www-file.huawei.com/-/media/corporate/pdf/sustainability/2017-huawei-sustainability-report-en.pdf?la=en → pages 1

[12] Y. Cui, V. K. N. Lau, and F. Zhang, "Grid power-delay tradeoff for energy harvesting wireless communication systems with finite renewable energy storage," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1651–1666, Aug. 2015. → pages 2, 11, 12

[13] I. Ahmed, A. Ikhlef, D. W. K. Ng, and R. Schober, "Power allocation for an energy harvesting transmitter with hybrid energy sources," *IEEE Trans. Commun.*, vol. 12, no. 12, pp. 6255–6267, Dec. 2013. → pages 11, 12

[14] Y. Mao, J. Zhang, and K. B. Letaief, "A Lyapunov optimization approach for green cellular networks with hybrid energy supplies," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2463–2477, Dec. 2015. → pages 1, 2, 11

[15] F. Yuan, Q. T. Zhang, S. Jin, and H. Zhu, "Optimal harvest-use-store strategy for energy harvesting wireless systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 698–710, Feb. 2015. → pages 2, 11

[16] X. Kang, Y. K. Chia, C. K. Ho, and S. Sun, "Cost minimization for fading channels with energy harvesting and conventional energy," *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4586–4598, Aug. 2014. → pages 2, 11

[17] D. Zhai, M. Sheng, X. Wang, and Y. Li, "Leakage-aware dynamic resource allocation in hybrid energy powered cellular networks," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4591–4603, Nov. 2015. → pages 2, 11, 12

[18] S. Bu, F. R. Yu, Y. Cai, and X. P. Liu, "When the smart grid meets energy-efficient communications: Green wireless cellular networks powered by the smart grid," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 3014–3024, Aug. 2012. → pages 2, 12

[19] J. Xu and R. Zhang, "Cooperative energy trading in CoMP systems powered by smart grids," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2142–2153, Apr. 2016. → pages 3, 12

[20] Y. Dong, M. J. Hossain, J. Cheng, and V. C. M. Leung, "Dynamic cross-layer beamforming in hybrid powered communication systems with harvest-use-trade strategy," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8011–8025, Dec. 2017. → pages 3, 64, 82, 84

[21] Zhi-Quan Luo and Wei Yu, "An introduction to convex optimization for communications and signal processing," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1426–1438, 2006. → pages 3

[22] X. Huang, T. Han, and N. Ansari, "Smart grid enabled mobile networks: Jointly optimizing BS operation and power distribution," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1832–1845, June 2017. → pages 3, 12

[23] M. Grant and S. P. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014. → pages 3

[24] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Glob. Optim.*, vol. 47, no. 1, pp. 29–51, 2009. → pages 3, 155

[25] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010. → pages 3

[26] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951. → pages 3

[27] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014. → pages 3, 4, 7

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009. → pages 4

[29] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 2018. → pages 4, 5, 9, 10, 19, 45, 46

[30] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, USA: Morgan & Claypool, 2010. → pages 3, 4, 5, 22, 23, 66, 67, 91, 92, 151, 152, 158, 159

[31] B. Li, M. Ma, and G. B. Giannakis, "On the convergence of SARAH and beyond," *arXiv preprint arXiv:1906.02351*, June 2019. → pages 4, 6, 17

[32] B. Li, L. Wang, and G. B. Giannakis, "Almost tune-free variance reduction," *arXiv preprint arXiv:1908.09345*, Aug. 2019. → pages

[33] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, Mar. 2015. → pages 4, 6, 17, 27, 29

[34] Y. Dong, J. Cheng, M. J. Hossain, and V. C. M. Leung, "Secure distributed on-device learning networks with Byzantine adversaries," *IEEE Netw.*, vol. 33, no. 6, pp. 180–187, 2019. → pages 40

[35] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv preprint arXiv:1909.07972*, Sept. 2019. → pages 4, 6, 17

[36] E. Balevi and J. G. Andrews, "One-bit OFDM receivers via deep learning," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4326–4336, June 2019. → pages 4

[37] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10 760–10 772, Nov. 2018. → pages 4

[38] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming.* Athena Scientific Belmont, MA, 1996, vol. 5. → pages 5, 9, 19, 21, 45, 47

[39] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. → pages 5, 10, 11, 47, 103

[40] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, Oct. 2016. → pages 5, 10

[41] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. International Conference on Learning Representations*, San Juan, Puerto Rico, May 2016, pp. 1–10. → pages 5, 10, 11

[42] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 44:1–44:25, Dec. 2017. → pages 7, 15, 18, 27, 41

[43] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, USA, Dec. 2017, pp. 119–129. → pages 7, 8, 15, 18, 19, 25, 26, 27, 41

[44] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. International Conference on Machine Learning*, Stockholmsmässan, Stockholm, Sweden, July 2018, pp. 5650–5659. → pages 7, 8, 18, 40

[45] ——, "Defending against saddle point attack in Byzantine-robust distributed learning," in *Proc. International Conference on Machine Learning*, vol. 97, Long Beach, California, USA, June 2019, pp. 7074–7084. → pages 8

[46] L. Su and J. Xu, "Securing distributed gradient descent in high dimensional statistical learning," in *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 1, Mar. 2019, pp. 12:1–12:41. → pages 7

[47] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proc. International Conference on Machine Learning*, Stockholmsmässan, Stockholm, Sweden, July 2018, pp. 3521–3530. → pages 8, 18, 19, 26

[48] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Proc. Advances in Neural Information Processing Systems*, Montreal, CA, Dec. 2018, pp. 4614–4624. → pages 8, 26, 103

[49] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proc. International Conference on Machine Learning (ICML)*, Long Beach, California, USA, June 2019, pp. 6893–6901. → pages 8, 26

[50] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "DRACO: Byzantine-resilient distributed training via redundant gradients," in *Proc. International Conference on Machine Learning*, Stockholmsmässan, Stockholm, Sweden, July 2018, pp. 903–912. → pages 8, 19, 26

[51] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, Jan. 2019, pp. 1544–1551. → pages 8, 15, 18, 26, 27, 28, 35, 41

[52] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. Advances in Neural Information Processing Systems*, Palais des Congrès de Montréal, Montréal, Dec. 2014, pp. 19–27. → pages 9

[53] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *J. American Statistical Association*, vol. 114, no. 526, pp. 668–681, 2019. → pages 9

[54] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Advances in Neural Information Processing Systems*, Montreal, CA, Dec. 2018, pp. 5050–5060. → pages 9

[55] J. Sun, T. Chen, G. B. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2019, pp. 3365–3375. → pages 9

[56] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. International Conference on Learning Representations*, Addis Ababa, Ethiopia, Apr. 2019. → pages 9

[57] 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 *Proc. AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, Jan. 2019, pp. 5693–5700. → pages

[58] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *Proc. International Conference on Machine Learning*, vol. 97, Long Beach, California, USA, June 2019, pp. 7184–7193. → pages 9

[59] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. AAAI Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, Madison, Wisconsin, USA, Jul. 1998, pp. 746–752. → pages 10

[60] A. Nair *et al.*, "Massively parallel methods for deep reinforcement learning," in *Deep learning workshop, International Conference on Machine Learning*, Lille, France, Jul. 2015. → pages

[61] S. Kar, J. M. F. Moura, and H. V. Poor, "QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1848–1862, Jan. 2013. → pages 11, 46, 47, 48

[62] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. International Conference on Machine Learning*, Stockholmsmässan, Stockholm, Sweden, Jul. 2018, pp. 9340–9371. → pages 10, 11, 50

[63] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Quebec, CA, Dec. 2018, pp. 9649–9660. → pages 10, 11, 50

[64] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. International Conference on Machine Learning*, New Brunswick, New Jersey, USA, Jun. 1994, pp. 157–163. → pages 10

[65] G. Arslan and S. Yuksel, "Decentralized Q-learning for stochastic teams and games," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1545–1558, Apr. 2017. → pages

[66] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Machine Learning Research*, vol. 4, pp. 1039–1069, Dec. 2003. → pages

[67] J. Perolat, B. Scherrer, B. Piot, and O. Pietquin, "Approximate dynamic programming for two-player zero-sum Markov games," in *Proc. International Conference on Machine Learning*, Lille, France, Jul. 2015, pp. 1321–1329. → pages 10

[68] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, California, USA, Dec. 2017, pp. 6379–6390. → pages 10

[69] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Proc. Conference on Theoretical Aspects of Rationality and Knowledge*, San Francisco, California, USA, Mar. 1996, pp. 195–210. → pages 10

[70] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored MDPs," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, British Columbia, CA, Dec. 2002, pp. 1523–1530. → pages 10

[71] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of markov decision processes," in *Proc. Conference on Uncertainty in Artificial Intelligence*, Stanford, California, USA, Jul. 2000, pp. 32–37. → pages 10

[72] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, Aug. 2013. [Online]. Available: https://doi.org/10.1177/0278364913495721 → pages 10

[73] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robotics Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004. → pages 10

[74] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3–4, pp. 279–292, May 1992. → pages 11, 20, 46

[75] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," in *Proc. Advances in Neural Information Processing Systems*, Denver, Colorado, USA, Dec. 1994, pp. 703–710. → pages

[76] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Machine Learning*, vol. 16, no. 3, pp. 185–202, Sept. 1994. → pages 20, 46, 47

[77] C. Szepesvári, "The asymptotic convergence-rate of Q-learning," in *Proc. Advances in Neural Information Processing Systems*, Denver, Colorado, USA, Dec. 1998, pp. 1064–1070. → pages 11

[78] M. J. Kearns and S. P. Singh, "Finite-sample convergence rates for Q-learning and indirect algorithms," in *Proc. Advances in Neural Information Processing Systems*, Denver, Colorado, USA, Dec. 1999, pp. 996–1002. → pages 11

[79] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *J. Machine Learning Research*, vol. 5, pp. 1–25, Dec. 2003. → pages

[80] C. L. Beck and R. Srikant, "Error bounds for constant step-size Q-learning," *Systems & Control Letters*, vol. 61, no. 12, pp. 1203–1208, Dec. 2012. → pages 20, 47

[81] M. J. Wainwright, "Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for Q-learning," *arXiv preprint arXiv:1905.06265*, May 2019. → pages 11, 20, 46, 47, 51, 56, 102

[82] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proc. International Conference on Machine Learning*, Helsinki, Finland, Jul. 2008, pp. 664–671. → pages 11, 21, 46, 47, 50, 51

[83] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, "Finite sample analyses for TD(0) with function approximation," in *Proc. AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, Feb. 2018, pp. 6144–6153. → pages 11, 47, 51

[84] R. Srikant and L. Ying, "Finite-time error bounds for linear stochastic approximation andtd learning," in *Proc. Conference on Learning Theory*, Phoenix, Arizona, USA, Jun. 2019, pp. 2803–2830. → pages 57

[85] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," in *Proc. Conference on Learning Theory*, 2018, pp. 1691–1692. → pages

[86] G. Wang, B. Li, and G. B. Giannakis, "A multistep Lyapunov approach for finite-time analysis of biased stochastic approximation," *arXiv preprint arXiv:1909.04299*, Nov. 2019. → pages 50, 51, 57, 137

[87] Z. Chen, S. Zhang, T. T. Doan, S. T. Maguluri, and J.-P. Clarke, "Performance of Q-learning with linear function approximation: Stability and finite-time analysis," *arXiv preprint arXiv:1905.11425*, Oct. 2019. → pages 11, 21, 22, 46, 47, 50, 51, 56, 57, 102, 137

[88] G. Qu and A. Wierman, "Finite-time analysis of asynchronous stochastic approximation and Q-learning," *arXiv preprint arXiv:2002.00260*, Feb. 2020. → pages

[89] S. Zou, T. Xu, and Y. Liang, "Finite-sample analysis for SARSA and Q-Learning with linear function approximation," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, British Columbia, CA, Dec. 2019, pp. 8665–8675. → pages 49, 50

[90] P. Xu and Q. Gu, "A finite-time analysis of Q-learning with neural network function approximation," *arXiv preprint arXiv:1912.04511*, Dec. 2019. → pages 11

[91] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning," in *Proc. International Conference on Machine Learning*, Long Beach, California, USA, Jun. 2019, pp. 1626–1635. → pages 11, 50

[92] J. Sun, G. Wang, G. B. Giannakis, Q. Yang, and Z. Yang, "Finite-sample analysis of decentralized temporal-difference learning with linear function approximation," in *Proc. International Conference on Artificial Intelligence and Statistics*, Palermo, Italy, Jun. 2020. → pages 11, 22, 50, 57

[93] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 3872–3882, Aug. 2013. → pages 11

[94] J. Leithon, T. J. Lim, and S. Sun, "Online energy management strategies for base stations powered by the smart grid," in *Proc. IEEE SmartGridComm*, Oct. 2013, pp. 199–204. → pages 12

[95] M. Sheng, D. Zhai, X. Wang, Y. Li, Y. Shi, and J. Li, "Intelligent energy and traffic coordination for green cellular networks with hybrid energy supply," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1631–1646, Feb. 2017. → pages 11, 12

[96] K. Peng, H. Huang, S. Wan, and V. C. M. Leung, "End-edge-cloud collaborative computation offloading for multiple mobile users in heterogeneous edge-server environment," *Wireless Netw.*, to be published, 2020. → pages 11

[97] S. Yin, Z. Qu, and S. Li, "Achievable throughput optimization in energy harvesting cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 407–422, Mar. 2015. → pages 11, 12

[98] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2018. → pages 11, 13

[99] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in OFDMA systems with hybrid energy harvesting base station," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3412–3427, July 2013. → pages 11, 12

[100] F. Guo, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Joint optimization of caching and

association in energy-harvesting-powered small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6469–6480, July 2018. → pages 11, 12

[101] C. Hu, J. Gong, X. Wang, S. Zhou, and Z. Niu, "Optimal green energy utilization in MIMO systems with hybrid energy supplies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3675–3688, Aug. 2015. → pages 12

[102] W. Lee, L. Xiang, R. Schober, and V. W. S. Wong, "Direct electricity trading in smart grid: A coalitional game analysis," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 7, pp. 1398–1411, July 2014. → pages 12

[103] J. Xu and R. Zhang, "CoMP meets smart grid: A new communication and energy cooperation paradigm," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2476–2488, June 2015. → pages 12

[104] G. Wang, V. Kekatos, A. J. Conejo, and G. B. Giannakis, "Ergodic energy management leveraging resource variability in distribution grids," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4765–4775, Nov. 2016. → pages

[105] M. J. Farooq, H. Ghazzai, A. Kadri, H. ElSawy, and M.-S. Alouini, "A hybrid energy sharing framework for green cellular networks," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 918–934, Feb. 2017. → pages

[106] B. Li, T. Chen, X. Wang, and G. B. Giannakis, "Real-time energy management in microgrids with reduced battery capacity requirements," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1928–1938, Mar. 2019. → pages 12, 13

[107] X. Wang, Y. Zhang, G. B. Giannakis, and S. Hu, "Robust smart-grid-powered cooperative multipoint systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6188–6199, Nov. 2015. → pages 12

[108] X. Wang, Y. Zhang, T. Chen, and G. B. Giannakis, "Dynamic energy management for smart-grid-powered coordinated multipoint systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1348–1359, May 2016. → pages 12, 13

[109] X. Zhang, M. R. Nakhai, and W. N. S. F. Wan Ariffin, "A bandit approach to price-aware energy management in cellular networks," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1609–1612, July 2017. → pages 12, 13

[110] S. Hu, Y. Zhang, X. Wang, and G. B. Giannakis, "Weighted sum-rate maximization for MIMO downlink systems powered by renewables," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5615–5625, Aug. 2016. → pages 12

[111] X. Wang, T. Chen, X. Chen, X. Zhou, and G. B. Giannakis, "Dynamic resource allocation for smart-grid powered MIMO downlink transmissions," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3354–3365, Dec. 2016. → pages 12

[112] X. Wang, X. Chen, T. Chen, L. Huang, and G. B. Giannakis, "Two-scale stochastic control for integrated multipoint communication systems with renewables," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1822–1834, May 2018. → pages 13

[113] H. Yu, M. H. Cheung, L. Huang, and J. Huang, "Power-delay tradeoff with predictive scheduling in integrated cellular and Wi-Fi networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 735–742, Apr. 2016. → pages 13

[114] A. Sadeghi, G. Wang, and G. B. Giannakis, "Deep reinforcement learning for adaptive caching in hierarchical content delivery networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1024–1033, 2019. → pages 13

[115] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton university press, 2015. → pages 20, 45

[116] F. S. Melo and M. I. Ribeiro, "Q-learning with linear function approximation," in *International Conference on Computational Learning Theory*, San Diego, California, USA, Jun. 2007, pp. 308–322. → pages 21

[117] A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, and J. P. How, "A tutorial on linear function approximators for dynamic programming and reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 6, no. 4, pp. 375–451, 2013. → pages 22

[118] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st ed. Springer Publishing Company, Incorporated, 2014. → pages 28, 101

[119] F. Latorre, P. Rolland, and V. Cevher, "Lipschitz constant estimation of neural networks via sparse polynomial optimization," *Proc. International Conference on Learning Representations*, to be published, 2020. → pages 29

[120] C. Hu, W. Pan, and J. T. Kwok, "Accelerated gradient methods for stochastic optimization and online learning," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 2009, pp. 781–789. → pages 30, 32, 101

[121] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Opt.*, vol. 19, no. 4, pp. 1574–1609, 2009. → pages 33

[122] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994. [Online]. Available: https://cs.nyu.edu/~roweis/data.html → pages 40

[123] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/ → pages 40

[124] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, Sept. 2017. [Online]. Available: https://www.kaggle.com/koushikk/fmnist → pages 40

[125] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Springer, 2009, vol. 48. → pages 46

[126] R. S. Sutton, H. R. Maei, and C. Szepesvári, "A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, British Columbia, CA, Dec. 2009, pp. 1609–1616. → pages 47

[127] D. A. Levin and Y. Peres, *Markov chains and mixing times*. American Mathematical Society, 2017, vol. 107. → pages 47, 138

[128] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, 2018. → pages 50

[129] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Proc. International Conference on Machine Learning*, Tahoe City, California, USA, Jul. 1995, pp. 30–37. → pages 50

[130] R. S. Sutton, "Open theoretical questions in reinforcement learning," in *European Conference on Computational Learning Theory*, Nordkirchen, Germany, Mar. 1999, pp. 11–17. → pages 50

[131] G. J. Gordon, "Stable function approximation in dynamic programming," in *Proc. International Conference on Machine Learning*, Tahoe City, California, USA, Jul. 1995, pp. 261–268. → pages 50

[132] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, no. 5, May 1997. → pages 51

[133] K. You and L. Xie, "Network topology and communication data rate for consensusability of discrete-time multi-agent systems," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2262–2275, 2011. → pages 55

[134] H. Li, J. Xu, R. Zhang, and S. Cui, "A general utility optimization framework for energy-harvesting-based wireless communications," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 79–85, Apr. 2015. → pages 61, 81

[135] A. Attar, H. Li, and V. C. M. Leung, "Green last mile: how fiber-connected massively distributed antenna systems can save energy," *IEEE Wireless Commun.*, vol. 18, no. 5, pp. 66–74, Oct. 2011. → pages 62

[136] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005. → pages 63

[137] D. Krishnaswamy, "Game theoretic formulations for network-assisted resource management in wireless networks," in *Proc. IEEE VTC-Fall*, vol. 3, Vancouver, Canada, Sept. 2002, pp. 1312–1316. → pages 63

[138] A. Wiesel, Y. C. Eldar, and S. Shamai, "Linear precoding via conic optimization for fixed MIMO receivers," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 161–176, Jan. 2006. → pages 68

[139] W. Yu and T. Lan, "Transmitter optimization for the multi-antenna downlink with per-antenna power constraints," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2646–2660, June 2007. → pages 68

[140] Taesang Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 528–541, Mar. 2006. → pages 73

[141] M. R. A. Khandaker, K. Wong, Y. Zhang, and Z. Zheng, "Probabilistically robust SWIPT for secrecy MISOME systems," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 211–226, Jan. 2017. → pages 74, 94

[142] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9780898718829 → pages 74, 94

[143] *Medium Access Control (MAC) protocol specification (TS 36.321 Release 16)*, June 2020. → pages 91

[144] Y. Dong, M. J. Hossaini, J. Cheng, and V. C. M. Leung, "Robust energy efficient beamforming in misome-swipt systems with proportional secrecy rate," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 1, pp. 202–215, Jan. 2019. → pages 93

[145] E. F. Beckenbach and R. Bellman, *Inequalities*. Springer Science & Business Media, 2012, vol. 30. → pages 126, 127

[146] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1686–1716, Aug. 2015. → pages 141, 148

Appendices

Appendix A

Related Proofs of Chapter 3

A.1 Proof of Lemma 3.1

Based on the strong convexity of f_0, we obtain

$$
\begin{aligned}
f_0(\boldsymbol{u}_0) \geq & f_0(\boldsymbol{u}_{0,k}) + \left\langle \nabla f_0(\boldsymbol{u}_{0,k}), \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\rangle + \frac{\delta_0}{2} \left\| \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\|^2 \\
\geq & f_0(\boldsymbol{w}_{0,k}) - \frac{L_0}{2} \left\| \boldsymbol{w}_{0,k} - \boldsymbol{u}_{0,k} \right\|^2 + \left\langle \nabla f_0(\boldsymbol{u}_{0,k}), \boldsymbol{u}_0 - \boldsymbol{w}_{0,k} \right\rangle + \frac{\delta_0}{2} \left\| \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\|^2 \qquad \text{(A.1a)} \\
= & f_0(\boldsymbol{w}_{0,k}) - \frac{L_0}{2\alpha_{0,k}^2} \left\| \boldsymbol{h}_{0,k} \right\|^2 + \frac{\delta_0}{2} \left\| \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\|^2 + \left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k} - \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{w}_{0,k} \right\rangle \qquad \text{(A.1b)} \\
\geq & f_0(\boldsymbol{w}_{0,k}) + \frac{2\alpha_{0,k} - L_0}{2\alpha_{0,k}^2} \left\| \boldsymbol{h}_{0,k} \right\|^2 + \frac{\delta_0}{2} \left\| \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\|^2 \\
& + \left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\rangle - \left\langle \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{w}_{0,k} \right\rangle - \left\| \sum_{n=1}^{N} \boldsymbol{g}_{n,k} \right\| \frac{\left\| \boldsymbol{h}_{0,k} \right\|}{\alpha_{0,k}} \qquad \text{(A.1c)}
\end{aligned}
$$

where (A.1a) follows the Lipschitz continuous gradient of f_0; the RHS (A.1b) uses the definition (cf. (3.6b)) $\boldsymbol{h}_{0,k} := \alpha_{0,k}(\boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k}) = \nabla f_0(\boldsymbol{u}_{0,k})$; while the RHS of (A.1c) also relies on $\langle \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k} \rangle \geq -\frac{1}{\alpha_{0,k}} \| \sum_{n=1}^{N} \boldsymbol{g}_{n,k} \| \| \boldsymbol{h}_{0,k} \|$. Substituting $\boldsymbol{h}_{0,k}$ into (A.1c) completes the proof.

A.2 Proof of Lemma 3.2

Using the proximal operator definition, rewrite (3.8b) as

$$
\boldsymbol{w}_{n,k} = \arg\min_{\boldsymbol{u}_n} \left\{ \left\langle \nabla f(\boldsymbol{u}_{n,k}; x_{n,k}), \boldsymbol{u}_n - \boldsymbol{u}_{n,k} \right\rangle + \frac{\alpha_{n,k}}{2} \left\| \boldsymbol{u}_n - \boldsymbol{u}_{n,k} \right\|^2 + \gamma p_n(\boldsymbol{w}_{0,k} - \boldsymbol{u}_n) \right\}. \qquad \text{(A.2)}
$$

Based on the update in (A.2), $\gamma \nabla_{\boldsymbol{w}_n} p_n(\boldsymbol{w}_{0,k} - \boldsymbol{w}_{n,k}) = -\boldsymbol{g}_{n,k}$, and the definition of gradient

noise $\boldsymbol{\zeta}_{n,k}$ in (3.9), we obtain

$$\boldsymbol{h}_{n,k} = \alpha_{n,k}\left(\boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k}\right) = \nabla f\left(\boldsymbol{u}_{n,k}; x_{n,k}\right) - \boldsymbol{g}_{n,k} \tag{A.3}$$

$$\nabla f_n\left(\boldsymbol{w}_{n,k}\right) = \boldsymbol{h}_{n,k} + \boldsymbol{g}_{n,k} - \boldsymbol{\zeta}_{n,k}. \tag{A.4}$$

Since f_n is Lipschitz continuous, we deduce that

$$\begin{aligned} f_n\left(\boldsymbol{u}_{n,k}\right) \geq & f_n\left(\boldsymbol{w}_{n,k}\right) - \left\langle \nabla f_n\left(\boldsymbol{u}_{n,k}\right), \boldsymbol{w}_{n,k} - \boldsymbol{u}_{n,k} \right\rangle - \frac{L_n}{2}\left\|\boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k}\right\|^2 \\ = & f_n\left(\boldsymbol{w}_{n,k}\right) - \left\langle \nabla f_n\left(\boldsymbol{u}_{n,k}\right), \boldsymbol{w}_{n,k} - \boldsymbol{u}_{n,k} \right\rangle - \frac{L_n}{2\alpha_{n,k}^2}\left\|\boldsymbol{h}_{n,k}\right\|^2. \end{aligned} \tag{A.5}$$

Based on the strong convexity of f_n, we further obtain

$$f_n(\boldsymbol{u}_n) \geq f_n\left(\boldsymbol{u}_{n,k}\right) + \left\langle \nabla f_n\left(\boldsymbol{u}_{n,k}\right), \boldsymbol{u}_n - \boldsymbol{u}_{n,k} \right\rangle + \frac{\delta_n}{2}\left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2. \tag{A.6}$$

Summing (A.5) and (A.6), we arrive at

$$f_n(\boldsymbol{u}_n) - f_n\left(\boldsymbol{w}_{n,k}\right) \tag{A.7a}$$

$$\geq \left\langle \nabla f_n\left(\boldsymbol{u}_{n,k}\right), \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle - \frac{L_n}{2\alpha_{n,k}^2}\left\|\boldsymbol{h}_{n,k}\right\|^2 + \frac{\delta_n}{2}\left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2 \tag{A.7b}$$

$$\geq \left\langle \boldsymbol{h}_{n,k} + \boldsymbol{g}_{n,k} - \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle - \frac{L_n}{2\alpha_{n,k}^2}\left\|\boldsymbol{h}_{n,k}\right\|^2 + \frac{\delta_n}{2}\left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2 \tag{A.7c}$$

$$\geq \frac{2\alpha_{n,k} - L_n}{2\alpha_{n,k}^2}\left\|\boldsymbol{h}_{n,k}\right\|^2 + \frac{\delta_n}{2}\left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2 - \left\langle \boldsymbol{\zeta}_{n,k} - \boldsymbol{g}_{n,k}, \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle + \left\langle \boldsymbol{h}_{n,k}, \boldsymbol{u}_n - \boldsymbol{u}_{n,k} \right\rangle \tag{A.7d}$$

where the RHS of (A.7c) is due to (A.4), and the RHS of (A.7d) follows from (A.3).

Finally, substituting into (A.7d) completes the proof.

A.3 Proof of Lemma 3.3

With $\boldsymbol{h}_{0,k}$ as in (A.1b), construct a strongly convex function with modulus $\delta_0 + \alpha_{0,k}\beta_k$ in slot k, as

$$\phi_{0,k}(\boldsymbol{u}_0) := \left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\rangle + \frac{\delta_0}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{u}_{0,k}\right\|^2 + \frac{\alpha_{0,k}\beta_k}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{v}_{0,k-1}\right\|^2. \tag{A.8}$$

According to (3.6c), $\boldsymbol{v}_0^k$ is the minimizer of (A.8). Strong convexity implies that $\phi_{0,k}(\boldsymbol{v}_{0,k}) \leq \phi_{0,k}(\boldsymbol{u}_0) - \frac{\delta_0+\alpha_{0,k}\beta_k}{2}\left\|\boldsymbol{v}_{0,k}-\boldsymbol{u}_0\right\|^2$. Thus, upon expanding $\phi_{0,k}(\boldsymbol{v}_{0,k})$ and $\phi_{0,k}(\boldsymbol{u}_0)$, we have

$$\begin{aligned}
&\left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_0 - \boldsymbol{u}_{0,k} \right\rangle + \frac{\delta_0}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{u}_{0,k}\right\|^2 \\
\geq &\left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{v}_{0,k} - \boldsymbol{u}_{0,k} \right\rangle + \frac{\alpha_{0,k}\beta_k + \delta_0}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{v}_{0,k}\right\|^2 \\
&+ \frac{\alpha_{0,k}\beta_k}{2}\left\|\boldsymbol{v}_{0,k} - \boldsymbol{v}_{0,k-1}\right\|^2 - \frac{\alpha_{0,k}\beta_k}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{v}_{0,k-1}\right\|^2 .
\end{aligned} \tag{A.9}$$

Similar to (A.9), and with $\boldsymbol{h}_{n,k}$ as in (A.3), we obtain

$$\begin{aligned}
\left\langle \boldsymbol{h}_{n,k}, \boldsymbol{u}_n - \boldsymbol{u}_{n,k} \right\rangle + \frac{\delta_n}{2}\left\|\boldsymbol{u}_n - \boldsymbol{u}_{n,k}\right\|^2 \geq &\left\langle \boldsymbol{h}_{n,k}, \boldsymbol{v}_{n,k} - \boldsymbol{u}_{n,k} \right\rangle + \frac{\alpha_{n,k}\beta_k + \delta_n}{2}\left\|\boldsymbol{u}_n - \boldsymbol{v}_{n,k}\right\|^2 \\
&+ \frac{\alpha_{n,k}\beta_k}{2}\left\|\boldsymbol{v}_{n,k} - \boldsymbol{v}_{n,k-1}\right\|^2 - \frac{\alpha_{n,k}\beta_k}{2}\left\|\boldsymbol{u}_n - \boldsymbol{v}_{n,k-1}\right\|^2 .
\end{aligned} \tag{A.10}$$

Substituting (A.9) and (A.10) into (3.13), we thus find

$$\begin{aligned}
F(\boldsymbol{w}_k) - F(\boldsymbol{u}) \leq &\frac{\left\|\sum_{n=1}^{N} \boldsymbol{g}_{n,k}\right\|}{\alpha_{0,k}}\left\|\boldsymbol{h}_{0,k}\right\| - \sum_{n=0}^{N_{\mathrm{R}}} \frac{2\alpha_{n,k} - L_n}{2\alpha_{n,k}^2}\left\|\boldsymbol{h}_{n,k}\right\|^2 + \sum_{n=0}^{N_{\mathrm{R}}} \frac{\lambda_{1,n,k}}{\beta_k} \\
&+ \sum_{n=0}^{N_{\mathrm{R}}} \left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_n - \boldsymbol{w}_{n,k} \right\rangle + \left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_{0,k} - \boldsymbol{v}_{0,k} \right\rangle + \sum_{n=1}^{N_{\mathrm{R}}} \left\langle \boldsymbol{h}_{n,k}, \boldsymbol{u}_{n,k} - \boldsymbol{v}_{n,k} \right\rangle
\end{aligned} \tag{A.11}$$

where $\lambda_{1,n,k}$ is defined as

$$\lambda_{1,n,k} := \frac{\alpha_{n,k}\beta_k^2}{2}\left\|\boldsymbol{u}_n - \boldsymbol{v}_{n,k-1}\right\|^2 - \frac{\delta_n\beta_k + \alpha_{n,k}\beta_k^2}{2}\left\|\boldsymbol{u}_n - \boldsymbol{v}_{n,k}\right\|^2 - \frac{\alpha_{n,k}\beta_k^2}{2}\left\|\boldsymbol{v}_{n,k} - \boldsymbol{v}_{n,k-1}\right\|^2 . \tag{A.12}$$

Setting $\boldsymbol{u} = \boldsymbol{w}_{k-1}$ in (3.13), and dropping the non-positive terms $-\frac{\delta_n}{2}\|\boldsymbol{w}_{n,k-1} - \boldsymbol{w}_{n,k}\|^2$, we arrive at

$$\begin{aligned}
F(\boldsymbol{w}_k) - F(\boldsymbol{w}_{k-1}) \leq &\frac{\left\|\sum_{n=1}^{N} \boldsymbol{g}_{n,k}\right\|}{\alpha_{0,k}}\left\|\boldsymbol{h}_{0,k}\right\| - \sum_{n=0}^{N_{\mathrm{R}}} \frac{2\alpha_{n,k} - L_n}{2\alpha_{n,k}^2}\left\|\boldsymbol{h}_{n,k}\right\|^2 + \sum_{n=0}^{N_{\mathrm{R}}} \left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{w}_{n,k-1} - \boldsymbol{w}_{n,k} \right\rangle \\
&+ \left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_{0,k} - \boldsymbol{w}_{0,k-1} \right\rangle + \sum_{n=1}^{N_{\mathrm{R}}} \left\langle \boldsymbol{h}_{n,k}, \boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k-1} \right\rangle .
\end{aligned} \tag{A.13}$$

Using (A.11) and (A.13), the convex combination $\beta_k(F(\boldsymbol{w}_k) - F(\boldsymbol{u})) + (1-\beta_k)(F(\boldsymbol{w}_k) - F(\boldsymbol{w}_{k-1}))$

is bounded as

$$\begin{aligned} &F(\boldsymbol{w}_k) - F(\boldsymbol{u}) - (1-\beta_k)(F(\boldsymbol{w}_{k-1}) - F(\boldsymbol{u})) \\ \leq &\frac{\left\| \sum_{n=1}^{N} \boldsymbol{g}_{n,k} \right\|}{\alpha_{0,k}} \left\| \boldsymbol{h}_{0,k} \right\| - \sum_{n=0}^{N_{\mathrm{R}}} \frac{2\alpha_{n,k} - L_n}{2\alpha_{n,k}^2} \left\| \boldsymbol{h}_{n,k} \right\|^2 + \sum_{n=0}^{N_{\mathrm{R}}} \lambda_{1,n,k} + \sum_{n=0}^{N_{\mathrm{R}}} \lambda_{2,n,k} + \sum_{n=0}^{N_{\mathrm{R}}} \lambda_{3,n,k} \end{aligned} \tag{A.14}$$

where

$$\lambda_{2,n,k} := \left\langle \boldsymbol{\zeta}_{n,k}, \beta_k \boldsymbol{u}_n + (1-\beta_k)\boldsymbol{w}_{n,k-1} - \boldsymbol{w}_{n,k} \right\rangle \tag{A.15}$$

and

$$\lambda_{3,n,k} := \begin{cases} \left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N} \boldsymbol{g}_{n,k}, \boldsymbol{u}_{0,k} - \beta_k \boldsymbol{v}_{0,k} - (1-\beta_k)\boldsymbol{w}_{0,k-1} \right\rangle, n = 0 \\ \left\langle \boldsymbol{h}_{n,k}, \boldsymbol{u}_{n,k} - \beta_k \boldsymbol{v}_{n,k} - (1-\beta_k)\boldsymbol{w}_{n,k-1} \right\rangle, n = 1, \ldots, N_{\mathrm{R}}. \end{cases} \tag{A.16}$$

Based on (3.8a), we obtain

$$(1-\beta_k)\boldsymbol{w}_{n,k} = \boldsymbol{u}_{n,k} - \beta_k \boldsymbol{v}_{n,k}, n = 0, 1, \ldots, N_{\mathrm{R}}. \tag{A.17}$$

Substituting (A.17) into (A.15), it holds for $n = 0, 1, \ldots, N_{\mathrm{R}}$ that

$$\lambda_{2,n,k} = \beta_k \left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_n - \boldsymbol{v}_{n,k-1} \right\rangle + \left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_{n,k} - \boldsymbol{w}_{n,k} \right\rangle \leq \beta_k \left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_n - \boldsymbol{v}_{n,k-1} \right\rangle + \frac{\sqrt{\left\| \boldsymbol{\zeta}_{n,k} \right\|^2}}{\alpha_{n,k}} \left\| \boldsymbol{h}_{n,k} \right\| \tag{A.18}$$

where (A.18) follows from Hölder's inequality [145].

Taking expectation on both sides of (A.18) for terms $n = 1, \ldots, N_{\mathrm{R}}$, we obtain

$$\mathbb{E}_{X_{n,1:K}} \left[\lambda_{2,n,k} \right] \leq \mathbb{E}_{X_{n,1:K}} \left[\beta_k \left\langle \boldsymbol{\zeta}_{n,k} \boldsymbol{u}_n - \boldsymbol{v}_{n,k-1} \right\rangle + \frac{\sqrt{\left\| \boldsymbol{\zeta}_{n,k} \right\|^2}}{\alpha_{n,k}} \left\| \boldsymbol{h}_{n,k} \right\| \right] = \frac{\sigma_n}{\alpha_{n,k}} \left\| \boldsymbol{h}_{n,k} \right\| \tag{A.19}$$

where the equality is due to the facts

$$\mathbb{E}_{X_{n,1:K}} \left[\sqrt{\left\| \boldsymbol{\zeta}_{n,k} \right\|^2} \right] \leq \mathbb{E}_{X_{n,1:K-1}} \sqrt{\mathbb{E}_{X_{n,K}} \left[\left\| \boldsymbol{\zeta}_{n,k} \right\|^2 \right]} = \sigma_n \tag{A.20}$$

and since Assumption 3.4 dictates $\mathbb{E}_{X_{n,K}}[\boldsymbol{\zeta}_{n,k}] = 0$, we have

$$\mathbb{E}_{X_{n,1:K}}\left[\left\langle \boldsymbol{\zeta}_{n,k}, \boldsymbol{u}_n - \boldsymbol{v}_n^{k-1}\right\rangle\right] = \mathbb{E}_{X_{n,1:K-1}}\left\langle \mathbb{E}_{X_{n,K}}\left[\boldsymbol{\zeta}_{n,k}\right], \boldsymbol{u}_n - \boldsymbol{v}_{n,k-1}\right\rangle = 0. \tag{A.21}$$

Based on the Young's inequality [145], $\lambda_{2,0,k}$ is bounded as

$$\lambda_{2,0,k} \leq \frac{\beta_k}{2\epsilon}\left\|\boldsymbol{\zeta}_{0,k}\right\|^2 + \frac{\epsilon\beta_k}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{v}_{0,k-1}\right\|^2 + \frac{\left\|\boldsymbol{\zeta}_{0,k}\right\|}{\alpha_{0,k}}\left\|\boldsymbol{h}_{0,k}\right\| \tag{A.22}$$

with $\epsilon \in (0, \infty)$.

Substituting (A.17) into (A.16), we obtain for $n = 1, \ldots, N$ that

$$\lambda_{3,n,k} = \beta_k\left\langle \boldsymbol{h}_{n,k}, \boldsymbol{v}_{n,k-1} - \boldsymbol{v}_{n,k}\right\rangle \leq \frac{1}{2\alpha_{n,k}}\left\|\boldsymbol{h}_{n,k}\right\|^2 + \frac{\alpha_{n,k}\beta_k^2}{2}\left\|\boldsymbol{v}_{n,k-1} - \boldsymbol{v}_{n,k}\right\|^2 \tag{A.23}$$

where the inequality is due to Young's inequality [145].

Substituting (A.17) into $\lambda_{3,0,k}$, we deduce

$$\begin{aligned}\lambda_{3,0,k} =& \beta_k\left\langle \boldsymbol{h}_{0,k} + \sum_{n=1}^{N}\boldsymbol{g}_{n,k}, \boldsymbol{v}_{0,k-1} - \boldsymbol{v}_{0,k}\right\rangle \\ \leq& \frac{1}{2\alpha_{0,k}}\left\|\boldsymbol{h}_{0,k} + \sum_{n=1}^{N}\boldsymbol{g}_{n,k}\right\|^2 + \frac{\alpha_{0,k}\beta_k^2}{2}\left\|\boldsymbol{v}_{0,k-1} - \boldsymbol{v}_{0,k}\right\|^2 \\ \leq& \frac{2}{3\alpha_{0,k}}\left\|\boldsymbol{h}_{0,k}\right\|^2 + \frac{2}{\alpha_{0,k}}\left\|\sum_{n=1}^{N}\boldsymbol{g}_{n,k}\right\|^2 + \frac{\alpha_{0,k}\beta_k^2}{2}\left\|\boldsymbol{v}_{0,k-1} - \boldsymbol{v}_{0,k}\right\|^2\end{aligned} \tag{A.24}$$

where first inequality is due to Young's inequality [145], and the second inequality is based on the fact that

$$\left\|\boldsymbol{h}_{0,k} + \sum_{n=1}^{N}\boldsymbol{g}_{n,k}\right\|^2 \leq \frac{4}{3}\left\|\boldsymbol{h}_{0,k}\right\|^2 + 4\left\|\sum_{n=1}^{N}\boldsymbol{g}_{n,k}\right\|^2. \tag{A.25}$$

Substituting (A.12), (A.18) and (A.22)–(A.24) into the RHS of (A.14), we obtain

$$F(\boldsymbol{w}_k) - F(\boldsymbol{u}) - (1-\beta_k)(F(\boldsymbol{w}_{k-1}) - F(\boldsymbol{u})) \leq \sum_{n=0}^{N_{\mathrm{R}}}\left(\lambda_{4,n,k} + \lambda_{5,n,k}\right) + \frac{2}{\alpha_{0,k}}\left\|\sum_{n=1}^{N}\boldsymbol{g}_{n,k}\right\|^2 + \frac{\beta_k}{2\epsilon}\left\|\boldsymbol{\zeta}_{0,k}\right\|^2 \tag{A.26}$$

where

$$\lambda_{4,n,k} := \begin{cases} \frac{\left\|\sum_{n=1}^{N} \boldsymbol{g}_{n,k}\right\| + \|\boldsymbol{\zeta}_{0,k}\|}{\alpha_{0,k}} \left\|\boldsymbol{h}_{0,k}\right\| - \frac{2\alpha_{0,k}-3L_0}{6\alpha_{0,k}^2} \left\|\boldsymbol{h}_{0,k}\right\|^2, n = 0 \\ \frac{\sigma_n}{\alpha_{n,k}} \left\|\boldsymbol{h}_{n,k}\right\| - \frac{\alpha_{n,k}-L_n}{2\alpha_{n,k}^2} \left\|\boldsymbol{h}_{n,k}\right\|^2, n = 1, \ldots, N \end{cases}$$

and

$$\lambda_{5,n,k} := \begin{cases} \frac{\epsilon\beta_k+\alpha_{0,k}\beta_k^2}{2} \left\|\boldsymbol{u}_0 - \boldsymbol{v}_{0,k-1}\right\|^2 - \frac{\delta_0\beta_k+\alpha_{0,k}\beta_k^2}{2} \left\|\boldsymbol{u}_0 - \boldsymbol{v}_{0,k}\right\|^2, n = 0 \\ \frac{\alpha_{n,k}\beta_k^2}{2} \left\|\boldsymbol{u}_n - \boldsymbol{v}_{n,k-1}\right\|^2 - \frac{\delta_n\beta_k+\alpha_{n,k}\beta_k^2}{2} \left\|\boldsymbol{u}_n - \boldsymbol{v}_{n,k}\right\|^2, n = 1, \ldots, N. \end{cases} \tag{A.27}$$

Using the inequality $-ax^2 + bx \leq \frac{b^2}{4a}$ and the power of $\|\sum_{n=1}^{N} \boldsymbol{g}_{n,k}\| + \|\boldsymbol{\zeta}_{0,k}\|$ in (3.14), we can bound $\lambda_{4,n,k}$ as

$$\lambda_{4,n,k} \leq \lambda_{6,n,k} = \begin{cases} \frac{3\sigma_0^2}{2(2\alpha_{0,k}-3L_0)}, n = 0 \\ \frac{\sigma_n^2}{2(\alpha_{n,k}-L_n)}, n = 1, \ldots, N. \end{cases} \tag{A.28}$$

Substituting (A.28) into (A.26) and setting $\boldsymbol{u} = \boldsymbol{u}^*$ lead to (3.15).

A.4 Proof of Theorem 3.4

Dividing both sides of (3.15) by β_k^2, we can write

$$\frac{1}{\beta_k^2}(F(\boldsymbol{w}_k) - F(\boldsymbol{u}^*)) \leq \frac{1-\beta_k}{\beta_k^2}(F(\boldsymbol{w}_{k-1}) - F(\boldsymbol{u}^*)) + \sum_{n=0}^{N_\mathrm{R}} \frac{\lambda_{5,n,k} + \lambda_{6,n,k}}{\beta_k^2} + \frac{2\gamma^2 N^2 G}{\alpha_{0,k}\beta_k^2} + \frac{\gamma^2 N_\mathrm{B}^2 G}{2\epsilon\beta_k}. \tag{A.29}$$

Setting $\beta_k = \frac{2}{k+2}$, we can readily verify that

$$\frac{1-\beta_k}{\beta_k^2} \leq \frac{1}{\beta_{k-1}^2}. \tag{A.30}$$

Summing (A.29) over $k = 1, \ldots, K$, it follows after straightforward manipulations that

$$\begin{aligned} \frac{1}{\beta_k^2}(F(\boldsymbol{w}_k) - F(\boldsymbol{u}^*)) \leq & F(\boldsymbol{w}_0) - F(\boldsymbol{u}^*) + \sum_{k=1}^{K} \frac{\gamma^2 N_\mathrm{B}^2 G}{2\epsilon\beta_k} + \frac{\frac{\epsilon}{\beta_1} + \alpha_{0,1}}{2} \left\|\boldsymbol{u}_0^* - \boldsymbol{v}_{0,0}\right\|^2 + \sum_{n=1}^{N_\mathrm{R}} \frac{\alpha_{n,1}}{2} \left\|\boldsymbol{u}_n^* - \boldsymbol{v}_{n,0}\right\|^2 \\ & + \sum_{k=1}^{K-1} \sum_{n=0}^{N_\mathrm{R}} \lambda_{7,n,k} + \sum_{k=1}^{K} \sum_{n=0}^{N_\mathrm{R}} \lambda_{8,n,k} \end{aligned} \tag{A.31}$$

where

$$\lambda_{7,n,k} := \begin{cases} \frac{1}{2}\left(\alpha_{0,k+1} - \alpha_{0,k} + \frac{\epsilon}{\beta_{k+1}} - \frac{\delta_0}{\beta_k}\right)\left\|\boldsymbol{u}_0^* - \boldsymbol{v}_{0,k}\right\|^2, n = 0 \\ \frac{1}{2}\left(\alpha_{n,k+1} - \alpha_{n,k} - \frac{\delta_n}{\beta_k}\right)\left\|\boldsymbol{u}_n^* - \boldsymbol{v}_{n,k}\right\|^2, n = 1, \ldots, N_{\mathrm{R}} \end{cases} \tag{A.32}$$

and

$$\lambda_{8,n,k} := \begin{cases} \frac{8\gamma^2 N^2 G + 3\sigma_0^2}{2\left(2\alpha_{0,k} - 3L_0\right)\beta_k^2}, n = 0 \\ \frac{\sigma_n^2}{2\left(\alpha_{n,k} - L_n\right)\beta_k^2}, n = 1, \ldots, N_{\mathrm{R}}. \end{cases} \tag{A.33}$$

To analyze the convergence of FRPG, we introduce the following constraints

$$\frac{\delta_0}{\beta_k} - \frac{\epsilon}{\beta_{k+1}} > 0 \tag{A.34a}$$

$$\frac{\delta_0}{\beta_k} - \frac{\epsilon}{\beta_{k+1}} \geq \alpha_{0,k+1} - \alpha_{0,k} \tag{A.34b}$$

$$\alpha_{0,k} = \frac{3}{2}\left(\frac{c_0}{\beta_k^2} + L_0\right) \tag{A.34c}$$

$$\frac{\delta_n}{\beta_k} \geq \alpha_{n,k+1} - \alpha_{n,k}, n = 1, \ldots, N_{\mathrm{R}} \tag{A.34d}$$

$$\alpha_{n,k} = \frac{c_n}{\beta_k^2} + L_n, n = 1, \ldots, N_{\mathrm{R}} \tag{A.34e}$$

where $c_n > 0$ with $n = 0, 1, \ldots, N_{\mathrm{R}}$; and (A.34a) with $\beta_k = \frac{2}{k+2}$ imply that $\epsilon < \frac{3}{4}\delta_0$. Without loss of generality, we set $\epsilon = \frac{1}{2}\delta_0$. Based on (A.34b) and (A.34c), we have $c_0 \leq \frac{4}{21}\delta_0$. Hence, $\alpha_{0,k}$ is given by $\alpha_{0,k} = \frac{\delta_0}{14}(k+2)^2 + \frac{3}{2}L_0$. From (A.34d) and (A.34e), we deduce that $c_n \leq \frac{6}{7}\delta_n$, which implies that $\alpha_{n,k} = \frac{3\delta_n}{14}(k+2)^2 + L_n$ with $n = 1, \ldots, N$. As a result, we find

$$\alpha_{n,k} = \begin{cases} \frac{\delta_0}{14}(k+2)^2 + \frac{3}{2}L_0, n = 0 \\ \frac{3\delta_n}{14}(k+2)^2 + L_n, n = 1, \ldots, N_{\mathrm{R}}. \end{cases} \tag{A.35}$$

Based on (A.35) and $\epsilon = \frac{1}{2}\delta_0$, we simplify $\sum_{k=1}^{K-1}\lambda_{7,n,k}$ as

$$\sum_{k=1}^{K-1}\lambda_{7,n,k} \leq \lambda_{9,n} := \begin{cases} \left(\frac{3}{8}\delta_0 + \frac{1}{2}\alpha_{0,1}\right)\left\|\boldsymbol{u}_0^* - \boldsymbol{v}_{0,0}\right\|^2, n = 0 \\ \frac{1}{2}\alpha_{n,1}\left\|\boldsymbol{u}_n^* - \boldsymbol{v}_{n,0}\right\|^2, n = 1, \ldots, N_{\mathrm{R}}. \end{cases} \tag{A.36}$$

Using (A.35), $\lambda_{8,n,k}$ reduces to

$$\lambda_{8,n,k} = \lambda_{10,n} := \begin{cases} \frac{7\gamma^2 N^2 G + \frac{21}{8}\sigma_0^2}{\delta_0}, n = 0 \\ \frac{7\sigma_n^2}{12\delta_n}, n = 1, \ldots, N_{\mathrm{R}}. \end{cases} \tag{A.37}$$

Substituting $\beta_k = \frac{2}{k+2}$ and (A.35)–(A.37) into (A.31), we establish the convergence rate of FRPG as

$$F(\boldsymbol{w}_k) - F(\boldsymbol{u}^*) \leq \frac{4}{(K+2)^2}\left(F(\boldsymbol{w}_0) - F(\boldsymbol{u}^*) + \sum_{n=0}^{N_{\mathrm{R}}} \lambda_{9,n}\right) + \frac{4K}{(K+2)^2}\sum_{n=0}^{N_{\mathrm{R}}} \lambda_{10,n} + O\left(\frac{\gamma^2 N_{\mathrm{B}}^2 G}{\delta_0}\right) \tag{A.38}$$

where $O(x)$ represents a polynomial of x.

A.5 Proof of Theorem 3.8

Setting $\beta^i = \frac{2}{i+2}$ so that $\frac{1-\beta[i]}{\beta[i]^2} \leq \frac{1}{\beta[i-1]^2}$; summing (3.31) over $i = 1, \ldots, I$; and, multiplying by $\beta[I]^2$, we obtain

$$\begin{aligned} &\frac{1}{T}\sum_{k=1}^{T} F(\boldsymbol{w}_k[I]) - F(\boldsymbol{u}^*) \\ \leq &\beta^2[I]\left(\frac{1}{T}\sum_{k=1}^{T} F(\boldsymbol{w}_k[0]) - F(\boldsymbol{u}^*)\right) + \beta^2[I]\sum_{n=0}^{N_{\mathrm{R}}}\sum_{i=1}^{I}\lambda_{14,n}[i] + \beta[I]^2\sum_{i=1}^{I}\frac{\gamma^2 N_{\mathrm{B}}^2 G}{2\epsilon\beta[i]} \\ &+ \beta^2[I]\left(\frac{\epsilon + \alpha_0[1]\beta[1]}{2\beta[1]}\left\|\boldsymbol{u}_0^* - \boldsymbol{v}_0[0]\right\|^2 + \frac{1}{2}\sum_{i=1}^{I-1}\left(\alpha_0[i+1] - \alpha_0[i] + \frac{\epsilon}{\beta[i+1]} - \frac{\delta_0}{\beta[i]}\right)\left\|\boldsymbol{u}_0^* - \boldsymbol{v}_0[i]\right\|^2\right) \\ &+ \frac{\beta^2[I]}{T}\sum_{n=1}^{N_{\mathrm{R}}}\left(\frac{\alpha_n[i]}{2}\left\|\boldsymbol{u}_n^* - \boldsymbol{v}_n[0]\right\|^2 + \frac{1}{2}\sum_{i=1}^{I-1}\left(\alpha_n[i+1] - \alpha_n[i] - \frac{\delta_n}{\beta[i]}\right)\left\|\boldsymbol{u}_n^* - \boldsymbol{v}_n[i]\right\|^2\right) \end{aligned} \tag{A.39}$$

Based on (A.39), we introduce the following constraints in order to guarantee the convergence of LFRPG

$$\frac{\delta_0}{\beta[i]} - \frac{\epsilon}{\beta[i+1]} > 0 \tag{A.40a}$$

$$\frac{\delta_0}{\beta[i]} - \frac{\epsilon}{\beta[i+1]} \geq \alpha_0[i+1] - \alpha_0[i] \tag{A.40b}$$

$$\frac{\delta_n}{\beta[i]} \geq \alpha_n[i+1] - \alpha_n[i], n = 1, \ldots, N_{\mathrm{R}} \tag{A.40c}$$

$$\alpha_0[i] = \frac{3}{2}\left(\frac{c_0}{\beta^2[i]} + L_0\right) \tag{A.40d}$$

$$\alpha_n[i] = \frac{c_n}{\beta^2[i]} + L_n, n = 1, \ldots, N_{\mathrm{R}}. \tag{A.40e}$$

Eq. (A.40a) implies that $\epsilon < \frac{3}{4}\delta_0$, based on which we select $\epsilon = \frac{1}{2}\delta_0$. Using (A.40b) and (A.40d), we obtain $c_0 \leq \frac{4}{21}\delta_0$; and based on (A.40c) and (A.40e), we find $c_n \leq \frac{6}{7}\delta_n$. Thus, we set the stepsize α_n^i as

$$\alpha_n[i] = \begin{cases} \frac{\delta_0}{14}(i+2)^2 + \frac{3}{2}L_0, n = 0 \\ \frac{3\delta_n}{14}(i+2)^2 + L_n, n = 1, \ldots, N_{\mathrm{R}}. \end{cases} \tag{A.41}$$

We now can establish convergence of $\bar{\boldsymbol{w}}[I] = T^{-1}\sum_{k=1}^{T}\boldsymbol{w}_k[I]$ as

$$\begin{aligned} F(\bar{\boldsymbol{w}}[I]) - F(\boldsymbol{u}^*) \leq & \frac{1}{T}\sum_{k=1}^{T} F(\boldsymbol{w}_k[I]) - F(\boldsymbol{u}^*) \\ \leq & \frac{2\lambda_{16}}{T(I+2)^2} + \frac{\lambda_{17}}{(I+2)^2} + \frac{I\lambda_{18}}{(I+2)^2} + O\left(\frac{\gamma^2 N_{\mathrm{B}}^2 G}{\delta_0}\right) \end{aligned} \tag{A.42}$$

where λ_{16}, λ_{17} and λ_{18} are defined respectively as

$$\lambda_{16} := \sum_{n=1}^{N_{\mathrm{R}}} \alpha_n[1] \left\|\boldsymbol{u}_n^* - \boldsymbol{v}_n[0]\right\|^2 \tag{A.43}$$

$$\lambda_{17} := \left(\frac{3}{2}\delta_0 + 2\alpha_0[1]\right)\left\|\boldsymbol{u}_0^* - \boldsymbol{v}_0[0]\right\|^2 + \frac{4}{T}\sum_{k=1}^{T} F(\boldsymbol{w}_k[0]) - 4F(\boldsymbol{u}^*) \tag{A.44}$$

$$\lambda_{18} := \sum_{n=1}^{N_{\mathrm{R}}} \frac{7\sigma_n^2}{3\delta_n} + \frac{11\sigma_0^2 + 28\gamma^2 N^2 G}{\delta_0}. \tag{A.45}$$

Appendix B

Related Proofs of Chapter 4

B.1 Proof of Equivalence Between (4.4) and (4.6)

Before we proceed the proof, we introduce the definition of inner product of two functions. Denote the any two functions $f_1(s, \boldsymbol{a})$ and $f_2(s, \boldsymbol{a})$, the inner product ot two functions is defined as

$$\langle f_1, f_2 \rangle_X := \sum_{\boldsymbol{a} \in \mathcal{A}} \sum_{s \in \mathcal{S}} f_1(s, \boldsymbol{a}) f_2(s, \boldsymbol{a}) \pi^s \mu_{\boldsymbol{a}}^s p_{\boldsymbol{a}}^{s,s'}. \tag{B.1}$$

Therefore, the X-induced norm is denoted by

$$\|f_1\|_X = \sqrt{\langle f_1, f_1 \rangle_X}. \tag{B.2}$$

We investigate the equivalence between (4.4) and (4.6) where the projection is with respect to the induced norm $\|\cdot\|_X$. The projected Bellman equation in (4.4) is recast as

$$\text{proj}_{\mathcal{Q}} \left\{ \mathbb{B}\left[\tilde{Q}(s, \boldsymbol{a})\right] - \tilde{Q}(s, \boldsymbol{a}) \right\} = 0, s \in \mathcal{S} \text{ and } \boldsymbol{a} \in \mathcal{A}. \tag{B.3}$$

Based on (B.3) and the definition of projection operator, we obtain

$$\left\langle \phi_d(s, \boldsymbol{a}), \mathbb{B}\left[\tilde{Q}(s, \boldsymbol{a})\right] - \tilde{Q}(s, \boldsymbol{a}) \right\rangle_X = 0, d = 1, \ldots, D. \tag{B.4}$$

Using (B.1), we obtain the following derivations

$$
\begin{aligned}
&\left\langle \phi_d(s,\boldsymbol{a}), \mathbb{B}\left[\tilde{Q}(s,\boldsymbol{a})\right] - \tilde{Q}(s,\boldsymbol{a})\right\rangle_X = 0 \\
\Leftrightarrow &\sum_{\boldsymbol{a}\in\mathcal{A}}\sum_{s\in\mathcal{S}} \phi_d(s,\boldsymbol{a})\pi^s \mu_{\boldsymbol{a}}^s \left(\frac{1}{N}\sum_{n=1}^{N} r_n(s,\boldsymbol{a}) + \gamma \sum_{s'\in\mathcal{S}} p_{\boldsymbol{a}}^{s,s'} \max_{\boldsymbol{a}'\in\mathcal{A}} \tilde{Q}(s',\boldsymbol{a}') - \tilde{Q}(s,\boldsymbol{a})\right) = 0 \\
\Leftrightarrow &\sum_{\boldsymbol{a}\in\mathcal{A}}\sum_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}} \pi^s \mu_{\boldsymbol{a}}^s p_{\boldsymbol{a}}^{s,s'} \phi_d(s,\boldsymbol{a}) \left(\frac{1}{N}\sum_{n=1}^{N} r_n(s,\boldsymbol{a}) + \gamma \max_{\boldsymbol{a}'\in\mathcal{A}} \tilde{Q}(s',\boldsymbol{a}') - \tilde{Q}(s,\boldsymbol{a})\right) = 0 \\
\Leftrightarrow &\mathbb{E}_X\left[\phi_d(s,\boldsymbol{a})\left(\frac{1}{N}\sum_{n=1}^{N} r_n(s,\boldsymbol{a}) + \gamma \max_{\boldsymbol{a}'\in\mathcal{A}} \tilde{Q}(s',\boldsymbol{a}') - \tilde{Q}(s,\boldsymbol{a})\right)\right] = 0 \\
\Leftrightarrow &\mathbb{E}_X\left[\phi(s,\boldsymbol{a})\left(\frac{1}{N}\sum_{n=1}^{N} r_n(s,\boldsymbol{a}) + \gamma \max_{\boldsymbol{a}'\in\mathcal{A}} \tilde{Q}(s',\boldsymbol{a}') - \tilde{Q}(s,\boldsymbol{a})\right)\right] = 0
\end{aligned}
\tag{B.5}
$$

where the expectation is taken over the triple $X = (s, \boldsymbol{a}, s')$.

Since each agent n maintains a local estimation of Q-function as $\tilde{Q}(s,\boldsymbol{a}) = \boldsymbol{\phi}^\top(s,\boldsymbol{a})\boldsymbol{w}_n$, we obtain (4.6) by substituting the approximate Q-function into the last equation in (B.5).

B.2 Proof of Lemma 4.1

Let $\boldsymbol{w}$ and $\boldsymbol{w}'$ be two vectors in $\mathbb{R}^{Nd}$. Recalling (4.12) and $X_k = (S_k, A_k, S_{k+1})$, we obtain $\boldsymbol{f}(\boldsymbol{w}; X_k) - \boldsymbol{f}(\boldsymbol{w}'; X_k)$ as

$$
\boldsymbol{f}(\boldsymbol{w}; X_k) - \boldsymbol{f}(\boldsymbol{w}'; X_k) = \begin{bmatrix} \vdots \\ \gamma\phi_k\left(\left\langle \hat{\phi}_{n,k}, \boldsymbol{w}_n\right\rangle - \left\langle \check{\phi}_{n,k}, \boldsymbol{w}'_n\right\rangle\right) \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ \phi_k\left\langle \phi_k, \boldsymbol{w}_n - \boldsymbol{w}'_n\right\rangle \\ \vdots \end{bmatrix} \tag{B.6}
$$

where $\hat{\phi}_{n,k} = \arg\max_{\boldsymbol{a}'\in\mathcal{A}} \boldsymbol{\phi}^\top(S_{k+1},\boldsymbol{a}')w_n$ and $\check{\phi}_{n,k} = \arg\max_{\boldsymbol{a}'\in\mathcal{A}} \boldsymbol{\phi}^\top(S_{k+1},\boldsymbol{a}')w'_n$.

Based on (B.6), we obtain

$$
\|\boldsymbol{f}(\boldsymbol{w}; X_k) - \boldsymbol{f}(\boldsymbol{w}'; X_k)\| \leq \left\| \begin{bmatrix} \vdots \\ \gamma\phi_k\left(\left\langle \hat{\phi}_{n,k}, \boldsymbol{w}_n\right\rangle - \left\langle \check{\phi}_{n,k}, \boldsymbol{w}'_n\right\rangle\right) \\ \vdots \end{bmatrix} \right\| + \left\| \begin{bmatrix} \vdots \\ \phi_k\left\langle \phi_k, \boldsymbol{w}_n - \boldsymbol{w}'_n\right\rangle \\ \vdots \end{bmatrix} \right\| \tag{B.7}
$$

Recalling $\hat{\phi}_{n,k} = \arg\max_{\boldsymbol{a}'\in\mathcal{A}} \langle \boldsymbol{\phi}(S_{k+1},\boldsymbol{a}'), \boldsymbol{w}_n\rangle$ and $\check{\phi}_{n,k} = \arg\max_{\boldsymbol{a}''\in\mathcal{A}} \left\langle \boldsymbol{\phi}^(S_{k+1},\boldsymbol{a}''), \boldsymbol{w}'_n\right\rangle$, the

upper and lower bounds of the term $\left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_n \right\rangle - \left\langle \check{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}'_n \right\rangle$ are derived as

$$\max_{\boldsymbol{a}' \in \mathcal{A}} \langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}'), \boldsymbol{w}_n \rangle - \max_{\boldsymbol{a}'' \in \mathcal{A}} \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}''), \boldsymbol{w}'_n \right\rangle \leq \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}'), \boldsymbol{w}_n - \boldsymbol{w}'_n \right\rangle, \boldsymbol{a}' \in \mathcal{A} \tag{B.8}$$

and

$$\max_{\boldsymbol{a}' \in \mathcal{A}} \langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}'), \boldsymbol{w}_n \rangle - \max_{\boldsymbol{a}'' \in \mathcal{A}} \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}''), \boldsymbol{w}'_n \right\rangle \geq \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}''), \boldsymbol{w}_n - \boldsymbol{w}'_n \right\rangle, \boldsymbol{a}'' \in \mathcal{A}. \tag{B.9}$$

Based on (B.8), (B.9) and Cauchy-Schwarz inequality, we obtain

$$\left| \max_{\boldsymbol{a}' \in \mathcal{A}} \langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}'), \boldsymbol{w}_n \rangle - \max_{\boldsymbol{a}'' \in \mathcal{A}} \left\langle \boldsymbol{\phi}(S_{k+1}, \boldsymbol{a}''), \boldsymbol{w}'_n \right\rangle \right|^2 \leq \left\| \boldsymbol{w}_n - \boldsymbol{w}'_n \right\|^2. \tag{B.10}$$

The power of the first term in (B.7) is upper-bounded as

$$\left\| \begin{bmatrix} \vdots \\ \gamma \phi_k \left(\left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_n \right\rangle - \left\langle \check{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}'_n \right\rangle \right) \\ \vdots \end{bmatrix} \right\|^2 = \gamma^2 \sum_{n=1}^{N} \left\| \phi_k \left(\left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_n \right\rangle - \left\langle \check{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}'_n \right\rangle \right) \right\|^2 \tag{B.11a}$$

$$\leq \gamma^2 \sum_{n=1}^{N} \left| \left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_n \right\rangle - \left\langle \check{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}'_n \right\rangle \right|^2 \tag{B.11b}$$

$$\leq \gamma^2 \sum_{n=1}^{N} \left\| \boldsymbol{w}_n - \boldsymbol{w}'_n \right\|^2 := \gamma^2 \left\| \boldsymbol{w} - \boldsymbol{w}' \right\|^2 \tag{B.11c}$$

where inequality (B.11b) follows from $\|\boldsymbol{\phi}(s_n, a_n)\| \leq 1$, and inequality (B.11c) follows from (B.10).

Following similar arguments in (B.11), the power of the second term in (B.7)

$$\left\| \begin{bmatrix} \vdots \\ \phi_k \left\langle \phi_k, \boldsymbol{w}_n - \boldsymbol{w}'_n \right\rangle \\ \vdots \end{bmatrix} \right\|^2 \leq \left\| \boldsymbol{w} - \boldsymbol{w}' \right\|^2. \tag{B.12}$$

Based on (B.11c) and (B.12), we conclude

$$\left\| \boldsymbol{f}(\boldsymbol{w}; X_k) - \boldsymbol{f}(\boldsymbol{w}'; X_k) \right\| \leq (1 + \gamma) \left\| \boldsymbol{w} - \boldsymbol{w}' \right\| := L \left\| \boldsymbol{w} - \boldsymbol{w}' \right\|. \tag{B.13}$$

where $L = 1 + \gamma$.

Using (B.13), we obtain

$$\|\boldsymbol{f}(\boldsymbol{w}; X_k)\| = \|\boldsymbol{f}(\boldsymbol{w}; X_k) - \boldsymbol{f}(\boldsymbol{w}^*; X_k) + \boldsymbol{f}(\boldsymbol{w}^*; X_k)\| \leq L \|\boldsymbol{w} - \boldsymbol{w}^*\| + LG \tag{B.14}$$

where G is the upper bound of $\|\boldsymbol{f}(\boldsymbol{w}; X_k)\| / L$.

We obtain the upper bound of $\|\boldsymbol{f}(\boldsymbol{w}; X_k)\|$ as

$$\|\boldsymbol{f}(\boldsymbol{w}; X_k)\| = \left\| \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_k \left(r_{n,k} + \gamma \left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_n^* \right\rangle - \left\langle \boldsymbol{\phi}_k, \boldsymbol{w}_n^* \right\rangle \right) \\ \vdots \end{bmatrix} \right\| \tag{B.15a}$$

$$\leq \left\| \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_k r_{n,k} \\ \vdots \end{bmatrix} \right\| + \left\| \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_k \left(\gamma \left\langle \hat{\boldsymbol{\phi}}_{n,k}, \boldsymbol{w}_n^* \right\rangle - \left\langle \boldsymbol{\phi}_k, \boldsymbol{w}_n^* \right\rangle \right) \\ \vdots \end{bmatrix} \right\| \tag{B.15b}$$

$$\leq \sqrt{N} r_{\max} + (1 + \gamma) \|\boldsymbol{w}^*\| \tag{B.15c}$$

where the inequality (B.15b) follows from the triangle inequality, and inequality (B.15c) follows the fact that $r_n(S_k, A_k) \leq r_{\max}$, $\|\boldsymbol{\phi}_k\| \leq 1$, and $\|\hat{\boldsymbol{\phi}}_{n,k}\| \leq 1$, $n = 1, \ldots, N$. Therefore, we conclude that $G \leq \sqrt{N}(\|\boldsymbol{w}^*\| + r_{\max}/L)$.

Following similar procedures and the fact $\mathbb{E}_X \left[\bar{\boldsymbol{f}}(\boldsymbol{w}^*; X) \right] = 0$, we have

$$\left\| \mathbb{E}_X \left[\bar{\boldsymbol{f}}(\boldsymbol{w}^*; X) \right] - \mathbb{E}_X \left[\bar{\boldsymbol{f}}(\boldsymbol{w}^*; X) \right] \right\| \leq L \|\boldsymbol{w} - \boldsymbol{w}^*\| \tag{B.16}$$

and

$$\left\| \bar{\boldsymbol{f}}(\boldsymbol{w}^*; X) \right\| \leq L \|\boldsymbol{w} - \boldsymbol{w}^*\| + LG. \tag{B.17}$$

B.3 Proof of Lemma 4.2

Recalling the vector $\bar{\boldsymbol{f}}(\boldsymbol{w}) = \mathbb{E}_X[\bar{\boldsymbol{f}}(\boldsymbol{w}; X)]$ is obtained as

$$\bar{\boldsymbol{f}}(\boldsymbol{w}) = \mathbb{E}_X \begin{bmatrix} \vdots \\ \boldsymbol{\phi}(s,\boldsymbol{a})\boldsymbol{g}(\boldsymbol{w}_n; X) \\ \vdots \end{bmatrix}. \tag{B.18}$$

Then, we have

$$(\boldsymbol{w}-\boldsymbol{w}')^\top\big(\bar{\boldsymbol{f}}(\boldsymbol{w})-\bar{\boldsymbol{f}}(\boldsymbol{w}')\big) = \sum_{n=1}^{N}\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)^\top\big(\mathbb{E}_X[\boldsymbol{\phi}(s,\boldsymbol{a})\boldsymbol{g}(\boldsymbol{w}_n;X)] - \mathbb{E}_X\big[\boldsymbol{\phi}(s,\boldsymbol{a})\boldsymbol{g}(\boldsymbol{w}_n';X)\big]\big). \tag{B.19}$$

Therefore, we obtain the nth term on the right-hand side of (B.19) as

$$\begin{aligned}
&\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)^\top\big(\mathbb{E}_X[\boldsymbol{\phi}(s,\boldsymbol{a})\boldsymbol{g}(\boldsymbol{w}_n;X)] - \mathbb{E}_X\big[\boldsymbol{\phi}(s,\boldsymbol{a})\boldsymbol{g}(\boldsymbol{w}_n';X)\big]\big) \qquad \text{(B.20)}\\
&= \big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)^\top\mathbb{E}_X\Big[\gamma\boldsymbol{\phi}(s,\boldsymbol{a})\Big(\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')\boldsymbol{w}_n - \max_{\boldsymbol{a}''\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}'')\boldsymbol{w}_n'\Big) - \boldsymbol{\phi}^\top(s,a)\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\Big]\\
&= \gamma\mathbb{E}_X\Big[\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)^\top\boldsymbol{\phi}(s,\boldsymbol{a})\Big(\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')\boldsymbol{w}_n - \max_{\boldsymbol{a}''\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}'')\boldsymbol{w}_n'\Big)\Big] - \mathbb{E}_X\big[\boldsymbol{\phi}^\top(s,\boldsymbol{a})\big(\boldsymbol{w}_n-w_n'\big)\big]^2\\
&\le \gamma\sqrt{\mathbb{E}_X[\boldsymbol{\phi}^\top(s,\boldsymbol{a})(\boldsymbol{w}_n-\boldsymbol{w}_n')]^2}\sqrt{\mathbb{E}_X\Big[\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')\boldsymbol{w}_n - \max_{\boldsymbol{a}''\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}'')\boldsymbol{w}_n'\Big]^2}\\
&\quad - \mathbb{E}_X\big[\boldsymbol{\phi}^\top(s,\boldsymbol{a})\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\big]^2\\
&\le \gamma\sqrt{\mathbb{E}_X[\boldsymbol{\phi}^\top(s,\boldsymbol{a})(\boldsymbol{w}_n-\boldsymbol{w}_n')]^2}\sqrt{\mathbb{E}_X\Big[\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')(\boldsymbol{w}_n-\boldsymbol{w}_n')\Big]^2} - \mathbb{E}_X\big[\boldsymbol{\phi}^\top(s,\boldsymbol{a})\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\big]^2\\
&\le \sqrt{\mathbb{E}_X[\boldsymbol{\phi}^\top(s,\boldsymbol{a})(\boldsymbol{w}_n-\boldsymbol{w}_n')]^2}\,\frac{\gamma\mathbb{E}_X\Big[\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\Big]^2 - \mathbb{E}_X\big[\boldsymbol{\phi}^\top(s,\boldsymbol{a})\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\big]^2}{\gamma\sqrt{\mathbb{E}_X\Big[\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s,\boldsymbol{a})(\boldsymbol{w}_n-\boldsymbol{w}_n')\Big]^2} + \sqrt{\mathbb{E}_X[\boldsymbol{\phi}^\top(s,\boldsymbol{a})(\boldsymbol{w}_n-\boldsymbol{w}_n')]^2}}\\
&\le \frac{\gamma^2\mathbb{E}_X\Big[\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\Big]^2 - \mathbb{E}_X\big[\boldsymbol{\phi}^\top(s,\boldsymbol{a})\big(\boldsymbol{w}_n-\boldsymbol{w}_n'\big)\big]^2}{\frac{\gamma\sqrt{\mathbb{E}_X[\max_{\boldsymbol{a}'\in\mathcal{A}}\boldsymbol{\phi}^\top(s',\boldsymbol{a}')(\boldsymbol{w}_n-\boldsymbol{w}_n')]^2}}{\sqrt{\mathbb{E}_X[\boldsymbol{\phi}^\top(s,\boldsymbol{a})(w_n-w_n')]^2}}+1}\\
&\le -\frac{c_0}{2-c_0}\big\|\boldsymbol{w}_n-\boldsymbol{w}_n'\big\|^2 := -L\delta\big\|\boldsymbol{w}_n-\boldsymbol{w}_n'\big\|^2
\end{aligned}$$

where $\delta = \frac{c_0}{(2-c_0)L}$.

Stacking the N terms on the RHS of (B.20), we obtain

$$(\boldsymbol{w}-\boldsymbol{w}')^{\top}\big(\bar{\boldsymbol{f}}(\boldsymbol{w})-\bar{\boldsymbol{f}}(\boldsymbol{w}')\big) \leq -L\delta\left\|\boldsymbol{w}-\boldsymbol{w}'\right\|^2. \tag{B.21}$$

B.4 Proof of Lemma 4.3

To analyze the properties of accumulated gradient noise $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$, we introduce an auxiliary function $\hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k)$ such that

$$\bar{\boldsymbol{w}}_{k+T} = \bar{\boldsymbol{w}}_k + \boldsymbol{\Lambda}\sum_{t=k}^{T+k}\alpha_t \boldsymbol{f}(\boldsymbol{w}_k; X_t) + \hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1}). \tag{B.22}$$

Based on (4.14) and (B.22), we obtain the relation between $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ and $\hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ as

$$\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1}) = \hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1}) + \boldsymbol{\Lambda}\sum_{t=k}^{T+k}\alpha_t\big[\boldsymbol{f}(\boldsymbol{w}_k; X_t) - \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\big] \tag{B.23}$$

Taking expectation of (B.23) over the sample trajectory $X_{k:k+T-1}$ conditioning on $\boldsymbol{w}_k$, we have

$$\mathbb{E}[\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})] = \mathbb{E}\left[\boldsymbol{\Lambda}\sum_{t=k}^{T+k}\alpha_t\big[\boldsymbol{f}(\boldsymbol{w}_k; X_t) - \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\big]\right] + \mathbb{E}\Big[\hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1})\Big] \tag{B.24}$$

Based on the triangle inequality, we have

$$\left\|\mathbb{E}[\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})]\right\| \leq \left\|\mathbb{E}\left[\boldsymbol{\Lambda}\sum_{t=k}^{T+k-1}\alpha_t\big[\boldsymbol{f}(\boldsymbol{w}_k; X_t) - \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\big]\right]\right\| + \left\|\mathbb{E}\Big[\hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1})\Big]\right\|. \tag{B.25}$$

B.4.1 The Upper Bound of the First Term of (B.25)

Recalling Markov chain $\{S_k\}_k$ is irreducible and aperiodic. Based on [87, Lemma 3.11] and [86, eqs. (58) and (59)], we conclude that $\{X_k\}_k$ is an irreducible and aperiodic Markov chain. Using the irreducible and aperiodic properties and taking expectation over $X_{k:k+T-1}$, we have the following derivations

$$\left\|\mathbb{E}\left[\boldsymbol{\Lambda}\sum_{t=k}^{T+k-1}\alpha_t\big[\boldsymbol{f}(\boldsymbol{w}_k; X_t) - \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\big]\right]\right\| \tag{B.26a}$$

$$= \left\| \sum_{t=k}^{T+k-1} \alpha_t \big(\mathbb{E}[\boldsymbol{\Lambda} \boldsymbol{f}(\boldsymbol{w}_k; X_t)] - \mathbb{E}_{X_t} \big[\boldsymbol{\Lambda} \bar{\boldsymbol{f}}(\boldsymbol{w}_k; X_t) \big] \big) \right\| \tag{B.26b}$$

$$= \left\| \sum_{t=k}^{T+k-1} \alpha_t \sum_{X} \Big(P(X_t = X \,|\boldsymbol{w}_k) - \pi^s \mu_{\boldsymbol{a}}^s p_{\boldsymbol{a}}^{s,s'} \Big) \boldsymbol{\Lambda} \bar{\boldsymbol{f}}(\boldsymbol{w}_k; X) \right\| \tag{B.26c}$$

$$\leq \sum_{t=k}^{T+k-1} \alpha_t \sum_{X} \Big| P(X_t = X \,|\boldsymbol{w}_k) - \pi^s \mu_{\boldsymbol{a}}^s p_{\boldsymbol{a}}^{s,s'} \Big| \left\| \boldsymbol{\Lambda} \bar{\boldsymbol{f}}(\boldsymbol{w}_k; X) \right\| \tag{B.26d}$$

$$\leq L[\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G] \sum_{t=k}^{T+k-1} \alpha_t \sum_{X} \Big| P(X_t = X | \boldsymbol{w}_k) - \pi^s \mu_{\boldsymbol{a}}^s p_{\boldsymbol{a}}^{s,s'} \Big| \tag{B.26e}$$

$$\leq \alpha_k L T \lambda_1(T,k) [\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G] \tag{B.26f}$$

where (B.26c) follows that the random variable X is independent of $\boldsymbol{w}_k$ and the fact $\boldsymbol{\Lambda} \boldsymbol{f}(\boldsymbol{w}_k; X) = \boldsymbol{\Lambda} \boldsymbol{f}(\boldsymbol{w}_k; X)$, the inequality (B.26d) follows the facts $\|a\boldsymbol{w} + b\boldsymbol{w}'\| \leq |a| \, \|\boldsymbol{w}\| + |b| \, \|\boldsymbol{w}'\|$ and $\|\boldsymbol{\Lambda}\| = 1$, the inequality (B.26e) follows (B.17) and $\|\boldsymbol{\Lambda}\| = 1$, and (B.26f) follows from the decaying stepsize and Theorem 4.9 in [127]. Moreover, $\lambda_1(T,k)$ is defined as

$$\frac{1}{T} \sum_{t=k}^{T+k-1} 2c_1 \rho^t \leq \frac{2c_1 \rho^k}{T(1-\rho)} := \lambda_1(T,k) \tag{B.27}$$

where $c_1 > 0$ and $\rho \in (0,1)$.

B.4.2 The Upper Bound of the Second Term of (B.25)

Setting $T \leftarrow T+1$ in (B.22), we have that

$$\bar{\boldsymbol{w}}_{k+T+1} = \bar{\boldsymbol{w}}_k + \boldsymbol{\Lambda} \sum_{t=k}^{k+T} \alpha_t \boldsymbol{f}(\boldsymbol{w}_k; X_t) + \hat{\boldsymbol{\zeta}}_{T+1}(\boldsymbol{w}_k; X_{k:k+T}). \tag{B.28}$$

Subtracting (B.22) from (B.28), we obtain

$$\bar{\boldsymbol{w}}_{k+T+1} - \bar{\boldsymbol{w}}_{k+T} = \alpha_{k+T} \boldsymbol{\Lambda} \boldsymbol{f}(\boldsymbol{w}_k; X_{k+T}) + \hat{\boldsymbol{\zeta}}_{T+1}(\boldsymbol{w}_k; X_{k:k+T}) - \hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1}) \tag{B.29}$$

Substituting the iteration (4.12) into (B.29) and performing several algebraic manipulations, we obtain

$$\hat{\boldsymbol{\zeta}}_{T+1}(\boldsymbol{w}_k; X_{k:k+T}) = \hat{\boldsymbol{\zeta}}_T(\boldsymbol{w}_k; X_{k:k+T-1}) + \alpha_{k+T} \boldsymbol{\Lambda} [\boldsymbol{f}(\boldsymbol{w}_{k+T}; X_{k+T}) - \boldsymbol{f}(\boldsymbol{w}_k; X_{k+T})]. \tag{B.30}$$

Based on the triangle inequality and the fact $\|\mathbf{\Lambda}\| = 1$, we have

$$\left\|\hat{\boldsymbol{\zeta}}_{T+1}(\boldsymbol{w}_k; X_{k:k+T})\right\| \tag{B.31a}$$

$$\leq \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| + \alpha_{k+T} \left\|\boldsymbol{f}(\boldsymbol{w}_{k+T}; X_{k+T}) - \boldsymbol{f}(\boldsymbol{w}_k; X_{k+T})\right\| \tag{B.31b}$$

$$\leq \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| + \alpha_{k+T} L \left\|\boldsymbol{w}_{k+T} - \boldsymbol{w}_k\right\| \tag{B.31c}$$

$$= \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| + \alpha_{k+T} L \left\|\mathbf{\Lambda} \sum_{t=k}^{T+k-1} \alpha_t \boldsymbol{f}(\boldsymbol{w}_k; X_t) + \hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| \tag{B.31d}$$

$$\leq (1 + \alpha_{k+T} L) \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| + \alpha_{k+T} L \sum_{t=k}^{T+k-1} \alpha_t \left\|\boldsymbol{f}(\boldsymbol{w}_k; X_t)\right\| \tag{B.31e}$$

$$\leq (1 + \alpha_{k+T} L) \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| + \alpha_k^2 L^2 T [\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G] \tag{B.31f}$$

$$\leq \alpha_k^2 L^2 T^2 \lambda_2(T) [\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G] \tag{B.31g}$$

where the inequality (B.31g) follows from the facts $\hat{\boldsymbol{\zeta}}_1(\boldsymbol{w}_k; X_k) = 0$ and decaying stepsize α_k with $\lambda_2(T+1) := T^{-2} \sum_{t=1}^{T} t(1 + \alpha_1 L)^{T-t}$.

Therefore, the second term of (B.25) is upper-bounded by

$$\left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\| \leq \alpha_k^2 L^2 T^2 \lambda_2(T) [\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G]. \tag{B.32}$$

Substituting (B.26f) and (B.32) into (B.25), we have

$$\|\mathbb{E}\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})\| \leq \alpha_k L T [\lambda_1(T,k) + \alpha_k L T \lambda_2(T)] [\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G]. \tag{B.33}$$

We obtain the power of $\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})$ as

$$\|\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})\|^2 \tag{B.34a}$$

$$= \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1}) + \mathbf{\Lambda} \sum_{t=k}^{T+k-1} \alpha_t \left[\boldsymbol{f}(\boldsymbol{w}_k; X_t) - \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\right]\right\|^2 \tag{B.34b}$$

$$\leq 3 \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\|^2 + 3 \left\|\sum_{t=k}^{T+k-1} \alpha_t \boldsymbol{f}(\boldsymbol{w}_k; X_t)\right\|^2 + 3\alpha_k^2 L^2 T^2 \|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2 \tag{B.34c}$$

$$\leq 3 \left\|\hat{\boldsymbol{\zeta}}_{T}(\boldsymbol{w}_k; X_{k:k+T-1})\right\|^2 + 3\alpha_k^2 L^2 T^2 [\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G]^2 + 3\alpha_k^2 L^2 T^2 \|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2 \tag{B.34d}$$

$$\leq 3\alpha_k^2 L^2 T^2 \left[3 + 2\alpha_k^2 L^2 T^2 \lambda_2^2(T)\right] \|\boldsymbol{w}_k - \boldsymbol{w}^*\|^2 + 6\alpha_k^2 L^2 T^2 G^2 \left[1 + \alpha_k^2 L^2 T^2 \lambda_2^2(T)\right] \tag{B.34e}$$

where the inequality (B.34c) follows the facts (B.16), decaying stepsize α_k and $(a+b+c)^2 \leq 3a^2+3b^2+3c^2$, the inequality (B.34d) follows from (B.17), and the inequality (B.34e) is obtained by substituting (B.32) into (B.34d) and performing several algebraic manipulations.

B.5 Proof of Lemma 4.4

We recast the iteration (4.12) as

$$\bar{\boldsymbol{w}}_{k+1} = \bar{\boldsymbol{w}}_k + \alpha_k \boldsymbol{\Lambda} \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) = \bar{\boldsymbol{w}}_k + \alpha_k \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k) + \alpha_k \boldsymbol{\Lambda} \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) - \alpha_k \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)\,. \tag{B.35}$$

We derive the upper bound of the term $\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)$ as

$$\begin{aligned}
&\|\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)\| && \text{(B.36a)}\\
&\leq \|\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) - \boldsymbol{\Lambda} \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)\| + \|\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)\| && \text{(B.36b)}\\
&\leq L \|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\| + \|\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)\| && \text{(B.36c)}\\
&\leq L \|\boldsymbol{w}_k - \bar{\boldsymbol{w}}_k\| + \sqrt{N} r_{\max} && \text{(B.36d)}\\
&= L \|\boldsymbol{\Delta} \boldsymbol{w}_k\| + \sqrt{N} r_{\max} && \text{(B.36e)}
\end{aligned}$$

where (B.36b) is based on triangle inequality, (B.36c) is based on Lemma 4.1, and (B.36d) follows from the following facts

$$\begin{aligned}
&\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k) && \text{(B.37)}\\
&= \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_{n,k} \left[\frac{1}{N} \sum_{n=1}^{N} r_{n,k} + \left(\gamma \hat{\boldsymbol{\phi}}_{n,k} - \boldsymbol{\phi}_{n,k} \right)^{\top} \bar{w}_{n,k} \right] \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_{n,k} \left[r_{n,k} + \left(\gamma \hat{\boldsymbol{\phi}}_{n,k} - \boldsymbol{\phi}_{n,k} \right)^{\top} \bar{\boldsymbol{w}}_{n,k} \right] \\ \vdots \end{bmatrix}\\
&= \begin{bmatrix} \vdots \\ \boldsymbol{\phi}_{n,k} \left[\frac{1}{N} \sum_{n=1}^{N} r_{n,k} - r_{n,k} \right] \\ \vdots \end{bmatrix} && \text{(B.38)}
\end{aligned}$$

with $\left\| \frac{1}{N} \sum_{n=1}^{N} r_{n,k} - r_{n,k} \right\| \leq r_{\max}$ and $\left\| \boldsymbol{\phi}_{n,k} \right\| \leq 1$.

Subtracting (B.35) from (4.13), we obtain

$$\boldsymbol{\Delta w}_{k+1} = (\boldsymbol{B} \otimes I_d)\, \boldsymbol{\Delta w}_k + \alpha_k [\nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)] - \alpha_k [\boldsymbol{\Lambda} \nabla \boldsymbol{f}(\boldsymbol{w}_k; X_k) - \nabla \boldsymbol{f}(\bar{\boldsymbol{w}}_k; X_k)] \quad \text{(B.39)}$$

B.5.1 Decaying Stepsize

Denote the second-largest singular value of $\boldsymbol{B}$ by c_2. Based on $\|(\boldsymbol{B} \otimes I_d)\, \boldsymbol{\Delta w}_k\| \leq c_2 \|\boldsymbol{\Delta w}_k\|$, (B.39) and Lemma 4.1, we have

$$\|\boldsymbol{\Delta w}_{k+1}\| \leq (c_2 + 2\alpha_k L) \|\boldsymbol{\Delta w}_k\| + \alpha_k \sqrt{N} r_{\max} \leq (c_2 + 2\alpha_1 L) \|\boldsymbol{\Delta w}_k\| + \alpha_k \sqrt{N} r_{\max} \quad \text{(B.40)}$$

where $\alpha_1 \geq \alpha_k$.

Telescoping the series in (B.40), we have

$$\|\boldsymbol{\Delta w}_k\| \leq (c_2 + 2\alpha_1 L)^k \|\boldsymbol{\Delta w}_1\| + \sqrt{N} r_{\max} \sum_{t=1}^{k-1} \alpha_t (c_2 + 2\alpha_1 L)^{k-t-1} \quad \text{(B.41a)}$$

$$\leq (c_2 + 2\alpha_1 L)^k \|\boldsymbol{\Delta w}_1\| + \bar{c}_3 \sqrt{N} r_{\max} e_k \quad \text{(B.41b)}$$

where (B.41b) follows Theorem 2.8 in [146] with $e_k = \max\{\alpha_k, ((1 + c_2 + 2\alpha_1 L)/2)^k\}$ and positive constant $\bar{c}_3$.

When the decaying stepsize $\alpha_k = \bar{\alpha} L^{-1}/k$ is used, we have $(1 + c_2 + 2\alpha_1 L)/2 \leq (1 + c_2 + 2\bar{\alpha})/2$. We set $c_2 + 2\bar{\alpha} < 1$ such that the consensus error converges. To simplify (B.41b), we introduce the following lemma.

Lemma B.1. *There always exists a positive constant such that $c'\rho^k \leq c''/k$ where $\rho \in (0, 1)$.*

Proof Define a function $f(k) = \log(\frac{c''}{c'}) - \log(k) - k\log(\rho)$. Setting the first-order derivative of $f(k)$ equal to zero, namely $\partial_k f(k) = -\frac{1}{k} - \log(\rho) = 0$. We have $k = \frac{1}{-\log(\rho)}$. Substituting $k = \frac{1}{-\log(\rho)}$ into $f(k)$, we obtain the minimum value of $f(k)$ as $f_{\min} = \log(\frac{\exp(1)c''}{c'} \log(\frac{1}{\rho}))$. When $c'' \geq \frac{c'}{\exp(1)\log(\frac{1}{\rho})}$ and $\rho \in (0, 1)$, we have $f(k) \geq f_{\min} \geq 0$ and $c'\rho^k \leq \frac{c''}{k}$.

Based on Lemma B.1, there exists

$$c_3 \geq \bar{c}_3 \max\left\{\frac{1}{L}, \frac{1}{c \exp(1) \log(\frac{2}{1 + c_2 + 2\bar{\alpha}})}\right\} \quad \text{(B.42)}$$

such that $\bar{c}_3 e_k \le \alpha_k L c_3$.

Based on (B.42), we obtain

$$\|\boldsymbol{\Delta w}_k\| \le (c_2 + 2\alpha_1 L)^k \|\boldsymbol{\Delta w}_1\| + \alpha_k L\sqrt{N} r_{\max} c_3 \le (c_2 + 2\bar{\alpha})^k \|\boldsymbol{\Delta w}_1\| + \alpha_k L\sqrt{N} r_{\max} c_3. \tag{B.43}$$

B.5.2 Constant Stepsize

When the decaying stepsize $\alpha_k = \bar{\alpha} L^{-1}$ is used with $\bar{\alpha} < (1 - c_2)/4$, we obtain the convergence rate of consensus error from (B.41a) as

$$\left\|\boldsymbol{\Delta w}^k\right\| \le (c_2 + 2\bar{\alpha})^k \left\|\boldsymbol{\Delta w}^1\right\| + \frac{2\bar{\alpha}\sqrt{N} r_{\max}}{L(1 - c_2)}. \tag{B.44}$$

B.6 Proof of Lemma 4.5

To perform the finite-sample analysis, we define a T-step Lyapunov function as

$$\mathbb{C}_{T,k} = \frac{1}{2} \sum_{t=k}^{T+k-1} \|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\|^2 . \tag{B.45}$$

Therefore, the drift of T-step Lyapunov function is obtained as

$$\begin{aligned}
\mathbb{C}_{T,k+1} - \mathbb{C}_{T,k} &= \frac{1}{2} \sum_{t=k}^{T+k-1} \|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\|^2 - \frac{1}{2} \sum_{t=k}^{T+k-1} \|\bar{\boldsymbol{w}}_t - \boldsymbol{w}^*\|^2 && \text{(B.46a)} \\
&= \frac{1}{2} \|\bar{\boldsymbol{w}}_{k+T} - \boldsymbol{w}^*\|^2 - \frac{1}{2} \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2 && \text{(B.46b)} \\
&= \frac{1}{2} \|\bar{\boldsymbol{w}}_{k+T} - \bar{\boldsymbol{w}}_k\|^2 + (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top (\bar{\boldsymbol{w}}_{k+T} - \bar{\boldsymbol{w}}_k) && \text{(B.46c)} \\
&= \frac{1}{2} \left\| \boldsymbol{\Lambda} \sum_{t=k}^{T+k-1} \alpha_t \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k) + \boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1}) \right\|^2 \\
&\quad + \left\langle \bar{\boldsymbol{w}}_k - \boldsymbol{w}^*, \boldsymbol{\Lambda} \sum_{t=k}^{T+k-1} \alpha_t \bar{\boldsymbol{f}}(\boldsymbol{w}_k) + \boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1}) \right\rangle && \text{(B.46d)} \\
&= \alpha_k^2 T^2 \left\|\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\right\|^2 + \|\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})\|^2 \\
&\quad + \langle \bar{\boldsymbol{w}}_k - \boldsymbol{w}^*, \boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1}) \rangle + \sum_{t=k}^{T+k-1} \alpha_t \left\langle \bar{\boldsymbol{w}}_k - \boldsymbol{w}^*, \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k) \right\rangle && \text{(B.46e)}
\end{aligned}$$

where the equality (B.46d) follows from (4.14), and (B.46e) follows from the facts $\langle \boldsymbol{\Lambda}, \bar{\boldsymbol{w}}_k - \boldsymbol{w}^* \rangle =$

$\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*$, $(a+b)^2 \le 2a^2 + 2b^2$ and decaying stepsize.

Based on (B.16), the first term of (B.46e) is upper-bounded by

$$\alpha_k^2 T^2 \left\|\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k)\right\|^2 \le \alpha_k^2 L^2 T^2 \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2. \tag{B.47}$$

Given $\boldsymbol{w}_k$, we observe that the third term in (B.46e) is a function of joint observation trajectory $X_{k:k+T-1}$. Taking expectation over $X_{k:k+T-1}$ conditioning on $\boldsymbol{w}_k$, the conditional expectation of the third term in (B.46e) is upper-bounded by

$$\begin{aligned}
&\mathbb{E}[(\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})] && \text{(B.48a)}\\
&= (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \mathbb{E}[\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})] && \text{(B.48b)}\\
&\le \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| \, \|\mathbb{E}[\boldsymbol{\zeta}_T(\boldsymbol{w}_k; X_{k:k+T-1})]\| && \text{(B.48c)}\\
&\le \alpha_k LT[\lambda_1(T,k) + \alpha_k LT\lambda_2(T)] \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| \, (\|\boldsymbol{w}_k - \boldsymbol{w}^*\| + G) && \text{(B.48d)}\\
&= \alpha_k LT[\lambda_1(T,k) + \alpha_k LT\lambda_2(T)] \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| \, (\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| + G)\\
&\quad + \alpha_k LT[\lambda_1(T,k) + \alpha_k LT\lambda_2(T)] \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| \, \|\boldsymbol{\Delta}\boldsymbol{w}_k\| && \text{(B.48e)}\\
&\le \alpha_k LT[2\alpha_k LT + 2\lambda_1(T,k) + 2\alpha_k LT\lambda_2(T)] \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2 + \frac{\lambda_1^2(T,k) + \alpha_k^2 L^2 T^2 \lambda_2^2(T)}{4T^2} \|\boldsymbol{\Delta}\boldsymbol{w}_k\|^2\\
&\quad + \alpha_k LTG^2[2\lambda_1(T,k) + 2\alpha_k LT\lambda_2(T)] && \text{(B.48f)}
\end{aligned}$$

where the inequality (B.48b) follows the Cauchy-Schwarz inequality for dot product, (B.48c) is based on (B.33), and (B.48f) follows from the elementary inequality $ab \le a^2 + \frac{b^2}{4}$.

Based on Lemma 4.2 and the fact $\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}^*) = \mathbb{E}_X\left[\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}^*; X)\right] = 0$, we obtain the upper bound of the fourth term in (B.46e) as

$$\begin{aligned}
&\sum_{t=k}^{T+k-1} \alpha_t (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k) && \text{(B.49a)}\\
&= \sum_{t=k}^{T+k-1} \alpha_t (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \left[\nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k) - \nabla \bar{\boldsymbol{f}}(\bar{\boldsymbol{w}}_k) + \nabla \bar{\boldsymbol{f}}(\bar{\boldsymbol{w}}_k) - \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}^*)\right] && \text{(B.49b)}\\
&\le \alpha_k LT \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| \, \|\boldsymbol{\Delta}\boldsymbol{w}_k\| + \sum_{t=k}^{T+k-1} \alpha_t (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \left[\nabla \bar{\boldsymbol{f}}(\bar{\boldsymbol{w}}_k) - \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}^*)\right] && \text{(B.49c)}\\
&\le \alpha_k LT \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| \, \|\boldsymbol{\Delta}\boldsymbol{w}_k\| - \sum_{t=k}^{T+k-1} \alpha_t L\delta \, \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2 && \text{(B.49d)}
\end{aligned}$$

$$\leq \alpha_k^2 L^2 T^2 \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 + \frac{1}{4T^2}\left\|\boldsymbol{\Delta w}_k\right\|^2 - \sum_{t=k}^{T+k-1} \alpha_t L\delta \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 \tag{B.49e}$$

where (B.49c) follows from the Cauchy-Schwarz inequality, (B.49d) follows from Lemma 4.2, and (B.49e) follows from the elementary inequality $ab \leq a^2 + \frac{b^2}{4}$.

B.6.1 Decaying Stepsize

Using Riemann sum and the decaying stepsize $\alpha_k = cL^{-1}/k$, we obtain the lower bound of $\sum_{t=k}^{k+T-1} \alpha_t$ as

$$\sum_{t=k}^{k+T-1} \frac{\alpha_t}{\alpha_k} \geq \int_0^T \frac{k}{k+t} dt = k \log\left(\frac{k+T}{k}\right) \geq \log(1+T). \tag{B.50}$$

Substituting (B.50) into (B.49e), we obtain

$$\sum_{t=k}^{T+k-1} \alpha_t (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k) \leq \alpha_k^2 L^2 T^2 \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 + \frac{1}{4T^2}\left\|\boldsymbol{\Delta w}_k\right\|^2 - \alpha_k L\delta \log(1+T) \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2. \tag{B.51}$$

Combining (B.34e), (B.47), (B.48), and (B.51) and taking expectation conditioning on $\boldsymbol{w}_k$, we obtain

$$\mathbb{E}[\mathbb{C}_{T_\epsilon,k+1} - \mathbb{C}_{T_\epsilon,k}] \leq \alpha_k LT\lambda_6(T,\alpha_k) \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 + \lambda_3(T,\alpha_k)\left\|\boldsymbol{\Delta w}_k\right\|^2 + 2\alpha_k LTG^2\lambda_4(T,\alpha_k) \tag{B.52}$$

where $\lambda_3(T,\alpha_k)$, $\lambda_4(T,\alpha_k)$, and $\lambda_6(T,\alpha_k)$ are respectively defined as

$$\lambda_3(T,\alpha_k) = 18\alpha_k^2 L^2 T^2 + 12\alpha_k^4 L^4 T^4 \lambda_2^2(T) + \frac{1}{4T^2}\left[1 + \lambda_1^2(T,k) + \alpha_k^2 L^2 T^2 \lambda_2^2(T)\right] \tag{B.53}$$

$$\lambda_4(T,\alpha_k) = \lambda_1(T,k) + 3\alpha_k LT + \alpha_k LT\lambda_2(T) + 3\alpha_k^3 L^3 T^3 \lambda_2^2(T) \tag{B.54}$$

and

$$\lambda_6(T,\alpha_k) = 2\lambda_1(T,k) - \delta\frac{\log(1+T)}{T} + 2\alpha_k LT\left[11 + \lambda_2(T) + 6\alpha_k^2 L^2 T^2 \lambda_2^2(T)\right]. \tag{B.55}$$

Based on (B.27), we have $2\lambda_1(T,k) - \delta T^{-1}\log(1+T) = T^{-1}[4c_1\rho^k(1-\rho)^{-1} - \delta\log(1+T)]$. Moreover, (B.27) is a monotonically increasing term of k. Therefore, there exist $\epsilon > 0$ and T_ϵ such that

$$2\lambda_1(T_\epsilon, 1) - \delta\frac{\log(1+T_\epsilon)}{T_\epsilon} \leq -2\epsilon. \tag{B.56}$$

Given T_ϵ, the third term of $\lambda_6(T_\epsilon, \alpha_k)$ is a monotonically increasing function of α_k. Hence, there exists an $\alpha_\epsilon > 0$ such that

$$2\alpha_k L T_\epsilon \left[11 + \lambda_2(T_\epsilon) + 6\alpha_k^2 L^2 T_\epsilon^2 \lambda_2^2(T_\epsilon)\right] \leq \epsilon \tag{B.57}$$

where $\alpha_k \leq \alpha_\epsilon$.

Combining the facts (B.56) and (B.57), we conclude $\lambda_6(T_\epsilon, \alpha_k) \leq -\epsilon$. As a result, we obtain

$$\mathbb{E}[\mathbb{C}_{T_\epsilon,k+1} - \mathbb{C}_{T_\epsilon,k}] \leq -\epsilon\alpha_k L T_\epsilon \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 + \lambda_3(T_\epsilon, \alpha_k) \left\|\boldsymbol{\Delta w}^k\right\|^2 + 2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon, \alpha_k). \tag{B.58}$$

Taking iterated expectation over $\boldsymbol{w}_k$, we have

$$\mathbb{E}\left[\mathbb{C}_{T_\epsilon,k+1} - \mathbb{C}_{T_\epsilon,k}\right] \leq -\epsilon\alpha_k L T_\epsilon \mathbb{E}\left[\left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2\right] + \lambda_3(T_\epsilon, \alpha_k)\mathbb{E}\left[\left\|\boldsymbol{\Delta w}_k\right\|^2\right] + 2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon, \alpha_k). \tag{B.59}$$

B.6.2 Constant Stepsize

Using the constant stepsize $\alpha_k = \bar{\alpha} L^{-1}$, we obtain the upper bound of the fourth term in (B.46e) as

$$\sum_{t=k}^{T+k-1} \alpha_t (\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*)^\top \nabla \bar{\boldsymbol{f}}(\boldsymbol{w}_k) \leq \alpha_k^2 L^2 T^2 \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 + \frac{1}{4T^2} \left\|\boldsymbol{\Delta w}_k\right\|^2 - \alpha_k L T \delta \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2. \tag{B.60}$$

Combining (B.34e), (B.47), (B.48), and (B.60) and taking expectation conditioning on $\boldsymbol{w}_k$, we obtain

$$\mathbb{E}[\mathbb{C}_{T_\epsilon,k+1} - \mathbb{C}_{T_\epsilon,k}] \leq \alpha_k L T \bar{\lambda}_6(T, k) \left\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\right\|^2 + \lambda_3(T, \alpha_k) \left\|\boldsymbol{\Delta w}_k\right\|^2 + 2\alpha_k L T G^2 \lambda_4(T, \alpha_k) \tag{B.61}$$

where $\bar{\lambda}_6(T, k)$ is obtained as

$$\bar{\lambda}_6(T, k) = 2\lambda_1(T, k) - \delta + 2\bar{\alpha} T \left[11 + \lambda_2(T) + 6\bar{\alpha}^2 T^2 \lambda_2^2(T)\right]. \tag{B.62}$$

Based on (B.27), there exist ϵ, T_ϵ and $\bar{\alpha} \leq \alpha_\epsilon L$ such that $2\lambda_1(T_\epsilon, 1) - \delta \leq -2\epsilon$ and $2\bar{\alpha} T_\epsilon [11 + \lambda_2(T_\epsilon) + 6\bar{\alpha}^2 T^2 \lambda_2^2(T_\epsilon)] \leq \epsilon$. Moreover, (B.27) is a monotonically increasing term of k. Therefore,

we obtain

$$\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k+1} - \mathbb{C}_{T_\epsilon,k}\big] \leq -\epsilon\alpha_k L T_\epsilon \mathbb{E}\Big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\Big] + \lambda_3(T_\epsilon,\alpha_k)\mathbb{E}\big[\|\boldsymbol{g}\boldsymbol{w}_k\|^2\big] + 2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon,\alpha_k). \tag{B.63}$$

B.7 Proof of Theorem 4.6

Based on the iteration (4.14), we obtain the relation between $\|\bar{\boldsymbol{w}}_{k+T} - \boldsymbol{w}^*\|$ and $\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|$ as

$$\begin{aligned}
\|\bar{\boldsymbol{w}}_{k+T} - \boldsymbol{w}^*\| &\leq (1+\alpha_{k+T-1}L)\,\|\bar{\boldsymbol{w}}_{k+T-1} - \boldsymbol{w}^*\| + \alpha_{k+T-1}LG && \text{(B.64a)}\\
&\leq (1+\alpha_k L)\,\|\bar{\boldsymbol{w}}_{k+T-1} - \boldsymbol{w}^*\| + \alpha_k LG && \text{(B.64b)}\\
&\leq (1+\alpha_k L)^T\,\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\| + \alpha_k LG\sum_{\tau=0}^{T-1}(1+\alpha_k L)^\tau && \text{(B.64c)}
\end{aligned}$$

where the inequality (B.64b) follows from $\alpha_{k+T-1} \leq \alpha_k$, and the inequality (B.64c) is obtained by telescoping the iterates in (B.64b).

Based on the elementary inequality $(a+b)^2 \leq 2a^2 + 2b^2$, we have

$$\|\bar{\boldsymbol{w}}_{k+T} - \boldsymbol{w}^*\|^2 \leq 2(1+\alpha_k L)^{2T}\,\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2 + 2\alpha_k^2 L^2 G^2\left(\sum_{\tau=0}^{T-1}(1+\alpha_k L)^\tau\right)^2. \tag{B.65}$$

Taking summation of (B.65) over $T = 0,\ldots,T_\epsilon - 1$ and dividing both sides by two, we have

$$\begin{aligned}
\mathbb{C}_{T_\epsilon,k} &\leq \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2 \sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_k L)^{2\tau} + \alpha_k^2 L^2 G^2 \sum_{j=1}^{T_\epsilon-1}\left(\sum_{\tau=0}^{j-1}(1+\alpha_k L)^\tau\right)^2 && \text{(B.66a)}\\
&\leq \|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2 \sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_k L)^{2\tau} + \alpha_k^2 L^2 G^2 \sum_{j=0}^{T_\epsilon-1}\left(\sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_k L)^\tau\right)^2 && \text{(B.66b)}
\end{aligned}$$

where the inequality (B.66b) follows from $j \leq T_\epsilon$.

Recalling the fact $\bar{\boldsymbol{w}}_k = \boldsymbol{\Lambda}\boldsymbol{w}_k$ and $\|\boldsymbol{\Lambda}\boldsymbol{w}_k - \boldsymbol{w}^*\|^2 \leq \|\boldsymbol{w}_k - \boldsymbol{w}^*\|$, inequality (B.66b) also verifies the boundedness condition of $\mathbb{E}[\mathbb{C}_{T_\epsilon,1}]$ as long as the term $\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|^2$ is bounded.

Based on (B.66b), we have

$$\frac{\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k}\big]}{\sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_1 L)^{2\tau}} \leq \frac{\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k}\big]}{\sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_k L)^{2\tau}} \tag{B.67a}$$

$$\leq \mathbb{E}\Big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\Big] + \alpha_k^2 L^2 G^2 \frac{\sum_{j=0}^{T_\epsilon-1}\Big(\sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_k L)^{\tau}\Big)^2}{\sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_k L)^{2\tau}} \tag{B.67b}$$

$$\leq \mathbb{E}\Big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\Big] + \alpha_k^2 L^2 G^2 T_\epsilon^2 \tag{B.67c}$$

where the inequality (B.67c) follows from the elementary inequality $(\sum_{\tau=0}^{T_\epsilon-1} x_\tau)^2 \leq T_\epsilon \sum_{\tau=0}^{T_\epsilon-1} x_\tau^2$.

Based on the fact $1+x \leq \exp(x)$, we have

$$\sum_{\tau=0}^{T_\epsilon-1}(1+\alpha_1 L)^{2\tau} \leq \frac{(1+\alpha_1 L)^{2T_\epsilon}}{\alpha_1 L} \leq \frac{\exp(2\alpha_1 L T_\epsilon)}{\alpha_1 L}. \tag{B.68}$$

Substituting (B.68) into (B.67c), we have

$$\frac{\alpha_1 L}{\exp(2\alpha_1 L T_\epsilon)}\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k}\big] \leq \mathbb{E}\Big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\Big] + \alpha_k^2 L^2 T_\epsilon^2 G^2. \tag{B.69}$$

Substituting (B.69) into (B.59), we obtain

$$\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k+1}\big] \leq \left[1 - \frac{\epsilon\alpha_1\alpha_k L^2 T_\epsilon}{\exp(2\alpha_1 L T_\epsilon)}\right]\mathbb{E}\big[\mathbb{C}_{T_\epsilon,k}\big] + \lambda_3(T_\epsilon,\alpha_k)\mathbb{E}\big[\|\boldsymbol{\Delta w}_k\|^2\big] + \epsilon\alpha_k^3 L^3 T_\epsilon^3 G^2 + 2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon,\alpha_k). \tag{B.70}$$

Based on Lemma 4.4, we have

$$\mathbb{E}\big[\|\boldsymbol{\Delta w}_k\|^2\big] \leq 2(c_2+2c)^{2k}\mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] + 2Nc_3^2 r_{\max}^2 \alpha_k^2 L^2 \tag{B.71a}$$

$$\leq 2\alpha_k^2 L^2\Big(c_9^2\mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] + Nc_3^2 r_{\max}^2\Big) \tag{B.71b}$$

where the inequality (B.71b) follows from Lemma B.1 with $c_9 \geq \bar{\alpha}^{-1}\exp(-1)\log^{-1}(\frac{1}{c_2+2\bar{\alpha}})$.

Recalling the definitions of $\lambda_1(T_\epsilon,k)$ in (B.27) and $\lambda_4(T_\epsilon,\alpha_k)$ in (B.53). Using Lemma B.1, we have

$$2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon,\alpha_k) \leq 2\alpha_k L T_\epsilon G^2\big[\alpha_k L c_{10} + 3\alpha_k L T_\epsilon + \alpha_k L T_\epsilon \lambda_2(T_\epsilon) + 3\alpha_k^3 L^3 T_\epsilon^3 \lambda_2^2(T_\epsilon)\big] \tag{B.72}$$

where c_{10} is obtained as

$$c_{10} \geq \bar{\alpha}^{-1} \exp(-1) \log^{-1}\left(\frac{T_\epsilon(1-\rho)}{2c_1}\right). \tag{B.73}$$

Based on (B.71b) and (B.72), we obtain the upper bound of the last three terms on the right-hand side of (B.70) as

$$\lambda_3(T_\epsilon, \alpha_k)\mathbb{E}\left[\|\boldsymbol{\Delta w}_k\|^2\right] + \epsilon\alpha_k^3 L^3 T_\epsilon^3 G^2 + 2\alpha_k L T_\epsilon G^2 \lambda_4(T_\epsilon, \alpha_k) \tag{B.74a}$$

$$\leq 2\alpha_k^2 L^2 \left(c_9^2 \mathbb{E}\left[\|\boldsymbol{\Delta w}_1\|^2\right] + N c_3^2 r_{\max}^2\right) + \epsilon\alpha_k^3 L^3 T_\epsilon^3 G^2 \tag{B.74b}$$

$$+ 2\alpha_k L T_\epsilon G^2 \left[\alpha_k L c_{10} + 3\alpha_k L T_\epsilon + \alpha_k L T_\epsilon \lambda_2(T_\epsilon) + 3\alpha_k^3 L^3 T_\epsilon^3 \lambda_2^2(T_\epsilon)\right] \tag{B.74c}$$

$$\leq \alpha_k^2 L^2 \left[2c_9^2 \mathbb{E}\left[\|\boldsymbol{\Delta w}_1\|^2\right] + 2N c_3^2 r_{\max}^2 + \epsilon\alpha_k L T_\epsilon^3 G^2 + 2c_{10} T_\epsilon G^2 + 6T_\epsilon + 2T_\epsilon \lambda_2(T_\epsilon) + 6\alpha_k L T_\epsilon^3 \lambda_2^2(T_\epsilon)\right] \tag{B.74d}$$

$$\leq \alpha_k^2 L^2 c_4 \tag{B.74e}$$

where c_4 is obtained as

$$c_4 = 2c_9^2 \mathbb{E}\left[\|\boldsymbol{\Delta w}_1\|^2\right] + 2N c_3^2 r_{\max}^2 + \epsilon\bar{\alpha} T_\epsilon^3 G^2 + 2c_{10} T_\epsilon G^2 + 6T_\epsilon + 2T_\epsilon \lambda_2(T_\epsilon) + 6\bar{\alpha} T_\epsilon^3 \lambda_2^2(T_\epsilon). \tag{B.75}$$

Substituting (B.74e) into (B.70) and setting $\alpha_k = \bar{\alpha} L^{-1}/k$, we obtain

$$\mathbb{E}\left[\mathbb{C}_{T_\epsilon, k+1}\right] \leq \left[1 - \frac{\epsilon\bar{\alpha}^2 T_\epsilon}{\exp(2\bar{\alpha} T_\epsilon) k}\right] \mathbb{E}\left[\mathbb{C}_{T_\epsilon, k}\right] + \frac{c_4}{k^2}. \tag{B.76}$$

Recalling the facts $\bar{\alpha} < \min\{L\alpha_\epsilon, \frac{1-c_2}{2}\}$ and ϵ is a small positive constant, we have $\frac{\epsilon\bar{\alpha}^2 T_\epsilon}{\exp(2\bar{\alpha} T_\epsilon)} \leq 1$. Based on Lemma 2.3 in [146], we conclude that

$$\mathbb{E}\left[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\right] \leq 2\mathbb{E}\left[\mathbb{C}_{T_\epsilon}^k\right] \leq \frac{2c_5}{k} \tag{B.77}$$

where c_5 is obtained as

$$c_5 = \max\left\{\mathbb{E}\left[\mathbb{C}_{T_\epsilon, 1}\right], \frac{\exp(2\bar{\alpha} T_\epsilon) c_4}{\epsilon\bar{\alpha}^2 T_\epsilon}\right\}. \tag{B.78}$$

B.8 Proof of Theorem 4.8

Based on (B.44), constant stepsize $\alpha_k \equiv \bar{\alpha} L^{-1}$ and $\bar{\alpha} < (1 - c_2)/4$, we obtain

$$\mathbb{E}\big[\|\boldsymbol{\Delta w}_k\|^2\big] \leq 2(c_2 + 2\bar{\alpha})^{2k} \mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] + \frac{8N\bar{\alpha}^2 r_{\max}^2}{L^2(1 - c_2)} \tag{B.79}$$

Following similar arguments in Appendix B.7, there exist ϵ and T_ϵ satisfying $2\lambda_1(T_\epsilon, k) - \delta \leq -2\epsilon$ such that

$$\mathbb{E}\big[\mathbb{C}_{T_\epsilon, k+1}\big] \leq c_6 \mathbb{E}\big[\mathbb{C}_{T_\epsilon, k}\big] + \lambda_3\Big(T_\epsilon, \frac{\bar{\alpha}}{L}\Big) \mathbb{E}\big[\|\boldsymbol{\Delta w}_k\|^2\big] + \epsilon \bar{\alpha}^3 T_\epsilon^3 G^2 + 2cT_\epsilon G^2 \lambda_4\Big(T_\epsilon, \frac{\bar{\alpha}}{L}\Big) \tag{B.80}$$

where c_6 is obtained as

$$c_6 = 1 - \frac{\epsilon \bar{\alpha} T_\epsilon}{\sum_{\tau=0}^{T_\epsilon - 1} (1 + \bar{\alpha})^{2\tau}}. \tag{B.81}$$

Substituting (B.79) into (B.80) and performing several algebraic manipulations, we have

$$\mathbb{E}\big[\mathbb{C}_{T_\epsilon, k+1}\big] \leq c_6 \mathbb{E}\big[\mathbb{C}_{T_\epsilon, k}\big] + 2\lambda_3\Big(T_\epsilon, \frac{\bar{\alpha}}{L}\Big) \mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] (c_2 + 2\bar{\alpha})^{2k} + \frac{4\bar{\alpha} c_1 G^2}{1 - \rho} \rho^k + \bar{\alpha}^2 N c_7 \tag{B.82}$$

where c_7 is obtained as

$$c_7 = \frac{8 r_{\max}^2}{L^2(1 - c_2)} \lambda_3\Big(T_\epsilon, \frac{\bar{\alpha}}{L}\Big) + \frac{T_\epsilon^2 G^2}{N} \big[6 + 2\lambda_2(T_\epsilon) + \epsilon \bar{\alpha} T_\epsilon + 6cT_\epsilon \lambda_2^2(T_\epsilon)\big]. \tag{B.83}$$

Using (B.82), we obtain the upper bound of T_ϵ-step Lyapunov function as

$$\begin{aligned}
&\mathbb{E}\big[\mathbb{C}_{T_\epsilon, k+1}\big] \\
&\leq c_6^k \mathbb{E}\big[\mathbb{C}_{T_\epsilon, 1}\big] + 2 \sum_{t=1}^{k} c_6^{k-t} (c_2 + 2\bar{\alpha})^{2(t-1)} \lambda_3\Big(T_\epsilon, \frac{\bar{\alpha}}{L}\Big) \mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] + \sum_{t=1}^{k} c_6^{k-t} \rho^{t-1} \frac{4\bar{\alpha} c_1 G^2}{1 - \rho} + \bar{\alpha}^2 N c_7 \sum_{t=1}^{k} c_6^{k-t} \\
&\leq c_6^k \mathbb{E}\big[\mathbb{C}_{T_\epsilon, 1}\big] + \frac{\sum_{\tau=0}^{T_\epsilon - 1} (1 + \bar{\alpha})^{2\tau}}{\epsilon T_\epsilon} \bar{\alpha} N c_7 + \frac{c_6^k - (c_2 + 2\bar{\alpha})^{2k}}{c_6 - (c_2 + 2\bar{\alpha})^2} 2\lambda_3\Big(T_\epsilon, \frac{\bar{\alpha}}{L}\Big) \mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big] + \frac{c_6^k - \rho^k}{c_6 - \rho} \frac{4\bar{\alpha} c_1 G^2}{1 - \rho} \\
&\leq c_6^k \mathbb{E}\big[\mathbb{C}_{T_\epsilon}^1\big] + \frac{N}{2} \lambda_5(k) + \frac{\bar{\alpha} N}{\epsilon T_\epsilon} \sum_{\tau=0}^{T_\epsilon - 1} (1 + \bar{\alpha})^{2\tau} c_7
\end{aligned} \tag{B.84}$$

where $\lambda_5(k)$ is obtained as

$$\lambda_5(k) = \frac{4\lambda_3\big(T_\epsilon, \frac{\bar{\alpha}}{L}\big)\mathbb{E}\big[\|\boldsymbol{\Delta w}_1\|^2\big]\left[c_6^k - (c_2 + 2\bar{\alpha})^{2k}\right]}{N\big[c_6 - (c_2 + 2\bar{\alpha})^2\big]} + \frac{8\bar{\alpha}c_1G^2\big(c_6^k - \rho^k\big)}{N(c_6 - \rho)(1-\rho)}. \tag{B.85}$$

Since $\frac{1}{2}\mathbb{E}[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2] \leq \mathbb{E}[\mathbb{C}_{T_\epsilon,k}]$, we have

$$\frac{1}{N}\mathbb{E}\Big[\|\bar{\boldsymbol{w}}_k - \boldsymbol{w}^*\|^2\Big] \leq \frac{2\mathbb{E}\big[\mathbb{C}_{T_\epsilon,1}\big]}{N}c_6^{k-1} + \lambda_5(k-1) + \frac{\bar{\alpha}N}{\epsilon T_\epsilon}\sum_{\tau=0}^{T_\epsilon-1}(1+\bar{\alpha})^{2\tau}c_7. \tag{B.86}$$

Appendix C

Related Proofs of Chapter 5

C.1 Upper Bound of Lyapunov Drift-Plus-Penalty Function

Substituting (5.9) into (5.15), we have

$$D(\boldsymbol{q}_k^{\mathrm{A}}) = \frac{1}{2}\mathbb{E}_{\iota_{1,k}}\left[\left\|\boldsymbol{q}_{k+1}^{\mathrm{A}}\right\|^2 - \left\|\boldsymbol{q}_k^{\mathrm{A}}\right\|^2 | \boldsymbol{q}_k^{\mathrm{A}}\right] \tag{C.1a}$$

$$\leq \frac{1}{2}\sum_{n=1}^{N}\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k}^2 + r_{n,k}^2 | \boldsymbol{q}_k^{\mathrm{A}}\right] + \frac{1}{2}\sum_{n=1}^{N} q_{n,k}^{\mathrm{A}}\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k} - r_{n,k} | \boldsymbol{q}_k^{\mathrm{A}}\right] \tag{C.1b}$$

$$\leq N + \frac{1}{2}\sum_{n=1}^{N} q_{n,k}^{\mathrm{A}}\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k} - r_{n,k} | \boldsymbol{q}_k^{\mathrm{A}}\right] \tag{C.1c}$$

where (C.1b) follows from the fact $([a-b]^+ + c)^2 \leq a^2 + b^2 + c^2 + 2a(c-b)$ with $a, b, c \geq 0$, and (C.1c) follows from the facts (5.8) and $\nu_{n,k} \in (0, 1)$.

C.2 Proof of Theorem 5.2

The major steps follow the proof of [30, Theorem 4.2], and we will only sketch the proof for our formulated optimization problem. The optimization problem (5.17) is non-convex; therefore, we are motivated to seek a suboptimal solution to problem (5.17) within the feasible region. Recalling Lemma 5.1, we have the following inequality

$$D(\boldsymbol{q}_k^{\mathrm{A}}) + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}}) | \boldsymbol{q}_k^{\mathrm{A}}\right] \leq N + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}}) | \boldsymbol{q}_k^{\mathrm{A}}\right] + \sum_{n=1}^{N} q_{n,k}^{\mathrm{A}}\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k} - r_{n,k} | \boldsymbol{q}_k^{\mathrm{A}}\right]. \tag{C.2}$$

Note that $\boldsymbol{q}_k^{\mathrm{A}}$ is independent of the random sources in $\iota_{1,k}$, we can simplify (C.2) as

$$D(\boldsymbol{q}_k^{\mathrm{A}}) + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})\right] \leq N + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})\right] + \sum_{n=1}^{N} q_{n,k}^{\mathrm{A}}\mathbb{E}_{\iota_{1,k}}\left[\nu_{n,k} - r_{n,k}\right] \tag{C.3}$$

When the traffic arrival rate vector $[\bar{\nu}_1, \ldots, \bar{\nu}_N]$ is in the stable region of system, and the random sources in $\iota_{1,k} = \{\boldsymbol{h}_{n,k}, E_k^{\mathrm{HAV}}, \nu_{n,k}\}_{n=1}^N$ is independent and identically distributed over slots, we have

$$\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})\right] \leq F^{\mathrm{SOPT}} + \epsilon \tag{C.4a}$$

$$\mathbb{E}_{\iota_{1,k}}\left[r_{n,k}\right] \geq \bar{\nu}_n + \epsilon \tag{C.4b}$$

where F^{SOPT} is the maximum suboptimal value of (5.17), and ϵ can be chosen arbitrarily close to zero [30, Appendix 4.A].

Due to the minimum requirement of SINR, the expected GEE is lower-bounded as

$$\mathbb{E}\left[F(P_k^{\mathrm{BST}})\right] \geq F^{\min}. \tag{C.5}$$

Substituting (C.4) into (C.3) and setting $\epsilon_0 \to 0$, we obtain

$$D(\boldsymbol{q}_k^{\mathrm{A}}) + V\mathbb{E}_{\iota_{1,k}}\left[F(P_k^{\mathrm{BST}})\right] \leq N + VF^{\mathrm{SOPT}} - \epsilon \sum_{n=1}^{N} q_{n,k}^{\mathrm{A}}. \tag{C.6}$$

In order to prove the first part of Theorem 5.2, we first take the iterated expectation over endogenous (i.e., $\boldsymbol{q}_k^{\mathrm{A}}$) and exogenous (i.e., $\iota_{1,k}$) random sources and perform some algebraic manipulations of (C.6) as

$$\mathbb{E}\left[D(\boldsymbol{q}_k^{\mathrm{A}})\right] \leq N + V\left(F^{\mathrm{SOPT}} - \mathbb{E}\left[F(P_k^{\mathrm{BST}})\right]\right) - \epsilon \sum_{n=1}^{N} \mathbb{E}\left[q_{n,k}^{\mathrm{A}}\right]. \tag{C.7a}$$

Recalling the definition of drift function in (5.15), we have

$$\frac{1}{2}\mathbb{E}\left[\left\|\boldsymbol{q}_{k+1}^{\mathrm{A}}\right\|^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|\boldsymbol{q}_k^{\mathrm{A}}\right\|^2\right] \leq N + V\left(F^{\mathrm{SOPT}} - \mathbb{E}\left[F(P_k^{\mathrm{BST}})\right]\right) - \epsilon \sum_{n=1}^{N} \mathbb{E}\left[q_{n,k}^{\mathrm{A}}\right] \tag{C.8a}$$

$$\leq N + V\left(F^{\mathrm{SOPT}} - F^{\min}\right) - \epsilon \sum_{n=1}^{N} \mathbb{E}\left[q_{n,k}^{\mathrm{A}}\right] \tag{C.8b}$$

where (C.8b) follows from (C.5).

Taking telescope summing of (C.8b) over $k = 0, \ldots, K-1$ and dropping the nonnegative term

$q_{n,k}^{\mathrm{A}}$, $n = 1, \ldots, N$, we have

$$\mathbb{E}\left[\left\|\boldsymbol{q}_K^{\mathrm{A}}\right\|^2\right] \leq 2K\left[N + V\left(F^{\mathrm{SOPT}} - F^{\min}\right)\right] + \mathbb{E}\left[\left\|\boldsymbol{q}_0^{\mathrm{A}}\right\|^2\right] \tag{C.9}$$

Since $\mathbb{E}[\|\boldsymbol{q}_K^{\mathrm{A}}\|^2] - \mathbb{E}^2[\|\boldsymbol{q}_K^{\mathrm{A}}\|] \geq 0$ and $q_{n,K}^{\mathrm{A}} \leq \|\boldsymbol{q}_K^{\mathrm{A}}\|$, we have

$$\mathbb{E}\left[q_{m,k}^{\mathrm{A}}\right] \leq \mathbb{E}\left[\|\boldsymbol{q}_K^{\mathrm{A}}\|\right] \leq \sqrt{2K\left[N + V\left(F^{\mathrm{SOPT}} - F^{\min}\right)\right] + \mathbb{E}\left[\left\|\boldsymbol{q}_0^{\mathrm{A}}\right\|^2\right]}. \tag{C.10}$$

Based on (C.10), we observe that the backlog of each access queue increase at a rate of $O(\sqrt{K})$. Therefore, we conclude that access queues are mean rate stable.

Taking telescope summing of (C.8a) over $k = 0, \ldots, K-1$ and perform several algebraic manipulations, we have

$$V\sum_{k=0}^{K-1}\mathbb{E}\left[F(P_k^{\mathrm{BST}})\right] \leq KN + KVF^{\mathrm{SOPT}} + \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{q}_0^{\mathrm{A}}\|^2\right]. \tag{C.11}$$

Dividing both side of (C.11) by KV and setting $K \to \infty$, we obtain (5.18).

Taking telescope summing of (C.8b) over $k = 0, \ldots, K-1$, we obtain

$$\begin{aligned}\epsilon\sum_{k=0}^{K-1}\sum_{n=1}^{N}\mathbb{E}\left[q_{n,k}^{\mathrm{A}}\right] \leq & KN + KV\left(F^{\mathrm{SOPT}} - F^{\min}\right) + \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{q}_0^{\mathrm{A}}\|^2\right] - \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{q}_K^{\mathrm{A}}\|^2\right] \\ \leq & KN + KV\left(F^{\mathrm{SOPT}} - F^{\min}\right) + \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{q}_0^{\mathrm{A}}\|^2\right].\end{aligned} \tag{C.12}$$

Dividing both side of (C.12) by ϵK and setting $K \to \infty$, we obtain (5.19).

C.3 Proof of Theorem 5.3

Introducing a set of auxiliary variables $c_{3,n,k} > 0$, we obtain an equivalent form of the optimization problem (5.21) as

$$\min_{\{\boldsymbol{w}_{n,k}, \lambda_{1,n,k}, c_{3,n,k}\}_{n=1}^{N}} \quad VF(P_k^{\mathrm{BST}}) - \sum_{n=1}^{N} q_{n,k}^{\mathrm{A}} c_{3,n,k} \tag{C.13a}$$

$$\text{s.t. } \mathrm{SINR}_{n,k} \geq \lambda_{1,n,k}, n = 1, \ldots, N \tag{C.13b}$$

$$\mathrm{SINR}_{n,k} \geq \gamma_n^{\mathrm{REQ}}, n = 1, \ldots, N \tag{C.13c}$$

$$\sum_{n=1}^{N} \left\| \boldsymbol{w}_{n,k} \right\|^2 \leq P^{\max} \tag{C.13d}$$

$$c_{3,n,k}\left[1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])\right] \leq 1, n=1,\ldots,N. \tag{C.13e}$$

The KKT conditions related to the proof are listed as

$$c_{4,n,k}\left(c_{3,n,k}\left[1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])\right]-1\right)=0, n=1,\ldots,N \tag{C.14a}$$

$$c_{4,n,k}\left[1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])\right]-q_{n,k}^{\mathrm{A}}=0, n=1,\ldots,N \tag{C.14b}$$

$$c_{4,n,k} \geq 0, n=1,\ldots,N. \tag{C.14c}$$

Based on (C.14b), we obtain the expression for $c_{4,n,k}$ as

$$c_{4,n,k}=\frac{q_{n,k}^{\mathrm{A}}}{1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])}>0. \tag{C.15}$$

Hence, we can obtain the expression of γ_n from (C.14a) as

$$c_{3,n,k}=\frac{1}{1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])}. \tag{C.16}$$

Using (C.15) and (C.16), we obtain the following optimization problem

$$\min_{\{\boldsymbol{w}_{n,k},\lambda_{1,n,k}\}_{n=1}^{N}} \sum_{n=1}^{N} c_{4,n,k}\left(c_{3,n,k}\left[1+\exp(-c_{1,n}[10\log_{10}(\mathrm{SINR}_{n,k})-c_{2,n}])\right]-1\right)+VF(P_k^{\mathrm{BST}}) \tag{C.17a}$$

$$\text{s.t. } \mathrm{SINR}_{n,k} \geq \lambda_{1,n,k}, n=1,\ldots,N \tag{C.17b}$$

$$\mathrm{SINR}_{n,k} \geq \gamma_n^{\mathrm{REQ}}, n=1,\ldots,N \tag{C.17c}$$

$$\sum_{n=1}^{N} \left\| \boldsymbol{w}_{n,k} \right\|^2 \leq P^{\max}. \tag{C.17d}$$

Together with (C.15) and (C.16), the optimization problem (C.17) shares the same KKT conditions with the optimization problem (5.21).

C.4 Activeness of Constraints in (5.26)

Recalling the fact in (5.22), we only need to discuss the relation between (5.26d) and (5.26e). Let $\{\boldsymbol{w}_{n,k}^*, \lambda_{1,n,k}^*, \lambda_{2,n,k}^*\}_{n=1}^N$ be the set of optimal solution. Suppose that some of the constraints in (5.26e) are inactive, and denote the set of indices of inactive constraints by

$$\mathcal{IDX} = \left\{ n \middle| \sqrt{\sigma_n^2 + \sum_{l \neq n}^{N} \left| \boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{l,k}^* \right|^2} < \lambda_{2,n,k}^*, n = 1, \ldots, N \right\}. \tag{C.18}$$

Then, there is a positive constant $c_{5,n,k} > 1$ such that $\sqrt{\sigma_n^2 + \sum_{l \neq n}^{N} \left| \boldsymbol{h}_{n,k}^{\mathrm{H}} \boldsymbol{w}_{l,k}^* \right|^2} = \lambda_{2,n,k}^* / c_{5,n,k}$, $n \in \mathcal{IDX}$. To keep corresponding constraints in (5.26d) unchanged, we set $\lambda_{1,n,k}^* \leftarrow c_{5,n,k}^2 \lambda_{1,n,k}^*$. Recalling the fact $c_{5,n,k} > 1$, $c_{5,n,k}^2 \lambda_{1,n,k}^* > \lambda_{1,n,k}^*$. Then, the term $c_{5,n,k}^2 \lambda_{1,n,k}^*$ achieves a smaller objective value of (5.26a), which contradicts optimality of $\lambda_{1,n,k}^*$. Hence, we conclude that the constraints in (5.26e) are active.

C.5 Convergence Property of SABF Algorithm

The successive approximation procedures are used in Algorithm 6. Let $\mathcal{F}^\tau$ and $\mathcal{OBJ}^\tau$ respectively denote the approximate feasible region and optimal objective value of problem (5.30) in each iteration τ. Based on lines 6–8, the solution $\{\boldsymbol{w}_{n,k}^{\tau-1}, \lambda_{1,n,k}^{\tau-1}, c_{3,n,k}^{\tau-1}, c_{4,n,k}^{\tau-1}\}_{n=1}^N$ is in the feasible region of next iteration τ, namely $\mathcal{F}^\tau$. Since the solution $\{\boldsymbol{w}_{n,k}^{\tau}, \lambda_{1,n,k}^{\tau}, c_{3,n,k}^{\tau}, c_{4,n,k}^{\tau}\}_{n=1}^N$ is a minimizer to problem (5.30) in the feasible region $\mathcal{F}^\tau$, we obtain

$$\mathcal{OBJ}^\tau \leq \mathcal{OBJ}^{\tau-1}. \tag{C.19}$$

Since the maximum transmission power of BST is $P^{\max}$, the objective value of (5.30) is lower-bounded. Thus, Algorithm 6 generates a set of solution $\{\boldsymbol{w}_{n,k}^{\tau}, \lambda_{1,n,k}^{\tau}, c_{3,n,k}^{\tau}, c_{4,n,k}^{\tau}\}_{n=1}^N$ that converges as τ goes to infinity. Since the objection function of (5.30) is convex, we obtain that the convergent solution $\{\boldsymbol{w}_{n,k}^{\infty}, \lambda_{1,n,k}^{\infty}, c_{3,n,k}^{\infty}, c_{4,n,k}^{\infty}\}_{n=1}^N$ is a KKT point of the problem (5.30) via similar arguments in [24, Proposition 3.2]. Based on Theorem 5.3, we conclude that the solution $\{\boldsymbol{w}_{n,k}^{\infty}, \lambda_{1,n,k}^{\infty}\}_{n=1}^N$ is a KKT point of the problem (5.21).

Appendix D

Related Proofs of Chapter 6

D.1 Upper Bound of Two Time-Scale Lyapunov Drift-Plus-Penalty Function

Taking the telescoping summation over $k = 1, \ldots, T$ for the (m, n)th access queue in (6.9), we obtain the one-frame dynamic equation of the (m, n)th access queue as

$$q^{\mathrm{A}}_{m,n,1}[i+1] = q^{\mathrm{A}}_{m,n,1}[i] + \sum_{k=1}^{T} \nu_{m,n,k}[i] - \sum_{k=1}^{T} r_{m,n,k}[i]. \tag{D.1}$$

Based on (D.1), the one-frame drift of the (m, n)th access queue is upper-bounded as

$$\frac{1}{2}\left[\left(q^{\mathrm{A}}_{m,n,1}[i+1]\right)^2 - \left(q^{\mathrm{A}}_{m,n,1}[i]\right)^2\right] \leq \frac{(\nu^{\max})^2 + (r^{\max})^2}{2} T + q^{\mathrm{A}}_{m,n,1}[i] \sum_{k=1}^{T}\left[\nu_{m,n,k}[i] - r_{m,n,k}[i]\right] \tag{D.2}$$

where the inequality holds due to the facts in (6.11).

Following a similar argument, we obtain the upper-bound of the one-frame drift of the (m, n)th processing queue as

$$\frac{1}{2}\left[\left(q^{\mathrm{U}}_{m,n,1}[i+1]\right)^2 - \left(q^{\mathrm{U}}_{m,n,1}[i]\right)^2\right] \leq \frac{(s^{\max})^2 + (r^{\max})^2}{2} T + q^{\mathrm{U}}_{m,n,1}[i] \sum_{k=1}^{T}\left[r_{m,n,k}[i] - s_{m,n,k}[i]\right] \tag{D.3}$$

where the inequality holds due to the facts in (6.11).

Based on (D.2) and (D.3), we obtain the upper-bound of one-frame Lyapunov drift-plus-penalty function conditioning on $\boldsymbol{q}^{\mathrm{A}}[i]$ and $\boldsymbol{q}^{\mathrm{U}}[i]$ in (6.19).

D.2 Proof of Theorem 6.1

Let $\{\boldsymbol{w}^*_{m,n,k}[i], a^*_{m,n}[i], \varpi^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ denote minimizer to RHS of (6.19) under the constraints in (6.6) and (6.12)–(6.14). Let $\{\tilde{\boldsymbol{w}}_{m,n,k}[i], \tilde{a}_{m,n}[i], \tilde{\varpi}_{m\to l,k}[i]\}_{m,l,n,k,i}$ denote the set of feasible resource allocation variables such that $\bar{\boldsymbol{\nu}} + \epsilon\mathbf{1} \leq \mathbb{E}_{\iota_{2,k}[i]}[\boldsymbol{r}_k[i]] \leq \bar{\boldsymbol{s}} - \epsilon\mathbf{1}$.

Substituting the minimizer $\{\boldsymbol{w}^*_{m,n,k}[i], a^*_{m,n}[i], \varpi^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ into the RHS of (6.19), we obtain

$$D\big(\boldsymbol{q}^{\mathrm{A}}[i], \boldsymbol{q}^{\mathrm{U}}[i]\big) + V\sum_{m=1}^{M}\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}\Big[F(P^{\mathrm{BST}}_{m,k}[i])|\boldsymbol{q}^{\mathrm{A}}[i], \boldsymbol{q}^{\mathrm{U}}[i]\Big] \tag{D.4}$$

$$\leq Tc_1 + V\sum_{m=1}^{M}\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}\Big[F(P^{\mathrm{BST}}_{m,k}[i])|\boldsymbol{q}^{\mathrm{A}}[i], \boldsymbol{q}^{\mathrm{U}}[i]\Big] + \sum_{k=1}^{T}\mathbb{E}^{\top}_{\iota_{2,k}[i]}\Big[\boldsymbol{\nu}_k[i] - \boldsymbol{r}_k[i]|\boldsymbol{q}^{\mathrm{A}}[i]\Big]\boldsymbol{q}^{\mathrm{A}}[i] + \sum_{k=1}^{T}\mathbb{E}^{\top}_{\iota_{2,k}[i]}\Big[\boldsymbol{r}_k[i] - \boldsymbol{s}_k[i]|\boldsymbol{q}^{\mathrm{U}}[i]\Big]\boldsymbol{q}^{\mathrm{U}}[i] \tag{D.5}$$

$$\leq Tc_1 + V\sum_{m=1}^{M}\sum_{k=1}^{T}\mathbb{E}_{\iota_{2,k}[i]}\Big[F(P^{\mathrm{BST}}_{m,k}[i])|\boldsymbol{q}^{\mathrm{A}}[i], \boldsymbol{q}^{\mathrm{U}}[i]\Big] - \epsilon T\mathbf{1}^{\top}_{MN\times 1}\Big[\boldsymbol{q}^{\mathrm{A}}[k] + \boldsymbol{q}^{\mathrm{U}}[k]\Big] \tag{D.6}$$

where the inequality (D.6) follows the fact that $\{\boldsymbol{w}^*_{m,n,k}[i], a^*_{m,n}[i], \varpi^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ is a feasible solution under the constraints in (6.6) and (6.12)–(6.14).

Note the endogenous (i.e., $\boldsymbol{q}^{\mathrm{A}}[i]$ and $\boldsymbol{q}^{\mathrm{U}}[i]$) and exogenous (i.e., $\iota_{2,k}[i]$) random sources are independent. Rearranging (D.6) and taking iterated expectation over all random sources, we obtain an upper bound of the one-frame Lyapunov drift function as

$$\mathbb{E}\Big[D\big(\boldsymbol{q}^{\mathrm{A}}[i], \boldsymbol{q}^{\mathrm{U}}[i]\big)\Big] \leq Tc_1 + 2TVF - \epsilon T\mathbf{1}^{\top}_{MN\times 1}\mathbb{E}\Big[\boldsymbol{q}^{\mathrm{A}}[i] + \boldsymbol{q}^{\mathrm{U}}[i]\Big] \tag{D.7}$$

based on the bounded GEE (6.20).

Taking telescoping summation over $i = 0, 1, \ldots, I-1$ for (D.7) and performing several algebraic manipulations, we obtain the upper bound of queue backlogs as

$$\epsilon T\sum_{i=1}^{I}\mathbf{1}^{\top}\mathbb{E}\Big[\boldsymbol{q}^{\mathrm{A}}[i] + \boldsymbol{q}^{\mathrm{U}}[i]\Big] \tag{D.8a}$$

$$\leq \frac{1}{2}\mathbb{E}\Big[\|\boldsymbol{q}^{\mathrm{A}}[0]\|^2 - \|\boldsymbol{q}^{\mathrm{A}}[I]\|^2 + \|\boldsymbol{q}^{\mathrm{U}}[0]\|^2 - \|\boldsymbol{q}^{\mathrm{U}}[I]\|^2\Big] + ITc_1 + 2ITVF \tag{D.8b}$$

$$\leq \frac{1}{2}\mathbb{E}\Big[\|\boldsymbol{q}^{\mathrm{A}}[0]\|^2 + \|\boldsymbol{q}^{\mathrm{U}}[0]\|^2\Big] + ITc_1 + 2ITVF \tag{D.8c}$$

where (D.8c) is due to the nonnegative term $\|\boldsymbol{q}^{\mathrm{A}}[I]\|^2 + \|\boldsymbol{q}^{\mathrm{U}}[I]\|^2$.

Dividing both sides of (D.8c) by ϵIT, we obtain

$$\frac{1}{I}\sum_{i=1}^{I}\mathbf{1}^{\top}\mathbb{E}\left[\boldsymbol{q}^{\mathrm{A}}[i]+\boldsymbol{q}^{\mathrm{U}}[i]\right] \leq \frac{c_1+2VF}{\epsilon} + \frac{1}{2\epsilon IT}\mathbb{E}\left[\|\boldsymbol{q}^{\mathrm{A}}[0]\|^2 + \|\boldsymbol{q}^{\mathrm{U}}[0]\|^2\right]. \tag{D.9}$$

Note that the initial queue backlogs $\boldsymbol{q}^{\mathrm{A}}[0]$ and $\boldsymbol{q}^{\mathrm{U}}[0]$ are fixed. Setting $I \to \infty$, we obtain

$$\limsup_{I\to\infty}\frac{1}{I}\sum_{i=1}^{I}\mathbf{1}^{\top}\mathbb{E}\left[\boldsymbol{q}^{\mathrm{A}}[i]+\boldsymbol{q}^{\mathrm{U}}[i]\right] \leq \frac{c_1+2VF}{\epsilon} < \infty. \tag{D.10}$$

The backlogs of access queues and processing queues are nonnegative due to the constraints in (6.12) and queue dynamic functions in (6.9) and (6.10). Based on the nonnegative queue backlogs and (D.10), we conclude that

$$\limsup_{I\to\infty}\frac{1}{I}\sum_{i=1}^{I}\mathbb{E}\left[q^{\mathrm{A}}_{m,n,1}[i]+q^{\mathrm{U}}_{m,n,1}[i]\right] \leq \frac{c_1+2VF}{\epsilon} < \infty \tag{D.11}$$

such that the constraints in (6.15) are satisfied.

Now, we prove the inequalities in (6.22). Based on (D.10), we obtain that the nonnegative queue backlogs of access queues and processing queues satisfy

$$\limsup_{I\to\infty}\frac{1}{I}\sum_{i=1}^{I}\mathbf{1}^{\top}_{MN\times 1}\mathbb{E}\left[\boldsymbol{q}^{\mathrm{A}}[i]\right] \leq \frac{c_1+2VF}{\epsilon} < \infty \tag{D.12}$$

and

$$\limsup_{I\to\infty}\frac{1}{I}\sum_{i=1}^{I}\mathbf{1}^{\top}_{MN\times 1}\mathbb{E}\left[\boldsymbol{q}^{\mathrm{U}}[i]\right] \leq \frac{c_1+2VF}{\epsilon} < \infty. \tag{D.13}$$

Based on (D.12), (D.13), and Theorem 2.8 in [30], we conclude that the access queues and processing queues are mean-rate stable. Furthermore, the necessary conditions for mean-rate stable access queues and processing queues are obtained as [30, Theorem 2.5]

$$\bar{\nu}_{m,n} \leq \limsup_{I\to\infty}\frac{1}{IT}\sum_{i=1}^{I}\sum_{k=1}^{T}\mathbb{E}\left[r_{m,n,k}[i]\right] \leq \bar{s}_{m,n}. \tag{D.14}$$

Hence, we obtain a relaxed time-average GEE (R-TAGEE) minimization problem as

$$F^* = \min_{\{\boldsymbol{w}_{m,n,k}[i], a_{m,n}[i], \varpi_{m\to l,k}[i]\}_{m,l,n,k,i}} \lim_{I\to\infty} \frac{1}{IT} \sum_{i=1}^{I}\sum_{k=1}^{T}\sum_{m=1}^{M} \mathbb{E}\big[F(P_{m,k}^{\mathrm{BST}}[i])\big] \tag{D.15a}$$

$$\text{s.t. (6.6), (6.12) – (6.14) and (D.14)} \tag{D.15b}$$

where F^* is the optimal value of the R-TAGEE minimization problem.

Since the constraints in (6.16b) are a subset of the constraints in (D.15b), the optimal value of TAGEE minimization problem (6.16) is lower-bounded by the optimal value of R-TAGEE minimization problem (D.15) as

$$F^* \le \lim_{I\to\infty} \frac{1}{IT} \sum_{i=1}^{I}\sum_{k=1}^{T}\sum_{m=1}^{M} \mathbb{E}\big[F(P_{m,k}^{\mathrm{BST}}[i])\big]. \tag{D.16}$$

Therefore, we establish the first inequality in (6.22).

Based on the arguments in [30], when the random sources in $\iota_{2,k}[i]$ are independent and identically distributed over different slots, there is an optimal solution $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}$ to R-TAGEE minimization problem (D.15) that almost-surely satisfies $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}$ is a function of current random sources $\iota_{2,k}[i]$ and $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ guarantees that $\bar{v}_{m,n} \le \mathbb{E}_{\iota_{2,k}[i]}\big[r_{m,n,k}[i]\big] \le \bar{s}_{m,n}$ and $F^* = \sum_{m=1}^{M} \mathbb{E}\Big[F(P_{m,k}^{\mathrm{BST}}[i])\Big]$.

Note that $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ is not a minimizer to the RHS of (D.5). Substituting $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ into (D.5) and taking iterated expectation, we obtain

$$\mathbb{E}\big[D(\boldsymbol{q}^{\mathrm{A}}[i], \boldsymbol{q}^{\mathrm{U}}[i])\big] + V\sum_{m=1}^{M}\sum_{k=1}^{T} \mathbb{E}\big[F(P_{m,k}^{\mathrm{BST}}[i])\big] \tag{D.17a}$$

$$\le Tc_1 + V\sum_{k=1}^{T} \mathbb{E}[\boldsymbol{\nu}_k[i] - \boldsymbol{r}_k[i]] + \sum_{k=1}^{T} \mathbb{E}[\boldsymbol{r}_k[i] - \boldsymbol{s}_k[i]] \tag{D.17b}$$

$$\le Tc_1 + TVF^* \tag{D.17c}$$

where the value of left-hand side (D.17b) is obtained by $\{\boldsymbol{w}^*_{m,n,k}[i], a^*_{m,n}[i], \varpi^*_{m\to l,k}[i]\}_{m,l,n,k,i}$; and the inequality (D.17c) is based on the two facts: 1) $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ is a function of current random sources $\iota_{2,k}[i]$; and 2) $\{\tilde{\boldsymbol{w}}^*_{m,n,k}[i], \tilde{a}^*_{m,n}[i], \tilde{\varpi}^*_{m\to l,k}[i]\}_{m,l,n,k,i}$ guarantees

that $\bar{\nu}_{m,n} \leq \mathbb{E}_{\iota_{2,k}[i]}\left[r_{m,n,k}[i]\right] \leq \bar{s}_{m,n}$ and $F^* = \sum_{m=1}^{M} \mathbb{E}\Big[F(P_{m,k}^{\mathrm{BST}}[i])\Big]$.

Taking telescoping summation over $i = 0, 1, \ldots, I-1$ over (D.17c) and dividing both sides by ITV, we obtain

$$\frac{1}{IT}\sum_{i=1}^{I}\sum_{k=1}^{T}\sum_{m=1}^{M}\mathbb{E}\big[F(P_{m,k}^{\mathrm{BST}}[i])\big] - \frac{1}{2ITV}\left\|\boldsymbol{q}^{\mathrm{A}}[0]\right\|^2 - \frac{1}{2ITV}\left\|\boldsymbol{q}^{\mathrm{U}}[0]\right\|^2 \leq F^* + \frac{c_1}{V}. \tag{D.18}$$

Letting $I \to \infty$, we obtain the second inequality of (6.22) due to the fixed backlogs $\boldsymbol{q}^{\mathrm{A}}[0]$ and $\boldsymbol{q}^{\mathrm{U}}[0]$.

D.3 Proof of the Activeness of Constraints in (6.32c)

Let $\{\boldsymbol{w}_{m,n,k}^*[i], \varpi_{m\to l,k}^*[i]\}$ denote the set of optimal beamforming vectors and exchanged NRE variables given $\theta_k[i]$. Suppose that the (m,n)th constraint in (6.32c) is inactive, i.e.,

$$\frac{\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,n,k}^*[i]}{\sqrt{\exp(q_{m,n,k}^{\mathrm{A}}[i]\theta_k[i]) - 1}} > \sqrt{I_{m,n,k}^{\mathrm{INTRA}}[i] + I_{m,n,}^{\mathrm{INTER}}[i] + \sigma_{m,n}^2}. \tag{D.19}$$

Hence, we introduce an auxiliary variable $c_{6,m,n}$ such that

$$\gamma_{m,n}\frac{\boldsymbol{h}_{m,n,k}^{\mathrm{H}}[i]\boldsymbol{w}_{m,n,k}^*[i]}{\sqrt{\exp(q_{m,n,k}^{\mathrm{A}}[i]\theta_k[i]) - 1}} = \sqrt{I_{m,n,k}^{\mathrm{INTRA}}[i] + I_{m,n,}^{\mathrm{INTER}}[i] + \sigma_{m,n}^2}. \tag{D.20}$$

Based on (D.19) and (D.20), we obtain $c_{6,m,n} < 1$. Setting a new solution $\{\tilde{\boldsymbol{w}}_{m,n,k}^*[i], \tilde{\varpi}_{m\to l,k}^*[i]\}$ such that $\tilde{\boldsymbol{w}}_{m,n,k}[i] = c_{6,m,n}\boldsymbol{w}_{m,n,k}^*[i]$ and $\tilde{\varpi}_{m\to l,k}^*[i] = \varpi_{m\to l,k}^*[i]$. The new solution satisfies all the constraints in (6.32b)–(6.32e). Moreover, the new solution $\{\tilde{\boldsymbol{w}}_{m,n,k}^*[i], \tilde{\varpi}_{m\to l,k}^*[i]\}$ obtains a smaller objective value than that of $\{\boldsymbol{w}_{m,n,k}^*[i], \varpi_{m\to l,k}^*[i]\}$ since $c_{6,m,n} < 1$. This observation contradicts with the assumption. Hence, we conclude that the constraints in (6.32c) are active.

Appendix E

Other Contributions

During my Ph.D research at UBC, I also have some other contributions. The following publications are listed in the Appendix since they are not directly related to the thesis.

J5: Y. Dong, M. J. Hossain, J. Cheng, and V. C. M. Leung, "Robust energy efficient beamforming in MISOME-SWIPT systems with proportional secrecy rate," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 1, pp. 202–215, Jan. 2019.

J6: Y. Dong, A. E. Shafie, M. J. Hossain, J. Cheng, N. Al-Dhahir, and V. C. M. Leung, "Secure beamforming in full-duplex MISO-SWIPT systems with multiple eavesdroppers," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6559–6574, Oct. 2018.

J7: Y. Dong, M. Z. Hassan, J. Cheng, M. J. Hossain and V. C. M. Leung, "An edge computing empowered radio access network with UAV-mounted FSO fronthaul and backhaul: Key challenges and approaches," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 154–160, June 2018.

J8: Y. Dong, X. Ge, M. J. Hossain, J. Cheng, and V. C. M. Leung, "Proportional fairness based beamforming and signal splitting for MISO-SWIPT systems," *IEEE Communications Letters*, vol. 21, no. 5, pp. 1135–1138, May 2017.

C3: Y. Dong, M. J. Hossain, J. Cheng and V. C. M. Leung, "Robust secrecy energy efficient beamforming in MISOME-SWIPT systems with proportional fairness," in *Proceedings of IEEE Global Communications Conference*, Abu Dhabi, UAE, pp. 1–6, Dec. 2018.

C4: Y. Dong, M. J. Hossain, J. Cheng, and V. C. M. Leung, "Joint precoding and power control in small-cell networks with proportional-rate MISO-BC backhaul," in *Proceedings of IEEE Global Communications Conference*, Waikoloa, HI, USA, pp. 1–6, Dec. 2019.

C5: Y. Dong, A. E. Shafie, M. J. Hossain, J. Cheng, N. Al-Dhahir, and V. C. M. Leung,"Secure beamforming in full-duplex SWIPT systems with loopback self-interference cancellation," in *Proceedings of IEEE International Conference on Communications*, Kansas City, MO, USA, pp. 1–6, May 2018.

C6: Y. Dong, J. Cheng, M. J. Hossain, and V. C. M. Leung, "Extracting the most weighted throughput in UAV empowered wireless systems with nonlinear energy harvester," in *Proceedings of Biennial Symposium on Communications*, Toronto, Canada, pp. 1–5, May 2018.

C7: Y. Dong, M. J. Hossain, J. Cheng and V. C. M. Leung, "Fronthaul-aware group sparse precoding and signal splitting in SWIPT C-RAN," in *Proceedings of IEEE Global Communications Conference*, Singapore, pp. 1–6, Dec. 2017.

C8: Y. Dong, M. J. Hossain, J. Cheng and V. C. M. Leung, "Joint RRH selection and beamforming in distributed antenna systems with energy harvesting," in *Proceedings of International Conference on Computing, Networking and Communications*, Silicon Valley, CA, USA, pp. 1–6, Jan. 2017.