Resource Allocation for Next Generation Wireless Networks

by

Ticao Zhang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
Aug 6th, 2022

Keywords: Optimization, Machine learning, Wireless communication

Approved by

Xiaowen Gong, Assistant Professor of Electrical and Computer Engineering
SueAnne Griffith, Assistant Professor of Electrical and Computer Engineering
Yin Sun, Assistant Professor of Electrical and Computer Engineering
Tao Shu, Associate Professor of Computer Science and Software Engineering
Shiwen Mao, Professor of Electrical and Computer Engineering

Abstract

The rapid development of various emerging applications, such as machine learning, virtual reality, and internet-of-things (IoT) brought massive data traffic. Next generation (6G) wireless networks are expected to support extremely high data rates and a wide variety of applications. To fulfill the requirement of 6G, enabling technologies such as intelligent reflecting surface (IRS) and machine learning have been proposed. In this dissertation, we explore the critical resource allocation problems in 6G wireless networks with the two enabling technologies.

The first part of the dissertation focuses on wireless resource allocation in IRS-assisted networks. Two examples are explored. In the first example, we studied a fairness-aware resource allocation in a rate splitting (RS) network assisted by IRS. Joint active beamforming at the BS and passive beamforming at the IRS design is proposed so that the minimum user rate can be maximized. In the second example, an IRS-assisted federated learning system was studied. An efficient resource allocation algorithm based on optimization theory is proposed to jointly configure the communication and computing parameters to minimize the system energy consumption.

The second part of this dissertation investigates resource allocation problems for wireless networks with machine learning via two examples. In the first example, the energy efficiency of a device-to-device (D2D) network is investigated. A deep learning approach is proposed to allocate the power resources to maximize the sum rate of all devices. In the second example, we studied a downlink resource block allocation problem in a radio access network (RAN) via deep reinforcement learning (DRL).

Our results suggest that IRS has great potential in improving the system network performance. Optimization methods still play vital roles in resource allocation for next-generation

wireless networks. Meanwhile, machine learning approaches would be indispensable tools to address some challenges that the optimization methods could not handle. At the end of this dissertation, we present a workflow to address the general wireless resource allocation problems. Future research directions are also given.

Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Shiwen Mao, who provided me helpful guidance and continuous support during Ph.D. study. His constant encouragement, serious attitude and enthusiasm towards good research has motived me to do my best. He has been so helpful and considerate in my life. I feel so grateful and lucky to have him as advisor.

I would also like to thank my dissertation committee, Prof. Xiaowen Gong, Prof. Yin Sun and Prof. SueAnne Griffith for their valuable comments and and generous support. I am also indebted to Prof. Tao Shu for serving as the university reader and reviewing my work. In addition, I thank Prof. Tao Jiang, Prof. Shi Jin and Prof. Chaokai Wen, who provided me with valuable suggestions and instructions in the field of wireless communication.

I want to express my thankfulness to my current and former colleagues: Xuyu Wang, Mingjie Feng, Ningkai Tang, Kefan Xiao, Chao Yang, Yu Wang, Xiangyu Wang, Wenqian Zhang, Lingxiao Wang, Junwei Ma, Ziqi Wang, Zhitao Yu for their support and inspiring discussions. I also thank my friends: Matthew, Evan, Tony, Ruben, Abigail, Taylor, Tom, Ann, Stanley, the Non's family, Prof. Eming Arthur Lee and all the friends in Auburn for the pleasant, memories and friendship, which would be a treasure for the rest of my life.

Finally, I would like to extend my thanks with all my heart to my mother, Yanhong Fu and my father Lunxiang Zhang for their continuous support and tremendous care, without whom I wouldn't make it today.

The work was supported by the US National Science Foundation (NSF) under grant CNS-2107190, ECCS-1923717 and CNS-2148382.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AE Auto-Encoder

AirComp Over-the-Air Computation

AI Artificial Intelligence

AO Alternative Optimization

BCD Block Coordinate Descent

BS Base Station

CDF Cumulative Distribution Functions

CDMA Code Division Multiple Access

CNN Convolutional Neural Network

CQI Channel Quality Indicator

CSI Channel State Information

D2D Device-to-Device

DC Difference of Convex

DDPG Deep Deterministic Policy Gradient

DNN Deep Neural Network

DQN Deep Q-Network

DRL Deep Reinforcement Learning

DT Decision Tree

DVS Dynamic Voltage Scaling

EE Energy-efficiency

eMBB Enhanced Mobile Broadband

FDMA Frequency Division Multiple Access

FET Field Effect Transistor

FPGA Field Programmable Gate Array

GLI Geographical Location Information

GMM Gaussian Mixture Model

HARQ Hybrid Automatic Retransmission Request

HoL Head of Line

IoE Internet of Everything

IoT Internet of Things

IRS intelligent Reflecting Surface

KKT Karush-Kuhn-Tucker

LSTM Long Short-Term Memory

LTE Long Term Evolution

MDP Markov decision process

MEC Mobile Edge Computing

mMTC Massive Machine Type Communication

mmWave Millimeter Wave

NB Naive Bayesian

NOMA Non-Orthogonal Multiple Access

PCA Principal Component Analysis

PDR Packet Drop Rate

PF Proportional Fairness

PIN Positive Intrinsic Negative

QoS Quality of Service

RAN Radio Access Network

RBG Resource Block Group

RBM Restricted Boltzmann Machine

RB Resource Block

RL Reinforcement Learning

RNN Recurrent Neural Network

RS Rate Splitting

SAG Stochastic Average Gradient

SCA Successive Convex Approximate

SDMA Space Division Multiple Access

SDP Semidefinite Programming

SPCA Successive Pseudo Convex Approximation

SWIFT Simultaneous Power and Energy Transfer

THz Terahertz

UAV Unmanned Arial Vehicle

UDN Ultra Dense Network

UE User Equipment

uRLLC ultra-Reliable and Low Latency Communications

V2V Vehicle-to-Vehicle

VLC Visible Light Communication

VR Virtual Reality

WMMSE Weighted Minimum Mean Squared Error

ZF Zero-Forcing

Chapter 1

Introduction

1.1 Evolving Towards 6G Communication

Wireless communication originates from Marconi's pioneering demonstration of telegraphy in the 19th century and was theoretically supported by Information theory proposed by Shannon in the year 1948 [1]. Since the 1980s, the continuous development of information and communication technology has played important roles in improving the wireless communication performance. As shown in Figure 1.1, in 1979, the voice communication was supported by 1G analog wireless network, which was then replaced by digital communication in 2G in the year 1991. In 1998, 3G technology (CDMA) provides a way to support mobile data services. In 4G long term evolution (LTE) network in 2008, multiple input multiple output (MIMO) architecture together with orthogonal frequency multiple access (OFDM) were applied to achieve a higher spectrum efficiency. We are now at the stage where 5G technology is being deployed commercially.

Compared with 4G wireless communication, 5G technology get a 1000 times data rate increase, a round trip delay less than 1ms and a 5$\sim$15 times increase in terms of spectrum efficiency [2]. The 5G system include three major communication scenarios: enhanced mobile broadband (eMBB), massive machine type communication (mMTC), and ultra-reliable and low latency communications (uRLLC). Various technologies such as millimeter wave (mmWave), massive multiple-input multiple-output (MIMO) and ultra-dense network (UDN) have been proposed to achieve the requirement of 5G communication [3].

However, 5G will not meet all the requirements of future communication in 2030s. Upcoming technologies such as artificial intelligence (AI), virtual reality (VR) and internet of everything (IoE) may require ultra-high data rates and reliability, and low latency. Existing

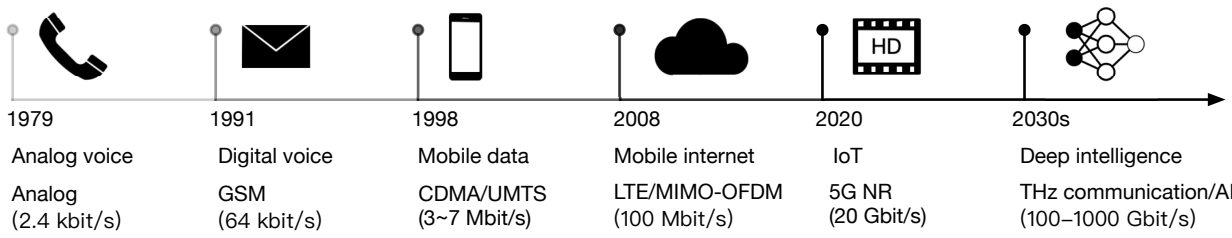


Figure 1.1: The development trend of wireless communication

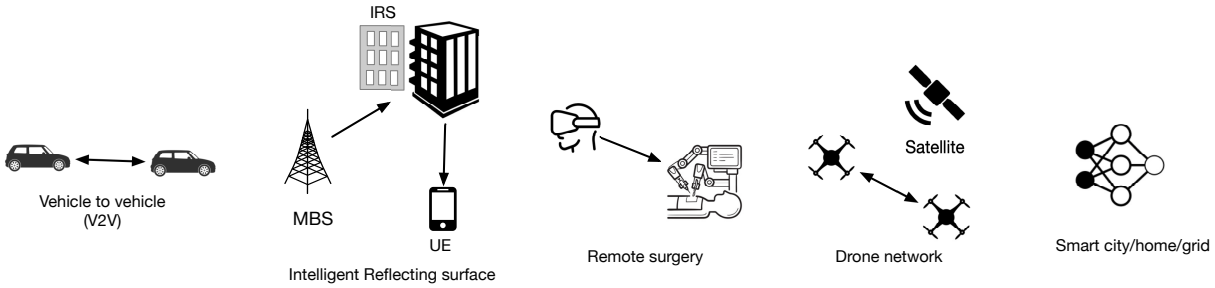


Figure 1.2: Applications of next-generation wireless networks

5G's eMBB, mMTC and uRLLC may not satisfy the future various applications as shown in Fig 1.2. Some fundamental issues such as higher system capacity (100Gbps), higher spectrum efficiency, lower latency (less than 0.1ms), full coverage (100%) and improved quality of service (QoS) need to be addressed. For example, a user who wears a VR glass to play immersive games will demand an ultra-high data rates (at least 25Mbit/s) with low latency (less than 20ms). Autonomous driving in future would require extensive connectivity, high throughput, high reliability, and low latency.

6G communication, with the full support of machine learning and enabling communication technologies such as intelligent reflecting surface (IRS), multiple access technology, unmanned aerial vehicles, cell-free communication, integrated sensing and communication, wireless optical technology and terahertz communication have been proposed to fulfill the future communication demands of a fully connected and intelligent digital world.

A vision of 6G communication is shown in Figure 1.3 [4]. First, it can be seen that 6G wireless communication will be a space-air-ground-sea integrated network which consists of

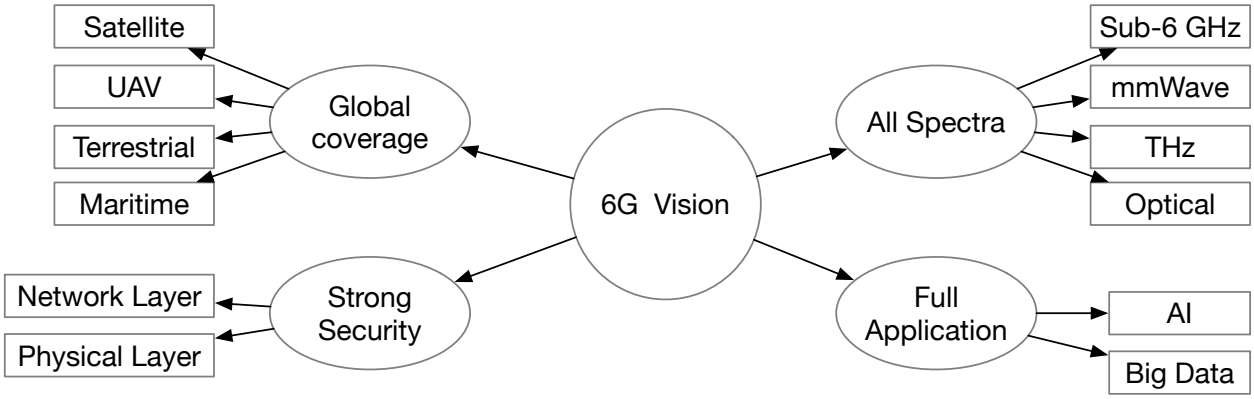


Figure 1.3: A vision of 6G communication

satellite, unmanned arial vehicle (UAV), ground communication and maritime communication. These technologies will provide a full global coverage. Second, past generation wireless networks utilize micro-wave communication, whose resources are fully used. 6G wireless communication will explore all spectra, including sub-6GHz, mmwave, Terahertz (THz) and optical (visible light) frequency bands. Furthermore, it is envisioned that machine learning will provide intelligence for wireless networks. The explosive increase of wireless devices brings about huge dataset resources. Leveraging machine learning and big data, 6G will enable more applications such as smart city/home/healthcare. Finally, 6G communication will promise strong security for both physical layer and network layer.

Discussions for 6G communication is still ongoing within the wireless community. Several research papers have investigated the vision of 6G wireless communication. For example, [5, 6] focuses on the role of machine learning plays in 6G wireless communication. [7] described the challenges and potentials of terahertz (THz) communication in the development of 6G communication. [8] identifies future research directions for intelligent reflecting surface (IRS) assisted communication, with an envision that IRS will play critical roles in 6G like that of massive MIMO plays in 5G. [9] suggested that 6G will be human centric. The security, secrecy and privacy would become key features. [4, 10] provides a comprehensive survey of recent advances and future trends in 6G communication. The vision, enabling technologies and new paradigm shifts in 6G are presented in [11].

5G is being deployed commercially now while countries have already started its research in 6G [12]. China started its research into 6G in 2018 and plan to launch it around 2029. China launched a satellite to test THz transmission in 2020. The US also started 6G research in 2018 to open the higher frequency spectrum for experiment use. Companies like Apple, AT&T and Google are on board in the Next G Alliance. The European Union launched a three-year research project on the basic 6G technologies in 2017. Finland started 6Genesis project for 6G networks, aiming to develop and implement key enabling technologies in 2018. Meanwhile, German's Next Generation Mobile Networks Alliance launched a 6G research project in 2020. South Korea is investing 11.7 billion dollars into developing a digital economy that includes 6G with companies such as LG, Samsung, and SK Telecom.

1.2 Key 6G Communication Technologies

To achieve the target 6G performance, enabling technologies including visible light communication, THZ communication, cell-free massive MIMO, space-air-ground-sea integrated network, holographic radio communication, intelligent reflecting surface (IRS), machine learning, etc., will be used in 6G communication networks. In this section, we will mainly introduce IRS and machine learning, which have been regarded as the two of the paradigm-shifting and revolutionary candidate technologies in 6G.

1.2.1 Intelligent Reflecting Surface

Ultra-dense network (UDN) and massive MIMO have been regarded as key technologies in 5G networks to improve the system spectrum efficiency. However, they also bring the huge hardware cost and energy consumption. To improve the performance, the concept of an intelligent information network with controllable channel is proposed in 6G. IRS has been regarded as an enabling technology to reconfigure the radio signal propagation in wireless links [13–15]. It is a promising enabler for smart wireless communication for B5G/6G wireless systems.

Generally, IRS is a planar antennas array that consists of a large number of passive reflecting elements. Each element can intelligently reconfigure the amplitude and phase of the incident signals. As a result, the signal propagation channel can be smartly coordinated to achieve a desired distribution. An IRS structure is shown in Figure [**?**]. As can be seen, the IRS consists of three layers and a controller chip. The first (outer) layer is composed of a large number of tunable elements, which is printed on a dielectric substrate to manipulate the incident signals. The second layer, a copper plate is deployed in the middle to reduce the signal energy leakage. The third layer is a control circuit board, which can excite and modify the amplitudes/phase of the reflecting elements. The control circuit board is controlled by a controller chip, which is implemented by a field-programmable gate array (FPGA). The base station (BS) can communicate with the IRS controller chip through wired/wireless links.

There are three ways to control the IRS reflection, namely, 1) mechanical actuation, 2) functional materials, and 3) electronic devices. The first approach uses mechanical rotation and translation to control the reflecting elements. The second approach uses liquid crystal and graphene. In practice, the third approach which uses positive-intrinsic-negative (PIN) diodes and field-effect transistors (FETs) is the most widely used method due to its fast response time and relatively low hardware cost and energy consumption. The PIN diode can switch to *on* and *off* state with different biasing voltages. Moreover, a typical channel coherence time is on the orders of millisecond (ms). As a comparison, the diode switching time is on the order of microsecond (μs), hence, the IRS can be well deployed for varying channels. Besides the phase control, it is even more cost-effective for amplitude adjustment, which can be achieved by load resistance adjustment.

Due to its flexibility in deployment, low implementation cost and low power consumption, IRS is expected to play vital roles in performance optimization in 6G wireless networks. There are reveral research on this topic. IRS reflection modeling and hardware design was studied [16]. Later, this research area has been explored in terms of theoretical IRS signal

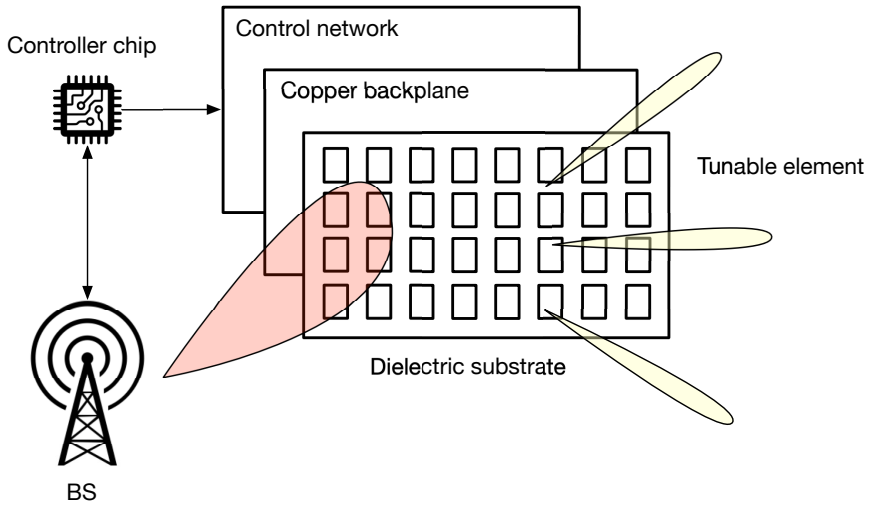


Figure 1.4: The structure of IRS

and channel modeling [14], practical IRS beamforming design [17], and prototype deployment [18]. The beamforming design includes both passive beamforming at the IRS and active beamforming at the transmitter, which is optimized based on different objectives, such as power minimization [17], rate maximization [19], energy efficiency maximization [20], etc. Recently, IRS has been investigated for physical layer security [21], simultaneous power and energy transfer (SWIFT) [22], mobile edge computing [23], etc. The tutorial [14] covers an overview of IRS technology, including its applications in next generation wireless communication, hardware architecture as well as the future challenges.

1.2.2 Machine Learning

Intelligence will be the fundamental characteristics in 6G wireless networks. From 1G to 4G communication, there was no use of AI for communication. 5G involves very limited AI. Moving forward, 6G communication will be fully supported by AI for communication. The adoption of machine learning and AI makes it possible to create an intelligent wireless network. Machine learning has made breakthrough in sensing, mining, and prediction, which can all contribute to enhancing the performance of 6G wireless networks [24].

Current wireless communication technology heavily relies on mathematical models that define the system structure. Such models are based on some assumptions, which may not be accurate enough. Moreover, in some cases, there may be no mathematical models for the communication system. Conventional methods may not be able to meet the requirement of 6G wireless communications. On the other hand, the explosive increase of wireless devices brings about huge dataset resources, which is considered as an enabler for machine learning. Advancement in computing capacity, e.g., GPU, also makes it possible to execute training algorithms at a fast speed. Motivated by this, machine learning has been regarded as another a key technology in 6G.

Generally, machine learning can be categorized into deep learning and reinforcement learning (RL) based on how the model is trained. In terms of deep learning, it is essentially an autonomous system where data is used to find a pattern or make predictions about the data. For example, Apple's Face ID trains an algorithm by scanning our face. Each time we log in with Face ID, the neural engine performs analysis to predict whether it is us or not with the captured face data points. Based on whether deep learning requires labeled data in the training process, deep learning can be further categorized into supervised deep learning and unsupervised deep learning, as shown in table 1.1:

- Supervised deep learning: supervised learning uses a set of labeled samples to learn a mapping between the input and output spaces. Depending on whether the output is continuous or not, supervised learning can be categorized as *regression* and *classification*. Deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), naive Bayesian (NB), decision tree (DT), and support vector regression are typical techniques in supervised deep learning.

- Unsupervised deep learning: Unsupervised learning focuses on classifying unlabeled samples into different clusters. It is mainly used for data dimension reduction in the continuous cases or clustering in the discrete cases. Restricted Boltzmann machine

Table 1.1: Supervised vs. Unsupervised Learning

	Supervised learning	Unsupervised learning
Training data	labeled	unlabeled
Discrete case	classification	clustering
Continuous case	regression	dimensionality reduction
Accuracy of results	high	less accurate
Number of classes	known	not known

(RBM), auto-encoder (AE), Gaussian mixture model (GMM), principal component analysis (PCA), and k-means clustering are usually used in unsupervised deep learning.

Reinforcement learning is an autonomous system that learns decision making via trial and error. An agent is trained to learn to make decisions from the interactions with the environment. It is like learning to ride a bike where people fall off or make too heavy moves, but the feedback of what worked and what didn't work overtime makes people learn to ride eventually. In RL, the agent learns from the feedback whether the performed action would result a better performance. Then the action that worked is reinforced and the algorithm is modified to deliver the best results.

Both deep learning and RL are autonomous learning system. The difference is deep learning learns a patten from a training set and the learned result is applied to a new set, while RL learns to make decisions in a dynamic way based on the feedback from the interactions with the environment. Note that deep learning and RL may work together. In RL, a deep neural network can be used to learn the Q-table mapping, which results in deep reinforcement learning (DRL).

The role of machine learning plays in 6G has been introduced in [24]. It was shown that current wireless systems rely heavily on mathematical models of the communication system. Such models may be inaccurate, and it may challenging to model some complex wireless systems. Moreover, the optimization of wireless networks may be inefficient and fail

in meeting the 6G requirement. Machine learning is expected to model the system and be a promising tool to provide network intelligence.

Machine learning can be trained and deployed in different layers of wireless networks. In physical layer, machine learning achieved great success in channel coding [25], positioning [26], channel estimations [27] and beamforming [28]. In transport layer, machine learning has shown satisfactory performances in congestion control [29] and network traffic prediction. In application layer, machine learning has been shown to be able to enhance the network performance management, UAV control [30] and virtual reality video transmission [31].

1.2.3 Other potential technologies

- Terahertz communication: Terahertz (THz) frequency band, which ranges from 0.1 to 10 THz, belongs to an unexplored radio spectrum. By increasing the transmission frequency to a high band, the bandwidth can be increased. However, it would also suffer from high path loss.

- Optical wireless technology: Optical wireless technology such as visible light communication (VLC) would be extensively used in 6G applications such as vehicle to vehicle communication, indoor positioning, and VR transmission.

- Unmanned aerial vehicles: In unmanned aerial vehicles (UAVs) or drones assisted communication, the BS can be installed in UAVs to extend the wireless connectivity. Due to its flexibility in deployment, this technology has been recognized as one of essential technologies for 6G communication.

- Cell-free massive MIMO: In 6G, the conventional cellular communication will shift to cell-free communication. Cell-free massive MIMO, which combines the advantages of distributed systems and massive MIMO, could solve the handover problem when users move from one cell to another cell. The massive, distributed antennas could further improve the energy-efficiency.

- Space-air-ground-sea Integrated Networks: Space-air-ground-sea integrated networks integrates satellite communication networks, aerial networks, marine communication networks and terrestrial networks to provide a global coverage.

- Proactive caching and mobile edge computing: Storing the video content at the edge, proactive caching could reduce the access delay and traffic offloading. Leveraging the computing capability of the edge, mobile edge computing (MEC) makes it possible to support computation-intensive applications with stringent delay requirement in 6G.

1.3 Resource Allocation for Next Generation Wireless Networks

1.3.1 Wireless Resource Allocation

Due to the wireless technology, we are embracing a rising popularity of mobile applications and services, such as 4K video streaming, virtual reality, UAV trajectory and indoor localization. These applications usually have diverse requirements on the networks. For example, to support the automatic vehicle driving, the communication latency should be made as low as possible to ensure the safety of the passenger. Virtual reality video transmission put stringent requirements on both the network bandwidth and transmission latency to ensure a smooth experience for the VR users. Wireless sensor networks powered with battery requires a high system energy efficiency. As a result, Allocating the limited wireless resources will be crucial in supporting these applications.

In practical communication system, resource allocation is a general concept, which includes channel access management, power allocation, bandwidth allocation, user association, energy management, beamforming design, etc. The taxonomy of resource allocation is shown in Figure 1.5. It can be categorized based on the following principles.

- The purposes: Depending on the type of resources, resource allocation can be categorized based on the optimization parameters. For example, power allocation, beamforming design, bandwidth allocation, user selection, resource block allocation, subcarrier

allocation and so on. In some cases, the optimization problem may involve more than one resources, for example, the joint power and bandwidth allocation.

- The network types: Resource allocation can be performed in diverse network types. Different network types face different resource allocation challenges. For example, in massive MIMO system, a proper beamforming design is needed to exploit the multiplexing gain provided by the antennas. In ultra-dense networks, it is important to mitigate the interferences when allocating resources.

- The performance criteria: Based on different performance criteria, resource allocation problems can be formulated differently. For example, in a battery-powered wireless sensor network, the criteria can be chosen as maximizing the energy-efficiency (EE). In a vehicle communication system, the communication latency should be minimized for the sake of safety.

- The solving approaches: Resource allocation can be viewed as an optimization problem. The solving approach can be categorized. Conventional methods that relies on mathematical models achieved great success in the design and optimization from 1G to 5G. These methods include optimization method, game theory, graph theory, etc. Machine learning method is another approach emerged recently to enhance the performance of 6G [32–34]. It does not need the accurate modeling of the problem and does not require complex iterations.

1.3.2 Optimization Methods

In this subsection, we will mainly introduce some of the basics of the most widely used model-based approaches, convex optimization and non-convex optimization.

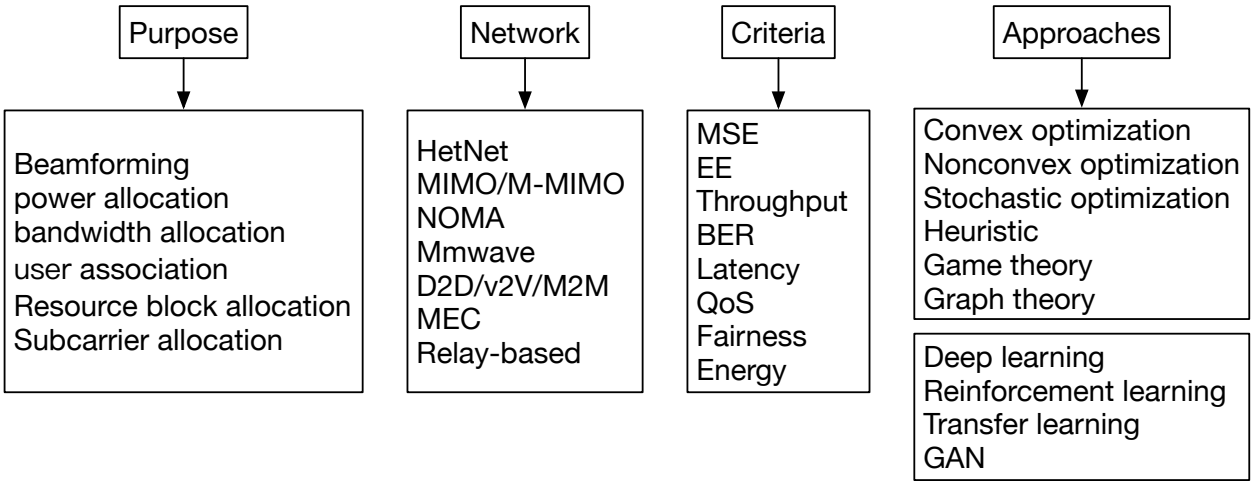


Figure 1.5: Taxonomy for resource allocation problems

Convex Optimization

Convex optimization refers to the minimization of a convex objective function subject to convex constraints. It is widely used in the design and optimization of wireless networks. In convex optimization, if we can find a local optimal solution, then the solution would also be global optimum. Before proceeding, we introduce some basic optimization concept.

- Convex sets: a set $S \subseteq \mathbb{R}^N$ is said to be convex if for any two points $x, y \in S$, the line segment joining the two points x and y also lies in S, i.e.,

$$\lambda x + (1-\lambda)y \in S, \quad \forall \lambda \in [0,1]. \tag{1.1}$$

  For example, the unit ball $S = \{x||x|^2 \leq 1\}$ is a convex set since any line segment joining any two points in the ball also lies in the ball. However, the surface of the ball $S = \{x||x| = 1\}$ is a non-convex set since any line segment joining the two points on the surface of the ball will not be in the surface of the ball.

- Convex function: let $f : \mathbb{R}^N \to \mathbb{R}$ be a differentiable function with a continuous gradient. A function $f(\mathbf{x})$ is convex if

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{y}), \quad \forall\lambda \in [0,1], \tag{1.2}$$

Typical convex functions include $|x|, x^2, ||\mathbf{Hx}||^2, \mathbf{Hx}$ where $x, \mathbf{H}, \mathbf{X}$ are scalar/matrix/vector, respectively. We say that f is concave if $-f$ is convex. If f is convex, we can construct a global under-estimator of f via the first-order Taylor series expansion

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}). \tag{1.3}$$

Moreover, if a function is twice differentiable, then the Hessian matrix $\nabla^2 f(\mathbf{x}) \succeq 0$.

- Convex optimization problems: a generic optimization problem has the following forms.

$$\begin{aligned} &\min f_0(\mathbf{x}) \\ &\text{s.t. } f_i(\mathbf{x}) \leq 0, i = 1, 2, ..., m, \\ &\quad h_j(\mathbf{x}) = 0, j = 1, 2, ..., r, \\ &\quad \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{1.4}$$

where $f_0(\mathbf{x})$ is the objective function, f_i is the inequality constraint and h_j is the equality constraint. The optimization problem is said to be convex if 1) $f_i(i = 0, 1, 2, ..., m)$ are convex functions; 2) h_j are affine functions; and 3) $\mathcal{X}$ is convex. If a problem is convex, efficient interior-point optimization algorithms can be used to find the global optimal solution.

- Lagrangian duality and Karush-Kuhn-Tucker (KKT) condition: Consider the optimization problem in (1.4) and introduce the dual variables λ_i, μ_j, then we get the

Lagrangian function

$$L(\mathbf{x}, \lambda, \mu) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{r} \mu_j h_j(\mathbf{x}). \tag{1.5}$$

A necessary condition for $\mathbf{x}^*$ to be a local optimal solution is that there exists some (λ^*, μ^*) such that

$$f_i(\mathbf{x}^*) \leq 0, \forall i = 1, 2, ..., m \tag{1.6}$$

$$h_j(\mathbf{x}^*) = 0, \forall j = 1, 2, ..., r \tag{1.7}$$

$$\lambda_i^* \geq 0, \forall i = 1, 2, ..., m \tag{1.8}$$

$$\lambda_i^* f_i(\mathbf{x}^*) = 0, \forall i = 1, 2, ..., m \tag{1.9}$$

$$\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{r} \mu_j \nabla h_j(\mathbf{x}^*) = 0 \tag{1.10}$$

Collectively, the condition (1.6)-(1.10) are called the KKT condition for optimality. KKT conditions are necessary conditions for optimality, regardless of the convexity of the problem. If the problem is convex, then the KKT conditions will be both necessary and sufficient conditions.

- Semidefinite matrix and semidefinite programming (SDP): $\mathbf{X} \in \mathbb{C}^{N \times N}$ is a positive semidefinite if $\mathbf{v}^H \mathbf{X} \mathbf{v} \geq 0$ for any $\mathbf{v} \in \mathbb{C}^N$. If $\mathbf{X}$ is also a symmetric matrix, then we denote $\mathbf{X}$ as $\mathbf{X} \succeq 0$. An semidefinite programming (SDP) problem has the form

$$\min \quad \mathbf{CX} \tag{1.11}$$

$$\text{s.t.} \quad \text{tr}(\mathbf{A}_k \mathbf{X}) = b_k, \; k = 1, 2, ..., K. \tag{1.12}$$

$$\mathbf{X} \succeq \mathbf{0}. \tag{1.13}$$

where $\mathbf{C}, \mathbf{X}_k$ and $\mathbf{X}$ are all symmetric matrices and b_k is a scalar. SDP is convex optimization which optimizes a linear objective function over the intersection of the

cone of positive semidefinite matrices with an affine space. The interior point method can be used to find the optimal solution.

Non-convex Optimization

An optimization problem is said to be *nonconvex* if the problem violates any one of the conditions for a problem to be convex. Different from convex optimization which we have a variety of tools to handle, non-convex optimization problems bring about more challenges. Most of the non-convex optimization problems are NP-hard. In this subsection, we briefly introduce a few approaches that are used in non-convex optimization.

- Convex relaxation approach: non-convexity usually comes along with NP-hardness, which cannot be dealt with efficiently with existing mathematic tools. An efficient way is to relax the non-convex problem to a convex one, and then apply familiar algorithmic techniques. In general, these relaxations would bring a performance loss compared to the optimal solution. However, if the problem possesses some structure, the relaxation gap can be made quite small. In other words, solutions to the relaxed problem can be optimal for the original non-convex optimization problem. A widely used convex relaxation approach is successive convex approximate (SCA) method. Consider the following non-convex optimization problem

$$\begin{aligned} &\min f_0(\mathbf{x}) \\ &\text{s.t. } f_i(\mathbf{x}) \leq 0, i = 1, 2, ..., m, \\ &\mathbf{x} \in \mathcal{X}. \end{aligned} \tag{1.14}$$

where f_0 is convex, however, $f_i(\mathbf{x})$ are nonvex. Since directly solving this problem might be NP-hard, we want to solve it with a series of approximations $\tilde{f}_i(\mathbf{x}) \approx f_i(\mathbf{x})$. Specifically, if the approximations satisfy the following properties: 1) $f_i(\mathbf{x}) \leq \tilde{f}_i(\mathbf{x})$ for all $\mathbf{x}$, 2) $\nabla f_i(\mathbf{x}^t) = \nabla \tilde{f}_i(\mathbf{x}^t)$, and 3)$f_i(\mathbf{x}^t) = \tilde{f}_i(\mathbf{x}^t)$. Here $\mathbf{x}^t$ is the optimal solution

of the approximated problem in the t-th iteration, then the solutions generated by the series of the relaxed problem converge to the stationary solution of the original problem. In other words, the solution will satisfy the KKT condition of the original problem.

- Alternative optimization (AO): Alternative optimization is often used in settings where the optimization problem has two or more optimization variables. The problem is not jointly convex in terms of all the variables; however, it is convex in terms of each variable when other variables are fixed. This marginal convexity makes it possible to optimize the variables in an iterative way and find a stationary solution. The block coordinate descent (BCD) algorithm is an alternative optimization method widely used for the non-convex optimization problem with several block variables. In each iteration, a single block of variables is optimized, while the remaining blocks are fixed. If the problem is convex in terms of each block of variables, then each subproblem in each iteration can be optimally solved.

1.3.3 Machine Learning Approaches

Deep Learning

The goal of resource allocation is to manage the available network resources to maximize one or more network metrics. Formally speaking, denote the objective function as f and the available resources as $\mathbf{x} \in \mathcal{X}$, then the resource management problem can be formulated as

$$\max_{\mathbf{x}\in\mathcal{X}} \quad f(\mathbf{x}). \tag{1.15}$$

The conventional model-based approaches only work for the scenario where the problem has a tractable mathematical model. Moreover, even the model is accurate enough, it is still inevitable to solve the optimization problem based on a given set of system parameters, such as the realization of the channel, the users' locations and the users' demand. Once the system

parameters changed, the optimization problem needs to be solved again. Such problems are usually NP-hard; hence it is quite challenging to find efficient algorithms at all times in highly dynamic channels when the users' behavior and channel conditions are unpredictable. This motivates us to endow the network with artificial intelligence (AI) capability and to employ a machine learning approach where the system can dynamically determine the best resource management policy in the changing environment [35, 36].

The idea is that the resource allocation (1.15) can be regarded as a function mapping problem where the network state $\mathbf{s} \in \mathcal{S}$ is mapped to the resource management decision policy $\mathbf{x}$. The *universal approximate theorem* [37] shows that the mapping function $f(\mathbf{x})$ can be approximated arbitrarily well by a single fully connected layer with enough neurons and sigmoidal activation functions. In some cases, it may be hard to obtain the labeled data, unsupervised learning, on the other hand, which optimizes the objective function directly can be used to achieve a satisfactory performance. Namely we can define the negative value of the objective function $f(\mathbf{x})$ as the loss function. A neural network is trained to minimize the loss function, which is equivalent to maximize the objective function. Hence, an optimal resource allocation scheme can be obtained.

Reinforcement Learning

In some scenarios, it may be hard to express the objective value function with a closed-form expression. It this case, reinforcement learning (RL) shows great advantages in addressing this challenging task.

As shown in Figure1.6, an RL includes three part: the reward, the action and the state. In terms of resource allocation, the Action can be defined as the resource allocation scheme and the reward can be the objective function $f(\mathbf{x})$. When a system state is observed, an action is performed so that the agent can obtain a reward. The agent learns an optimal resource allocation policy via the interaction with the environment.

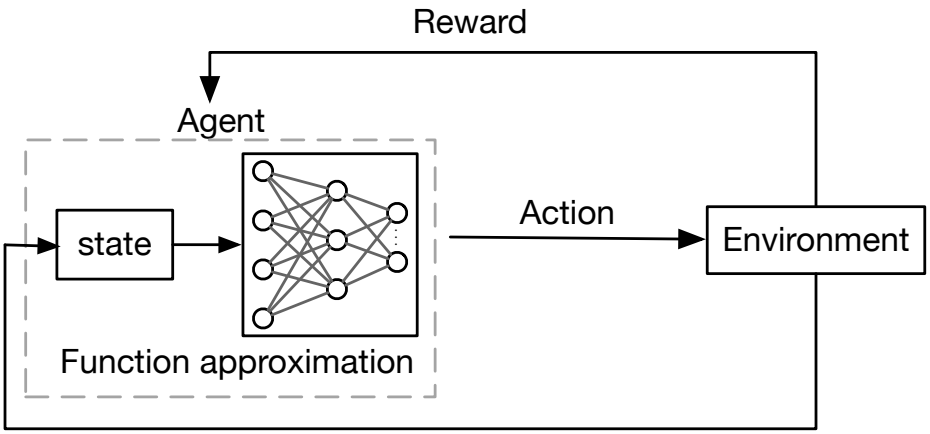


Figure 1.6: Reinforcement learning for resource allocation

Typical RL algorithms include the actor-critic algorithm, and the deep deterministic policy gradient (DDPG) algorithm and the Q-learning algorithm. Let's take Q-learning as an example. In Q-learning, the set of possible states is denoted as $\mathcal{S}$, and the set of discrete actions is denoted as $\mathcal{A}$. At each time instant t, the agent takes action $a^t \in \mathcal{A}$ when observing the state $s^t \in \mathcal{S}$ and receives a reward r^t. Then the system enters the next state s^{t+1}. The Q-learning algorithm aims to learn an optimal policy π which maps state s^t to action a^t, so that the reward over time can be maximized. A cumulative reward can be defined as

$$R^t = \sum_{\tau=0}^{\infty} \gamma^\tau \cdot r^{t+\tau}, \tag{1.16}$$

where $\gamma \in (0, 1]$ is a tradeoff scalar between the immediate and future rewards.

Under a policy π, the *Q-function* of the agent is defined as $Q_\pi(s, a) = \mathbb{E}_\pi[R^t | s^t = s, a^t = a]$. Q-learning aims to maximize the Q-function by maintaining a *Q-table*. However, when the state and action spaces become continuous and large, the problem becomes intractable. To address this problem, DRL uses a deep neural network (DNN) to approximate the mapping table. In time-varying and unpredictable networks, (D)RL has proved to be effective in dealing with real-time decision-making problems.

1.4 Overview of the Dissertation

In this dissertation, we aim to investigate resource allocation problems for next generation wireless networks. Specifically, two enabling technologies, IRS and machine learning will be considered. In Chapter 2 and Chapter 3, we study resource allocation problems in IRS-assisted networks. In Chapter 4 and Chapter 5, we investigated machine learning for wireless resource allocation.

In Chapter 2, we considered the resource allocation problem in an IRS-assisted rate splitting (RS) multiple access network. Attaining accurate channel state information (CSI) for the IRS cascaded channel is particularly challenging. Imperfect CSI is a major bottleneck to realize the substantial spectral efficiency benefit of IRS assisted networks. RS, a promising multiple access technology, has been shown to be able to achieve an improved spectrum efficiency and be robust to channel uncertainties. This chapter investigates the interplay between IRS and RS by considering the resource allocation in an IRS-assisted RS system. Joint active beamforming at the base station (BS) and the passive beamforming at the IRS is considered to maximize the minimum user rate. Both theoretical results and numerical results are provided to explain the interplay between RS and IRS.

In Chapter 3, we investigate the resource allocation problem in an IRS-assisted federated learning system. An energy consumption minimization problem is formulated. An iterative resource allocation algorithm is proposed to jointly configure the parameters with proven fast convergence. Simulation results validate that the proposed resource allocation algorithm can achieve significant energy savings, especially when the number of reflecting elements on the IRS is large and when the IRS is properly configured.

In Chapter 4, we studied the power allocation problem in a large wireless network with interfering links. A deep learning approach is proposed to manage the power control so that the sum rate of all links can be maximized. Specifically, we developed a deep learning based power control scheme, termed PowerNet, that uses the devices' geographical location information (GLI). Moreover, with a proper training, PowerNet transforms the

on-line complexity to off-line training, and is amenable for real-time services. Different from conventional deep neural network (DNN) that adopts fully connected structure, the proposed PowerNet leverages convolutional layers to better capture the interference pattern across different links in large wireless networks and utilizes deep residual learning to further enhance its robustness. Simulation results demonstrate that PowerNet can achieve a near-optimal performance at a remarkably high speed.

In Chapter 5, the channel resource block allocation problem in a downlink radio access network (RAN) was investigated with machine learning approaches. In RAN, packet scheduler plays an important role in satisfying the stringent delay requirements of a variety of applications. In this chapter, we show that optimal scheduling is a challenging combinatorial optimization problem, which is hard to solve within the channel coherence time with conventional optimization methods. We incorporate deep reinforcement learning (DRL) into the design of cellular packet scheduling. A delay-aware cellular traffic scheduling algorithm is developed to map the observed system state to scheduling decision. Due to the huge state space, a recurrent neural network (RNN) is utilized to approximate the optimal action-policy function. Simulation results show that the DRL-based packet scheduling can achieve the lowest average delay compared with conventional benchmarks.

The examples presented in this dissertation show the role of resource allocation plays in the optimization of next generation wireless networks. The two represented 6G technologies, IRS and machine learning are explored. We presented a workflow to deal with general resource allocation problems based on these examples. We conclude the dissertation and present future directions in chapter 6, .

Notation: The notation used in this paper is summarized as follows. Bold lower/upper case letters denote vectors and matrices, respectively. $\mathcal{CN}(\mu, \sigma^2)$ denotes the circularly symmetric complex Gaussian distribution with mean μ and variance σ^2. For any scalar a, $|a|$ denotes its absolute value. For any vector $\mathbf{a}$, a_i is the i-th element. $\mathbf{A}^*$, $\mathbf{A}^T$ and $\mathbf{A}^H$ represent the conjugate, transpose, and conjugate transpose of matrix $\mathbf{A}$, respectively.

Diag($\mathbf{A}$) stands for a vector whose elements are extracted from the diagonal of matrix $\mathbf{A}$. $\mathbf{A} \succeq \mathbf{0}$ means that $\mathbf{A}$ is a positive semidefinite (PSD) matrix. Rank($\mathbf{A}$) denotes the rank of matrix $\mathbf{A}$. $\arg(\cdot)$ returns the angle of a complex variable. Variables with star indicate optimal solutions. We use x, $\mathbf{x}$ and $\mathbf{X}$ to denote scalar, vector and matrix, respectively. $x_{i,j}$ denotes the (i,j)th element of $\mathbf{X}$; x_i is the ith element of $\mathbf{x}$ while $\mathbf{x} = (x_i)_{i=1}^N$, and $\mathbf{x}_{-j} = (x_i)_{i=1,i\neq j}^N$ denotes all elements of $\mathbf{x}$ except x_i.

Chapter 2

Resource Allocation For IRS-assisted Rate Splitting networks

2.1 Introduction

Intelligent reflecting surface (IRS) has recently emerged as a promising technology for 6G wireless communications due to its ability of reconfiguring the propagation environment [14, 20, 38]. IRS is a programmable meta-surface equipped with multiple low-cost programmable reflecting elements, whose amplitude and phase can be reconfigured. As a result, reflection of incoming signals can be programmed to enhance the wireless channel. Compared with conventional technologies such as amplify-and-forward (AF), massive multiple-input multiple-output (MIMO), and millimeter wave (mmWave) [39], IRS does not require an increased number of active radio frequency (RF) chains. Both the energy consumption and hardware cost are low. IRS has been regarded as a promising technology for 6G to make high spectrum efficient, yet cost-effective systems.

Meanwhile, rate splitting (RS) has been envisioned as a promising multiple access technology for beyond 5G wireless communication [40–42]. In RS, the message at the transmitter is split into private part and common part. The common part shared a common codebook known to all users, hence can be decoded by each user. The private part can be decoded by removing the interference from the common part and treating interfering signals from other users as noise. The flexibility to partially decode interference and partially treat the remaining interference as noise makes RS a promising PHY-layer transmission paradigm for non-orthogonal transmission, interference management and multiple access strategies in 6G [43]. RS has been shown in [44] to outperform and unify space division multiple access (SDMA) and non-orthogonal multiple access (NOMA) under wide range of network loads and user deployment. Recent research progress has shown that RS can achieve a

spectrum efficiency improvement [44], robust to imperfect channel state information (CSI) conditions [40,45] and outperforms NOMA and SDMA in secure downlink transmissions [46].

Inspired by the appealing advantages of IRS and RS, researchers have begun to study the interplay of integrating the two infrastructure level techniques. The benefits of integrating IRS and RS have been shown in terms of energy efficiency improvement [47, 48], max-min fairness [46], spectrum efficiency improvement [49] and outage probabilities performance improvement [50]. Apart from these benefits, [51] tries to answer the question "Why we should consider the integration of RS and IRS" by showing that RS and IRS share similarities and they complements each other. To be specific, both techniques perform beamforming. IRS plays passive beamforming to assist the signal transmission while the RS plays active beamforming at the BS to achieve better performance. A joint design should be considered when integrating the two techniques together. Moreover, IRS comprises only nearly passive elements. Accurate channel state information (CSI) for the IRS related channel is hard to obtain [52] while RS has been shown to be robust to CSI uncertainties [53]. Therefore, RS is a good fit to IRS systems with imperfect CSI [54].

Before the integration of RS and IRS, the joint design of IRS with other multiple access technologies has already been studied. In SDMA, the base station (BS) employs different linear precoding to serve different users simultaneously in the same frequency band for the downlink/uplink transmissions. [13] shows that conventional zero-forcing (ZF) beamforming is insufficient to null/suppress the interference from other users while an IRS can provide spatial interference nulling/cancellation capability to solve this problem. Compared with SDMA where interfering signals from other users are treated as noise, (power domain) NOMA allows users to be at the same time-frequency resource block and distinguishes them in the power domain. By doing so, NOMA has been regarded as a multiple access candidate technology to improve the system spectrum efficiency for future wireless networks [55]. The integration of IRS and NOMA has been shown to provide a cost-effective solution to achieve high energy efficiency [56], spectrum efficiency [19, 57] and increased coverage [58]. Another

multiple access technology, RS, has been shown in [44] to outperform and unify SDMA and NOMA under wide range of network loads and user deployment. Despite its appealing advantages and good fit with IRS technology, few technical contributions have been presented and the full possibilities of the integration of IRS and RS remain to be explored.

The RS beamforming design has been extensively studied. In [45], a sum rate maximization beamforming was studied with the weighted minimum mean squared error (WMMSE) method. In [59], a difference of convex functions (DC) programming was proposed to optimize the precoder covariance matrix. Secure and robust RS beamforming design was studied in [60] and [61] under imperfect CSI, respectively. Most of the previous works find a suboptimal solution with optimization techniques. Recently, a global optimal beamforming was proposed in [62] to optimize the energy efficiency in an RS system with branch and bound algorithm.

In terms of the joint beamforming design between IRS and RS, the authors in [63] explored the joint optimization in an IRS-assisted RS system. The successive convex approximation and semidefinite programming (SDP) is used to maximize the minimum rate of all users. This technique can return a rank-one solution for the IRS-assisted NOMA system [64]. However, the rank-one condition may not be satisfied in an IRS-assisted RS system [65]. [50] explores the on-off control for passive beamforming at the IRS. A closed-form expression for the outage probability for the cell-edge users is derived. [48] performs beamforming design to maximize the energy efficiency in an IRS-assisted RS system. [49] performs beamforming and IRS scatter matrix design to maximize the system spectral efficiency [49]. However, the above works all assume perfect CSI setting, which is unrealistic in practice. The mechanism and advantages of RS in combating the IRS related channel uncertainties have not yet studied so far.

This chapter tries to develop novel and efficient beamforming algorithms for IRS-assisted RS systems and explore the interplay between IRS and RS in the imperfect CSI case. The main contribution of this chapter is summarized as follows:

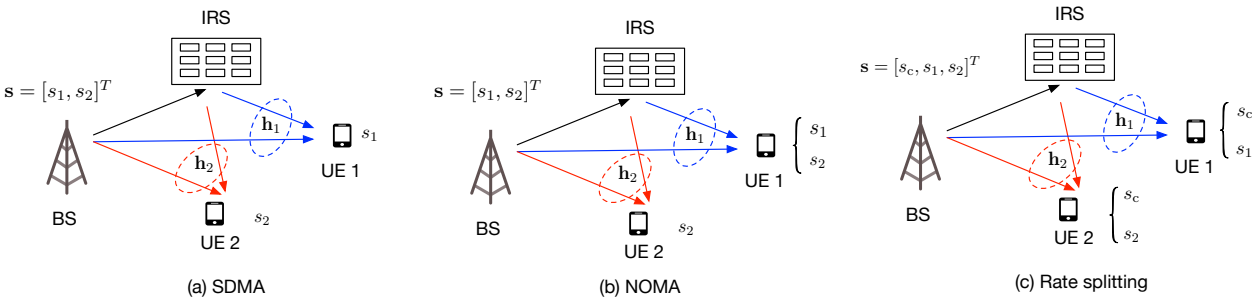


Figure 2.1: Three multiple access technologies assisted by IRS: (a) SDMA, (b) NOMA, and (c) Rate splitting.

1. We formulate a max-min rate optimization problem to explore the interplay between IRS and RS techniques. The joint beamforming design is considered under different IRS configurations.

2. We employed the weighted mean squared error minimization (WMMSE) algorithm to transform the non-convex max-min rate problem into a more tractable form. A block coordination descent (BCD) algorithm is proposed to optimize the variables in an alternative manner. The algorithm was extended to the imperfect CSI case to explore the robustness of RS in combating the IRS-related channel uncertainties.

3. We provided analysis on the impact of the system parameters and explain the interplay between RS and IRS. Numerical results shows that IRS-assisted RS is robust towards channel uncertainties and outperforms the conventional multiple access technology with/without IRS with respect to the max-min rate.

The remainder of this chapter is organized as follows. Section 2.2 presents the system model and formulates the max-min rate maximization problem. Section 2.3 introduces a rate-WMMSE relationship, based on which a BCD algorithm is developed for different IRS setups. The algorithm is extended to the imperfect CSI case in Section 2.4. Section 2.5 presents an asymptotic analysis on the impact of system parameters and interprets the role of IRS plays. Numerical results are provided in Section 2.6 to evaluate the effectiveness of the proposed algorithms. Finally, Section 2.7 concludes this chapter.

2.2 System Model and Problem Formulation

2.2.1 System Model

The differences between IRS-assisted RS, NOMA, and SDMA are illustrated in Fig. 2.1. The figure shows a two-user system where the BS aims to deliver message s_1 to User 1 and message s_2 to User 2. In the SDMA system, each user treats the signal from other users as interference, as shown in Fig. 2.1(a). In the NOMA system, User 1 completely decodes the message from User 2 using successive interference cancellation (SIC), hence an improved spectrum efficiency can be achieved. Unlike NOMA and SDMA, the RS system shown in Fig. 2.1(c) generates messages using a public codebook known to all users. Each user partially decodes the messages for other users and partially treats the messages from other users as interference.

Specifically, we consider a downlink multi-user multiple-input-single-output (MISO) communication system, which consists of a BS with M antennas, and K user equipments (UEs) each with a single antenna, denoted by $\mathcal{K} = \{1, 2, ..., K\}$. In RS, the message X_k intended for UE k is split into a private part $X_{\mathrm{p},k}$ and a common part $X_{\mathrm{c},k}$. The private parts are encoded independently into Gaussian data symbol streams, denoted as $[s_1, s_2, ..., s_k]^T \in \mathbb{C}^{K\times 1}$. Meanwhile, the common parts of all UEs $\{X_{\mathrm{c},1}, X_{\mathrm{c},2}, ..., X_{\mathrm{c},K}\}$ are combined into a common message X_{c}, which is encoded into a common stream s_{c} with a public codebook known to all UEs. As a result, the combined symbols are grouped into a vector $\mathbf{s} = [s_{\mathrm{c}}, s_1, s_2, ..., s_K]^T \in \mathbb{C}^{(K+1)\times 1}$. Each signal is assumed to have zero mean and unit variance, i.e., $\mathbb{E}[\mathbf{s}\mathbf{s}^H] = \mathbf{I}_{K+1}$. At the transmitter, the precoding matrix for all UEs is $\mathbf{w} = [\mathbf{w}_{\mathrm{c}}, \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K]$, where $\mathbf{w}_k \in \mathbb{C}^{M\times 1}$ is the precoding matrix for UE $k \in \mathcal{K}$ for its private data s_k, and $\mathbf{w}_{\mathrm{c}}$ is the precoding matrix for the common message s_{c}.

To assist signal transmission, an IRS with N reflecting elements is placed between the UEs and the BS. The equivalent channels from the BS to UE k, from the IRS to UE k, and from the BS to the IRS are denoted as $\mathbf{h}_{\mathrm{d},k} \in \mathbb{C}^{M\times 1}$, $\mathbf{h}_{\mathrm{r},k} \in \mathbb{C}^{N\times 1}$, and $\mathbf{G} \in \mathbb{C}^{N\times M}$,

respectively. The combined channel between the BS and UE k, $\mathbf{h}_k \in \mathbb{C}^{M\times 1}$, can be regarded as a combination of the direct and the reflected channels, i.e.,

$$\mathbf{h}_k \triangleq \mathbf{h}_{\mathrm{d},k} + \mathbf{G}^H \mathbf{\Theta} \mathbf{h}_{\mathrm{r},k}, \tag{2.1}$$

where $\mathbf{\Theta} \in \mathbb{C}^{N\times N}$ is the IRS phase shift matrix. Note that the IRS reflection phase-shift matrix $\mathbf{\Theta}$ is a diagonal matrix with diagonal elements $v_i = \beta_i e^{j\theta_i}$, where $\theta_i \in [0, 2\pi]$ and $\beta_i \in [0, 1]$, $1 \leq i \leq N$. We extract the diagonal elements of $\mathbf{\Theta}$ and let $\mathbf{v} = \mathrm{diag}\{\mathbf{\Theta}\} \in \mathbb{C}^{N\times 1}$. Then we have $\mathbf{h}_k = \mathbf{h}_{\mathrm{d},k} + \mathbf{H}_k^H \mathbf{v}$, where $\mathbf{H}_k = \mathrm{diag}\{\mathbf{h}_{\mathrm{r},k}^H\}\mathbf{G} \in \mathbb{C}^{N\times M}$. The reflection matrix $\mathbf{\Theta}$ captures the effective phase shifts of all the reflecting elements on the IRS. The phase shift unit can be adjusted by the IRS controller based on measured channel dynamics. Depending on the amplitude and phase shift of the reflecting elements, we consider two types of IRS in this chapter [64].

- IRS Φ_1 : the reflecting elements can be adjusted with arbitrary continuous amplitudes and phase, i.e., $\Phi_1 = \{\beta_i e^{j\theta_i} | \theta_i \in [0, 2\pi], \beta_i \in [0, 1]\}$.
- IRS Φ_2: The reflection amplitude is fixed and only the phase can be adjusted. If the phase can be adjusted continuously, the feasible set is expressed as $\Phi_2 = \{e^{j\theta_i} | \theta_i \in [0, 2\pi]\}$. This setting can be extended to quantized phase shift.

The signal received for UE k can be expressed as

$$y_k = \mathbf{h}_k^H \sum_{i\in\mathcal{M}} \mathbf{w}_i s_i + n_k, \quad \forall k \in \mathcal{K}, \tag{2.2}$$

where $\mathcal{M} = \mathcal{K} \cup \{\mathrm{c}\}$ denotes the combined set and $n_k \sim \mathcal{CN}(0, \sigma_0^2)$ is the additive white Gaussian noise at UE k.

At the receiver, each UE first decodes the common stream by treating all the private streams as noise. Then its private message is decoded by removing the decoded common stream with SIC. After the decoding process, the receiver recombines messages into original

messages. The decoding SINR for the common message and the private message for UE k at the receiver is given by

$$\gamma_{\mathrm{c},k} = \frac{|\mathbf{h}_k^H \mathbf{w}_\mathrm{c}|^2}{\sum_{i\in\mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_0^2}, \quad \forall k \in \mathcal{K}, \tag{2.3}$$

$$\gamma_{\mathrm{p},k} = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{i\in\mathcal{K}, i\neq k} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_0^2}, \quad \forall k \in \mathcal{K}. \tag{2.4}$$

Under Gaussian signaling, the achievable rate of UE k in decoding the common rate and the private rate are given by

$$r_{\mathrm{c},k} = \log_2(1+\gamma_{c,k}), \quad R_{\mathrm{p},k} = \log_2(1+\gamma_{p,k}). \tag{2.5}$$

To ensure that all UEs can decode the common message stream, the actual rate of the common message stream, denote as r_c, is constrained by each $r_{\mathrm{c},k}$, i.e.,

$$r_\mathrm{c} = \min_{k\in\mathcal{K}} r_{\mathrm{c},k}. \tag{2.6}$$

According to the RS decoding principle, the actual data stream r_c is shared by all UEs. By denoting $R_{\mathrm{c},k}$ as the general common rate allocated to UE k, we have

$$\sum_{k\in\mathcal{K}} R_{\mathrm{c},k} \leq r_\mathrm{c}, \quad R_{\mathrm{c},k} \geq 0. \tag{2.7}$$

After removing the common data stream, each UE decodes its own private message. Finally, the overall achievable data rate for UE k is given by

$$R_k = R_{\mathrm{c},k} + R_{\mathrm{p},k}, \quad \forall k \in \mathcal{K}. \tag{2.8}$$

2.2.2 Problem Formulation

In this chapter, we aim to maximize the minimum achievable rate of all UEs by jointly performing active beamforming at the BS as well as passive beamforming at the IRS. Specifically, the problem can be mathematically formulated as

$$\max_{\mathbf{w},\mathbf{R}_\mathrm{c},\mathbf{R}_\mathrm{p},\mathbf{v},s} \quad s \tag{2.9a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{M}} ||\mathbf{w}_k||^2 \leq P \tag{2.9b}$$

$$v_i \in \Phi, \ \forall i \tag{2.9c}$$

$$R_{\mathrm{p},k} + R_{\mathrm{c},k} \geq s \tag{2.9d}$$

$$R_{\mathrm{p},k} \leq \log_2(1+\gamma_{\mathrm{p},k}), \ \forall k \tag{2.9e}$$

$$\sum_{k\in\mathcal{K}} R_{\mathrm{c},k} \leq \log_2(1+\gamma_{\mathrm{c},k}), \ \forall k, \tag{2.9f}$$

where $\mathbf{w} = \{\mathbf{w}_i | i \in \mathcal{M}\}$, $\mathbf{R}_\mathrm{p} = \{R_{\mathrm{p},k} | k \in \mathcal{K}\}$, $\mathbf{R}_\mathrm{c} = \{R_{\mathrm{c},k} | k \in \mathcal{K}\}$, P is the total transmit power at the BS, and Φ denotes one of the set in Φ_1 or Φ_2.

The formulated problem is non-convex optimization where the non-convexity is due to $R_{\mathrm{p},k}$ and $R_{\mathrm{c},k}$. As can be seen from (2.5), the expression of $R_{\mathrm{p},k}$ involves the logarithmic operation of $\gamma_{\mathrm{p},k}$, which is concave. The inner function $\gamma_{\mathrm{p},k}$, as shown in (2.3), involves a quadratic-over-linear operation of $\mathbf{w}$, which is convex. As a results, the combined expression of $R_{\mathrm{p},k}$ is neither concave nor convex. Moreover, the beamforming vector $\mathbf{w}$ is coupled with the the IRS phase shift matrix $\mathbf{\Theta}$, which makes the problem more challenging.

2.3 Joint Beamforming Design with Perfect CSI

Problem (2.9) is a joint beamforming design problem, which is to find the active beamforming $\mathbf{w}$ at the BS as well as the passive beamforming $\mathbf{\Theta}$ at the IRS. To deal with the coupling between the passive beamforming and the active beamforming, we propose to use

the alternative optimization (AO) framework, which optimizes the active beamforming and the passive beamforming iteratively.

2.3.1 The SDR Approach

In [63–65], the SDR approach was proposed to address the RS beamforming design problem. The idea is to use the successive convex approximate (SCA) technique, which approximates the non-convex constraints with their convex surrogate function. Then the SDR method is used to solve the problem with convex-relaxed constraints. For the simplicity of analysis, we only introduce the active beamforming design at the BS when the IRS setting is fixed. The optimization of the IRS phase elements can be conducted similarly.

We rewrite the left-hand-side of (2.9e) and (2.9f) as

$$\begin{aligned} & R_{\mathrm{p},k} - \log_2\left(1 + \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{i\in\mathcal{K}, i\neq k} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_0^2}\right) \\ = & \underbrace{R_{\mathrm{p},k} - \log_2\left(\sum_{i\in\mathcal{K}} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) + \sigma_0^2\right)}_{g_{\mathrm{p},k}^{C}(\mathbf{W})} + \underbrace{\log_2\left(\sum_{i\in\mathcal{K}, i\neq k} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) + \sigma_0^2\right)}_{g_{\mathrm{p},k}^{\bar{C}}(\mathbf{W})} \triangleq g_{\mathrm{p},k}(\mathbf{W}), \end{aligned} \tag{2.10}$$

$$\begin{aligned} & \sum_{k\in\mathcal{K}} R_{\mathrm{c},k} - \log_2\left(1 + \frac{|\mathbf{h}_k^H \mathbf{w}_{\mathrm{c}}|^2}{\sum_{i\in\mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_0^2}\right) \\ = & \underbrace{\sum_{k\in\mathcal{K}} R_{\mathrm{c},k} - \log_2\left(\sum_{i\in\mathcal{M}} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) + \sigma_0^2\right)}_{g_{\mathrm{c},k}^{C}(\mathbf{W})} + \underbrace{\log_2\left(\sum_{i\in\mathcal{K}} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) + \sigma_0^2\right)}_{g_{\mathrm{c},k}^{\bar{C}}(\mathbf{W})} \triangleq g_{\mathrm{c},k}(\mathbf{W}), \end{aligned} \tag{2.11}$$

where $\mathbf{W} = \{\mathbf{W}_i\}_{i\in\mathcal{K}}$ is the set of covariance matrices and $\mathbf{W}_i = \mathbf{w}_i \mathbf{w}_i^H$. By mapping the beamforming vector to a high dimensional matrix, the inner part of the logarithm function becomes a linear function in terms of $\mathbf{W}$. Moreover, it can be easily verified that $g_{\mathrm{p},k}^{C}(\mathbf{W})$ and

$g_{\mathrm{c},k}^{C}(\mathbf{W})$ are convex functions in terms of $\mathbf{W}$, and $g_{\mathrm{p},k}^{\bar{C}}(\mathbf{W})$ and $g_{\mathrm{c},k}^{\bar{C}}(\mathbf{W})$ are concave functions of $\mathbf{W}$. As a result, $g_{\mathrm{p},k}(\mathbf{W})$ and $g_{\mathrm{c},k}(\mathbf{W})$ are neither concave nor convex. To facilitate the SCA approach, we need to convexify the nonconvex part of $g_{\mathrm{p},k}(\mathbf{W})$ and $g_{\mathrm{c},k}(\mathbf{W})$. A straightforward way is to convexify the concave functions $g_{\mathrm{p},k}^{\bar{C}}(\mathbf{W})$ and $g_{\mathrm{c},k}^{\bar{C}}(\mathbf{W})$ using the following inequality $\log_2(a) \leq -\log_2(b) + \frac{ab-1}{\log(2)}$ [66, Proposition 1].

The SCA approach optimizes the beamforming vector successively. Suppose $\mathbf{W}^t$ is the obtained solution in iteration t. Then in iteration $(t+1)$, $g_{\mathrm{p},k}^{\bar{C}}(\mathbf{W})$ and $g_{\mathrm{c},k}^{\bar{C}}(\mathbf{W})$ can be convexified as

$$g_{\mathrm{p},k}^{\bar{C}}(\mathbf{W}) \leq -\log_2\left(\Gamma_{\mathrm{p},k}(\mathbf{W}^t)\right) + \frac{\Gamma_{\mathrm{p},k}(\mathbf{W}^t)\left(\sum_{i\in\mathcal{K},i\neq k}\mathrm{Tr}\left(\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_i\right)+\sigma_0^2\right)-1}{\log(2)} \triangleq \hat{g}_{\mathrm{p},k}^{\bar{C}}(\mathbf{W};\mathbf{W}^t), \tag{2.12}$$

$$g_{\mathrm{c},k}^{\bar{C}}(\mathbf{W}) \leq -\log_2\left(\Gamma_{\mathrm{c},k}(\mathbf{W}^t)\right) + \frac{\Gamma_{\mathrm{c},k}(\mathbf{W}^t)\left(\sum_{i\in\mathcal{K}}\mathrm{Tr}\left(\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_i\right)+\sigma_0^2\right)-1}{\log(2)} \triangleq \hat{g}_{\mathrm{c},k}^{\bar{C}}(\mathbf{W};\mathbf{W}^t), \tag{2.13}$$

where $\Gamma_{\mathrm{p},k}(\mathbf{W}^t) = \left(\sum_{i\in\mathcal{K},i\neq k}\mathrm{Tr}\left(\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_i^t\right)+\sigma_0^2\right)^{-1}$ and $\Gamma_{\mathrm{c},k}(\mathbf{W}^t) = \left(\sum_{i\in\mathcal{K}}\mathrm{Tr}\left(\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_i^t\right)+\sigma_0^2\right)^{-1}$.

In iteration $(t+1)$, we solve the following SDP problem.

$$\max_{\mathbf{W},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{p}}} \quad s \tag{2.14a}$$

$$\text{s.t.} \quad R_{\mathrm{p},k} + R_{\mathrm{c},k} \geq s \tag{2.14b}$$

$$\sum_{k\in\mathcal{M}}\mathrm{Tr}(\mathbf{W}_i) \leq P \tag{2.14c}$$

$$\hat{g}_{\mathrm{p},k}(\mathbf{W};\mathbf{W}^t) \leq 0, \forall k \in \mathcal{K} \tag{2.14d}$$

$$\hat{g}_{\mathrm{c},k}(\mathbf{W};\mathbf{W}^t) \leq 0, \forall k \in \mathcal{K} \tag{2.14e}$$

$$\mathrm{rank}(\mathbf{W}_i) = 1,\ \forall i \in \mathcal{M} \tag{2.14f}$$

$$\mathbf{W}_i \succeq \mathbf{0},\ \forall i \in \mathcal{M}, \tag{2.14g}$$

where $\hat{g}_{\mathrm{p},k}(\mathbf{W};\mathbf{W}^t) = \hat{g}^{\bar{C}}_{\mathrm{p},k}(\mathbf{W};\mathbf{W}^t) + g^{C}_{\mathrm{p},k}(\mathbf{W})$ and $\hat{g}_{\mathrm{c},k}(\mathbf{W};\mathbf{W}^t) = \hat{g}^{\bar{C}}_{\mathrm{c},k}(\mathbf{W};\mathbf{W}^t) + g^{C}_{\mathrm{c},k}(\mathbf{W})$.

Note that the last two constraints (2.14f) and (2.14g) ensure that $\mathbf{W}_i = \mathbf{w}_i\mathbf{w}_i^H$ holds true. Despite we construct the convex surrogate function, Problem (2.14) is still non-convex. The non-convexity comes from the *rank-one* constraint (2.14f). SDR deals with this problem by dropping the rank-one constraint and obtaining the solution to the SDP problem with standard optimization tools. However, SDR may not provide the optimal rank-one solution for the considered RS system.

Theorem 1. *If* $\mathbf{W}^*$ *is the optimal solution to Problem (2.14) without constraint (2.14f), then* $\mathrm{Rank}(\mathbf{W}_k) \leq 1$, $k \in \mathcal{K}$. *However, the rank of* $\mathbf{W}_{\mathrm{c}}$ *may be larger than one, which is bounded as* $\mathrm{Rank}(\mathbf{W}_{\mathrm{c}}) \leq \min\{M, K\}$.

Proof. The proof is provided in Appendix A.1. □

The rank of the private covariance matrix $\mathbf{W}_k$ always satisfy the rank-one constraint, which means the beamforming vector $\mathbf{w}_k$ can always be recovered uniquely from the covariance matrix with eigen-value decomposition [64]. However, the rank of the common covariance matrix may not be one. In the worst case, the rank may be $\min\{M, K\}$. When recovering $\mathbf{w}_{\mathrm{c}}$ from $\mathbf{W}_{\mathrm{c}}$, there may be an energy loss. As a result, the performance could be affected.

2.3.2 The BCD-based Approach for IRS: Φ_1

Refs. [40, 42, 45] have proposed a weighted minimum mean square error (WMMSE) approach to solve RS beamforming design problem. In this subsection, we adapt it into the IRS-assisted system. The WMMSE method utilizes the relationship between mutual information and MMSE to find a stationary solution. Let the estimated common message of UE k be denoted as $\hat{s}_{\mathrm{c},k} = g_{\mathrm{c},k}y_k$ with $g_{\mathrm{c},k}$ being a scalar equalizer. Then an estimate of s_k can be expressed as $\hat{s}_k = g_{\mathrm{p},k}(y_k - \mathbf{h}_k^H\mathbf{w}_{\mathrm{c}}s_{\mathrm{c},k})$, since the interference from the common message can be removed at each UE. Then, the mean-squared errors (MSE) of the common

message and the private message can be expressed respectively as

$$\epsilon_{\mathrm{c},k} = \mathbb{E}[|\hat{s}_{\mathrm{c},k} - s_{\mathrm{c},k}|^2] = |g_{\mathrm{c},k}|^2 T_{\mathrm{c},k} - 2\mathrm{Re}(g_{\mathrm{c},k}\mathbf{h}_k^H \mathbf{w}_{\mathrm{c}}) + 1, \tag{2.15}$$

$$\epsilon_{\mathrm{p},k} = \mathbb{E}[|\hat{s}_k - s_k|^2] = |g_{\mathrm{p},k}|^2 T_{\mathrm{p},k} - 2\mathrm{Re}(g_{\mathrm{p},k}\mathbf{h}_k^H \mathbf{w}_k) + 1, \tag{2.16}$$

where

$$T_{\mathrm{c},k} = \mathbb{E}[|y_k|^2] = \sum_{i\in\mathcal{M}} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_0^2,$$
$$T_{\mathrm{p},k} = \mathbb{E}[|y_k|^2] - |\mathbf{h}_k^H \mathbf{w}_{\mathrm{c}}|^2 = \sum_{i\in\mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_i|^2 + \sigma_0^2.$$

To minimize the MSE of both the common and the private messages, we take the derivative of the MSE in (2.15) w.r.t. the scalar equalizer, i.e., $\partial\epsilon_{\mathrm{c},k}/\partial g_{\mathrm{c},k} = 0$ and $\partial\epsilon_{\mathrm{p},k}/\partial g_{\mathrm{p},k} = 0$. Then we obtain the optimal MMSE equalizer as

$$g_{\mathrm{c},k}^{\mathrm{MMSE}} = \mathbf{h}_k^H \mathbf{w}_{\mathrm{c}}/T_{\mathrm{c},k}, \quad g_{\mathrm{p},k}^{\mathrm{MMSE}} = \mathbf{h}_k^H \mathbf{w}_k/T_{\mathrm{p},k}. \tag{2.18}$$

Substituting the optimal equalizer (2.18) into the MSE (2.15), we obtain the minimum MSE as

$$\epsilon_{\mathrm{c},k}^{\mathrm{MMSE}} = 1 - \frac{|\mathbf{h}_k^H \mathbf{w}_{\mathrm{c}}|^2}{T_{\mathrm{c},k}}, \quad \epsilon_{\mathrm{p},k}^{\mathrm{MMSE}} = 1 - \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{T_{\mathrm{p},k}}. \tag{2.19}$$

By comparing (2.19) with (2.3) and (2.5), we obtain the rate-MMSE relationship as

$$r_{\mathrm{c},k} = -\log_2(\epsilon_{\mathrm{c},k}^{\mathrm{MMSE}}), \quad R_{\mathrm{p},k} = -\log_2(\epsilon_k^{\mathrm{MMSE}}). \tag{2.20}$$

Meanwhile, the augmented weighted MSE (WMSE) can be defined as

$$\xi_{\mathrm{c},k} = u_{\mathrm{c},k}\epsilon_{\mathrm{c},k}^{\mathrm{MMSE}} - \ln(u_{\mathrm{c},k}), \tag{2.21a}$$

$$\xi_{\mathrm{p},k} = u_{\mathrm{p},k}\epsilon_{\mathrm{p},k}^{\mathrm{MMSE}} - \ln(u_{\mathrm{p},k}), \tag{2.21b}$$

where $u_{\mathrm{c},k} > 0$ and $u_{\mathrm{p},k} > 0$ are the associated weights with UE k's MSE. The weighted MMSE (WMMSE) is defined as the minimum augmented WMSE over all possible weights, i.e., $\xi_{\mathrm{c},k}^{\mathrm{MMSE}} = \min_{u_{\mathrm{c},k}} \xi_{\mathrm{c},k}$ and $\xi_{\mathrm{p},k}^{\mathrm{MMSE}} = \min_{u_{\mathrm{p},k}} \xi_{\mathrm{p},k}$. Now we take derivative w.r.t. the weights in (2.21), i.e., $\partial\xi_{\mathrm{c},k}/\partial u_{\mathrm{c},k} = 0$ and $\partial\xi_{\mathrm{p},k}/\partial u_{\mathrm{p},k} = 0$. Then we obtain the optimal weight as

$$u_{\mathrm{c},k}^{\mathrm{MMSE}} = \frac{1}{\epsilon_{\mathrm{c},k}^{\mathrm{MMSE}}}, \quad u_{\mathrm{p},k}^{\mathrm{MMSE}} = \frac{1}{\epsilon_{\mathrm{p},k}^{\mathrm{MMSE}}}. \tag{2.22}$$

Combining (2.20), (2.21) and (2.22)), we obtain the rate-WMMSE relationship as

$$\xi_{\mathrm{c},k}^{\mathrm{MMSE}} = 1 + \ln\left(\epsilon_{\mathrm{c},k}^{\mathrm{MMSE}}\right) = 1 - r_{\mathrm{c},k}\ln 2, \tag{2.23a}$$

$$\xi_{\mathrm{p},k}^{\mathrm{MMSE}} = 1 + \ln\left(\epsilon_{\mathrm{p},k}^{\mathrm{MMSE}}\right) = 1 - R_{\mathrm{p},k}\ln 2. \tag{2.23b}$$

This will be the core relationship in the WMMSE algorithm.

With the rate-WMMSE relationship (2.23), the max-min rate problem in (2.9) can be equivalently rewritten as

$$\max_{\mathbf{w},\mathbf{G},\mathbf{U},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{p}},\mathbf{v}} \quad s \tag{2.24a}$$

$$\text{s.t.} \quad \frac{1-\xi_{\mathrm{c},k}}{\ln 2} \geq \sum_{k\in\mathcal{K}} R_{\mathrm{c},k} \tag{2.24b}$$

$$\frac{1-\xi_{\mathrm{p},k}}{\ln 2} \geq R_{\mathrm{p},k} \tag{2.24c}$$

$$R_{\mathrm{c},k} + R_{\mathrm{p},k} \geq s \tag{2.24d}$$

$$v_i \in \Phi_1, \, \forall i \tag{2.24e}$$

$$(2.9b), (2.7),$$

where $\mathbf{G} = \{g_{\mathrm{c},k}, g_{\mathrm{p},k} | k \in \mathcal{K}\}$ are the sets of scalar equalizers, $\mathbf{U} = \{u_{\mathrm{c},k}, u_{\mathrm{p},k} | k \in \mathcal{K}\}$ is the set of associated weights of UEs' MSE, and $\xi_{\mathrm{c},k}$ and $\xi_{\mathrm{p},k}$ are given in (2.21).

Theorem 2. *The stationary solution to Problem (2.24) that satisfies conditions (2.18) and (2.22) is also a stationary solution to Problem (2.9).*

Proof. We refer readers to [42] for a detailed proof. □

Substituting $\mathbf{h}_k = \mathbf{h}_{\mathrm{d},k} + \mathbf{H}_k^H \mathbf{v}$ into (2.17), we have

$$T_{\mathrm{c},k} = \mathbf{v}^H \mathbf{Q}_{\mathrm{c},k} \mathbf{v} + 2\mathrm{Re}\left(\mathbf{v}^H \mathbf{q}_{\mathrm{c},k}\right) + \alpha_{\mathrm{c},k} + \sigma_0^2,$$
$$T_{\mathrm{p},k} = \mathbf{v}^H \mathbf{Q}_{\mathrm{p},k} \mathbf{v} + 2\mathrm{Re}\left(\mathbf{v}^H \mathbf{q}_{\mathrm{p},k}\right) + \alpha_{\mathrm{p},k} + \sigma_0^2,$$

where $\mathbf{Q}_{\mathrm{c},k} = \sum_{i\in\mathcal{M}} \mathbf{H}_k \mathbf{w}_i \mathbf{w}_i^H \mathbf{H}_k^H$, $\mathbf{q}_{\mathrm{c},k} = \sum_{i\in\mathcal{M}} \mathbf{H}_k \mathbf{w}_i \mathbf{w}_i^H \mathbf{h}_{\mathrm{d},k}$, $\alpha_{\mathrm{c},k} = \sum_{i\in\mathcal{M}} |\mathbf{h}_{\mathrm{d},k}^H \mathbf{w}_i|^2$, $\mathbf{Q}_{\mathrm{p},k} = \sum_{i\in\mathcal{K}} \mathbf{H}_k \mathbf{w}_i \mathbf{w}_i^H \mathbf{H}_k^H$, $\mathbf{q}_{\mathrm{p},k} = \sum_{i\in\mathcal{K}} \mathbf{H}_k \mathbf{w}_i \mathbf{w}_i^H \mathbf{h}_{\mathrm{d},k}$, and $\alpha_{\mathrm{p},k} = \sum_{i\in\mathcal{K}} |\mathbf{h}_{\mathrm{d},k}^H \mathbf{w}_i|^2$.

Although Problem (2.24) is till non-convex, it has a block-wise convex property w.r.t. $\mathbf{w}$, $(\mathbf{G}, \mathbf{U})$, and $\mathbf{v}$. When $\mathbf{w}$ and $\mathbf{v}$ are fixed, the problem is convex w.r.t. $(\mathbf{G}, \mathbf{U})$. When $\mathbf{v}$ and $(\mathbf{G}, \mathbf{U})$ are fixed, we will obtain $\xi_{\mathrm{c},k}$ and $\xi_{\mathrm{p},k}$ as shown in (2.27), which is a second-order cone in terms of $\mathbf{w}$. When $\mathbf{w}$ and $(\mathbf{G}, \mathbf{U})$ are fixed, we obtain $\xi_{\mathrm{c},k}$ and $\xi_{\mathrm{p},k}$ as shown in (2.28),

which is also a second-order cone in terms of $\mathbf{v}$. Hence the block coordinate descent (BCD) method, which successively optimizes each of the variables, can be used. The details of the BCD algorithm are as follows.

1. *The optimization of* $(\mathbf{G}, \mathbf{U})$*:* The scaling factor $\mathbf{G}$ and the optimal weight $\mathbf{U}$ have closed-form expressions, which are given in (2.18) and (2.22), respectively.

2. *The optimization of* $\mathbf{w}$*:* The optimal $\mathbf{w}$ can be obtained by solving the following problem.

$$\begin{aligned} \max_{\mathbf{w},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{p}}} \quad & s \\ \text{s.t.} \quad & (2.24b),\ (2.24c),\ (2.24d),\ (2.9b), (2.7), \end{aligned} \tag{2.25}$$

   where $\xi_{\mathrm{c},k}$ and $\xi_{\mathrm{p},k}$ are given in (2.27).

3. *The optimization of* $\mathbf{v}$*:* The optimal $\mathbf{v}$ can be obtained by solving the following problem.

$$\begin{aligned} \max_{\mathbf{v},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{u}}} \quad & s \\ \text{s.t.} \quad & (2.24b),\ (2.24c),\ (2.24d),\ (2.7), \\ & |v_i|^2 \leq 1, \forall i. \end{aligned} \tag{2.26}$$

   where $\xi_{\mathrm{c},k}$ and $\xi_{\mathrm{p},k}$ are given in (2.28).

To summarize, the proposed algorithm successively optimizes the variables. Its complexity mainly comes from Step 2 and Step 3, which are two quadratically constrained convex optimization problems (QCCP). The QCCP problem in Step 2 consists of $d_1 = 4K + 2$ constraints and $d_2 = 1 + 2K + MK$ variables. Hence, the worst case computational complexity of solving this problem is $\mathcal{O}(d_1 d_2^2 + d_2^3)\sqrt{d_1}\log(1/\epsilon)$ [67] for a given stopping criterion ϵ. Similarly, the problem in Step 3 is also a QCCP, which has $d_3 = 4K + 1$ constraints and $d_4 = 1 + 2K + N^2$ variables. The computational complexity in Step

$$\xi_{\mathrm{c},k}(\mathbf{w}) = u_{\mathrm{c},k}\left[|g_{\mathrm{c},k}|^2\left(\sum_{i\in\mathcal{M}}\mathbf{w}_i^H\mathbf{h}_k\mathbf{h}_k^H\mathbf{w}_i+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{c},k}\mathbf{h}_k^H\mathbf{w}_{\mathrm{c}})+1\right]-\ln(u_{\mathrm{c},k}), \quad (2.27a)$$

$$\xi_{\mathrm{p},k}(\mathbf{w}) = u_{\mathrm{p},k}\left[|g_{\mathrm{p},k}|^2\left(\sum_{i\in\mathcal{K}}\mathbf{w}_i^H\mathbf{h}_k\mathbf{h}_k^H\mathbf{w}_i+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{p},k}\mathbf{h}_k^H\mathbf{w}_k)+1\right]-\ln(u_{\mathrm{p},k}). \quad (2.27b)$$

$$\begin{aligned}\xi_{\mathrm{c},k}(\mathbf{v}) =& u_{\mathrm{c},k}\left[|g_{\mathrm{c},k}|^2\left(\mathbf{v}^H\mathbf{Q}_{\mathrm{c},k}\mathbf{v}+2\mathrm{Re}\left(\mathbf{v}^H\mathbf{q}_{\mathrm{c},k}\right)+\alpha_{\mathrm{c},k}+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{c},k}\mathbf{h}_{\mathrm{d},k}^H\mathbf{w}_{\mathrm{c}}+g_{\mathrm{c},k}\mathbf{v}^H\mathbf{H}_k\mathbf{w}_{\mathrm{c}})+1\right]\\ &-\ln(u_{\mathrm{c},k}), \quad (2.28a)\\ \xi_{\mathrm{p},k}(\mathbf{v}) =& u_{\mathrm{p},k}\left[|g_{\mathrm{p},k}|^2\left(\mathbf{v}^H\mathbf{Q}_{\mathrm{p},k}\mathbf{v}+2\mathrm{Re}\left(\mathbf{v}^H\mathbf{q}_{\mathrm{p},k}\right)+\alpha_{\mathrm{p},k}+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{p},k}\mathbf{h}_{\mathrm{d},k}^H\mathbf{w}_k+g_{\mathrm{p},k}\mathbf{v}^H\mathbf{H}_k\mathbf{w}_k)+1\right]\\ &-\ln(u_{\mathrm{p},k}). \quad (2.28b)\end{aligned}$$

3 is $\mathcal{O}(d_3d_4^2+d_4^3)\sqrt{d_3}\log(1/\epsilon)$. Hence, the overall complexity of the BCD algorithm is $\mathcal{O}(I_{\mathrm{in}}(M^3K^{3.5}+(K+N^2)^3K^{0.5})\log(1/\epsilon))$, where I_{in} is the total number of outer iterations of the BCD algorithm.

Theorem 3. *The proposed BCD algorithm will converge after a finite number of steps.*

Proof. Denote the objective function in (2.24) as $F(\mathbf{G},\mathbf{U},\mathbf{w},\mathbf{v})$. Then we have

$$F(\mathbf{G}^{t-1},\mathbf{U}^{t-1},\mathbf{w}^t,\mathbf{v}^t)\le F(\mathbf{G}^t,\mathbf{U}^t,\mathbf{w}^t,\mathbf{v}^t)\le\ F(\mathbf{G}^t,\mathbf{U}^t,\mathbf{w}^{t+1},\mathbf{v}^t)\le F(\mathbf{G}^t,\mathbf{U}^t,\mathbf{w}^{t+1},\mathbf{v}^{t+1}),$$

where the first inequality holds since $\mathbf{U}^t$ is the optimal weight and $\mathbf{G}$ is the optimal MMSE equalizer to minimize the MSE. The second inequality holds since $\mathbf{w}^{t+1}$ is the optimal solution to Problem (2.25) and the third inequality holds since $\mathbf{v}^{t+1}$ is the optimal solution to Problem (2.42). □

2.3.3 The Penalty-based BCD Approach for IRS:Φ_2

In the IRS:Φ_2 case, we need to deal with the non-convex constraints with Φ_2. The problem becomes

$$\max_{\mathbf{w},\mathbf{G},\mathbf{U},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{p}},\mathbf{v}} \quad s, \tag{2.29a}$$

$$\text{s.t.} \quad (2.24b),\ (2.24c),\ (2.24d),\ (2.9b), (2.7)$$

$$v_i \in \Phi_2,\ \forall i. \tag{2.29b}$$

To simplify the optimization of $\mathbf{v}$ and facilitate parallel updating, we propose to use the penalty dual decomposition (PDD) framework. This method introduces an auxiliary variable $\mathbf{u}$. Hence Problem (2.29) is equivalently transformed to

$$\max_{\mathbf{w},\mathbf{G},\mathbf{U},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{p}},\mathbf{v}} \quad s \tag{2.30a}$$

$$\text{s.t.} \quad (2.24b),\ (2.24c),\ (2.24d),\ (2.9b),\ (2.7)$$

$$\mathbf{v} = \mathbf{u}, \tag{2.30b}$$

$$u_i \in \Phi_2,\ \forall i. \tag{2.30c}$$

To deal with the constraint $\mathbf{v} = \mathbf{u}$, Problem (2.29) is further transformed to

$$\max_{\mathbf{w},\mathbf{G},\mathbf{U},s,\mathbf{R}_{\mathrm{c}},\mathbf{R}_{\mathrm{p}},\mathbf{v}} \quad z = s - \frac{1}{2\rho}||\mathbf{v} - \mathbf{u} + \rho\boldsymbol{\mu}||^2 \tag{2.31a}$$

$$\text{s.t.} \quad (2.24b),\ (2.24c),\ (2.24d),\ (2.9b),\ (2.7)$$

$$u_i \in \Phi_2,\ \forall i, \tag{2.31b}$$

where ρ is a penalty coefficient and $\boldsymbol{\mu}$ is the dual variable vector associated with constraint $\mathbf{v} = \mathbf{u}$. To optimize the variable vector $\mathbf{u}$, we solve the following sub-problem.

$$\min_{\mathbf{u}} \quad ||\mathbf{v} - \mathbf{u} + \rho\boldsymbol{\mu}||^2 \tag{2.32}$$

$$\text{s.t.} \quad u_i \in \Phi_2, \forall i. \tag{2.33}$$

For this problem, we notice that u_i is decoupled in both the objective function and the constraint. Hence we can obtain the optimal solution in parallel. Note that

$$\begin{aligned} &||\mathbf{v} - \mathbf{u} + \rho\boldsymbol{\mu}||^2 \\ &= (\mathbf{v} + \rho\boldsymbol{\mu})^H(\mathbf{v} + \rho\boldsymbol{\mu}) + \mathbf{u}^H\mathbf{u} - 2\text{Re}(\mathbf{u}^H(\mathbf{v} + \rho\boldsymbol{\mu})). \end{aligned} \tag{2.34}$$

When $u_i \in \Phi_2$, we have $|u_i| = 1$. Hence $\mathbf{u}^H\mathbf{u} = N$. To minimize the term $||\mathbf{v} - \mathbf{u} + \rho\boldsymbol{\mu}||^2$, the phase of $\mathbf{u}$ and the phase of $\mathbf{v} + \rho\boldsymbol{\mu}$ should be the same, i.e.,

$$u_i = \exp\left\{j\angle\bar{v}_i\right\}, \tag{2.35}$$

where $\bar{v}_i$ is the element of $\bar{\mathbf{v}}$ and $\bar{\mathbf{v}} = \mathbf{v} + \rho\boldsymbol{\mu}$.

We design an iterative algorithm where the inner loop optimizes variables $(\mathbf{G}, \mathbf{U}, \mathbf{w}, \mathbf{v}, \mathbf{u})$ successively. In the outer loop, the dual variables are decreased by a factor of α. Finally, the dual variables are updated by

$$\boldsymbol{\mu} = \boldsymbol{\mu} + \frac{1}{\rho}(\mathbf{v} - \mathbf{u}). \tag{2.36}$$

As can be seen, algorithm 1 consists of an outer loop which decreases the penalty factor, and an inner loop which optimizes the variables successively. With the decrease of the penalty coefficient, the penalty term becomes larger and will eventually guarantee that $\mathbf{u} = \mathbf{v}$. For the inner loop, the complexity is the same as that of the BCD algorithm. Steps 11 and 12 in

the outer loop also have closed-form expressions. We conclude that the overall complexity of Algorithm 1 is $\mathcal{O}(I_{\text{out}}I_{\text{in}}(M^3K^{3.5}+(K+N^2)^3K^{0.5})\log(1/\epsilon))$.

Algorithm 1 Penalty-based BCD Algorithm

Require: P;
Ensure: $\mathbf{w}^*$, $\mathbf{v}^*$;
1: Initialize outer loop iteration index $I_{\text{out}}=0$; Initialize the beamforming vector with equal power allocation $\frac{P}{M(K+1)}$; Generate vector $\mathbf{v}$ and $\mathbf{u}$ with reflection amplitude 1 and random phases; Initialize $\boldsymbol{\mu}=\mathbf{0}$ and $\rho=200$.
2: **repeat**
3: Initialize inner loop iteration index $I_{\text{in}}=0$;
4: **repeat**
5: Update $\mathbf{G}^t$ and $\mathbf{U}^t$ based on (2.18) and (2.22);
6: Obtain $\mathbf{w}^{t+1}$ by solving convex optimization Problem (2.25);
7: Obtain $\mathbf{v}^{t+1}$ by solving Problem (2.31a);
8: Obtain $\mathbf{u}^{t+1}$ from (2.35) when $\Phi=\Phi_2$;
9: $I_{\text{in}}++$;
10: **until** $\frac{|z^t-z^{t-1}|}{|z^{t-1}|}\geq\epsilon$ or $I_{\text{in}}>I_{\text{in}}^{\max}$
11: Update $\boldsymbol{\mu}$ based on (2.36);
12: Update ρ by $\rho=c\rho$;
13: $I_{\text{out}}++$;
14: **until** $||\mathbf{u}-\mathbf{v}||_2\leq\epsilon$

2.4 Joint beamforming design with imperfect CSI

In practice, joint beamforming design requires precise CSI, which is quite challenging to obtain. In this section, we consider the scenario where the users know the perfect CSI while the BS only has the imperfect CSI estimation due to the feedback errors and mismatches.

We assume that the CSI error model is given by

$$\mathbf{H}_k=\hat{\mathbf{H}}_k+\Delta\mathbf{H}_k,$$
$$\mathbf{h}_{\text{d},k}=\hat{\mathbf{h}}_{\text{d},k}+\Delta\mathbf{h}_{\text{d},k}$$

where $\hat{\mathbf{H}}_k$ and $\hat{\mathbf{h}}_{\text{d},k}$ denotes the estimated cascaded channel and the estimated direct channel, respectively. $\Delta\mathbf{H}_k$ and $\Delta\mathbf{h}_{\text{d},k}$ denote the CSI error of the cascaded channel and the

direct channel, respectively. It is shown in [68] that when the number of the reflecting elements on the IRS is large, the distribution of $\text{vec}(\Delta\mathbf{H}_k)$ is approximated by Gaussian distribution. In the rest of this chapter, we assume that $\text{vec}(\Delta\mathbf{H}_k) \sim \mathcal{CN}(\mathbf{0}, \sigma^2_{\Delta H,k}\mathbf{I})$ and $\Delta\mathbf{h}_{\text{d},k} \sim \mathcal{CN}(\mathbf{0}, \sigma^2_{\Delta \text{d},k}\mathbf{I})$. Moreover, the CSI error variances decay at a speed proportional to the channel variances, i.e., $\sigma^2_{\Delta \text{d},k} = \frac{\delta}{M}||\mathbf{h}_{\text{d},k}||^2$ and $\sigma^2_{\Delta H,k} = \frac{\delta}{MN}||\mathbf{H}_k||^2_F$ where $\delta \in [0,1]$ measures the relative CSIT uncertainties [69, 70].

The BS is assumed to know the the estimated channel $\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{\text{d},k} + \hat{\mathbf{H}}_k^H\mathbf{v}$ and the conditional probability $p_{\mathbf{h}_k|\hat{\mathbf{h}}_k}(\mathbf{h}_k|\hat{\mathbf{h}}_k)$. According to [45], we can formulate the average rate (AR) optimization problem

$$\max_{\mathbf{w},\mathbf{R}_\text{c},\mathbf{R}_\text{p},\mathbf{v},s} \quad s \tag{2.37a}$$

$$\text{s.t.} \quad (2.9b), (2.9c) \tag{2.37b}$$

$$R_{\text{p},k} + R_{\text{c},k} \geq s \tag{2.37c}$$

$$R_{\text{p},k} \leq \mathbb{E}_{\mathbf{h}_k|\hat{\mathbf{h}}_k}[\log_2(1+\gamma_{\text{p},k})|\hat{\mathbf{h}}_k], \ \forall k \tag{2.37d}$$

$$\sum_{k\in\mathcal{K}} R_{\text{c},k} \leq \mathbb{E}_{\mathbf{h}_k|\hat{\mathbf{h}}_k}[\log_2(1+\gamma_{\text{c},k})|\hat{\mathbf{h}}_k], \ \forall k, \tag{2.37e}$$

To deal with the challenging stochastic optimization problem, we resort to the sample average approximation (SAA) method, which approximate the stochastic problem with a deterministic one. To be specific, for a given channel estimate $\hat{\mathbf{h}}_k$, we use set $\mathbb{H}^L$ to denote L i.i.d. channel realizations drawn from the conditional distribution $f_{\mathbf{h}_k|\hat{\mathbf{h}}_k}(\mathbf{h}_k|\hat{\mathbf{h}}_k)$.

$$\mathbb{H}^L = \{\mathbf{h}_k^l = \hat{\mathbf{h}}_k + \Delta\mathbf{h}_k^l | \hat{\mathbf{h}}_k, 1 \leq l \leq L, k \in \mathcal{K}\} \tag{2.38}$$

It was proven in [45] that when the sample size L is large enough, the stochastic rate can be approximated by the sample average. Hence if we define the sample average as

$$\bar{R}_{\text{p},k} = \frac{1}{L}\sum_{l=1}^{L} R_{\text{p},k}^l, \qquad \bar{R}_{\text{c},k} = \frac{1}{L}\sum_{l=1}^{L} R_{\text{c},k}^l$$

where $R_{\mathrm{p},k}^l$ and $R_{\mathrm{c},k}^l$ are the private rate and common rate associated with the channel realization $\{\mathbf{h}_k^l\}_k$. This leads to the SAA problem

$$\max_{\mathbf{w},\bar{\mathbf{R}}_\mathrm{c},\bar{\mathbf{R}}_\mathrm{p},\mathbf{v},s} \quad s \tag{2.39a}$$

$$\text{s.t.} \quad (2.9b), (2.9c) \tag{2.39b}$$

$$\bar{R}_{\mathrm{p},k} + \bar{R}_{\mathrm{c},k} \geq s \tag{2.39c}$$

$$\bar{R}_{\mathrm{p},k} \leq \frac{1}{L}\sum_{l=1}^{L} \log_2(1+\gamma_{\mathrm{p},k}^l),\ \forall k \tag{2.39d}$$

$$\sum_{k\in\mathcal{K}} \bar{R}_{\mathrm{c},k} \leq \frac{1}{L}\sum_{l=1}^{L} \log_2(1+\gamma_{\mathrm{c},k}^l),\ \forall k, \tag{2.39e}$$

where $\bar{\mathbf{R}}_\mathrm{p} = \{\bar{R}_{\mathrm{p},k} | k \in \mathcal{K}\}$, $\bar{\mathbf{R}}_\mathrm{c} = \{\bar{R}_{\mathrm{c},k} | k \in \mathcal{K}\}$, and $\gamma_{\mathrm{p},k}^l$ and $\gamma_{\mathrm{c},k}^l$ are the associated SNR with the channel realization $\mathbf{h}_k^l$. In the optimization process, the same beamforming vector $\mathbf{w}$ is fixed for all the channel realizations.

Although this problem is deterministic, it is still non-convex. Following a similar step in section 2.3, we approximate

$$\mathbb{E}_{\mathbf{h}_k|\hat{\mathbf{h}}_k}[\min_{u_{\mathrm{c},k}} \xi_{\mathrm{c},k} | \hat{\mathbf{h}}_k] \approx \min_{u_{\mathrm{c},k}} \bar{\xi}_{\mathrm{c},k} = \bar{\xi}_{\mathrm{c},k}^{\mathrm{MMSE}} = 1 - \bar{r}_{\mathrm{c},k} \ln 2$$

$$\mathbb{E}_{\mathbf{h}_k|\hat{\mathbf{h}}_k}[\min_{u_{\mathrm{p},k}} \xi_{\mathrm{p},k} | \hat{\mathbf{h}}_k] \approx \min_{u_{\mathrm{p},k}} \bar{\xi}_{\mathrm{p},k} = \bar{\xi}_{\mathrm{p},k}^{\mathrm{MMSE}} = 1 - \bar{R}_{\mathrm{p},k} \ln 2$$

where $\bar{r}_{\mathrm{c},k} = \frac{1}{L}\sum_{l=1}^{L}\log_2(1+\gamma_{c,k}^l)$. With this relationship, the SAA max-min rate problem (2.39) can be equivalently transformed into

$$\max_{\mathbf{w},\bar{\mathbf{G}},\bar{\mathbf{U}},s,\bar{\mathbf{R}}_{\mathrm{c}},\bar{\mathbf{R}}_{\mathrm{p}},\mathbf{v}} \quad s \tag{2.40a}$$

$$\text{s.t.} \quad \frac{1-\bar{\xi}_{\mathrm{c},k}}{\ln 2} \geq \sum_{k\in\mathcal{K}} \bar{R}_{\mathrm{c},k} \tag{2.40b}$$

$$\frac{1-\bar{\xi}_{\mathrm{p},k}}{\ln 2} \geq \bar{R}_{\mathrm{p},k} \tag{2.40c}$$

$$(2.39c), (2.24d), (2.24e), (2.9b), (2.7),$$

where $\bar{\mathbf{G}} = \left\{g_{\mathrm{c},k}^l, g_{\mathrm{p},k}^l | k \in \mathcal{K}, 1 \leq l \leq L\right\}$ and $\bar{\mathbf{U}} = \left\{u_{\mathrm{c},k}^l, u_{\mathrm{p},k}^l | k \in \mathcal{K}, 1 \leq l \leq L\right\}$ are the scaling equalizer sets. The optimal $g_{\mathrm{c},k}^l, g_{\mathrm{p},k}^l$ can be obtained from (2.18) based on a specific $\mathbf{h}_k^l$. The optimal $u_{\mathrm{c},k}^l, u_{\mathrm{p},k}^l$ can be obtained from (2.22) based on $\mathbf{h}_k^l$.

The block-wise convex property still holds. As a result, we have the SAA-BCD algorithm as follows.

1. *The optimization of* $(\bar{\mathbf{G}}, \bar{\mathbf{U}})$.

2. *The optimization of* $\mathbf{w}$*:* The optimal $\mathbf{w}$ can be obtained by solving the following problem.

$$\max_{\mathbf{w},s,\bar{\mathbf{R}}_{\mathrm{c}},\bar{\mathbf{R}}_{\mathrm{p}}} \quad s \tag{2.41}$$

$$\text{s.t.} \quad (2.40b),\ (2.40c),\ (2.24d),\ (2.9b), (2.7),$$

where $\bar{\xi}_{\mathrm{c},k}$ and $\bar{\xi}_{\mathrm{p},k}$ are given in (2.43).

$$\bar{\xi}_{\mathrm{c},k}(\mathbf{w}) = \frac{1}{L}\sum_{l=1}^{L} u_{\mathrm{c},k}^{l}\left[|g_{\mathrm{c},k}^{l}|^2\left(\sum_{i\in\mathcal{M}}\mathbf{w}_i^H\mathbf{h}_k^l\mathbf{h}_k^{l,H}\mathbf{w}_i+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{c},k}^{l}\mathbf{h}_k^{l,H}\mathbf{w}_{\mathrm{c}})+1\right]-\frac{1}{L}\sum_{l=1}^{L}\ln(u_{\mathrm{c},k}^{l}), \tag{2.43a}$$

$$\bar{\xi}_{\mathrm{p},k}(\mathbf{w}) = \frac{1}{L}\sum_{l=1}^{L} u_{\mathrm{p},k}^{l}\left[|g_{\mathrm{p},k}^{l}|^2\left(\sum_{i\in\mathcal{K}}\mathbf{w}_i^H\mathbf{h}_k^l\mathbf{h}_k^{l,H}\mathbf{w}_i+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{p},k}^{l}\mathbf{h}_k^{l,H}\mathbf{w}_{k})+1\right]-\frac{1}{L}\sum_{l=1}^{L}\ln(u_{\mathrm{p},k}^{l}). \tag{2.43b}$$

$$\begin{aligned}\bar{\xi}_{\mathrm{c},k}(\mathbf{v}) =&\frac{1}{L}\sum_{l=1}^{L} u_{\mathrm{c},k}^{l}\left[|g_{\mathrm{c},k}^{l}|^2\left(\mathbf{v}^H\mathbf{Q}_{\mathrm{c},k}^{l}\mathbf{v}+2\mathrm{Re}\left(\mathbf{v}^H\mathbf{q}_{\mathrm{c},k}^{l}\right)+\alpha_{\mathrm{c},k}^{l}+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{c},k}^{l}\mathbf{h}_{\mathrm{d},k}^{l,H}\mathbf{w}_{\mathrm{c}}+g_{\mathrm{c},k}^{l}\mathbf{v}^H\mathbf{H}_k^l\mathbf{w}_{\mathrm{c}})+1\right]\\ &-\frac{1}{L}\sum_{l=1}^{L}\ln(u_{\mathrm{c},k}^{l}),\end{aligned} \tag{2.44a}$$

$$\begin{aligned}\bar{\xi}_{\mathrm{p},k}(\mathbf{v}) =&\frac{1}{L}\sum_{l=1}^{L} u_{\mathrm{p},k}^{l}\left[|g_{\mathrm{p},k}^{l}|^2\left(\mathbf{v}^H\mathbf{Q}_{\mathrm{p},k}^{l}\mathbf{v}+2\mathrm{Re}\left(\mathbf{v}^H\mathbf{q}_{\mathrm{p},k}^{l}\right)+\alpha_{\mathrm{p},k}^{l}+\sigma_0^2\right)-2\mathrm{Re}(g_{\mathrm{p},k}^{l}\mathbf{h}_{\mathrm{d},k}^{l,H}\mathbf{w}_{k}+g_{\mathrm{p},k}^{l}\mathbf{v}^H\mathbf{H}_k^l\mathbf{w}_{k})+1\right]\\ &-\frac{1}{L}\sum_{l=1}^{L}\ln(u_{\mathrm{p},k}^{l}).\end{aligned} \tag{2.44b}$$

3. *The optimization of* $\mathbf{v}$*:* The optimal $\mathbf{v}$ can be obtained by solving the following problem.

$$\begin{aligned}\max_{\mathbf{v},s,\bar{\mathbf{R}}_{\mathrm{c}},\bar{\mathbf{R}}_{\mathrm{u}}}\quad & s \\ \text{s.t.}\quad & (2.40b),\ (2.40c),\ (2.24d),\ (2.7),\\ & |v_i|^2\le 1,\forall i.\end{aligned} \tag{2.42}$$

where $\bar{\xi}_{\mathrm{c},k}$ and $\bar{\xi}_{\mathrm{p},k}$ are given in (2.44).

The penalty-based BCD algorithm for the imperfect CSI case can be similarly derived. We leave this out due to space limitation.

2.5 Analysis of the interplay between IRS and RS

First, we fixed the channel related parameters and investigate the impact of several system parameters, including the number of BS antennas M, the number of UEs K, and the total BS transmit power P, on the MMF performance with respect to the degree of freedom (DoF) metric. DoF is a metric to quantify how well a communication scheme can exploit the spatial dimensions. The DoF of UE k is defined as

$$d_k = \lim_{P\to\infty} \frac{R_k(P)}{\log_2(P)}. \tag{2.45}$$

Accordingly, the MMF multiplexing gain is given by

$$d_{\mathrm{MMF}} = \min_{k\in\mathcal{K}} d_k. \tag{2.46}$$

The multiplexing gain d_k is the pre-log factor of the rate of UE k in the high SNR regime, which indicates the speed that the rate of UE k increases with the SNR. The MMF multiplexing gain d_{MMF} denotes the maximum multiplexing gain that can be simultaneously achieved by all UEs by exploiting the multiple antennas at the BS as well as the reflecting elements at the IRS. Despite that the multiplexing gain is defined in the high SNR regime, it is still helpful to gain insights on the performance limit in the finite SNR regime [53].

In an SDMA system, the common beamforming vector $\mathbf{w}_{\mathrm{c}}$ in (2.3) becomes zero. We now analyze the SINR of each private message. When $M \geq K$, zero-forcing (ZF) beamforming can be used to fully eliminate the interference in an $M \times K$ MIMO channel [71]. In this case, $\gamma_{\mathrm{p},k}$ scales at a speed $\mathcal{O}(P)$, and hence $d_{\mathrm{MMF}} = 1$. When $M < K$, the interference from other UEs cannot be eliminated; hence the rate will saturate at high SNR and $d_{\mathrm{MMF}} = 0$. Since SDMA is a special case of rate-splitting, when $M \geq K$, RS can also achieve a DoF of 1. For RS, when $M < K$, we assume that the common part is allocated with a power $p_{\mathrm{c}} = \mathcal{O}(P)$ and the private part is allocated with a power $p_k = \mathcal{O}(P^{\alpha})$. It follows that the received SINR

of the common stream at the UE scales as $\mathcal{O}(P^{(1-\alpha)})$, which brings a multiplexing gain of $(1-\alpha)$ to the common stream. The private part performs ZF beamforming to transmit K interference-free private streams. The received SINR of each private stream scales as $\mathcal{O}(P^{\alpha})$, resulting in a multiplexing gain of α. Note that the common part multiplexing gain $(1-\alpha)$ is divided equally among the $(K-M)$ UEs. To maximize d_{MMF}, we need to have $\frac{1-\alpha}{K-M}=\alpha$. As a result, the RS scheme achieves a max-min multiplexing gain $\alpha=\frac{1}{1+K-M}$ (see [72] for a detailed proof). That is, the MMF rate of RS scales as $\frac{1}{1+K-M}\log_2(P)$ when $M<K$, which characterizes the speed that the MMF rate increases with the SNR.

When considering the impact of the IRS, the coupling between the optimization variables makes the analysis quite complex. For SDMA, the reconfiguration of the channels does not affect the speed that $\gamma_{\mathrm{p},k}$ scales at, hence $d_{\mathrm{MMF}}=1$ for $M\geq K$ and $d_{\mathrm{MMF}}=0$ for $M<K$. For RS, when $M\geq K$, $d_{\mathrm{MMF}}=1$. When $M<K$, the received SINR of each private stream scales as $\mathcal{O}(P^{\alpha})$, resulting a multiplexing gain of α. However, the received SINR of the common stream at the UE still scale at a speed larger than $\mathcal{O}(P^{1-\alpha})$. This is because the use of IRS makes it possible to deliver strong common signal beams to the users, which brings a multiplexing gain larger than $(1-\alpha)$ to the common stream. We multiply a factor $\beta\geq 1$ to denote the improvement of the optimization of the IRS on the common stream multiplexing gain. This multiplexing gain is shared among the rest of the $K-M$ users. To maximize d_{MMF}, we have $\frac{\beta(1-\alpha)}{K-M}=\alpha$, hence RS with IRS can achieve a max-min multiplexing gain $\alpha=\frac{\beta}{K-M+\beta}(\beta\geq 1)$. It is practically challenging to get a closed form expression of β. However, we can still gain some insights by considering an extreme case where $M=1$ and $K>1$. In this case, the achievable SNR for the common data can be expressed as

$$\gamma_{\mathrm{c},k}=\frac{|\mathbf{w}_{\mathrm{c}}|^2}{\sum_k|\mathbf{w}_k|^2+\sigma_0^2/|\mathbf{h}_k|^2},$$

where $\mathbf{h}_k=\mathbf{h}_{\mathrm{d},k}+\mathbf{H}_k^H\mathbf{v}$. When the beamforming vector $\mathbf{w}_{\mathrm{c}}$ and $\mathbf{w}_k$ are fixed, by adjusting the reflecting elements on the IRS, it is possible to increase the value of $|\mathbf{h}_k|^2$ and increase

the value of $\min(\gamma_{c,k})$ for all k. Hence, an improved common data stream beamforming gain can be achieved. This beamforming gain is shared between the $K - M$ UEs, as a result, the d_{MMF} can be increased.

As IRS only consists of passive elements, accurately obtain the CSI is quite challenging. Beamforming design based on imperfect CSI would generally results in a degraded performance for IRS assisted networks. Fortunately, proposition 12 in [53] suggests that when $M < K$ and the CSI uncertainty is small, RS can achieve the same max min multiplexing gain as the perfect CSI case. In other words, RS has the advantage of being robust with the imperfect CSI. Therefore, it is a perfect fit for RS to integrate with IRS-assisted communication, even under the imperfect CSI setting. We will illustrate it via simulations.

2.6 Simulation Study

We consider an IRS assisted communication scenario as depicted in Fig. 2.2. The parameters and channels are set the same as [64]. In this x-y plane, the IRS is located at location $(50, 0)$m. The UEs are located randomly in a circle around center $(50, 5)$m with a radius of 3m. The BS is located at the origin $(0, 5)$m. The channel coefficients are a combination of distance-dependent large-scale fading and small-scale fading. The large scale path loss model follows $\text{PL}(d) = Ad^{-\alpha}$, where $A = -30$dB is the path loss at a reference distance 1m, d is the distance between the transmitter and receiver, and α is the path loss component. The path loss components for channels $\mathbf{h}_{\text{r},k}$, $\mathbf{h}_{\text{d},k}$, and $\mathbf{G}$ are set to 2.2, 3.5, and 2.2, respectively. For small-scale fading, the Rayleigh fading is assumed for the direct

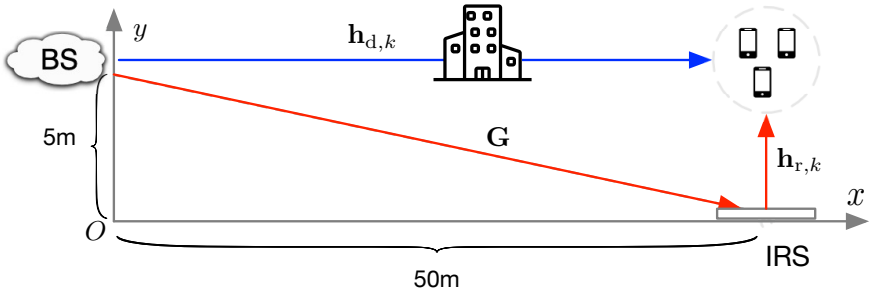


Figure 2.2: Illustration of the IRS-assisted network architecture.

channel $\mathbf{h}_{\mathrm{d},k}$ and the Rician fading is assumed for the IRS reflected channels, i.e.,

$$\mathbf{h}_{\mathrm{d},k} = \sqrt{\mathrm{PL}(d)}\mathbf{h}_{\mathrm{d},k}^{\mathrm{NLOS}}$$

$$\mathbf{G} = \sqrt{\frac{\mathrm{PL}(d)}{K_G+1}}\left(\sqrt{K_G}\mathbf{G}^{\mathrm{LOS}} + \mathbf{G}^{\mathrm{NLOS}}\right)$$

$$\mathbf{h}_{\mathrm{r},k} = \sqrt{\frac{\mathrm{PL}(d)}{K_R+1}}\left(\sqrt{K_R}\mathbf{h}_{\mathrm{r},k}^{\mathrm{LOS}} + \mathbf{h}_{\mathrm{r},k}^{\mathrm{NLOS}}\right)$$

where $K_G = K_R = 3$ is the Rician factors, $\mathbf{G}^{\mathrm{LOS}}$ and $\mathbf{h}_{\mathrm{r},k}^{\mathrm{LOS}}$ are line-of-sight (LoS) components, $\mathbf{h}_{\mathrm{d},k}^{\mathrm{NLOS}}$, $\mathbf{G}^{\mathrm{NLOS}}$ and $\mathbf{h}_{\mathrm{r},k}^{\mathrm{NLOS}}$ are the non-line-of-sight (NLOS) components.

The noise power σ^2 is set to -90dBm. The SNR metric is defined as P/σ_0^2. For the proposed BCD algorithm and Algorithm 1, the stopping criteria is $\epsilon = 0.01$. For Algorithm 1, the value of dual variables is $\rho = 200$ and the scaling constant is $c = 0.1$. Every simulated curve is obtained by averaging over 100 channel realizations.

We compare the performance of the proposed algorithm with three benchmark algorithms. The first is RS without IRS (termed RS: w/o IRS), where there is only a direct channel between the UE and BS. The second is IRS with random IRS (termed RS: random phase) where the phase shift at the IRS is randomly set. The active beamforming at the BS is optimized with the simplified BCD algorithm for the first two cases. The third benchmark is the conventional SDMA with IRS Φ_1 (termed SDMA:Φ_1). Note that SDMA is a special case of the proposed RS system, where the common rate for all UEs is set to zero. The

proposed BCD algorithm and Algorithm 1 can be easily adapted for the SDMA case. For our proposed algorithm, we consider RS with IRS:Φ_1 setting and RS with IRS:Φ_2 setting.

2.6.1 Perfect CSI Case

Convergence of the Proposed Algorithm

In Fig. 2.3, we present the convergence performance of the BCD algorithm and Algorithm 1 for $M = 2$, $K = 4$, $N = 20$, and $P = 20$dBm. For the BCD algorithm, the objective value is the max-min rate defined in (2.24). As can be seen, the max-min rates are monotonically increasing, and converges very quickly (i.e., 4-5 iterations for SDMA:Φ_1 and 10-15 iterations for RS:Φ_1). For Algorithm 1, the objective value is the max-min rate minus the penalty term (see (2.31a)). Both the outer loop iteration and the inner loop iteration of Algorithm 1 are counted. Due to the penalty factor, the objective function value fluctuates over iterations. This is because the inner loop is indeed the BCD algorithm, hence the objective value function keeps increasing in each inner loop. The convergence of the inner loop is guaranteed. However, the outer loop refines the penalty term, which may cause a decrease in the objective value. Such refinement brings fluctuations over iterations. Eventually, the penalty term will go to zero to ensure that $\mathbf{u} = \mathbf{v}$ and Algorithm 1 will converge after finite steps.

Impact of the Number of IRS Reflecting Elements

The max-min rate versus the number of reflecting elements on the IRS is plotted in Fig. 2.4. first, the performance of RS with IRS improves with the increase of N, while the SDMA scheme with IRS and the RS without IRS scheme remain nearly unchanged. This is expected since more reflecting elements on IRS for RS brings a performance gain, regardless of the setting of M and K. However, for SDMA:Φ_1, the MMF is 0 when $M < K$ as analyzed in Section V-A. This shows that the introduction of IRS does not bring any performance gain when $M < K$. It is quite straightforward that the performance of RS without IRS

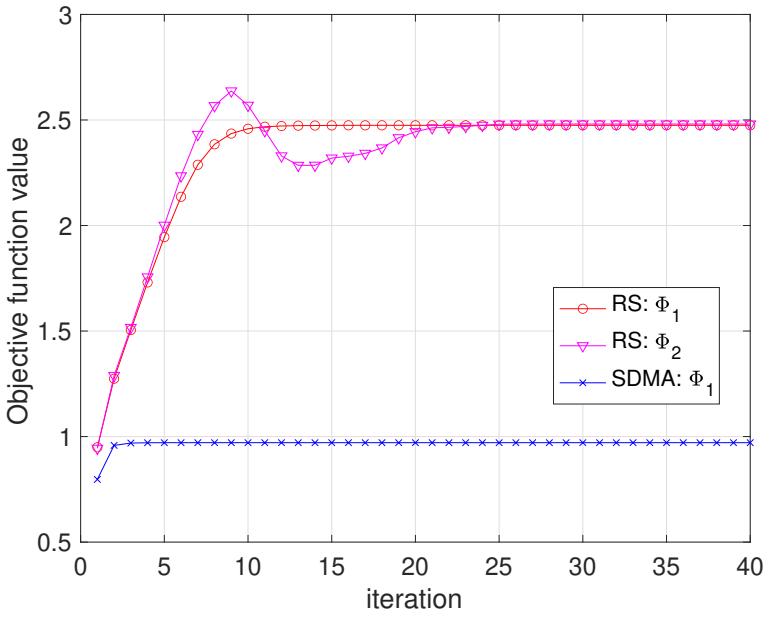


Figure 2.3: The objective function value versus the number of iterations for $M = 2$, $K = 4$, $N = 20$, and $P = 20$dBm.

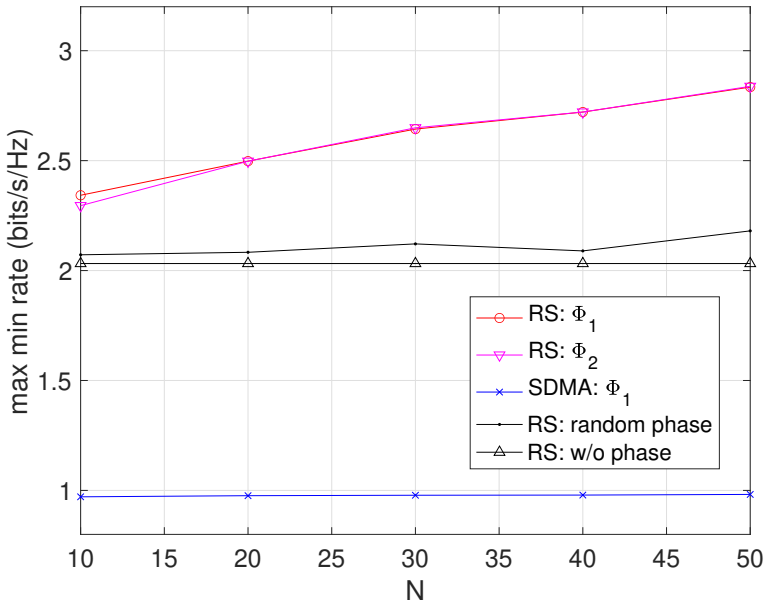


Figure 2.4: Max-min rate versus the number of IRS elements N for $M = 2$, $K = 4$, and $P = 20$dBm.

stays on a horizontal line. The performance of RS with randomly selected IRS phase shift improves slightly with the increases of N. From the perspective of IRS assumptions, it can be seen that the performance gap between the IRS:Φ_1 case and IRS Φ_2 is very small. This indicates that in practice we do not need an accurate IRS amplitude controller. Instead, an IRS where the phase shift can be adjusted works well.

Impact of the Transmit Power

Fig. 2.5 presents the max-min rate versus transmit power. As illustrated, it is observed that the max-min rate of RS increases with the increase of P. However, the performance improvement of the SDMA scheme is very small. This is because the MMF gain for SDMA is 0 under the setting $M < K$. In contrast, the RS scheme has a performance gain of $\frac{1}{1+K-M}\log_2(P)$, which increases with P at a logarithm speed in the high SNR regime. Second, when considering the impact of the IRS, we find that the performance of RS with ideal IRS is quite close to that with non-ideal IRS. Compared with RS without IRS and with random phase, RS:Φ_1 achieves a significant performance gain.

Impact of the Number of BS Antennas

The impact of the number of BS antennas on the max-min rate is shown in Fig. 3.9. The max-min rate for all schemes increases with M due to the MIMO multiplexing gain. Note that in the simulation, we have $K = 4$. As analyzed before, when $M < K$, the RS schemes with Φ_1 or Φ_2 both outperform SDMA. When $M \geq K$, the MIMO system can fully eliminate the interference; hence SDMA and RS achieve a very similar performance. Moreover, compared with RS without IRS and random IRS, RS with ϕ_1 achieves the highest performance gain. Therefore, in a MIMO system where $M = K$, RS does not bring significant gains compared with the conventional SDMA. However, in an overloaded system where $M < K$, RS offers significant benefits. Compared with RS without IRS and random IRS, when M is close to K, the performance gain brought by IRS can be maximized.

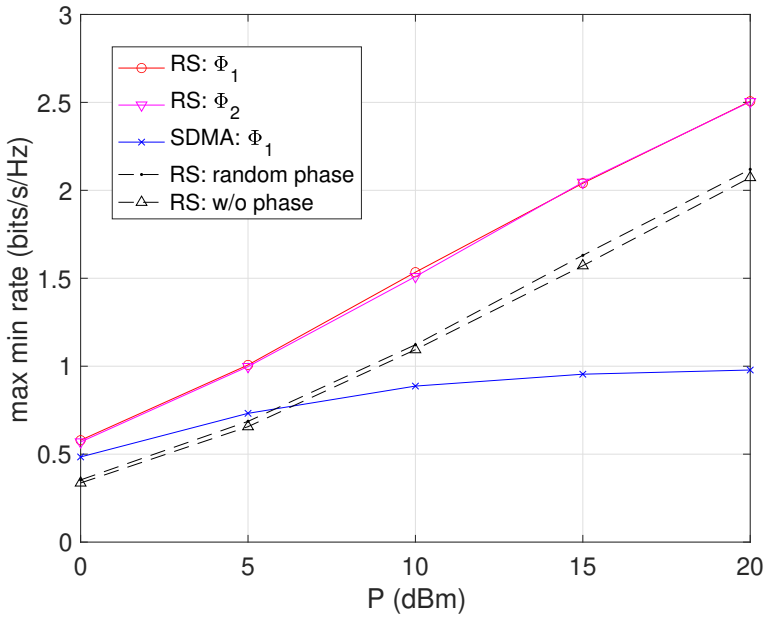


Figure 2.5: Max-min rate versus the transmit power P at the BS for $M = 2$, $K = 4$, and $N = 20$.

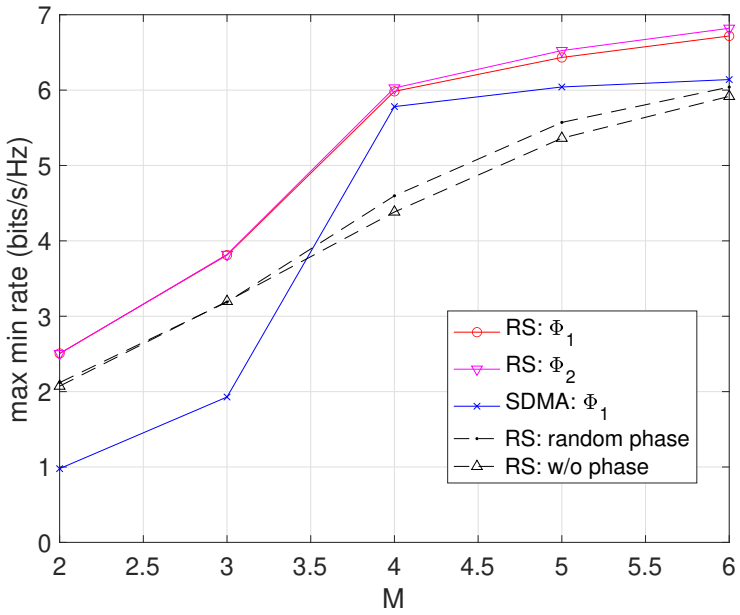


Figure 2.6: Max-min rate versus the number of antennas M at the BS for $K = 4$, $N = 20$, and $P = 20$dBm.

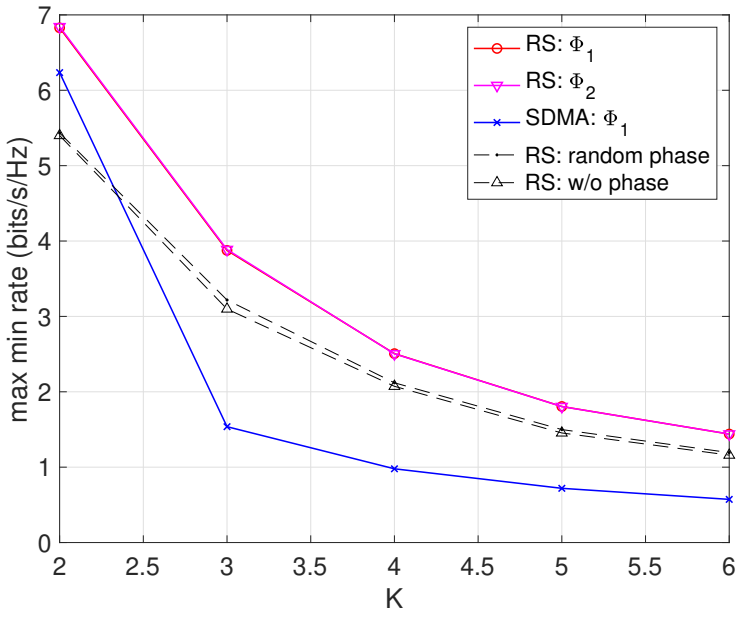


Figure 2.7: Max-min rate versus the number of users K for $M = 2$, $N = 20$, and $P = 20$dBm.

Impact of the Number of UEs

Fig. 3.8 illustrates the impact of the number of UEs on the max-min rate performance. The max-min rate decreases with the increase of the number of UEs due to inter-UE interference. In this experiment, we set $M = 2$. As can be seen, when $K > 2$, the performance of SDMA decreases significantly since $d_{\text{MMF}} = 0$. In contrast, the performance of RS with IRS decreases slowly. Compared with RS without IRS and random IRS, the performance gain is maximized when $M = K = 2$. When K increased to a quite large number, the IRS cannot effectively adjust the channel gain across different UEs. As a result, the performance gap between RS with Φ_1(or Φ_2) and RS without IRS becomes negligible.

Comparison with NOMA

Finally, we compare the performance of the proposed scheme with IRS-NOMA. IRS-NOMA is implemented with the SDR method in [64] by changing the objective function to max-min rate. In this experiment, we assume $M = 2$, $K = 3$, and $P = 20$dBm. Compared with RS, NOMA requires ordering the channels based on channel gains so that the UE with

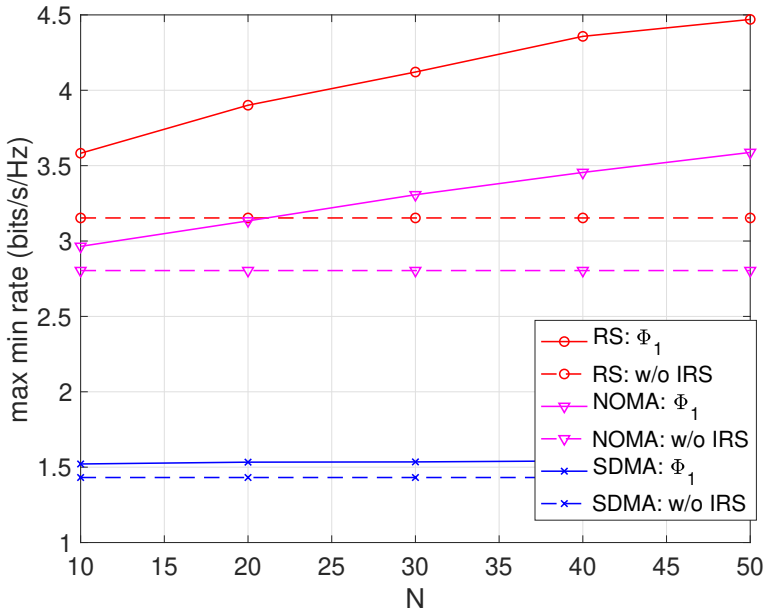


Figure 2.8: Max-min rate versus the number of reflecting elements N for $M = 2$, $K = 3$, and $P = 20$dBm.

a stronger channel gain can decode the UE's signal with a weaker channel gain. However, in IRS-assisted NOMA, the channels can be modified by the IRS. As a result, IRS-assisted NOMA needs to consider $K!$ different decoding orders, solve the beamforming separately for each decoding order, and select the best beamforming scheme. The complexity would be quite high especially when the number of UEs is large. Although some user ordering schemes are designed to reduce the complexity, they inevitably suffer certain performance loss [19]. Unlike NOMA, RS does not require channel ordering, and hence the complexity can be greatly reduced. When $M < K$, the introduction of IRS almost bring no MMF gain. We provide the max-min rate performance comparison in Fig. 2.8. With increased number of reflecting elements on the IRS, the max-min rates of SDMA, NOMA, and RS all increases. Compared with SDMA and NOMA, the RS rate increases at a higher speed and achieves the best performance. To be specific, when the number of reflecting elements on the IRS increases from 10 to 50, the max-min rate improvement for RS, NOMA and SDMA is given by about 0.9, 0.6 and 0.1 bits/s/Hz.

2.6.2 Imperfect CSI case

For each channel generation $\mathbf{h}_k$, we can have a channel estimation as $\hat{\mathbf{h}}_k = \mathbf{h}_k - \Delta\mathbf{h}_k$. Based on the conditional probability $f_{\mathbf{h}_k|\hat{\mathbf{h}}_k}(\mathbf{h}_k|\hat{\mathbf{h}}_k)$, we set $L = 100$ channel realizations. Note that $\mathbf{h}_k$ represents the actual channel experienced by user k but is unknown to the BS while $\hat{\mathbf{h}}_k^l (1 \leq l \leq L)$ can be used to calculate the sample average rate, which can approximate the ergodic max min rate performance.

Impact of the transmit power

We compare the ergodic max-min performance of RS:ϕ_1 with that of SDMA:ϕ_1 by changing the transmit power. In Fig. 2.9, Fig. 2.10 and Fig. **??**, we show their performance by fixing the number of transmit antennas on the IRS to be 20 and changing the relative channel estimation errors.

Fig. 2.9 shows the performance of a system where $M < K$. In this experiment, we have $M = 2$ and $K = 4$. As can be seen, with the increase of the transmit power, the performance of SDMA with IRS slowly increases while the performance of RS with IRS increases significantly, regardless of the value of the relative channel estimation error. The proposed scheme is quite robust towards channel estimation error. For example, when the transmit power is 20dBm and the relative CSI error is 0.1, the max-min rate of RS is around 2.2 bits/s/Hz, which is 88% of the rate achieved at perfect CSI case (2.5 bits/s/Hz). Even when the relative CSI error is quite large ($\delta = 0.5$, the proposed RS scheme still achieves 1.75 bits/s/Hz, which is around 70% of the perfect CSI case.)

Fig. 2.10 presents the performance of a system where $M = K$. In this plot, we set both M and K to be 3. As analyzed before, when $M \geq K$, RS and SDMA achieve the same multiplexing gain. This is confirmed when we observe that the curve for RS and SDMA almost overlapped under the perfect CSI case. However, when the relative CSI error gets larger, the performance of SDMA decreases significantly, the RS scheme is much more robust. To be specific, when the transmit power is 20dBm, the max-min rate of the perfect CSI case

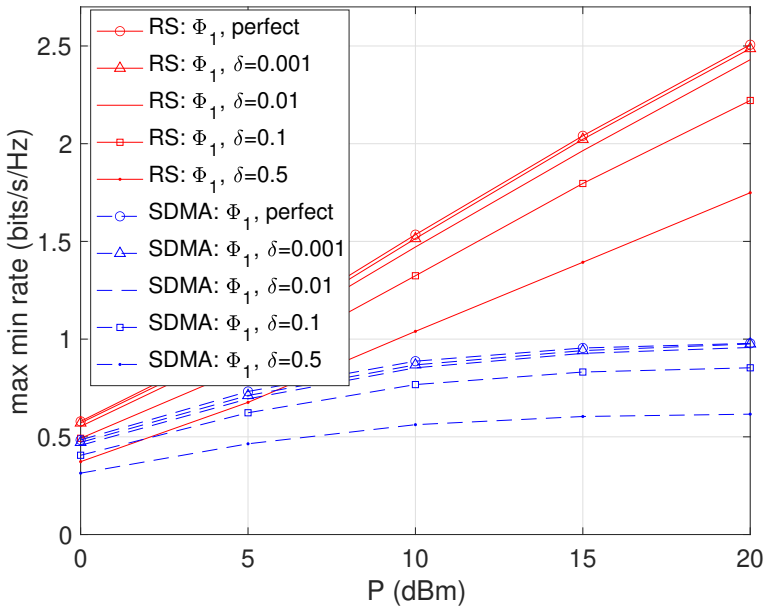


Figure 2.9: Max-min rate versus transmit power for $M = 2$, $K = 4$, and $N = 20$ under imperfect CSI case.

for RS and SDMA is around 6.4 bits/s/Hz and 6.0 bits/s/Hz, respectively. When the CSI error is large, i.e., $\delta = 0.5$, RS still achieves 2.84 bits/s/Hz, which is 44% the performance in the perfect CSI case. SDMA only achieves 1.57 bits/s/Hz, which is only 26% the performance of the perfect CSI case. This experiment suggests that in an IRS assisted communication scenario, RS is more robust to the CSI errors compared with SDMA, even when $M \geq K$.

Fig. **??** shows the performance of a system where $M > K$. SDMA and RS shows very similar performance in the low SNR regime under perfect CSI. However, as the relative CSI error grows, the performance of SDMA decreases significantly. When the transmit power is 20dBm, the max min rate for RS is around 5.0 bits/s/Hz (or 75% of the performance in the perfect CSI case) while the max min rate for SDMA is only 3.27 bits/s/Hz (or 50% of the performance on the perfect CSI case). This is due to the reason that when $M \geq K$, the transmit antennas at the BS and the reflecting elements on the IRS provide more multiplexing gains, which makes RS more robust in combating CSI errors.

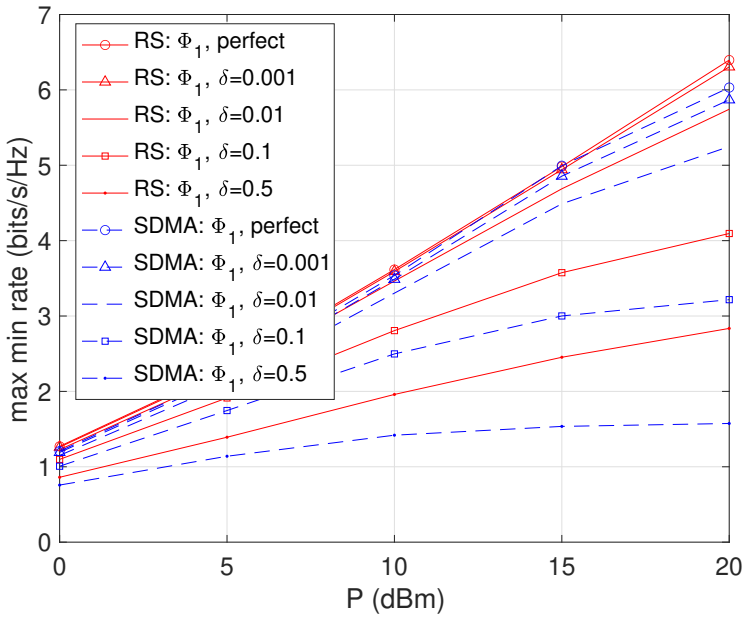


Figure 2.10: Max-min rate versus transmit power for $M = 3$, $K = 3$, and $N = 20$ under imperfect CSI case.

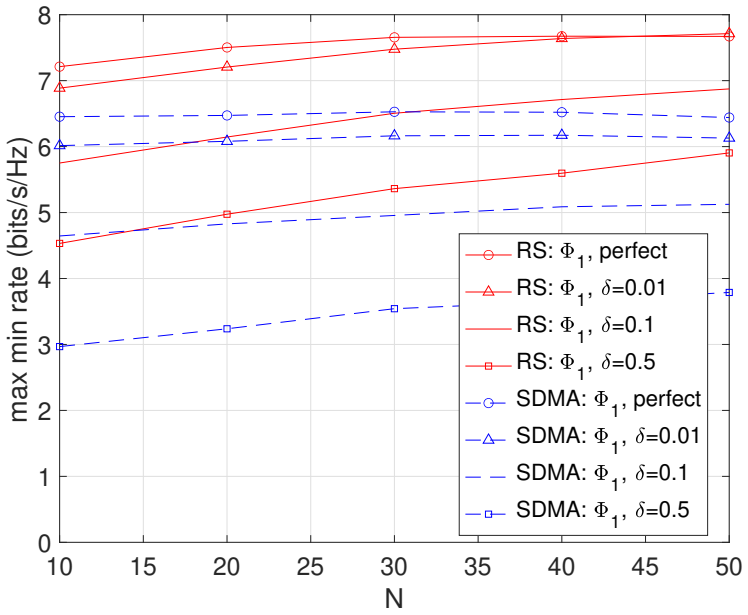


Figure 2.11: Max-min rate versus transmit power for $M = 4$, $K = 2$, and $P = 20$dbm under imperfect CSI case.

Impact of the number of IRS reflecting elements

We plot the max-min performance versus the number of reflecting elements on the IRS in Fig. 2.12. When the relative CSI error is large, i.e., $\delta = 0.5$, the performance of both SDMA and RS improves with the increase of N. To be specific, for RS, the performance gap from the perfect CSI case decreased from $7.2 - 4.5 = 2.7$ bits/s/Hz when $N = 10$ to $7.7 - 5.9 = 1.8$ bits/s/Hz when $N = 50$. For SDMA, the performance gap from the perfect CSI case decreased from $6.5 - 3 = 3.5$ bits/s/Hz when $N = 10$ to $6.4 - 3.8 = 2.6$ bits/s/Hz when $N = 50$. When the relative CSI error is small, i.e., $\delta = 0.01$, with the increase of N, for RS the performance gap from the perfect CSI case almost vanished. However, for SDMA, the performance gap always exists. This experiment suggest that it is more efficient to increase N for RS compared with SDMA.

The impact of IRS configuration

In this experiment, we investigate the impact of IRS configuration. We compared the performance of RS:ϕ_1 with two benchmark algorithms: RS with random phase and RS without IRS. When the transmit power is high, i.e. $P = 20$dBm, as shown in Fig. 2.13, the max-min rate decreases with the increase of the relative CSI error. Increasing the number of reflecting elements on the IRS can improve the performance of RS:ϕ_1. However, the performance of RS with random phase does not improve significantly when N increases. In the perfect CSI case, i.e., $\delta = 0$, RS:ϕ_1 (N=50) achieves considerable performance gain (7.2-4.7=2.5 bits/s/Hz) compared with RS without IRS. Even when the CSI error is $\delta = 0.01$, the performance gain is still 6.6-4.1=2.5 bits/s/Hz. When CSI error is large, i.e., $\delta = 0.5$, the performance gain becomes 3.6-2.3=1.3 bits/s/Hz. This shows that RS with properly configured IRS is quite robust toward a relatively small CSI error.

When the transmit power is low, i.e., $P = 0$dBM, the max-min rate performance of RS with different IRS configuration is shown in Fig. 2.14. RS with random phase achieves a small amount of performance gain compared with RS without IRS, no matter the relative

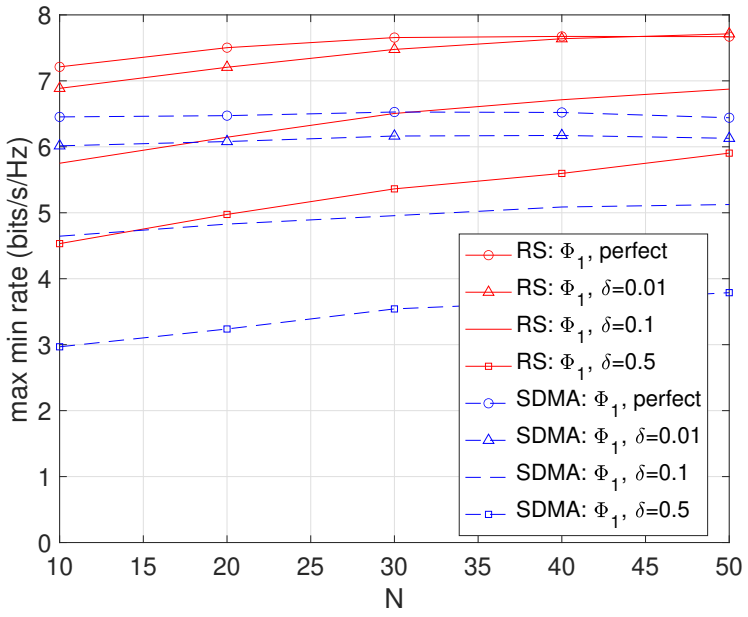


Figure 2.12: Max-min rate versus transmit power for $M = 4$, $K = 2$, and $P = 20$dbm under imperfect CSI case.

CSI error is small or large. RS:ϕ_1 achieves a max min rate at around 2 bits/s/Hz when $N = 50$ and $\delta = 0.01$, which is almost three times of the max min rate achieved by RS without IRS. This further demonstrate the benefits of integrating IRS and RS. Moreover, comparing Fig. 2.13 and Fig. 2.14, we notice that the benefit of increasing the reflecting elements on the IRS is more significant at low SNR regime or when the transmit power is small. To be specific, when $P = 0$dBm and $\delta = 0.01$, the performance gain of increasing N from 20 to 50 is (2.0-1.2)/1.2=66.7% for RS:ϕ_1 while this number is (6.6-5.7)/5.7 =15.8% when $P = 20$dBm and $\delta = 0.01$.

2.7 Conclusions

This chapter has investigated the resource allocation problem in an IRS-assisted RS system. A BCD algorithm is developed to maximize the minimum rate of all users under both perfect CSI and imperfect CSI setting. Our results show that RS has the advantage of being robust with imperfect CSI, which makes it a good fit for IRS assisted communication. With

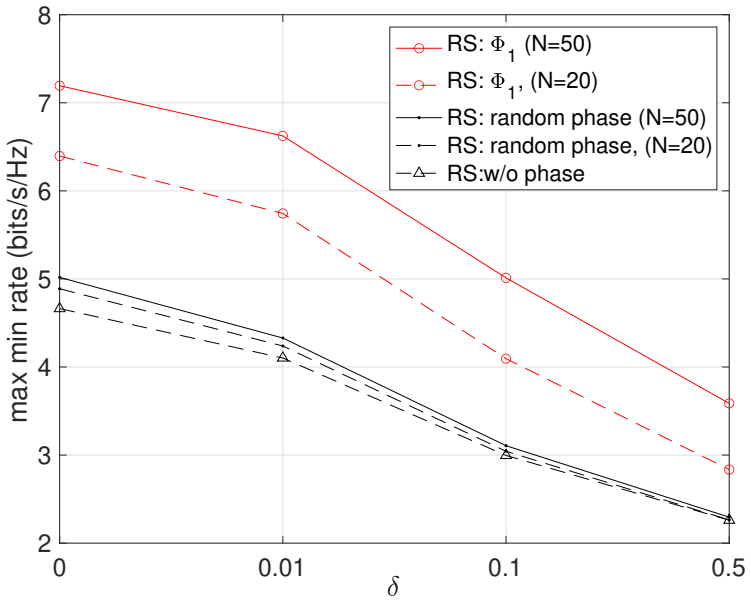


Figure 2.13: Max-min rate versus relative CSI error for $M = 3$, $K = 3$, and $P = 20$dbm.

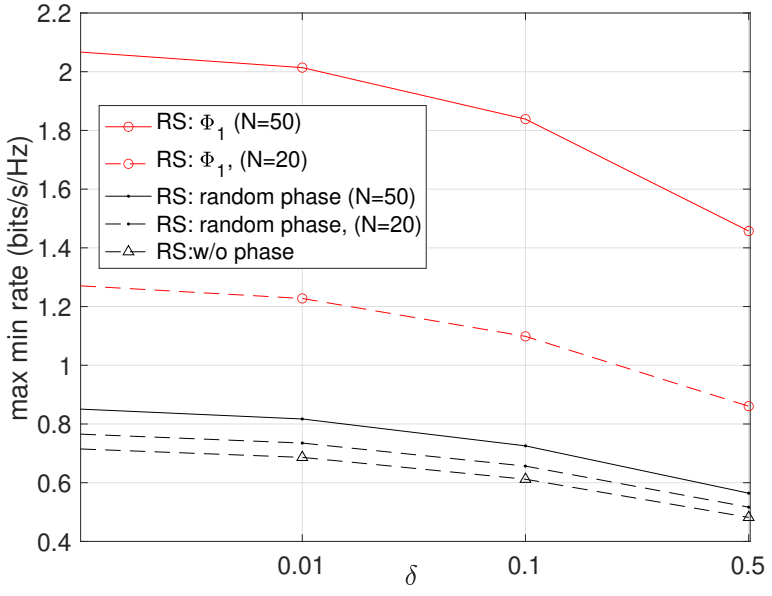


Figure 2.14: Max-min rate versus relative CSI error for $M = 3$, $K = 3$, and $P = 0$dbm.

the proposed method, IRS-assisted RS could achieve a satisfactory performance compared with conventional multiple access technologies, such as SDMA and NOMA with/without IRS.

Chapter 3

Energy-efficient Resource Allocation in IRS-assisted Federated Learning

3.1 Introduction

The emerging intelligent applications such as face recognition, autonomous driving, unmanned aerial vehicle (UAV), and indoor localization have imposed great challenges for Internet of Things (IoT) devices due to the computation-intensive and latency-sensitive features. The devices are generating a vast amount of data via their local sensors, e.g., GPS, accelerometer, and camera. It is envisioned that future networks should be able to utilize the local data at the mobile edge to perform intelligent inference and machine learning tasks. However, the paradigm changes from "connected things" to "connected intelligence" in the era of 6G brought about two main challenges [5]. First, the bandwidth is limited, aggregating the large volumes of data would cause network congestion. Second, data-privacy is becoming a critical issue in today's IoT and the Internet. As a result, it becomes more and more desirable to perform learning tasks at the end-IoT devices instead of sending raw data to the central cloud.

A new machine learning method, termed federated learning, has emerged as a promising solution for privacy-sensitive and low-latency solutions [73–75]. In federated learning, user data is stored locally. In each communication round, users perform local training based on their local data and then upload their trained model to the central server. After aggregating the local updates from all users, the central server distributes the new global model to the users. This process proceeds in an iterative way until convergence is reached. In this way, a global model, which is trained from the data stored on each device, can be obtained without data leakage or data being inferred from other users. This property makes federated learning one of the most promising technologies of future intelligent networks.

Nevertheless, so far, the potential of federated learning has not been fully exploited yet due to the stochastic nature of wireless channels. For example, cell edge users often suffer from communication links of poor quality or unfavorable wireless propagation conditions. Fortunately, the recent advances in reconfigurable wireless technology provide a new cost-effective means to enhance the performance of intelligent learning systems [17, 76]. To be specific, the intelligent reflecting surface (IRS) is composed of a large number of reflecting elements, whose amplitude and phase can be adjusted to create a favorable propagation environment [13–15]. The direct channel gain in combination with the reflection-aided beamforming gain can boost the local model uploading performance.

In this chapter, we investigate resource allocation for energy efficient communication in federated learning with IRS. There are several challenges. First, the IoT devices for federated learning are powered by batteries, which need to support both local training and model upload. How to save the battery power of each device becomes a critical issue. Second, the global model training accuracy depends on the number of training iterations. The wireless resources need to be properly allocated to meet the training accuracy requirement while also conserve energy. Third, with the involvement of IRS, the parameters become highly coupled. A joint design of the IRS parameters as well as the computing/communication parameters is of critical importance. The main contributions of this chapter include:

1. We investigate an IRS-assisted federated learning system, where the IRS reconfigures the communication channel so that the IoT devices can upload their model with a reduced power. As a result, the total energy consumption can be effectively reduced.

2. We formulate a joint local training and model uploading problem, which aims to minimize the energy consumption subject to the task completion time requirement. A low complexity iterative algorithm with proven fast convergence is proposed to optimize each variable iteratively. Most of the variables can be obtained numerically with the simple one-dimensional search algorithm, which makes it useful in practical systems.

We show that the main complexity of the algorithm comes from the optimization of IRS elements, which involves solving an SDP problem.

3. The convergence of the proposed algorithm is proved theoretically and verified numerically. Extensive simulations are performed to demonstrate the benefits brought by using IRS. Our results suggest that with the use of IRS, the energy consumption in federated learning of a battery powered IoT device network can be greatly reduced, especially, when the number of reflecting elements is large and the IRS is properly configured.

The remainder of this chapter is organized as follows. Section 3.2 introduces the relevant work and Section 3.3 presents the system model and problem statement. We start the design of the algorithm from the simplest case where there is only one device in Section 3.4. Then the algorithm is extended to a multi-device federated learning scenario in Section 3.5 and a low complex algorithm is proposed in Section 3.6. Numerical results are discussed in Section 3.7. Finally, Section 3.8 concludes this paper.

3.2 Related Works

Federated learning, first proposed in [73], is a distributed learning method that enables IoT devices to train a global model without sharing their own data with other users. Due to its advantages in protecting privacy, it has been successfully adopted in a wide range of application scenarios, such as semantic location, health prediction, or learning sentiment [75].

There are a few works focused on federated learning over wireless links. A communication and computation co-design approach for fast model aggregation is proposed in [77], which leverages the property of signal superimposition on wireless multiple access channels. This *over-the-air computation* (AirComp) framework is achieved by jointly considering the beamforming design and the device selection problem. A collaborative learning that considers limited wireless resources is first investigated in [78]. The impact of MAC layer bandwidth

and power limit on the performance of federated learning is investigated under the framework of AirComp. A general model that investigates the computation and communication latency trade-off in federated learning is proposed in [79]. The authors show that federated learning over wireless networks captures a trade-off between communication and computation. The previous research all focused on stochastic wireless channels. The benefits of configurable technology such as IRS on the performance of federated learning has not been fully investigated. Recent results in mobile edge computing show that the overall uplink transmission latency can be reduced [80] and the system throughput can be improved [23] with the IRS technology.

There are several works that investigate federated learning with IRS. In [77], the authors show that when federated learning meets IRS, the model aggregation error can be reduced via the enhanced signal provided by the IRS. AirComp and IRS have the potential to tackle the challenge of the communication bottleneck problem. The authors in [81] investigate the model aggregation performance in a federated learning system with IRS. A joint model device selection, beamforming, and IRS phase shift optimization algorithm is proposed. The proposed algorithm can schedule more devices in each communication round under certain accuracy requirement.

3.3 System Model and Problem Formulation

As shown in Fig. 3.1, we consider a single-cell federated learning communication system, where K single antenna IoT devices offload their locally trained models to an edge server hosted at a BS with M antennas through radio access links. The federated learning model is the same as that in [48, 82], where a global ML problem is solved at a central server with the training dataset partitioned over IoT devices.

We assume that each device k has a local training dataset with D_k data samples. The federated learning model is locally trained by each device's own dataset. Then the local model parameter is uploaded to the BS. After aggregation, the BS then broadcasts the global model

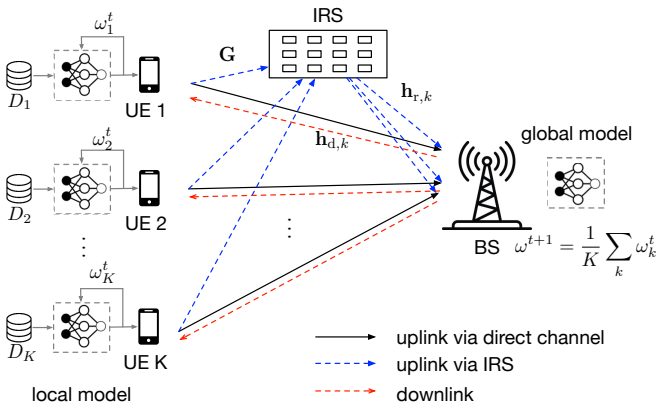


Figure 3.1: Illustration of the federated learning system with IRS.

to each participating device. This is called one round of training. Such communication round will be performed several times until the model achieves a required level of accuracy. *We aim to determine the resource allocation strategy to achieve an energy efficient design.*

3.3.1 Wireless Communication Model

We consider uplink frequency-division multiple access (FDMA) transmissions where the BS serves the users with orthogonal frequency bands. To assist the model uploading of mobile devices, an IRS with N reflecting elements is placed between the IoT devices and the BS. The equivalent channels from device k to the BS, from device k to the IRS, and from the IRS to the BS are denoted as $\mathbf{h}_{\mathrm{d},k} \in \mathbb{C}^{M\times 1}$, $\mathbf{h}_{\mathrm{r},k} \in \mathbb{C}^{N\times 1}$ and $\mathbf{G} \in \mathbb{C}^{M\times N}$, respectively. The IRS has a reflection phase-shift matrix $\mathbf{\Theta} \in \mathbb{C}^{N\times N}$, which is a diagonal matrix with $e^{j\theta_n}$ being its diagonal elements, $\theta_n \in [0, 2\pi]$, for all $1 \leq n \leq N$. $\mathbf{\Theta}$ captures the effective phase shifts of all the reflecting elements of the IRS. The phase shift unit can be adjusted by the IRS controller based on measured channel dynamics. The composite channel is therefore modeled as a combination of the direct channel and the reflected channel. The training update transmission between the IoT device and the cloud server happens in orthogonal frequency bands. Hence there is no interference between users. Then the uplink transmission

rate of the kth IoT device is given by

$$R_k = b_k \log_2 \left(1 + \frac{p_k |\mathbf{w}_k^H \mathbf{h}_k|^2}{N_0 |\mathbf{w}_k^H|^2 b_k}\right), \tag{3.1}$$

where b_k is the bandwidth allocated to device k, $\mathbf{n}$ is the additive white Gaussian noise (AWGN) with zero mean and noise power spectrum density N_0, $\mathbf{w}_k \in \mathbb{C}^{M\times 1}$ is the beamforming vector for device k, $\mathbf{h}_k \triangleq \mathbf{h}_{\mathrm{d},k} + \mathbf{G}\boldsymbol{\Theta}\mathbf{h}_{\mathrm{r},k} \in \mathbb{C}^{M\times 1}$ is the combined channel between device k and the BS.

3.3.2 Federated Learning Model

A federated learning process consists of three stages: local training, model aggregation, and model distribution. The entire training process differs from the conventional mobile edge computing system in three aspects. First, in mobile edge computing systems, a device can offload part of its work to the cloud while computing its own tasks asynchronously. However, for federated learning, each device must finish its local model training first, and then performs model uploading. Second, in federated learning, the cloud cannot aggregate the global model until each device offloads its local model to the cloud. This requires stringent synchronous processing and poses the latency requirement. This training process usually lasts for several rounds. Third, in federated learning, the uploaded model sizes should be the same for all the IoT devices, while the uploaded data sizes are usually different across different devices in general mobile edge computing. The models for the three stages of federate learning are provided in the following.

1) Local Training: When an application is executed on the IoT device, the energy consumption depends on the CPU workload of the device, which is characterized by the number of CPU cycles to complete this application. Assume c_k is the the number of CPU cycles required to process one bit and f_k is the number of CPU cycles per second for device k. Then the time required for carrying out the local model training can be expressed as $D_k c_k / f_k$

in each local training round. We assume that each device uses the stochastic average gradient (SAG) algorithm to train the local model to achieve a local level relative accuracy $\eta \in [0, 1]$. The number of local iterations is then given by [83]

$$L(\eta) = \ell_1 \ln(1/\eta), \tag{3.2}$$

where $\ell_1 > 0$ is a parameter depends on the data size and structure of the local problem. In [84], it is shown that the local level accuracy $\eta = 0$ describes an exact solution of the subproblem and $\eta = 1$ means that the local training has not been improved at all. In this case, the local training latency will be

$$t_k^L = L(\eta) D_k c_k / f_k. \tag{3.3}$$

Assume the IoT device uses a dynamic voltage scaling (DVS) scheme, so it can adjust its computational speed to save energy [85]. According to [85], the energy consumption per CPU cycle can be expressed as κf_k^2, where κ is a coefficient depending on the chip architecture. Then the energy consumption for local training can be expressed as

$$E_k^L = \kappa D_k c_k f_k^2 L(\eta). \tag{3.4}$$

2) Model Aggregation: After local model training, each IoT device then sends its local updates to the BS. Suppose S is the size of the offloading training model of a fixed dimension, which should be the same for all the IoT devices. The upload latency can be expressed as

$$t_k^U = S/R_k, \tag{3.5}$$

where R_k is given in (3.1). The energy consumption of model uploading for device k is expressed as

$$E_k^U = t_k^U(p_{c,k} + p_k), \tag{3.6}$$

where $p_{c,k}$ is a constant circuit power of the IoT device during the computational uploading process.

3) Model Distribution: The parameters related to the global model are updated via a simple linear processing at the cloud server hosted at the BS. The BS has a strong processing capability, and hence the processing time can be negligible. After the global model parameters are updated at the BS, the BS distributes the global model parameters to all the IoT devices. The broadcast time can also be negligible since the BS has high transmit power and large bandwidth.

To achieve a global accuracy ϵ, the number of global iterations is given by [84]

$$G(\eta) = \frac{\mathcal{O}(\ln(\frac{1}{\epsilon}))}{1-\eta}. \tag{3.7}$$

In this work, we consider a fixed, target global accuracy ϵ, so we can normalize $\mathcal{O}(\ln(\frac{1}{\epsilon}))$ to 1 without changing the nature of this problem.

To this end, the overall latency of IoT device k is composed of the local computation time and model uploading latency as

$$T_k = \frac{1}{1-\eta}(t_k^L + t_k^U). \tag{3.8}$$

Let T be the maximum training time for the entire federated learning algorithm. Then we have

$$T_k \leq T, \ \forall k. \tag{3.9}$$

The overall energy consumption for IoT device k over the entire federated learning process is

$$E_k = \frac{1}{1-\eta}(E_k^L + E_k^U). \tag{3.10}$$

3.3.3 Problem Formulation

To allow the IoT devices to save energy while also guaranteeing the training time/accuracy requirements of federated learning, we need to develop effective resource allocation algorithms. The energy minimization problem is thus formulated as follows.

$$\begin{aligned} (P1) \quad \min_{\eta, f_k, b_k, p_k, \mathbf{w}_k, \boldsymbol{\Theta}} \quad & \sum_k E_k \\ \text{s.t.} \quad & C1: T_k \leq T, \ \forall k \\ & C2: \eta \geq 0 \\ & C3: 0 \leq \theta_n \leq 2\pi, \ \forall n \\ & C4: 0 \leq p_k \leq P_{\max}, \ \forall k \\ & C5: \sum_k b_k \leq B, \end{aligned} \tag{3.11}$$

where constraint (C1) is the task completion time constraint; (C2), (C3), and (C4) specify the domain of η, θ_n, and p_k, respectively; constraint (C5) indicates that the combined occupied bandwidth should not exceed the total available bandwidth. This is a joint power, bandwidth, phase shift, accuracy control, and beamforming design problem. Problem (P1) has a non-convex and mixed structure where some variables are coupled. Obtaining a global optimal solution will be quite challenging.

3.4 Analysis of the Single Device System

First, we consider the simplest case where there is only one IoT device. Although such assumption is not practical in terms of federated learning, the results can still provide useful

insights on parameter optimization for a practical multiuser federated learning system. In the rest of this section, we set $k = 1$. The total energy consumption for device k is

$$E_k = \frac{1}{1-\eta}\left(\frac{S}{R_k}(p_{c,k} + p_k) + \kappa D_k c_k f_k^2 L(\eta)\right). \tag{3.12}$$

3.4.1 Design of the Device CPU Frequency

Theorem 4. *The optimal operating frequency for device k is given by*

$$f_k^* = \frac{L(\eta) D_k c_k}{T/G(\eta) - S/R_k}. \tag{3.13}$$

Proof. The proof is shown in appendix B.1. □

3.4.2 Design of Power Allocation

Next, we substitute the optimal solution f_k^* (3.13) into the original Problem (P1). We jointly optimize the power allocation when the local accuracy parameter η, the bandwidth b_k, the IRS parameters $\mathbf{\Theta}$ and $\mathbf{w}_k$ are known. The objective function becomes

$$E_k = G(\eta)\left(\frac{S}{R_k}(p_{c,k} + p_k)\kappa D_k c_k L(\eta)\left(\frac{L(\eta) D_k c_k}{T/G(\eta) - S/R_k}\right)^2\right),$$

where R_k is a function of p_k, b_k, and $\mathbf{\Theta}$. A direct optimization is quite hard. To solve this problem, we optimize each variable in an iterative manner. Specifically, we write

$$f_k^2 = \frac{f_k^{t,3}}{f_k^*} = \frac{f_k^{t,3}}{L(\eta) D_k c_k}\left(\frac{T}{G(\eta)} - \frac{S}{R_k}\right), \tag{3.14}$$

where f_k^t is the result in the tth iteration. The objective function (3.12) then assumes a simpler form as $E_k = G(\eta)\left(\frac{S}{R_k}(p_{c,k} + p_k) + \kappa f_k^{t,3}\left(\frac{T}{G(\eta)} - \frac{S}{R_k}\right)\right)$. When f_k is fixed, the

problem becomes

$$(P2a) \quad \min_{p_k} \quad \frac{p_k + p_{c,k} - A_k}{R_k} \tag{3.15}$$
$$\text{s.t.} \quad (C4),$$

where $A_k = \kappa f_k^{t,3}$ is a constant in each iteration step.

Theorem 5. *The optimal solution to (P2a) when $p_{c,k} - A_k > 0$ is given by*

$$p_k^* = \min\{p_k', P_{\max}\}, \tag{3.16}$$

where p_k' is the solution to $h(p_k) = \frac{a_k}{b_k + a_k p_k}(p_{c,k} + p_k - A_k) - \ln(1 + a_k p_k / b_k) = 0$.

Proof. The proof is shown in appendix B.2. □

If $p_{c,k} - A_k \leq 0$, then the objective function in the subproblem is negative when $p_k + p_{c,k} - A_k < 0$ and positive when $p_k + p_{c,k} - A_k > 0$. By investigating the monotonicity of the objective function, we find that $\frac{R_k}{A_k - p_{c,k} - p_k}$ is strictly increasing on the interval $[p_{k,\min}, P_{\max}]$. Hence the objective function is minimized when $p_k = p_{k,\min}$. In this paper, we only consider the case where $p_{c,k} - A_k > 0$ for simplicity. The case $p_{c,k} - A_k < 0$ can be similarly analyzed.

3.4.3 Design of Bandwidth Allocation and IRS Parameters

When the power and frequency parameters are fixed, we can see that minimizing energy consumption is equivalent to maximizing the achievable rate R_k. The subproblem becomes

$$(P2b) \quad \max_{b_k, \mathbf{w}_k, \boldsymbol{\Theta}} \quad R_k \tag{3.17}$$
$$\text{s.t.} \quad (C3), (C5).$$

First of all, we can prove that R_k is a concave function w.r.t. b_k on the interval $[0, B]$. The optimal bandwidth allocation b_k for the singe device case can also be obtained with a

bisection method by setting the first derivative of R_k to zero. To save space, we leave out this part of content.

Now, we optimize the IRS related parameters $\mathbf{\Theta}$ and $\mathbf{w}_k$. Maximizing the achievable rate R_k is equivalent to maximizing the corresponding SNR a_k. The problem becomes

$$\max_{\theta_n, \mathbf{w}_k} \quad \frac{p_k |\mathbf{w}_k^H \mathbf{h}_k|^2}{N_0 |\mathbf{w}_k^H|^2 b_k} \tag{3.18a}$$

$$\text{s.t.} \quad 0 \leq \theta_n \leq 2\pi, \tag{3.18b}$$

where $\mathbf{h}_k \triangleq \mathbf{h}_{\mathrm{d},k} + \mathbf{G}\mathbf{\Theta}\mathbf{h}_{\mathrm{r},k}$.

This problem can be solved by alternative optimization. Specifically, we first fix the IRS phase shift matrix $\mathbf{\Theta}$ and find the optimal detection vector $\mathbf{w}_k$. Without changing the nature of the problem, one can set $|\mathbf{w}_k^H|^2 = 1$ for simplicity. This problem becomes the well-known maximum ratio combing (MRC) detection problem. The SNR is maximized at

$$\mathbf{w}_k^* = \frac{\mathbf{h}_k}{|\mathbf{h}_k|}. \tag{3.19}$$

Next, for a fixed $\mathbf{w}_k^H$, we optimize the IRS phase vector. This problem is equivalent to the following problem.

$$\max \quad |\mathbf{w}_k^H (\mathbf{h}_{\mathrm{d},k} + \mathbf{G}\mathbf{\Theta}\mathbf{h}_{\mathrm{r},k})| \tag{3.20a}$$

$$\text{s.t.} \quad 0 \leq \theta_n \leq 2\pi, \tag{3.20b}$$

We follow a similar procedure as in [17] by rewriting

$$|\mathbf{w}_k^H (\mathbf{h}_{\mathrm{d},k} + \mathbf{G}\mathbf{\Theta}\mathbf{h}_{\mathrm{r},k})| \leq |\mathbf{w}_k^H \mathbf{h}_{\mathrm{d},k}| + |\mathbf{w}_k^H \mathbf{G}\mathbf{\Theta}\mathbf{h}_{\mathrm{r},k}| \leq |\mathbf{w}_k^H \mathbf{h}_{\mathrm{d},k}| + |\mathbf{w}_k^H \mathbf{G}\mathrm{diag}(\mathbf{h}_{\mathrm{r},k})| \tag{3.21}$$

where the first inequality is due to the triangle inequality and the equality holds if and only if $\arg(\mathbf{w}_k^H \mathbf{h}_{\mathrm{d},k}) = \arg(\mathbf{w}_k^H \mathbf{G}\mathbf{\Theta}\mathbf{h}_{\mathrm{r},k}) \triangleq \phi_0$. Note that $\mathrm{diag}(\mathbf{\Theta})$ is a diagonal matrix and we extract

its diagonal as a vector $\mathbf{v} = \text{diag}(\mathbf{\Theta})$, then $\mathbf{w}_k^H \mathbf{G}\mathbf{\Theta}\mathbf{h}_{\text{r},k} = \mathbf{w}_k^H \mathbf{G}\text{diag}(\mathbf{h}_{\text{r},k})\mathbf{v}$. Considering the constraint that $|v_n| = |e^{j\theta_n}| = 1$, the optimal solution to this problem is given by

$$\mathbf{v}^* = \exp\left\{j(\phi_0 - \arg(\mathbf{w}_k^H \mathbf{G}\text{diag}(\mathbf{h}_{\text{r},k})))\right\}, \tag{3.22}$$

where $\phi_0 = \arg(\mathbf{w}_k^H \mathbf{h}_{\text{d},k})$.

Remark 1. *IRS can strengthen the received signal power by aligning the cascaded channel with the direct channel compared with that without IRS.*

This alternating optimization method is appealing since it has a closed-form expression for both the IRS phase shift vector and the signal detection vector. Its convergence is guaranteed since each subproblem ensures that the objective function is non-decreasing over iterations and is bounded above as the second inequality in (3.21) suggests.

3.4.4 Design of the Accuracy Parameter

Finally, we optimize the accuracy parameter η. For simplicity of notation, the objective function can be rewritten as

$$f(\eta) = \frac{1}{1-\eta}\left(u + v\log\left(\frac{1}{\eta}\right)\right), \ \eta \in (0,1), \tag{3.23}$$

where $u = (p_{c,k} + p_k)S/R_k$ and $v = \kappa D_k c_k f_k^2 \ell_1$ are both positive numbers.

Theorem 6. *The optimal accuracy parameter η^* is the solution to $h(\eta) = 0$, where*

$$h(\eta) = -v(1-\eta) + u\eta - v\eta\ln(\eta). \tag{3.24}$$

Proof. The proof is shown in appendix B.3. □

The algorithm for single user training is presented in Algorithm 2. The mainly complexity comes from the alternative updates of $\mathbf{\Theta}$ and $\mathbf{w}_k$, whose complexity are on the order

Algorithm 2 Energy-efficient Optimization for Single Device

1: Initialize IRS phase shift matrix $\mathbf{\Theta}$ and the iteration number $t = 1$, $s = 1$;
2: **repeat**
3: Obtain $\mathbf{w}_k^s$ according to (3.19);
4: Obtain $\mathbf{v}^s$ according to (3.22);
5: $s = s + 1$;
6: **until** convergence
7: **repeat**
8: Obtain f_k^t according to (3.13);
9: Obtain p_k^t according to (3.16);
10: Obtain b_k^t ;
11: Obtain η^t according Theorem 6;
12: Calculate E_k based on (3.12);
13: $t = t + 1$;
14: **until** $\frac{|E_k^{t+1} - E_k^t|}{|E_k^t|} \leq \epsilon_1$ and (C1) is satisfied

of $\mathcal{O}(MN, N^2)$ and of $\mathcal{O}(MN^2)$, respectively. We can therefore claim that the overall complexity is $\mathcal{O}(I_1 MN^2)$, where I_1 is the iteration involved in Lines 2-6 in Algorithm 2.

3.5 Analysis of the Multiuser Federated Learning System

In this section, we consider the more practical multiuser federated learning system. The objective function becomes

$$E = \sum_k E_k = \sum_k G(\eta) \left(\frac{S}{R_k} (p_{c,k} + p_k) + \kappa D_k c_k f_k^2 L(\eta) \right). \tag{3.25}$$

3.5.1 Design of the Device CPU Frequency

First, we optimize the frequency when the training accuracy η is known. Minimizing the sum energy consumption of each device is equivalent to minimizing the individual energy consumption of each device. Again, for each device, E_k is an increasing function in terms of f_k. As a result, the frequency should be set as (3.13) to satisfy the latency constraint of each device.

3.5.2 Design of Power Allocation

From the objective function (3.25), we find that minimizing the energy consumption for all users is equivalent to

$$(P3a) \quad \min_{p_k} \quad \sum_k \frac{p_k + p_{c,k} - A_k}{R_k} \tag{3.26}$$
$$\text{s.t.} \quad (C3),$$

where $R_k = b_k \log_2 \left(1 + \frac{p_k |\mathbf{w}_k^H \mathbf{h}_k|^2}{N_0 |\mathbf{w}_k^H|^2 b_k}\right)$. Similarly, minimizing the sum of energy consumption is equivalent to minimizing the energy consumption of each device. Hence the optimal power allocation can be similarly obtained as (3.16), which is a one-dimensional search problem for each user.

3.5.3 Joint Design of Bandwidth Allocation and IRS Parameters

When the power and frequency are fixed in the last iteration, it is easy to verify that the optimal detection vector should be the same as the single device case as in (3.19), which maximizes the SNR for each device. Hence, the problem becomes

$$(P3b) \quad \min_{\boldsymbol{\Theta}, b_k} \quad \sum_k \frac{p_{c,k} + p_k - A_k}{b_k \log_2(1 + p_k |\mathbf{h}_k^H \mathbf{h}_k| / (N_0 b_k))} \tag{3.27}$$
$$\text{s.t.} \quad (C3), (C5),$$

where $\mathbf{h}_k = \mathbf{h}_{\mathrm{d},k} + \mathbf{G}\boldsymbol{\Theta}\mathbf{h}_{\mathrm{r},k}$.

The problem is difficult since the variables b_k and $\boldsymbol{\Theta}$ are coupled in the numerator and the problem is non-convex. Moreover, the objective function in (P3b) is still not straightforward with the phase shift vector $\boldsymbol{\Theta}$. Now we extract the diagonal elements of $\boldsymbol{\Theta}$ to have $\bar{\mathbf{v}} =$

$\text{diag}\{\boldsymbol{\Theta}\} \in \mathbb{C}^{N\times 1}$. Supposing $\mathbf{H}_k = \mathbf{G}\text{diag}\{\mathbf{h}_{\text{r},k}\} \in \mathbb{C}^{M\times N}$, we have

$$\mathbf{h}_k = \mathbf{h}_{\text{d},k} + \mathbf{H}_k\bar{\mathbf{v}} \tag{3.28}$$

$$|\mathbf{h}_k^H\mathbf{h}_k| = \mathbf{h}_{\text{d},k}^H\mathbf{h}_{\text{d},k} + \bar{\mathbf{v}}^H\mathbf{H}_k^H\mathbf{H}_k\bar{\mathbf{v}} + \mathbf{h}_{d,k}^H\mathbf{H}_k\bar{\mathbf{v}} + \bar{\mathbf{v}}^H\mathbf{H}_k^H\mathbf{h}_{d,k}. \tag{3.29}$$

By introducing an auxiliary matrix $\mathbf{R}_k \in \mathbb{C}^{(N+1)\times(N+1)}$ and an auxiliary vector $\mathbf{v} \in \mathbb{C}^{(N+1)\times 1}$, we further obtain

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{H}_k^H\mathbf{H}_k & \mathbf{H}_k^H\mathbf{h}_{d,k} \\ \mathbf{h}_{d,k}^H\mathbf{H}_k & 0 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \bar{\mathbf{v}} \\ 1 \end{bmatrix}. \tag{3.30}$$

Eqn. (3.29) can be further simplified as

$$|\mathbf{h}_k^H\mathbf{h}_k| = \mathbf{v}^H\mathbf{R}_k\mathbf{v} + \mathbf{h}_{\text{d},k}^H\mathbf{h}_{\text{d},k} = \text{Tr}(\mathbf{R}_k\mathbf{V}) + \mathbf{h}_{\text{d},k}^H\mathbf{h}_{\text{d},k} \triangleq f_k(\mathbf{V}), \forall k \in \mathcal{K}, \tag{3.31}$$

where $\mathbf{V} = \mathbf{v}\mathbf{v}^H \in \mathbb{C}^{(N+1)\times(N+1)}$. Then Problem (P3b) is equivalently transformed to (P3c), given by

$$\begin{aligned}
(P3c) \quad \min_{\mathbf{V},b_k} \quad & \sum_k \frac{p_{c,k} + p_k - A_k}{b_k \log_2(1 + p_k f_k(\mathbf{V})/(N_0 b_k))} && \text{(3.32a)} \\
\text{s.t.} \quad & V_{n,n} = 1 && \text{(3.32b)} \\
& \text{rank}(\mathbf{V}) = 1 && \text{(3.32c)} \\
& \mathbf{V} \succeq \mathbf{0} && \text{(3.32d)} \\
& (C5).
\end{aligned}$$

Note that constraints (3.32b) and (3.32d) ensure that $\mathbf{V} = \mathbf{v}\mathbf{v}^H$ holds true after optimization. Constraint (3.32c) is introduced to guarantee the unit modulus constraint when recovering $\mathbf{v}$ from $\mathbf{V}$. Due to the rank one constraint, this problem is non-convex in terms of the optimization variable $\mathbf{V}$. However, we have the following theorem.

Theorem 7. *After dropping the rank one constraint (3.32c), Problem (P3c) is convex in terms of b_k and $\mathbf{V}$, respectively.*

Proof. The proof is shown in B.4. □

Note that simply dropping the rank one constraint (3.32c) does not necessarily result in an optimal $\mathbf{v}^*$ due to the additional constraint (3.32b). In other words, the optimal solution to Problem (P3c) after dropping the rank one constraint might not be feasible. One way is to use the Gaussian randomization method as shown in [17] to find an approximated solution. A common way to recover $\mathbf{v}$ from $\mathbf{V}$ is to denote $\mathbf{V}$ as $\mathbf{V} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^H$, where $\mathbf{U} \in \mathbb{C}^{(N+1)\times(N+1)}$ is an unitary matrix and $\mathbf{\Sigma}$ is an eigenvalue diagonal matrix. A feasible solution is constructed as $\hat{\mathbf{v}} = \mathbf{V}\mathbf{\Sigma}^{1/2}\boldsymbol{\zeta}$, where $\boldsymbol{\zeta} \in \mathbb{C}^{(N+1)\times 1}$ is a randomly generated complex circularly symmetric Gaussian random variable with zero mean and unit variance. The solution can be recovered by $\mathbf{v}^* = \exp\left\{j \arg\left(\frac{\mathbf{v}}{v_{N+1}}\right)\right\}$ where v_{N+1} is the last element of vector $\mathbf{v}$. The optimal $\mathbf{V}^*$ can be further obtained from $\mathbf{v}^*$.

We denote the problem of (P3c) after dropping the rank one constraint (3.32c) as problem (P3c2'). Since this problem is in the form of sum-of-ratios, conventional fractional programming techniques such as the Dinkelbach's method cannot be used. To solve this problem, we first transform Problem (P3c2') into its equivalent form (P3d) by introducing auxiliary variable β_k.

$$(P3d) \quad \min_{\mathbf{V},\beta_k,b_k} \quad \sum_k \beta_k \tag{3.33}$$

$$\text{s.t.} \quad \frac{p_{c,k} + p_k - A_k}{b_k \log_2(1 + p_k f_k(\mathbf{V})/(N_0 b_k))} \leq \beta_k \tag{3.34}$$

$$(3.32b), (3.32d), (C5).$$

Theorem 8. *If* $\mathbf{V}^*$, $\{b_k^*\}$, *and* $\{\beta_k^*\}$ *are the optimal solution to (P3d), then there exists* $\{\lambda_k^*\}$ *such that* $\mathbf{V}^*$ *and* $\{b_k^*\}$ *are a solution to the following problem for* $\lambda_k = \lambda_k^*$ *and* $\beta_k = \beta_k^*$.

$$(P3d') \quad \min_{\mathbf{V}, b_k} \sum_k \lambda_k \left(p_{c,k} + p_k - A_k - \beta_k b_k \log_2 \left(1 + \frac{p_k f_k(\mathbf{V})}{N_0 b_k} \right) \right) \tag{3.35}$$
$$\text{s.t. } (3.32b), (3.32d), (C5),$$

and $\mathbf{V}^*$, $\{b_k^*\}$ *also satisfy the following system equation for* $\lambda_k = \lambda_k^*$ *and* $\beta_k = \beta_k^*$:

$$\lambda_k = \frac{1}{b_k^* \log_2(1 + \frac{p_k f_k(\mathbf{V}^*)}{N_0 b_k^*})} \tag{3.36a}$$
$$\beta_k = \frac{p_{c,k} + p_k - A_k}{b_k^* \log_2(1 + \frac{p_k f_k(\mathbf{V}^*)}{N_0 b_k^*})}. \tag{3.36b}$$

Proof. The proof is shown in appendix B.5 □

Theorem 8 shows that the solution to Problem (P3d) can be obtained by finding the solutions that satisfy the KKT conditions in (B.7) among the solutions to Problem (P3d'). More important, if the solution is unique, it will be the global optimal solution.

1) Finding $(\mathbf{V}^*, b_k^*)$ *When* λ_k *and* β_k *are given:* Note that for a fixed λ_k and β_k, Problem (P3) belongs to convex optimization. Hence effective algorithms can be designed to find the optimal solution $(\mathbf{V}^*, b_k^*)$. Since $\mathbf{V}$ and b_k are coupled, we propose to use the alternative optimization which optimizes each variable alternatively. First of all, when b_k is known, Problem (P3d') becomes a semidefinite programming (SDP), which can be solved with existing optimization tools such as CVX [86].

$$\max_{\mathbf{V}} \quad \sum_k \lambda_k \beta_k b_k \log_2 \left(1 + \frac{p_k f_k(\mathbf{V})}{N_0 b_k} \right) \tag{3.37}$$
$$\text{s.t.} \quad (3.32b), (3.32d).$$

Algorithm 3 Joint Optimization of $\mathbf{V}$ and b_k for Given λ_k and β_k

1: Initialization b_k^t and set $t = 1$;
2: **repeat**
3: Obtain $\mathbf{V}^t$ by solving problem (3.37) with SDP;
4: Recover $\mathbf{v}^t$ from $\mathbf{V}^t$ with Gaussian randomization algorithm;
5: Update the new variable $\mathbf{V}^t$;
6: Obtain b_k^t from (3.38);
7: $t = t + 1$;
8: **until** the objective function in (P3d') does not decrease

Theorem 9. *The optimal solution of b_k to (P3d') is given by*

$$b_k = \frac{p_k f_k(\mathbf{V})}{N_0 x_k}, \tag{3.38}$$

where

$$x_k = -\frac{1}{W_L(-e^{-C_k})} - 1, \tag{3.39}$$

and $W_L(\cdot)$ is the Lambert W *function. Note that the μ in the expression of C_k is the Lagrange multiplier for Problem (P3d') satisfying $\sum_k b_k = B$.*

Proof. The proof is shown in appendix B.6 □

The complete algorithm of finding the $\mathbf{V}$ and b_k when λ_k and β_k are given is presented in Algorithm 3.

2) Update Lagrange Multipliers λ_k and β_k: Now we update the Lagrange multipliers λ_k and β_k so that (3.36) will be satisfied. We follow a similar step as in [80, 87, 88] with the simple gradient method. Specifically, we choose initial values of the Lagrange variables and then a standard Newton-like method is used to update the Lagrange multipliers, as

$$\lambda_k^{t+1} = \lambda_k^t + \xi^{i^{(n)}} \nabla_1 \tag{3.40a}$$

$$\beta_k^{t+1} = \beta_k^t + \xi^{i^{(n)}} \nabla_2. \tag{3.40b}$$

Algorithm 4 Joint Optimization of $\mathbf{V}$ and b_k

1: Initialize λ_k^t and β_k^t according to (3.36);
2: Set $t = 1$;
3: **repeat**
4: When λ_k^t and β_k^t is given, obtain $\mathbf{V}^t$ and b_k^t with Algorithm 3;
5: Update λ_k^{t+1} and β_k^{t+1} according to (3.40);
6: $t = t + 1$;
7: **until** $\phi_k(\lambda_k^{t+1})$ and $\psi_k(\beta_k^{t+1})$ approaches zero;

Here t is the iteration index, $\xi^{i^{(n)}}$ is the step size, and ∇_1 and ∇_2 are the gradient directions for λ_k^t and β_k^t, respectively, given by

$$\nabla_1 = -\frac{\phi_k(\lambda_k)}{\phi'(\lambda_k)}, \quad \nabla_2 = -\frac{\psi_k(\beta_k)}{\psi'(\beta_k)}.$$

We also have

$$\begin{aligned}\phi_k(\lambda_k) &= \lambda_k b_k^* \log_2\left(1 + \frac{p_k f_k(\mathbf{V}^*)}{N_0 b_k^*}\right) - 1 \\ \psi_k(\lambda_k) &= \beta_k b_k^* \log_2\left(1 + \frac{p_k f_k(\mathbf{V}^*)}{N_0 b_k^*}\right) - (p_{c,k} + p_k - A_k),\end{aligned}$$

and n is the smallest integer among $\{1, 2, ..., \}$ satisfying

$$\sum_k |\phi_k(\lambda_k^{t+1})|^2 + \sum_k |\psi_k(\beta_k^{t+1})|^2 \leq (1 - \epsilon\xi^{i^{(n)}})^2 \left(\sum_k |\phi_k(\lambda_k^t)|^2 + \psi_k(\lambda_k^t)|^2\right), \tag{3.41}$$

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

Since $\left(1 - \epsilon\xi^{i^{(n)}}\right)^2$ will be a random number between $[0, 1]$, inequality (3.41) will ensure that $\phi_k(\lambda_k^{t+1})$ and $\psi_k(\beta_k^{t+1})$ both go to zero, which is exactly what the optimal solution in (3.36) suggests. The joint optimization algorithm is summarized in Algorithm 4.

Theorem 10. *Algorithm 4 will converge after a finite number of iteration steps.*

Proof. The proof is shown in appendix B.7 □

3.5.4 Design of the Accuracy Parameter

Finally, we optimize the accuracy parameter η. The objective function can be written as

$$f(\eta) = \frac{1}{1-\eta}\left(u + v\log\left(\frac{1}{\eta}\right)\right), \quad \eta \in (0,1), \tag{3.42}$$

where $u = \sum_k (p_{c,k} + p_k)\frac{S}{R}$ and $v = \sum_k \kappa D_k c_k f_k^2 \ell_1$. The optimal η can be similarly obtained as in the single user case with the bisection algorithm.

The complete algorithm for energy-efficient federated learning is presented in Algorithm 5. Note that the variables involved generally has a closed-form expression or can be obtained via simple one-dimensional search with neglect-able complexity except for variable b_k and $\mathbf{v}$, which requires solving an SDP problem. Generally, solving an SDP problem with the interior method or with general CVX solvers such as MOSEK [89] incurs high complexity. According to [90, Thm. 3.12], the complexity of solving an SDP problem with m constraints and an $n \times n$ variable matrix is $\mathcal{O}(\sqrt{n}\log(1/\epsilon)(mn^3 + m^2n^2 + m^3))$, where ϵ is the solution accuracy. In this problem, we have $n = N + 1$ and $m = N + 1$, hence the approximate complexity for solving one SDP problem would be $\mathcal{O}(\sqrt{N+1}\log(1/\epsilon)(N+1)^4)$. Suppose the iterations for Algorithm 3, Algorithm 4 and Algorithm 5 are I_2, I_3, and I_4, respectively. Then the proposed Algorithm 5 needs to solve a standard SDP problem (3.37) for $I_2 I_3 I_4$ times. Hence the total complexity of Algorithm 5 would be $\mathcal{O}(\sqrt{N+1}\log(1/\epsilon) I_2 I_3 I_4 (N+1)^4)$. When the number of the reflecting elements in the IRS becomes large, the total complexity would become considerably high.

3.6 Low Complexity Algorithm

As analyzed before, the complexity of the proposed Algorithm 5 mainly comes from solving the SDP problem (3.37). To reduce the complexity, or more specifically, to reduce

Algorithm 5 Energy-efficient Federated Learning

Initialize IRS phase shift matrix $\mathbf{\Theta}$ and the iteration number $t = 1$;
repeat
  Obtain f_k^t according to (3.13);
  Obtain p_k^t according to (3.16);
  Obtain b_k^t, $\mathbf{v}^t$ with Algorithm 4;
  Obtain η^t according to Theorem 3;
  Calculate the total energy consumption E^t;
  $t = t + 1$;
until $\frac{|E^{t+1}-E^t|}{|E^t|} \leq \epsilon_1$ and (C1) is satisfied

the complexity of getting $\mathbf{v}$ and b_k in Algorithm 3, we propose to leverage the majorization-minimization (MM) algorithm [91]. The idea is to find an easy-to-solve surrogate problem with a surrogate objective function to problem (3.37), and then solve this problem induced from the surrogate objective function instead of the original one. This approach can generate a sequence of sub-optimal solutions $\mathbf{v}^t$ at each iteration to approach the global optimal solution.

To proceed, we rewrite problem (3.37) as

$$\begin{aligned} \max_{\mathbf{v}} \quad & \sum_k \lambda_k \beta_k b_k g_k(\mathbf{v}) \\ \text{s.t.} \quad & v_{N+1} = 1; \quad |v_n| = 1, \quad \forall 1 \leq n \leq N, \end{aligned} \tag{3.43}$$

where $g_k(\mathbf{v}) = \log_2\left(1 + \frac{p_k\left(\mathbf{v}^H \mathbf{R}_k \mathbf{v} + \mathbf{h}_{\mathrm{d},k}^H \mathbf{h}_{\mathrm{d},k}\right)}{N_0 b_k}\right)$. To show the hidden convexity of $g_k(\mathbf{v})$, we have

$$g_k(\mathbf{v}) = -\log_2\left(1 - \frac{p_k\left(\mathbf{v}^H \mathbf{R}_k \mathbf{v} + \mathbf{h}_{\mathrm{d},k}^H \mathbf{h}_{\mathrm{d},k}\right)}{M_k}\right), \tag{3.44}$$

where $M_k = N_0 b_k + p_k \left(\mathbf{v}^H \mathbf{R}_k \mathbf{v} + \mathbf{h}_{\mathrm{d},k}^H \mathbf{h}_{\mathrm{d},k}\right)$. Then $g_k(\mathbf{v}, M_k)$ is jointly convex in terms of $\{\mathbf{v}, M_k\}$ [92]. Its lower bound surrogate function is given by

$$\begin{aligned} g_k(\mathbf{v}, M_k) &\geq g_k(\mathbf{v}^t, M_k^t) + \frac{\partial g_k}{\partial M_k}\big|_{M_k = M_k^t}(M_k - M_k^t) + (\mathbf{v} - \mathbf{v}^t)^H \frac{\partial g_k}{\partial \mathbf{v}}\bigg|_{\mathbf{v}=\mathbf{v}^t} \\ &= \mathrm{const}_k^t + \tau_k^t \mathbf{v}^H \mathbf{R}_k \mathbf{v} + 2\mathbf{v}^H \mathbf{r}_k^t \triangleq \tilde{g}_k(\mathbf{v}|\mathbf{v}^t), \end{aligned} \tag{3.45}$$

where $\tau_k^t = -\frac{p_k^2\left(\mathbf{v}^{t,H}\mathbf{R}_k\mathbf{v}^t + \mathbf{h}_{\mathrm{d},k}^H \mathbf{h}_{\mathrm{d},k}\right)}{M_k^t N_0 b_k \ln 2}$, $\mathbf{r}_k^t = \frac{p_k \mathbf{R}_k}{N_0 b_k \ln 2}\mathbf{v}^t$ and

$$\begin{aligned} \frac{\partial g_k}{\partial M_k}\big|_{M_k = M_k^t} &= -\frac{p_k\left(\mathbf{v}^{t,H}\mathbf{R}_k\mathbf{v}^t + \mathbf{h}_{\mathrm{d},k}^H \mathbf{h}_{\mathrm{d},k}\right)}{M_k^t N_0 b_k \ln 2} \\ \frac{\partial g_k}{\partial \mathbf{v}}\big|_{\mathbf{v}=\mathbf{v}^t} &= \frac{2p_k \mathbf{R}_k}{N_0 b_k \ln 2}\mathbf{v}^t. \end{aligned}$$

$\tilde{g}_k(\mathbf{v}|\mathbf{v}^t)$ is twice differentiable and concave. Moreover, we can verify that (i) $\tilde{g}_k(\mathbf{v}^t|\mathbf{v}^t) = g_k(\mathbf{v}^t)$; (ii)$\tilde{g}_k(\mathbf{v}|\mathbf{v}^t) \leq g_k(\mathbf{v})$; and (iii) $\nabla\tilde{g}_k(\mathbf{v}|\mathbf{v}^t) = \nabla\tilde{g}_k(\mathbf{v}^t)$. Hence $\tilde{g}_k(\mathbf{v})$ is minorized at any $\mathbf{v}^n$ with a function $\tilde{g}_k(\mathbf{v}|\mathbf{v}^t)$ [91, 92]. The MM method can be used to find a sequence of solutions to approach the global optimal solution with low complexity.

We can rewrite the objective function in (3.43) as $\sum_k \lambda_k \beta_k b_k \tilde{g}_k(\mathbf{v}|\mathbf{v}^t)$. Put the expression of $\tilde{g}_k(\mathbf{v}|\mathbf{v}^t)$ into the objective function and remove the constant. Accordingly, we need to solve the following problem at each iteration t.

$$\begin{aligned} \min_{\mathbf{v}} \quad & \mathbf{v}^H \mathbf{R}^t \mathbf{v} - 2\mathrm{Re}(\mathbf{v}^H \mathbf{r}^t) \\ \text{s.t.} \quad & v_{N+1} = 1; \quad |v_n| = 1, \ \forall 1 \leq n \leq N, \end{aligned} \tag{3.46}$$

where $\mathbf{R}^t = -\sum_k \lambda_k \beta_k b_k \tau_k^t \mathbf{R}_k$ and $\mathbf{r}^t = \sum_k \lambda_k \beta_k b_k \mathbf{r}_k^t$.

Proposition 1. *The objective function in (3.46) can be approximated by [91]:*

$$\begin{aligned} \mathbf{v}^H \mathbf{R}^t \mathbf{v} - 2\mathrm{Re}(\mathbf{v}^H \mathbf{r}^t) &\leq \mathbf{v}^H \mathbf{\Gamma}^t \mathbf{v} - 2\mathrm{Re}\left(\mathbf{v}^H\left[\mathbf{r}^n + (\mathbf{\Gamma}^t - \mathbf{R}^t)\mathbf{v}^t\right]\right) + \mathbf{v}^{t,H}(\mathbf{\Gamma}^t - \mathbf{R}^t)\mathbf{v}^t \\ &= \rho_{\max}(\mathbf{R}^t)\mathbf{v}^H\mathbf{v} - 2\mathrm{Re}\left(\mathbf{v}^H \tilde{\mathbf{r}}^t\right) + \mathrm{const}', \end{aligned} \tag{3.47}$$

where $\lambda_{\max}\{\mathbf{R}^t\}$ *is the maximum eigenvalue of matrix* $\mathbf{R}^t$, $\mathbf{\Gamma}^t = \lambda_{\max}\{\mathbf{R}^t\}\mathbf{I}_{N+1}$ *and* $\tilde{\mathbf{r}}^t = \mathbf{r}^t + (\lambda_{\max}\{\mathbf{R}^t\}\mathbf{I}_{N+1} - \mathbf{R}^t)\mathbf{v}^t$.

To minimize the objective function in (3.46), we can optimize its upper bound (3.47). Note that $\mathbf{v}^H\mathbf{v} = N + 1$ since $|v_n| = 1, \forall n$. Hence, we only need to maximize the term $2\text{Re}\left(\mathbf{v}^H\tilde{\mathbf{r}}^t\right)$. This term is maximized when the phase of $\mathbf{v}$ and the phase of $\tilde{\mathbf{r}}^t$ are the same, i.e.,

$$v_i = \exp\left\{j \arg(\tilde{r}_i^t)\right\}, \quad \forall 1 \leq i \leq N. \tag{3.48}$$

The phase vector has a closed-form expression (3.48). The complexity of the proposed algorithm mainly comes from computing the eigenvalues of matrix $\mathbf{R}^t \in \mathbb{C}^{(N+1)\times(N+1)}$, which has a complexity of $\mathcal{O}((N+1)^3)$. Hence the complexity would be $\mathcal{O}(I_2 I_3 I_4 (N+1)^3)$.

3.7 Simulation Study

In this section, simulation results are presented to validate the performance of the proposed IRS-assisted federated learning system. The federated learning parameters follow a similar setting as in [79,93]. The IRS related parameters are set based on the setting in [23]. Specifically, we consider an IRS assisted communication scenario as depicted in Fig. 3.2. In this x-y plane, the IRS is located at location $(20, x)$m. The default value of x is 20m in this paper. The IoT devices are located randomly in a disk area around center $(30, 0)$m with a radius of 2m. The BS is located at the origin (0,0)m. In this section, we will change the location of the IRS and investigate the impact of such changes on the overall system performance. The channel gains are a combination of distance-dependent large-scale fading and small-scale fading. The small-scale fading is assumed to be Rayleigh fading $\mathcal{CN}(0, 1)$. The large scale path loss model follows $Ad^{-\alpha}$, where $A = -30$dB is the path loss at a reference distance 1m, d is the distance between the transmitter and receiver, and α is the path loss component. The path loss components for channels $\mathbf{h}_{\mathrm{r},k}$, $\mathbf{h}_{\mathrm{d},k}$, and $\mathbf{H}$ are set to 2.2, 3.5, and 2.2, respectively. The noise power N_0 is set to $10^{-10}W/Hz$. The global training

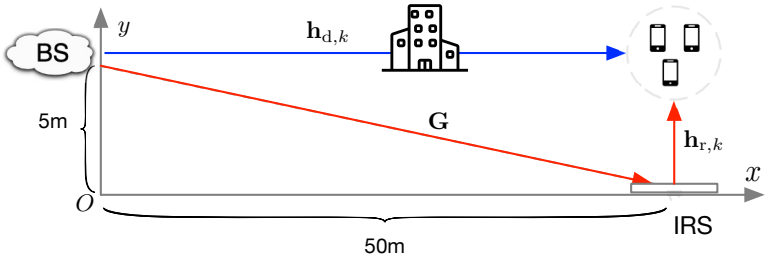


Figure 3.2: Illustrate the deployment of the IRS-assisted federated learning system.

Table 3.1: Federated Learning Parameter Setting

Parameter	*Notation*	*Value*
Local sample data size	D_k	[8,12] MB
Number of CPU cycles to process one bit	c_k	30 cycles/bit
Chip energy coefficient	κ	2×10^{-28}
Training completion deadline	T_k	40s
Upload model size	S	7850 bit
IoT device static power	$p_{\mathrm{c},k}$	0.5 W
Maximum operating frequency	$f_{\max}$	1 GHz
Maximum transmit power	$P_{\max}$	20 W
Bandwidth	B	1 MHz
Noise power	N_0	10^{-10} W/Hz

completion deadline is set as $T = 40$ s. For the bisection algorithm, the target accuracy is set to 10^{-5}. For Algorithm 2 and Algorithm 5, the stopping criteria is set to $\epsilon_1 = 0.01$. Each simulation result is the average of over 300 realizations.

The following two benchmark algorithms are also simulated for comparison purpose.

1. *IRS with Random Phase:* The IRS uses random phases. The detection vector $\mathbf{w}_k^t$, frequency f_k, power p_k, bandwidth b_k, and the local accuracy parameter η are optimally designed as in the proposed scheme.

2. *Without IRS:* There is only the direct channel between IoT devices and the BS. The other parameters are set as the same as in the IRS with Random Phase case.

3.7.1 Impact of the Number of Reflecting Elements N

First of all, we verify the performance of the proposed low-complexity algorithm by changing the number of reflecting elements on the IRS. In Fig. 3.3, we set $K = 5$, $M = 4$, and $T = 40$s and compare the energy consumption performance of different schemes. The SDP algorithm denotes Algorithm 5 where problem (3.37) is solved using SDP. It can be seen that with the increase of the number of reflecting elements on the IRS, the energy consumptions of the proposed low-complexity algorithm and the SDP algorithm both decrease. This is because the IRS can reconfigure the environment and help the devices to save model uploading power. A larger number of reflecting elements on the IRS generally brings a better performance. However, the processing complexity in optimizing the elements would also become quite high. Moreover, we find that the energy consumption curves for the case without IRS and IRS with random phase shift look like horizontal lines. This is straightforward as anticipated. For the case without IRS, changing the number of reflecting elements on the IRS will have no impact on the energy consumption performance. For the case IRS with random phase shift, the performance does not improve significantly since the channels are not properly configured.

In Fig. 3.4, we perform similar experiments with $K = 10$, $M = 5$, and $T = 40$s. When the number of reflecting elements on the IRS is increased to 50, the proposed algorithm can save up to about 55 Joule compared with no IRS deployment and IRS with random phase shift. These results again demonstrate the importance of jointly optimizing resource allocation and IRS beamforming. From both Fig. 3.3 and Fig. 3.4, the performance of the proposed algorithm and the SDP algorithm achieves very similar performance, but the former has a significantly lower complexity and runs much faster. Hence, for the rest simulations, we will only consider the proposed low-complexity algorithm.

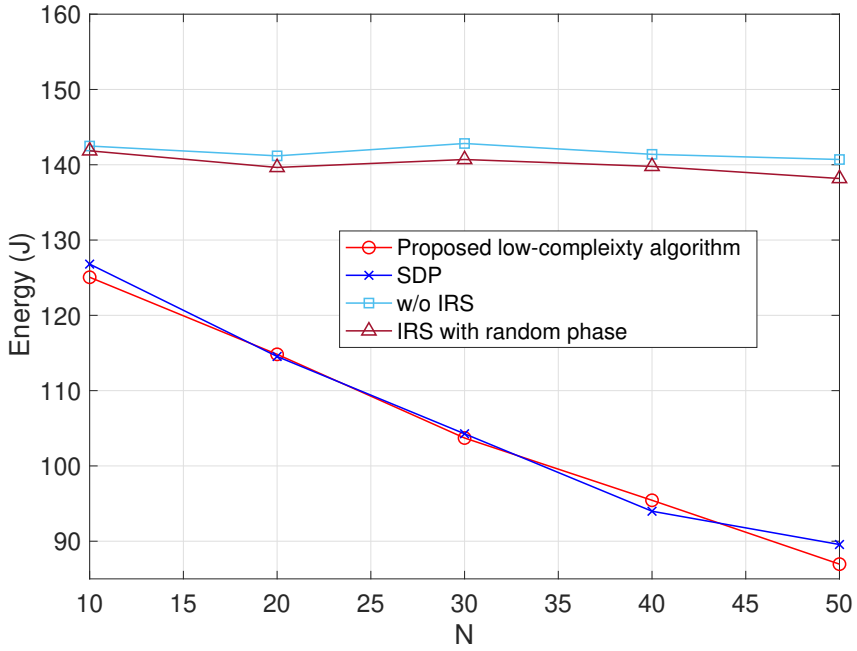


Figure 3.3: Total energy consumption versus N when $K = 5$, $M = 4$, and $T = 40$.

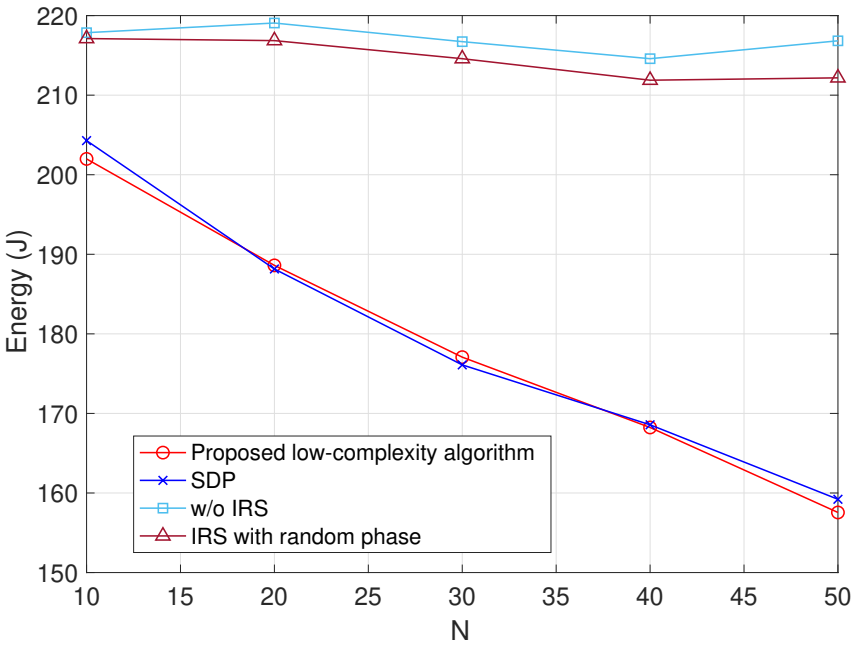


Figure 3.4: Total energy consumption versus N when $K = 10$, $M = 5$, and $T = 40$.

3.7.2 Convergence Behavior

In this section, we investigate the convergence of the proposed low-complexity algorithm. The convergence behavior of the general federated learning scheme is plotted in

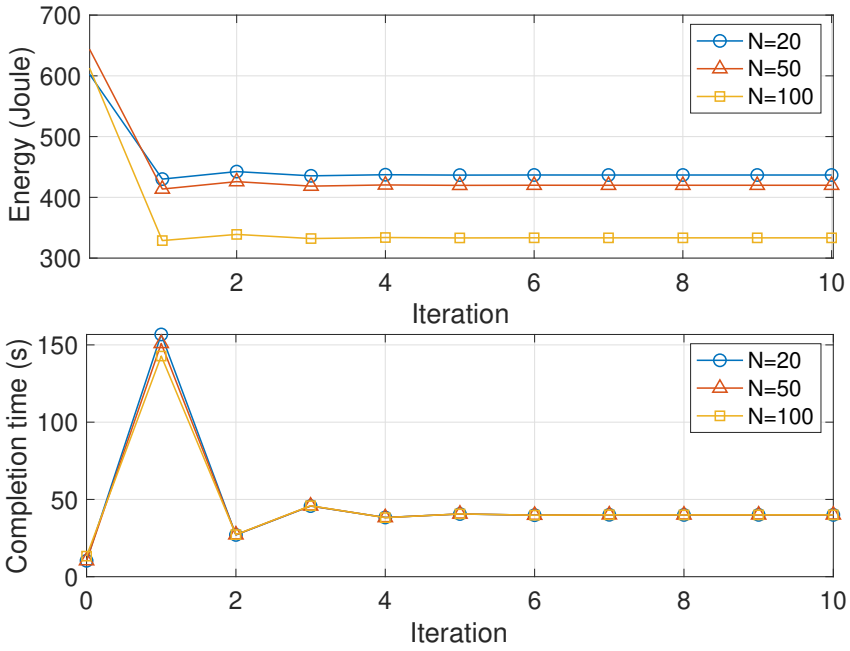


Figure 3.5: Convergence of the first device in a multiuser federated learning system with $K = 20$, $M = 4$, and $T = 40$.

Fig. 3.5. It can be seen that after 2-3 iterations, the energy consumption decreases to a low value. However, the completion time might violate the training deadline constraint. After several fine-tuning iterations, we can obtain a feasible solution that minimizes the energy consumption while also satisfying the completion time constraint.

We also change the number of the reflecting elements on the IRS. We find that when $N = 50$, the device saves more energy than the case when $N = 30$. Despite that, convergence of the training process does not change much when N is varied.

3.7.3 Energy and Time Consumption of Each Device

The power, bandwidth, and frequency allocation parameters for different devices are presented in Fig. 3.6. In this simulation, the devices are located very close to each other. Their operating frequency, power, and bandwidth seem not differ too much. The total energy consumption and time consumption over the entire training process is shown in Fig. 3.7. It can be seen that all the devices share the same latency, which is exactly $T = 40$s, while

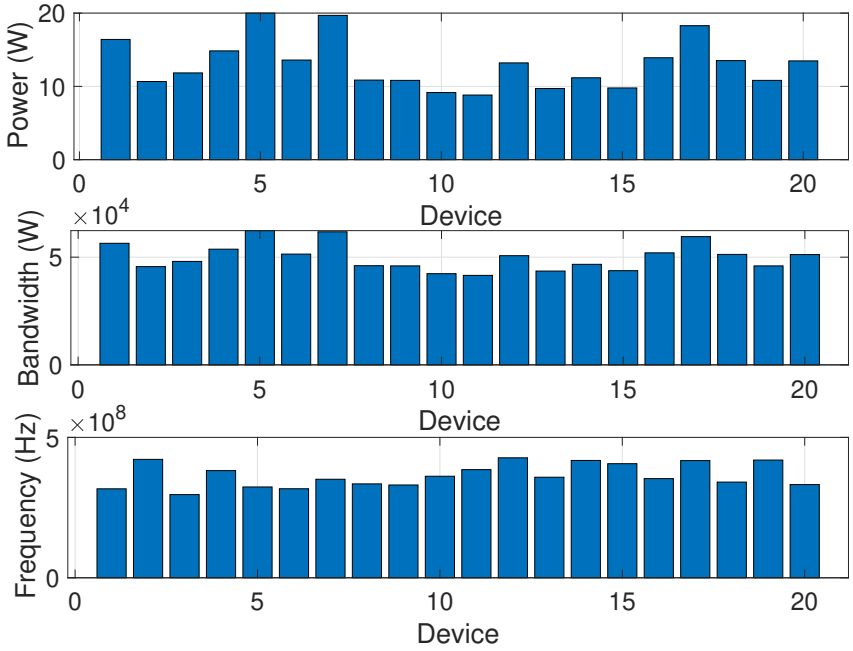


Figure 3.6: Power, bandwidth, and frequency allocation for different devices with $K = 20$, $M = 4$, $N = 20$, and $T = 40$.

the consumed energy differs. With our parameter setting, we also find that local training nearly does not consume much energy, but it accounts for almost 99% of latency. On the contrary, model uploading takes a lot of energy while it nearly takes no time. This setting is reasonable since in practice the model is trained locally by each device. The training usually takes several rounds which take time. Moreover, the device works on the lowest possible frequency, which further slows down the completion time. On the other hand, the devices are battery powered, the model update process consumes most of the energy. In this case, the deployed IRS can work as a passive, enhanced channel, which helps the devices to save their battery power.

3.7.4 Impact of the Number of Devices K

We investigate the impact of the number of devices in Fig. 3.8. We find that the energy consumption generally increases linearly with the number of devices involved. This is because in the multiuser system, only the bandwidth and the IRS reflecting elements are optimized

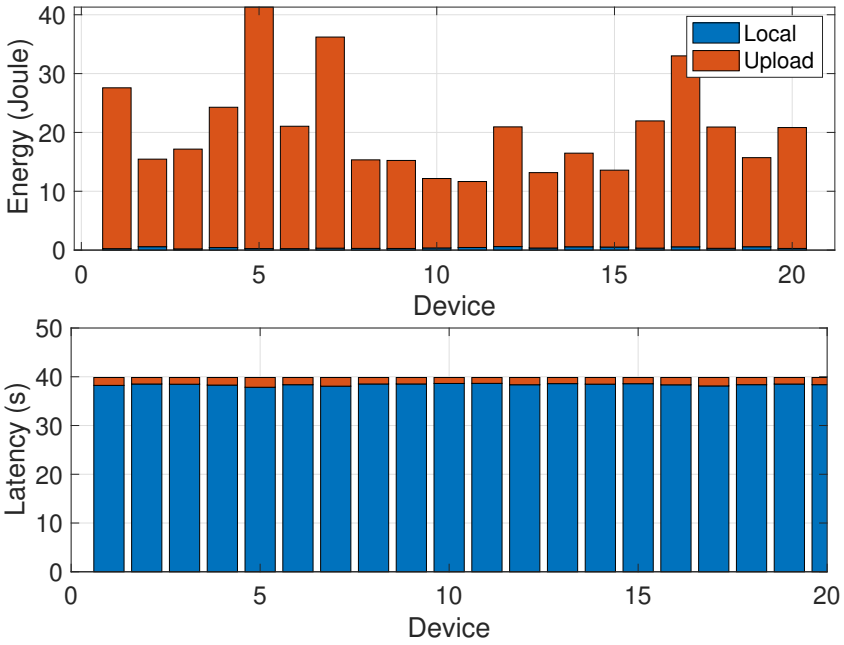


Figure 3.7: Total energy consumption and latency of local model training and model uploading for different devices with $K = 20$, $M = 4$, and $T = 40$.

jointly. Each device selects its own operating frequency and power. With increased number of devices, the proposed algorithm saves more energy than the two baseline algorithms. Moreover, the performance gap becomes larger as K is increased. This result demonstrates the advantages of the proposed algorithm in a communication system where the number of IoT devices is large.

3.7.5 Impact of the Number of Antennas M on the BS

Fig. 3.9 shows the impact of the number of BS antennas on energy saving of the federated learning system. As can be seen, with more receiving antennas on the BS, the system energy consumption can be greatly reduced. This is because the antennas on the BS provide additional multiplexing gain at the receiver so that each IoT device can reduce their transmit power for model uploading. Moreover, the performance gap between the proposed algorithm and the two benchmark algorithms will gradually vanish with increased M. This motivates

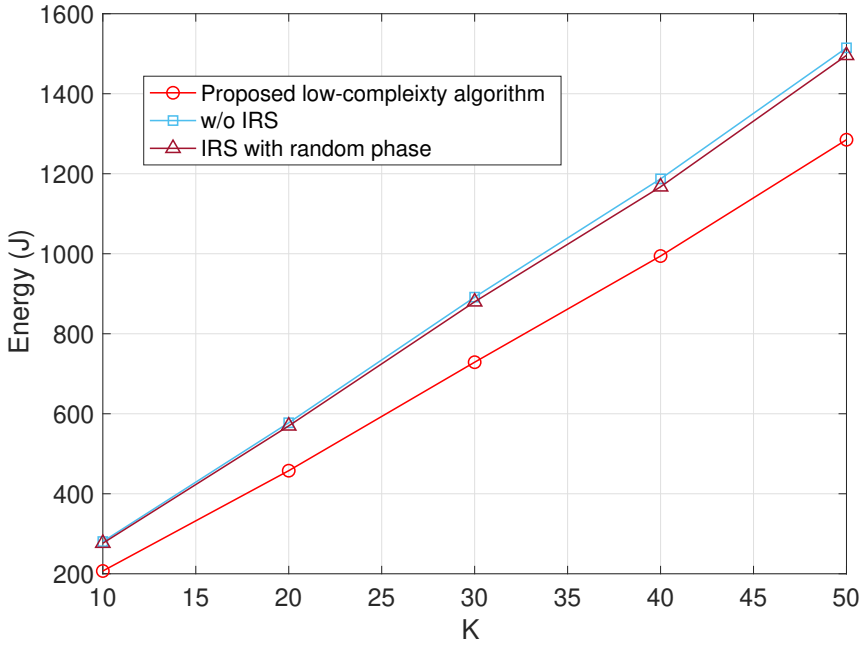


Figure 3.8: Total energy consumption versus K with $M = 4$, $N = 40$, and $T = 40$.

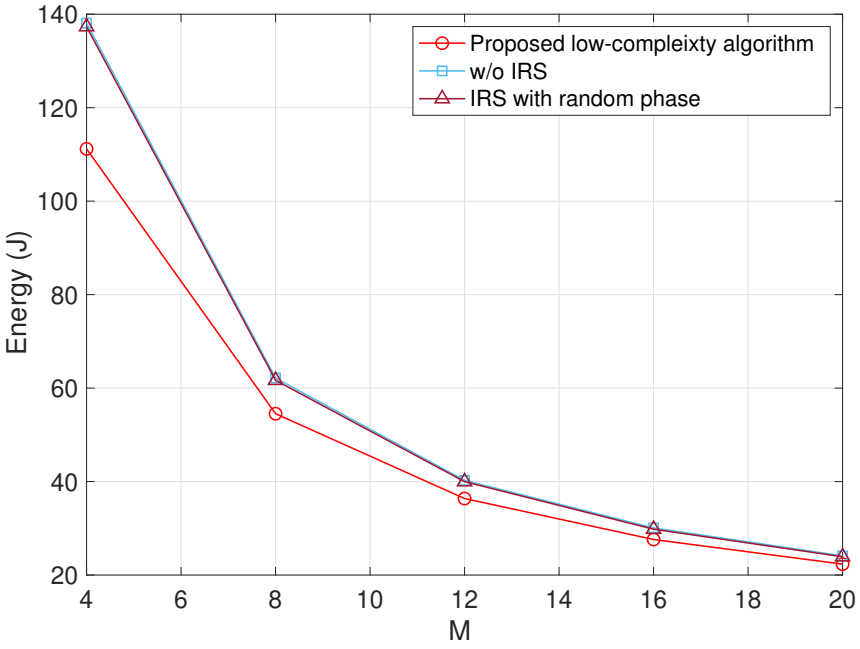


Figure 3.9: Total energy consumption versus M with $K = 5$, $N = 20$, and $T = 40$.

us to deploy an IRS with a larger number of reflecting elements, i.e., $N > M$, to harvest the reconfigured channel gain provided by the IRS.

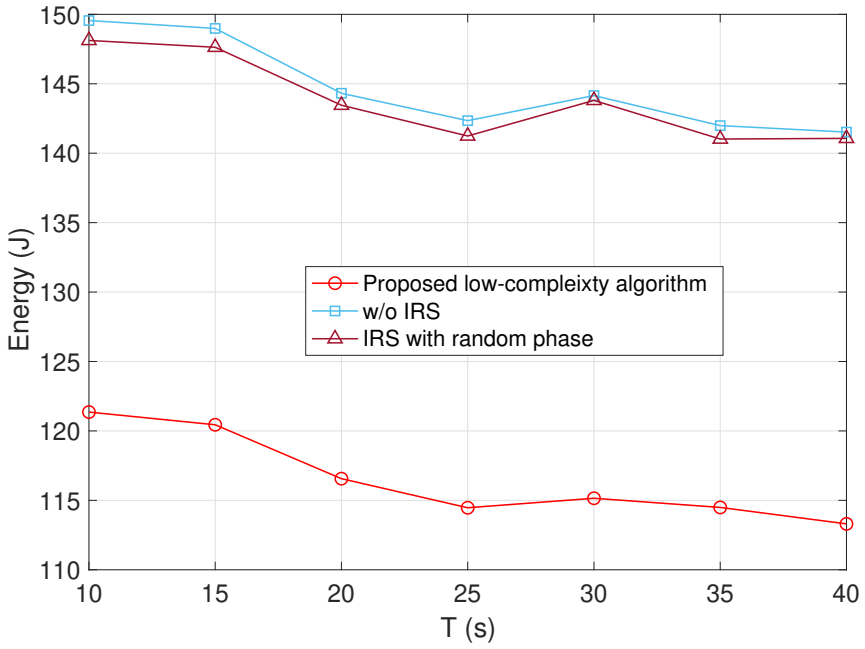


Figure 3.10: Total energy consumption versus T with $K = 5$, $N = 20$, and $M = 4$.

3.7.6 Impact of the Task Completion Time T

Fig. 3.10 shows the impact of task completion time T on energy saving of the federated learning system. It can be seen that the energy consumption slightly decreases with the increase of the completion time. This is because the devices always work on the lowest frequency to save energy and satisfy the task completion time. Moreover, in our setting the local computing takes a lot of time but only accounts for a small portion of energy consumption, while model uploading takes little time but consumes a lot of energy. In other words, the total energy consumption of the proposed federated learning system is insensitive to the task completion time.

3.7.7 Impact of the Bandwidth Constraint B

Fig. 3.11 shows the impact of the communication bandwidth on the system energy consumption. With the increase of the available bandwidth, each IoT device can reduce their transmit power or their uploading time to upload the same model. Hence the total

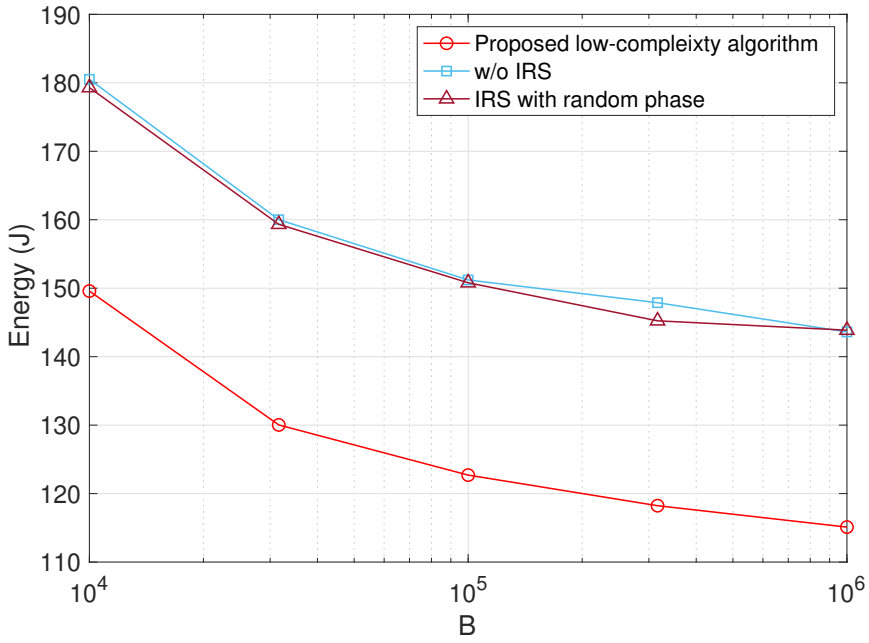


Figure 3.11: Total energy consumption versus bandwidth B with $K = 5$, $N = 20$, and $M = 4$.

energy consumption can be saved. As can be seen, the absolute value of the slope of these curves gradually goes to zero, which suggests that the impact of bandwidth is diminishing in the high bandwidth region. In other words, when the available bandwidth is large enough, the other communication/computing factors will become the major factor(s) that prevent the reduction of energy consumption.

3.7.8 Impact of the IRS Location and Path Loss on the Reflecting Channel

The impact of the IRS location on the system energy consumption is presented in Fig. 3.12, where x measures the distance between the IRS and BS. When the IRS is close to the IoT devices, the energy saving will be significant. The impact of the location of the IRS depends on the IRS reflected channel fading. In practice, the location of the IRS should be properly selected to reap the maximum benefit of the IRS technology. Similarly, the energy consumption versus the path loss of the reflecting channel is shown in Fig. 3.13. The default setting on the reflected channel is $\alpha = 2.2$ for $\mathbf{h}_{\mathrm{r},k}$ and $\mathbf{H}$. Now we change the value of the

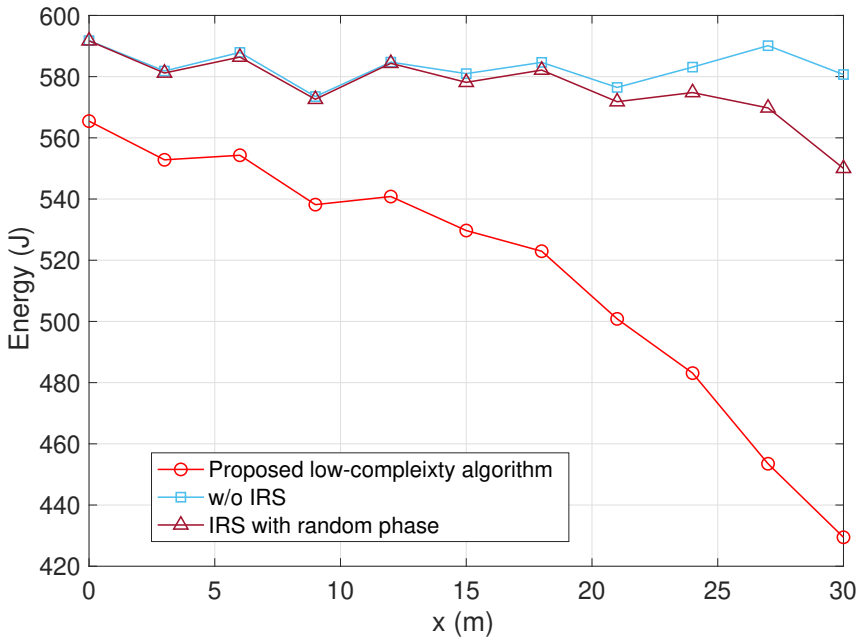


Figure 3.12: Energy consumption versus the location of the IRS when $N = 20$, $K = 20$, $M = 4$, and $T = 40$.

path loss from 2 to 2.5. We find that when the path loss on the reflected channel becomes larger, the energy saving becomes less. This is easy to explain. When the path loss on the reflected channel becomes larger, the channel enhancement effect of the IRS will become weaker. In the extreme case when the path loss on the reflected channel is infinitely large, i.e., the reflected channel is blocked, the deployment of IRS will make on difference.

Our simulation assumes that the repeated model uploading accounts for the major energy consumption of the federated learning system. In some systems, the local computing may take up the major energy consumption compared with the communication process. Different system factors such as local data size, model accuracy level, environment noise power level, and the CPU processing capability may have various effects on the system trade-offs: (i) between task completion time and the energy consumption, and (ii) energy consumption caused by communication and computation. The proposed algorithm provides a low complexity solution to explore these trade-offs.

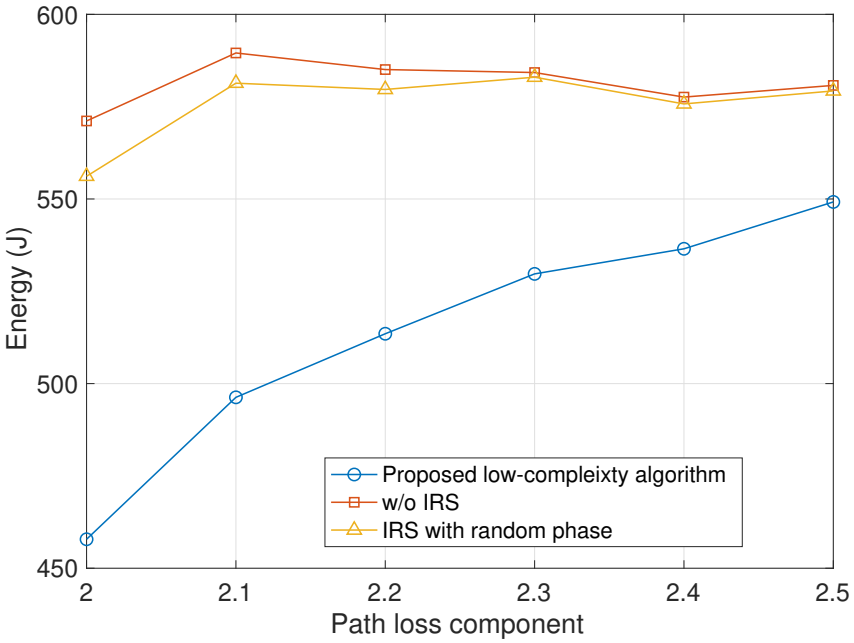


Figure 3.13: Energy consumption versus the path loss of the reflecting channel.

3.8 Conclusions

In this paper, we considered an energy-efficient federated learning framework where devices upload their locally trained models when assisted by an IRS. In this framework, an energy minimization problem was considered. We proposed an efficient parameter optimization algorithm to jointly optimize system parameters, such as the operating frequency of each device, transmit power, bandwidth, the IRS phases, and the local accuracy parameter. The proposed low-complexity algorithm can reasonably manage the energy resources by balancing the communication and local training costs. We have conducted extensive experiments to shed insight on the benefits on the use of IRS in federated learning systems.

Chapter 4

Energy-Efficient Resource Allocation In D2D Networks With Deep Learning

4.1 Introduction

Energy efficient power control is one of the most important issues for the sustainable development of future wireless networks [94, 95]. It is estimated that the energy-efficiency (EE) will increase 2000 times compared to the present networks [96] and the number of connected devices will reach 50 billion by 2020 [97]. The corresponding greenhouse gas emissions will bring a severe impact on global warming. On the other hand, restricting the connection of devices is unrealistic. In viewing of this, there has been continued interests in improving the EE of wireless network systems, i.e., maximizing the number of transmitted bits per joule consumption. This topic is of fundamental importance to a variety of practical communication scenarios, such as massive MIMO systems [98], wide-band systems [99], D2D networks [100], relay assisted MIMO networks [101], multi-cell and/or small-cell orthogonal frequency division multiple access (OFDMA) networks [102].

The EE of a wireless link is defined as a ratio, as

$$\text{EE}[bit/Joule] = \frac{\text{Rate}[bit/s]}{\text{Power consumption}[W]}. \tag{4.1}$$

Due to the fractional nature of energy-efficient performance metrics, conventional convex optimization theory cannot be applied directly. Instead, duality theory and fractional programming [103] provides a set of suboptimal solutions. Unfortunately, due to the existence of link interference, the numerator of EE is usually non-concave. The EE maximization problem is thus NP-hard in general [104, 105]. It is shown in [104] that a global optimal solution incurs an exponentially growing complexity. Due to limited computation capacity

and stringent delay requirements, especially in a large network, it is almost impossible to perform real-time optimal power control.

In viewing of this, several sub-optimal methods are proposed for EE maximization problems. One common approach is the interference cancellation technique. In [106], the multiuser interference is mitigated with the presence of a larger number of base station antennas. In [107], an iterative algorithm is developed to maximize the EE with orthogonal or semi-orthogonal subcarrier allocation schemes. However, these works require a large number of wireless resources (orthogonal channels) and often lead to a poor performance. Another line of approach is alternating optimization, which is not optimal but enjoys limited (typically polynomial) complexity. In [108], EE is optimized by solving a series of concave-convex fractional relaxations. This way, the difficult problem is tackled by solving a series of easier approximating problems. Following this idea, in [104], a sequential fractional programming algorithm is integrated into fractional programming to compute a suboptimal power control with an affordable complexity. Contributions in this sense also include [101, 109, 110], which consider multiple antenna system, millimeter-wave system, and full duplex systems, respectively.

However, most of the existing approaches use iterative algorithms. They do not lead to a simple online implementation and do not provide a closed-form solution. The computation demanding nature would make real-time deployment a challenging task, especially in the rapidly changing large-scale wireless environment. For example, in vehicle-to-vehicle (V2V) communication where the road safety and traffic efficiency directly depend on the network delay, simply relying on conventional methods that perform channel estimation first and then computing the optimal power control with iterative algorithms would waste a lot of time and channel resources. The state-of-art approach would not be able to meet the stringent delay requirement of the V2V communication system.

Nowadays, deep learning has achieved great success in computer vision, natural language processing, and many other applications. Recent results have already demonstrated that

deep learning can be viewed as an efficient tool in solving communication problems, such as channel estimation [111, 112], signal detection [113–115], channel modeling [116–118], beam selection [119,120], resource allocations [121–123], indoor fingerprinting [124,125], and smart congestion control [126]. Of all the existing works, we are particularly interested in making real-time resource allocation practical with the aid of deep learning. In [121], a small deep neural network (DNN) is adopted to approximate a popular interference management algorithm to maximize the sum-rate of a network with high-accuracy. The computation time is significantly reduced. Ref. [122] proposed a framework where a deep Q-network (DQN) is adopted to estimate a suitable schedule and then a DNN helps to allocate power based on this schedule to maximize the sum rate of a cellular network. [127] proposed an unsupervised learning method to tackle the problem of lack of ground truth. [123] develop a DNN based optimal power control to maximize the EE of a wireless network. The developed DNN based solution is shown to be virtually optimal with extremely low online complexity.

However, the current DNN based resource control algorithm is centralized. In order to perform the optimal power control, the BS needs to know the instant channel state information (CSI) on all the links in the network. This is sometimes unrealistic and would cause considerable delay, especially in large networks. Motivated by the fact that in some networks, the second order channel statistic varies slowly, and the CSI can be viewed as a function of the distance dependent path-loss, we investigate the possibility of training a spatial neural network (NN) with transmitter-receiver geography location information (GLI), which can be easily obtained by current global positioning systems (GPS) or indoor localization techniques [124, 125]. This way, we no longer need a complex channel estimation process, and the response time can be greatly reduced. Moreover, the learning ability of current DNNs degrades significantly with the increase of the problem size. Although increasing the size of the DNN can help to alleviate this problem, the learning power is till limited and sometimes the training loss does not decrease. To tackle this problem, we introduced convolutional

layers to better capture the interference pattern across different links. This kind of structure shows great learning ability in large size problems. Besides, motivated by the success of residual learning, a feedback connection is introduced to enhance the robustness of the developed NN. With the adoption of NN, the computational burden is transformed from on-line to off-line. The developed method is thus amenable for real-time applications.

Simulation results show that our proposed NN, which we call PowerNet, can achieve a better performance in terms of EE than the several benchmark schemes. Also, PowerNet shows great generalization ability and robustness in terms of both problem sizes and channel fading types. Our simulation results demonstrate that using only GLI to perform optimal power control is possible when channel fading is mainly characterized by distance based path-loss. The performance may decrease slightly when channel shadowing and fast fading effect is added. However, the complex channel estimation process can be avoided, and the time delay can be greatly reduced. Due to the parallel computation in NN, the deep learning-based method is almost 1000 times faster than the conventional iterative optimization algorithms.

This chapter is organized as follows. in Section 4.2, we introduce the system model and formulate our problems. In Section 4.3, a successive pseudo-convex approximation (SPCA) algorithm is proposed to find a sub-optimal solution. With the SPCA algorithm, we generate the geographical-distance and power-allocation pairs as training data. In Section 4.4, we present the proposed PowerNet to learn the mapping between the geographical-distance and the resulting power control schedule. In Section 4.5, the system simulation setup is introduced. In Section 4.6, our simulation study is presented. Section 4.7 concludes the chapter.

4.2 System Model and Problem Statement

Consider a cell area with N independent D2D links, denoted by $\mathcal{D}$, randomly located in the two-dimensional region. The transmitter and receiver pairs are indexed by $i \in \mathcal{D}$. Suppose the transmit power of the ith link is denoted as p_i and h_{ij} is the channel power gain

from the transmitter of the jth link to the receiver of the ith link, which can be modeled as

$$h_{ij} = g_{ij}\alpha_{ij}, \tag{4.2}$$

where g_{ij} is the small-scale fast fading power component and α_{ij} is the large-scale fading power component consisting of path-loss and shadowing.

Suppose $\mathbf{p} = [p_1, p_2, ..., p_N]$ is the power vector, then the weighted sum rate $R(\mathbf{p})$ and the total power consumption $P(\mathbf{p})$ can be expressed as

$$R(\mathbf{p}) = \sum_{i\in\mathcal{D}} w_i R_i(\mathbf{p}) \tag{4.3}$$

$$P(\mathbf{p}) = \sum_{i\in\mathcal{D}} (\beta p_i + P_{c,i}), \tag{4.4}$$

respectively, where

$$R_i(\mathbf{p}) = \log\left(1 + \frac{h_{ii}p_i}{\sum_{j\in\mathcal{D}, j\neq i} h_{ij}p_j + \sigma_n^2}\right), \tag{4.5}$$

is the data rate on link i, w_i is the weight of link i, β is the inefficiency of the link's power amplifier, $P_{c,i}$ is the fixed circuit power consumption of link i (including baseband, RF chain, phase shifters, and power amplifiers), and σ_n^2 is the background noise. Hence the EE maximization problem for the entire system can be formulated as

$$\text{(P1)} \quad \max_{p_i} \quad \eta_{\text{EE}}(\mathbf{p}) = \frac{R(\mathbf{p})}{P(\mathbf{p})} \tag{4.6}$$

$$\text{s.t.} \quad p_i \in [0, p_{\max}], \tag{4.7}$$

where (4.7) denotes the peak power constraint at each link.

The goal of this chapter is to develop an energy-efficient power control algorithm, while the challenges include

1. In large networks, it is time consuming and resource demanding to obtain the exact CSI for each link. Even if the CSI is obtained, it may change rapidly. The CSI updates process will consume a significant amount of resources.

2. The objective of (P1) is in the form of a sum of fractions. Such problems are in general NP-hard [128], and thus cannot be solved with a polynomial complexity using existing optimization methods.

3. Considering the fact that the current wireless transceiver design is typically executed at a timescale of milliseconds, the computationally demanding nature makes real-time implementation highly challenging. Indeed, any change in channel realizations or number of users will lead to a quite different power allocation. Therefore, it would be of great importance to develop an algorithm to solve (P1) within the channel coherence time.

To address the problem of lacking CSI, we model channel fading as a distance-dependent variable. This is generally reasonable since in most cases, the devices' relative positions already capture the main features of the channel. Hence, we can simply perform optimal power control based on geographical location information (GLI). In practice, the GLI can be obtained via current GPS or other positioning approaches with reduced cost. To enable real-time power control, we will introduce a spatial learning method to approximate the mapping from GLI to optimal power control. This way, the online computational cost will be transferred to off-line training. A much faster response can thus be achieved.

4.3 A Successive Pseudo-convex Approximation Approach

The global optimal solution to (P1) can be found with the branch and bound algorithm [129], which has an exponential complexity in terms of the number of variables [104]. The authors in [104] exploit the hidden monotonicity in the objective function to reduce the searching region from the entire feasible set to the problem boundary, but the complexity to

find the global optimal is still exponential. When the number of links is large, this algorithm may not be suitable for real-time operation. Therefore, we resort to a more practical yet also sub-optimal algorithm, which we call the successive pseudo-convex approximation (SPCA) approach [130] in this paper.

The main idea of SPCA is to approximate the objective function of (P1) with functions that have specific properties (e.g., convexity) and then to obtain the solution to the original problem by solving the approximation problem. Specifically, we expand the nonconvex sum rate function in the numerator of (4.6) with a first order Taylor series. Then the expanded function is positive concave. Since the denominator of (4.6) is a linear function, we retain it and do not make any changes. The objective function (P1) is thus approximated by a pseudo concave function (as the ratio of positive concave and linear functions), which can be solved by some iterative algorithms (see Definition 1). Also, this approximation ensures that the original problem (P1) and the approximated problem shares the same sets of stationary points (see Definition 2). Instead of searching for the stationary points of (P1) directly, we search the stationary points of the approximation problem. Pseudo concavity ensures that the resulted stationary points are global optimal for the approximated problems.

4.3.1 A Successive Pseudo Convex Approximation Approach

Definition 1. *A function $f(\mathbf{x})$ is pseudo convex if*

$$\nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) \geq 0 \Rightarrow f(\mathbf{y}) \geq f(\mathbf{x}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}. \tag{4.8}$$

Definition 2. *A point $\mathbf{y} \in \mathcal{X}$ is a stationary point of $f(\mathbf{y})$ if*

$$\nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{4.9}$$

Remark 2. *For convex (concave) optimization, stationary points are global optimal. For nonconvex optimization, stationary points are local optimal. Any stationary point of the pseudo convex optimization is also global optimal [131, Th. 9.3.3].*

The Main Idea

To begin with, we expand the sum rate function (4.3) at a reference point $\mathbf{p} = \mathbf{p}^t$ with a first order Taylor series as:

$$R(\mathbf{p}) \approx \widetilde{R}(\mathbf{p}; \mathbf{p}^t) = \sum_{i \in \mathcal{D}} w_i \widetilde{R}_i(p_i; \mathbf{p}^t), \tag{4.10}$$

where

$$\widetilde{R}_i(p_i; \mathbf{p}^t) \triangleq R_i(p_i; \mathbf{p}^t_{-i}) + (p_i - p_i^t) \underbrace{\sum_{j \in \mathcal{D}, j \neq i} \nabla_{p_i} R_j(\mathbf{p}^t)}_{\text{Price}_i}, \tag{4.11}$$

where $R_i(p_i; \mathbf{p}^t_{-i})$ denotes the rate function of the ith user with its transmitting power to be p_i while all the other powers are fixed to $\mathbf{p}^t_{-i} = \{p_j^t\}_{j \neq i}$, i.e.,

$$R_i(p_i; \mathbf{p}^t_{-i}) = \log \left(1 + \frac{h_{ii} p_i}{\sum_{j \in \mathcal{D}, j \neq i} h_{ij} p_j^t + \sigma_n^2} \right). \tag{4.12}$$

It can be seen that $\nabla_{p_j} \widetilde{R}_i(p_i; \mathbf{p}^t) = 0, \forall i \neq j$. Before proceeding, we show first *how good the approximated function (4.10) will be.*

Proposition 2. $\widetilde{R}(\mathbf{p}; \mathbf{p}^t)$ *is differentiable, both its value and its gradient is the same as that of* $R(\mathbf{p})$ *at* $\mathbf{p} = \mathbf{p}^t$.

Proof. It is obvious that $\tilde{R}(\mathbf{p};\mathbf{p}^t)|_{\mathbf{p}=\mathbf{p}^t} = R(\mathbf{p})|_{\mathbf{p}=\mathbf{p}^t}$. Due to the log function, the resulting first order expansion will also be differentiable. Now let us look at the derivative, which is

$$\nabla_{p_i}\widetilde{R}_i(p_i;\mathbf{p}^t)|_{p_i=p_i^t} = \nabla_{p_i}R_i(p_i;\mathbf{p}^t_{-i})|_{p_i=p_i^t} + \text{Price}_i = \nabla_{p_i}\left[\sum_{j\in\mathcal{D}} R_j(\mathbf{p}^t)\right] = \nabla_{p_i}R(\mathbf{p})|_{\mathbf{p}=\mathbf{p}^t}. \tag{4.13}$$

On the other hand, we have

$$\nabla_{p_i}\widetilde{R}(\mathbf{p};\mathbf{p}^t)|_{\mathbf{p}=\mathbf{p}^t} = \nabla_{p_i}\left[\sum_{j\in\mathcal{D}} \widetilde{R}_j(p_i;\mathbf{p}^t)\right]\Bigg|_{\mathbf{p}=\mathbf{p}^t} = \nabla_{p_i}\widetilde{R}_i(p_i;\mathbf{p}^t)|_{p_i=p_i^t}. \tag{4.14}$$

Comparing (4.13) and (4.14), we conclude that the approximating first order function shares the same gradient at the reference point $\mathbf{p} = \mathbf{p}^t$. □

Proposition 3. $\widetilde{R}(\mathbf{p};\mathbf{p}^t)$ *is concave.*

Proof. Due to the log nature, each $R_i(p_i;\mathbf{p}^t_{-i})$ is concave in p_i since

$$\frac{\partial R_i(p_i;\mathbf{p}^t_{-i})}{\partial p_i} = \frac{c_2}{c_1 + c_2 p_i}$$
$$\frac{\partial^2 R_i(p_i;\mathbf{p}^t_{-i})}{\partial^2 p_i} = -\frac{c_2^2}{(c_1 + c_2 p_i)^2} < 0,$$

where $c_1 = \sum_{j\in\mathcal{D}, j\neq i} h_{ij}p_j^t + \sigma_n^2$ and $c_2 = h_{ii}$. The second part in (4.11) is a linear function in terms of p_i. Hence each $\widetilde{R}_i(p_i;\mathbf{p}^t_{-i})$ is a concave function. $\widetilde{R}(\mathbf{p};\mathbf{p}^t)$ is the sum of finite concave functions, hence it is also concave. □

For the original function $R_i(\mathbf{p})$, it is concave in terms of p_i but non-concave (actually convex) in terms of p_j, for all $j \neq i$. By linearizing the non-concave part $\{R_j(\mathbf{p})\}_{j\in\mathcal{D}, j\neq i}$ w.r.t. p_i at $\mathbf{p}^t$, we obtain a strictly concave function in terms of p_i. Apart from concavity, the approximating function also approximates (P1) well, in terms of both gradient and value.

This way, the EE defined in (4.6) can be approximated by

$$\text{(P2)} \quad \tilde{\eta}_{\text{EE}}(\mathbf{p};\mathbf{p}^t) = \frac{\widetilde{R}(\mathbf{p};\mathbf{p}^t)}{P(\mathbf{p})}. \tag{4.15}$$

Remark 3. *Since the denominator in (4.15) is linear w.r.t.* $\mathbf{p}$*, the approximating function* $\tilde{\eta}_{\text{EE}}(\mathbf{p};\mathbf{p}^t)$ *will approximate the original function* $\eta_{\text{EE}}(\mathbf{p};\mathbf{p}^t)$ *well in terms of both value and gradient. Moreover, since the numerator now becomes concave,* $\eta_{\text{EE}}(\mathbf{p};\mathbf{p}^t)$ *will be pseudo-concave.*

After all the preparation, now we introduce the idea of the SPCA approach here. Our goal is to find an optimal point of (P1) with reduced cost. Due to the non-convexity of (P1), we can only find a sub-optimal solution by searching for the stationary point of (P1) (see remark 2). However, directly searching for the stationary point of (P1) is hard. Instead, we search for the stationary point of the approximating problem (P2). Although our function approximation in (4.15) does not guarantee that (P1) and (P2) shares the same stationary points, we can still use some nice properties of (P2) (see Remark 3) to find the stationary point of (P1). Specifically, (P2) is pseudo-concave, hence its stationary point can be easily found by searching for its maximal point. With the stationary point of (P2) together with the property that (P1) and (P2) have the same gradient and value at this point, we can search for the stationary point of (P1) at a reduced cost. The obtained stationary point will be a possible optimal (may still be sub-optimal) point of (P1).

Algorithm Design

Now we introduce an iterative algorithm to find the stationary point of (P1). As shown in Algorithm 6, we first choose an initial point $\mathbf{p} = \mathbf{p}^0$ and expand the rate function at this point. In Line 3, we search for the point $\mathbf{p}^*$ so that $\mathbf{p}^*$ maximizes the approximated function (P2). Due to pseudo concavity, $\mathbf{p}^*$ is a stationary point of (P2). If $\mathbf{p}^*$ happened to be the stationary point of (P1), the iteration stops. Otherwise, since (P1) and (P2) have the same

Algorithm 6 The Successive Pseudo Convex Optimization Algorithm

1: Initialize $t = 0$ and $\mathbf{p}^0 \in \mathcal{P}$;
2: **repeat**
3: Compute the optimal powel control of the approximating pseudo problem

$$\mathbf{p}^* = \arg\max_{\mathbf{p}\in\mathcal{P}} \quad \tilde{\eta}_{\mathrm{EE}}(\mathbf{p}; \mathbf{p}^t); \tag{4.18}$$

4: Compute the stepsize γ^t ;

$$\gamma^t = \arg\max_{0\leq\gamma\leq 1} \quad \eta_{\mathrm{EE}}(\mathbf{p}^t + \gamma(\mathbf{p}^* - \mathbf{p}^t)); \tag{4.19}$$

5: Update $\mathbf{p}^{t+1}$ by

$$\mathbf{p}^{t+1} = \mathbf{p}^t + \gamma^t(\mathbf{p}^* - \mathbf{p}^t); \tag{4.20}$$

6: $t = t + 1$;
7: **until** $||\mathbf{p}^{t+1} - \mathbf{p}^t||_2 <= \epsilon$

gradient and value at $\mathbf{p} = \mathbf{p}^0$,

$$(\mathbf{p}^* - \mathbf{p}) \cdot \nabla\widetilde{\eta}_{\mathrm{EE}}(\mathbf{p})|_{\mathbf{p}=\mathbf{p}^0} > 0, \tag{4.16}$$

implies that

$$(\mathbf{p}^* - \mathbf{p}) \cdot \nabla\eta_{\mathrm{EE}}(\mathbf{p})|_{\mathbf{p}=\mathbf{p}^0} > 0. \tag{4.17}$$

There must exist a point of (P1) between $\mathbf{p}^*$ and $\mathbf{p}^0$ that maximizes (P1) as shown in Fig. 4.1. Hence in Line 4, we use a step-size to linearize the points between $\mathbf{p}^*$ and $\mathbf{p}^0$ to reduce the searching space. Suppose we find a point $\mathbf{p}^1$ that maximizes (P1). Then we expand (P1) at point $\mathbf{p}^1$ again and continue the process. The sequence $\mathbf{p}^t$ generated by Algorithm 6 keeps on increasing the objective function of (P1). Since problem (P1) is nonempty and bounded, the monotone convergence theorem (MCT) ensures that $\mathbf{p}^t$ converges to a limit point. Due to Line 4, each limit point is a stationary point to (P1). This way, we find an optimal point of (P1), although it may not be the global optimal.

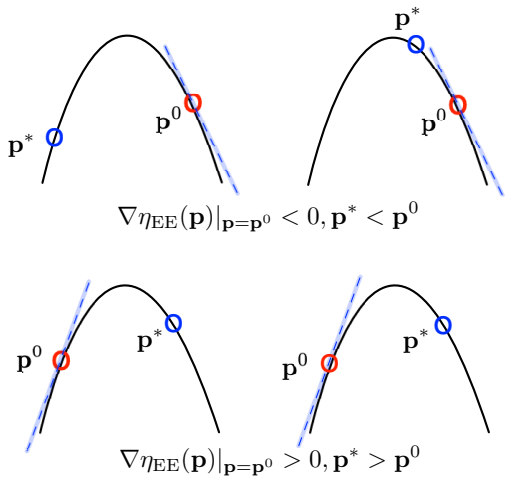


Figure 4.1: Illustration of (4.17).

Algorithm Implementation

The proposed algorithm can find a stationary point of (P1) very quickly and efficiently. To implement Algorithm 6, the main difficulty comes from solving problems (4.18) and (4.19). Problem (4.18) is in the form of fractional programming. Hence, we can use Dinkelbach's algorithm [132] to find the optimal solution in an iterative way. The objective function of problem (4.19) is nonconvex and hence it is non-trivial to solve. One promising solution is to reduce the step-size iteratively and search successively.

The Dinkelbach's algorithm is presented in Algorithm 7, where the function $F(\lambda, \mathbf{p}; \mathbf{p}^t)$ is defined as

$$F(\lambda, \mathbf{p}; \mathbf{p}^t) = \widetilde{R}(\mathbf{p}; \mathbf{p}^t) - \lambda P(\mathbf{p}). \tag{4.21}$$

In each iteration, the algorithm first finds the optimal $\mathbf{p}^*$ that maximizes function $F(\lambda^j, \mathbf{p}; \mathbf{p}^t)$. Then parameter λ^j is updated. It should be pointed out that the Dinkelbach's algorithm is guaranteed to find the optimal solution of problem (4.18) and converges at a super-linear speed. Moreover, problem (4.18) can be solved in parallel as we decompose F to multiple

Algorithm 7 Dinkelbach's Algorithm to Solve (4.18)

1: Initialize λ^0 with $F(\lambda^0) > 0$ and $j = 0$;
2: **repeat**
3: $\quad \mathbf{p}^* = \arg\max_{\mathbf{p}\in\mathcal{P}} \quad F(\lambda^j, \mathbf{p}; \mathbf{p}^t)$;
4: $\quad \lambda^{j+1} = \tilde{\eta}_{\mathrm{EE}}(\mathbf{p}^*; \mathbf{p}^t)$;
5: $\quad j = j + 1$;
6: **until** $|\lambda^j - \lambda^{j+1}| <= \epsilon$

parallel F_i as follows.

$$p_i^* = \arg\max_{p_i\in[0,p_{\max}]} \quad F_i(\lambda, p_i; \mathbf{p}^t), \quad \forall i \in \mathcal{D}, \tag{4.22}$$

where

$$F_i(\lambda, p_i; \mathbf{p}^t) = w_i \widetilde{R}_i(p_i; \mathbf{p}^t) - \lambda(\beta p_i + P_{c,i}), \forall i \in \mathcal{D}. \tag{4.23}$$

Note that each F_i is a concave function. Therefore, its maximum value can be obtained by setting its derivative to 0 or simply by solving a standard convex optimization problem very efficiently. Here we simply set the derivative to 0 to have

$$p_i = \frac{w_i}{\lambda\beta - w_i \mathrm{Price}_i} - \frac{\sum_{j\in\mathcal{D}, j\neq i} h_{ij} p_j^t + \sigma_n^2}{h_{ii}}. \tag{4.24}$$

To satisfy the power constraints, we choose $p_i = \max\{0, \min\{p_i, p_{\max}\}\}$ so that p_i falls into the interval $[0, p_{\max}]$. Note that this algorithm can be extended to an arbitrary power constraint interval $[p_{\min}, p_{\max}]$ simply by setting $p_i = \max\{p_{\min}, \min\{p_i, p_{\max}\}\}$.

For problem (4.19), an exhaustive search for the optimal value of λ is computationally prohibitive. To reduce complexity, we set the stepsize as $\gamma^t = \tau^m$, where m is the smallest nonnegative integer m satisfying the inequality in (4.25).

$$\eta_{\mathrm{EE}}(\mathbf{p}^t + \tau^m(\mathbf{p}^* - \mathbf{p}^t)) \leq \eta_{\mathrm{EE}}(\mathbf{p}^t) + \mu\tau^m \nabla\eta_{\mathrm{EE}}(\mathbf{p}^t)^T(\mathbf{p}^* - \mathbf{p}^t), \tag{4.25}$$

where $\mu \in (0,1)$ and $\tau \in (0,1)$ are two scalars. Note that we do not choose a constant stepsize because if the stepsize is too large, divergence may occur; if the stepsize is too small, the convergence rate may be very slow. In our simulation study, we choose $\mu = 0.01$ and $\tau = 0.5$. This successive searching algorithm helps to balance the tension between complexity and accuracy.

4.4 Deep learning-based Power Control

As can be seen, Algorithm 6 requires iterative loops with complex operations. Specifically, the outer loop refines $\mathbf{p}^t$ iteratively. In each loop, the iterative Dinkelbach's algorithm is applied to solve the sub-problem (4.18) and successive search is used to solve the sub-problem (4.19). These iterations significantly slow down the computational speed, which make it hard for real-time operations.

We aim to develop a real-time system that enables optimized power control with low complexity. Thanks to various advanced machine learning techniques, we can produce a model based on which the target values can be predicted. The essence is to learn a function offline and with the learned function the algorithm can be deployed online. Since the instant precise CSI is generally hard to obtain, we will use the GLI for computing the optimal power controls. Suppose d_{ij} is the distance between the transmitter of the jth link to the receiver of the ith link, we can first model the channel state h_{ij} as a function of d_{ij} and then get the corresponding power control with Algorithm 6 in the off-line stage. Note that Algorithm 6 is quite general. If precise CSI h_{ij} is provided, it computes the optimal power control solution. If only statistical or noise corrupted h_{ij} is available, the corresponding power would be sub-optimal, but still feasible. To enable real-time response in the on-line stage, we aim to find a function that maps $\{d_{ij}\}$ to $\{p_i\}$, given a training set of instance-label pairs $(\{d_{ij}\}, \{p_i\})$ $(i \in \mathcal{D})$.

4.4.1 Deep Neural Network Model

DNN has the ability to learn complex input-output relationships due to the universal approximations [133]. As shown in Fig. 4.2, it is composed of several layers. An input layer forwards the input data to the rest of the network, hidden layers process the input data and finally an output layer applies the final processing. DNN usually has more than one hidden layers. In this paper, we adopt a feedforward NN with fully-connected layers. An input vector $\mathbf{x}_0$ of dimension N_0 is feed to the network through the input layer, which also has N_0 neurons. Then it passes through L hidden layers, where layer l has N_l neurons. Finally, the output layer processes the information that comes from the last hidden layer. The neuron n $(n = 1, 2, ..., N_l)$ in layer l is modeled as

$$\mathbf{x}_l(n) = f_{n,l}(\mathbf{W}_{n,l}^T \mathbf{x}_{l-1} + b_{n,l}), \tag{4.26}$$

where $\mathbf{W}_{n,l} \in \mathbb{R}^{N_{l-1}}$ is the weight vector of the link between all the neurons in layer $l-1$ and the nth neuron in layer l, $b_{n,l}$ is the *bias term* of neuron n in layer l, and $f_{n,l}$ is the *activation function* which provides nonlinearity. The problem reduces to train the weights $\mathbf{W}_{n,l}$ and bias terms $b_{n,l}$ of the NN so that the input-output map of the NN emulates the desired input-output map. In this paper, the Rectified Linear Units (ReLU) function is used in the hidden layers. Additionally, to force the output satisfy constraints (4.7), we adopt the sigmoid function as the output activation function to map the generated power control to the interval $[0, p_{\max}]$. In this paper, we choose a DNN with three hidden layers with 200, 100, and 50 neurons in each layer, respectively.

4.4.2 Proposed PowerNet

Despite that DNN shows a promising performance in function approximation, it has several drawbacks. First of all, the interference pattern of neighboring links depends on the GLI, which is two-dimensional. While the input to a DNN should be one-dimensional. In

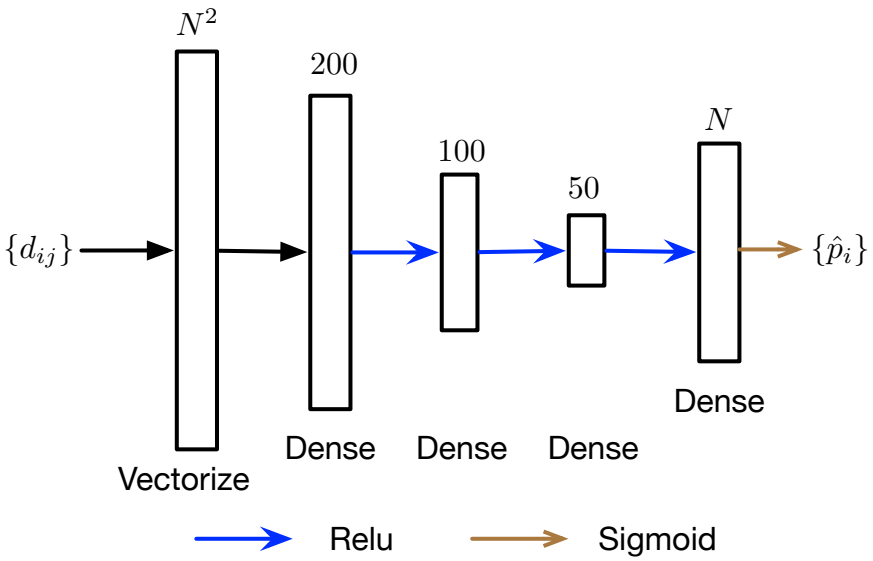


Figure 4.2: The structure of the DNN.

order to process the data, DNN vectorizes the GLI matrix as shown in Fig. 4.2 and then feedforward the data to the following neuron units. For small-sized problems, this operation may work well. However, for a large-sized problem, the vectorization process will inevitably lose some important features, leading to performance degradation. Moreover, when the problem size becomes large, a larger and deeper DNN is needed for sufficient learning power. A fully connected structure may not be efficient and optimal. In some cases, the training process may not converge if the parameters are not set properly. As a result, we will fail to get a proper trained NN.

In this paper, we exploit the popular convolutional neural networks (CNNs) to capture the spatial local correlation by enforcing a local connectivity pattern among the neurons of adjacent layers. The proposed DL architecture, named PowerNet, is presented in Fig. 4.3. As can be seen, the first part of PowerNet is a convolutional layer with two-dimensional GLI as input. The dimension of the convolutional layer is $N \times N \times 2$, where the values $S_1 \times S_2 \times S_3$ denotes the length, width, and the number of feature maps, respectively. We use kernels with dimension 3×3 to generate a feature map. Following the first convolutional layer, the features are fed into two residual learning blocks. Each residual learning block unit consists of three

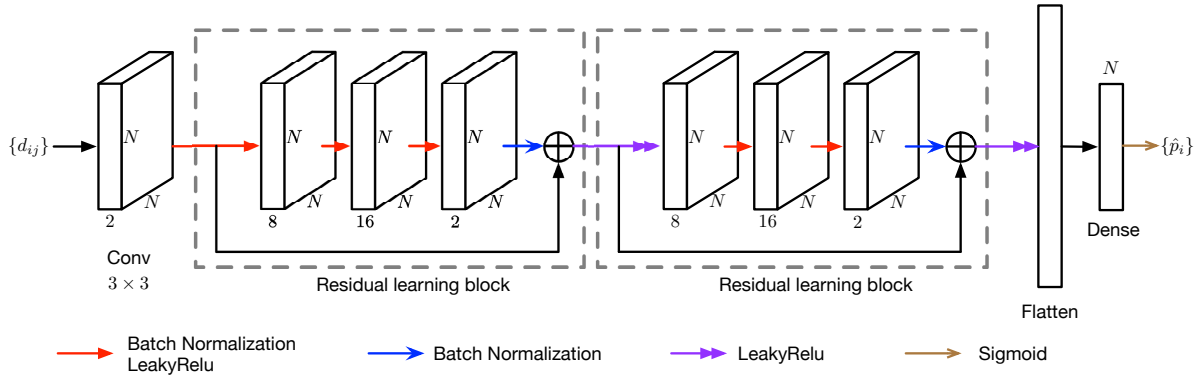


Figure 4.3: The structure of the proposed PowerNet.

layers. In each residual learning block unit, the first layer is the input layer and generates 8 feature maps. The second and the third layer generate 16 and 2 feature maps, respectively. Note that we introduced a shortcut connection between the input layer and the output layer of each residual block. This is inspired by the deep residual network to solve the the vanishing gradient problem caused by multiple stacked non-linear transformations [134,135]. After two such residual learning blocks, we use a flatten layer to connect the output of the residual learning block with the final dense layer. The power control output is generated after the nonlinear mapping in the final dense layer, which adopts the sigmoid activation function. In PowerNet, all kernels used are of dimension 3×3. LeakyRelu and batch normalization are used to provide nonlinearities.

4.5 System Setup

4.5.1 System Parameters

We simulate a square area of 1km×1km. The distance between the transmitter and receiver in a D2D link is uniformly distributed between [5,65] meters as shown in Fig. 4.4. The antenna height of each device is 1.5m. Antenna gain G_a is -2.5dB per device. The noise power spectral density is -174dBm/Hz and the noise figure is 7dB.

We adopt a short-range outdoor channel model ITU-1411 with 5MHz bandwidth at carrier frequency of 2.4GHz. In particular, if the BS antenna height is h_b, the mobile station antenna height is h_m, and the transmission wavelength is λ, then the transmission path-loss from the transmitter of the jth link to the receiver of the ith link (in dB) at distance d_{ij} is

$$L_{ij}[\text{dB}] = L_{\text{bp}} + 6 + \begin{cases} 20\log_{10}\left(\frac{d_{ij}}{R_{\text{bp}}}\right) & \text{if } d \leq R_{\text{bp}} \\ 40\log_{10}\left(\frac{d_{ij}}{R_{\text{bp}}}\right) & \text{if } d > R_{\text{bp}}, \end{cases} \tag{4.27}$$

where $R_{\text{bp}} = 4h_b h_m/\lambda$ denotes the breakpoint distance and $L_{\text{bp}} = |20\log_{10}\left(\frac{\lambda^2}{8\pi h_b h_m}\right)|$ denotes the basic transmission loss at the break point. Based on the choice of large-scale fading and small-scale fast fading, we consider three types of channel models:

1. Path Loss channel model: only the distance related path-loss is considered. The large scale power fading α_{ij} depends on the distance between the Tx and Rx in a D2D link.

2. Shadowing channel model: both the distance related path-loss and the shadowing effect are considered. α_{ij} consists of path-loss and shadowing.

3. Fast Fading channel model: path-loss, shadowing and small-scale fast fading power component are jointly considered. It is more approximated to the real-world fading channel.

The comparison of the three types of channel models are listed in Table 4.2 where $\xi \sim \mathcal{N}(0, \sigma^2)$ denotes the log-normal shadowing with σ being the standard deviation. In Path Loss channel model and Shadowing channel model, the fast fading component g_{ij} is not considered, hence its value is set as 1 while in Fast Fading channel model, g_{ij} is assumed to be exponentially distributed with unit mean. Based on these channel models, the total path-loss versus distance graph is presented in Fig. 4.5. It can be seen that the Path Loss channel model already captures the main trend of the total path-loss. Hence it is possible to perform power allocation simply based on the GLI metric. Comparing Fig. 4.5 with Fig. 4.6,

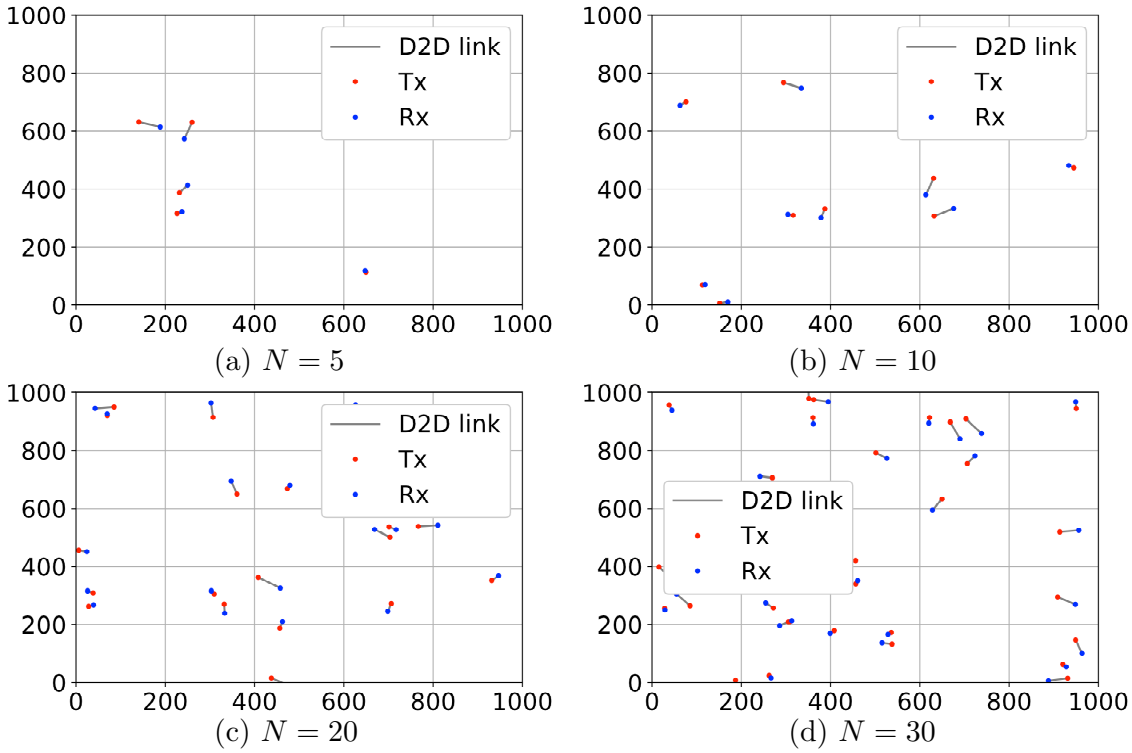


Figure 4.4: The layout of D2D links

we note that a larger standard deviation of the log-normal shadowing results in a larger fluctuation in the total path-loss. Moreover, both fast fading and shadowing cause certain randomness in the practical channel realizations. In this paper, we will generate Path Loss channel model based on the GLI provided and then calculate the optimal power control under the Path Loss channel model. A NN will be trained to learn a mapping from GLI to the optimal power control. The shadowing effect and fast fading effect will be added to investigate the generalization ability of the trained NN.

The static circuit power consumption is set to $P_{c,i} = 10$dBm and the amplifier inefficiency is set as $\beta = 1$. All devices have the same maximum transmit power $p_{\max} = 20$dBm and the weight $w_i = 1$, for all $i \in \mathcal{D}$. The parameters are listed in Table 5.1. The NN is implemented in Keras 2.2.4 with TensorFlow 1.8.0 as backend on a computer with a 3.7GHz

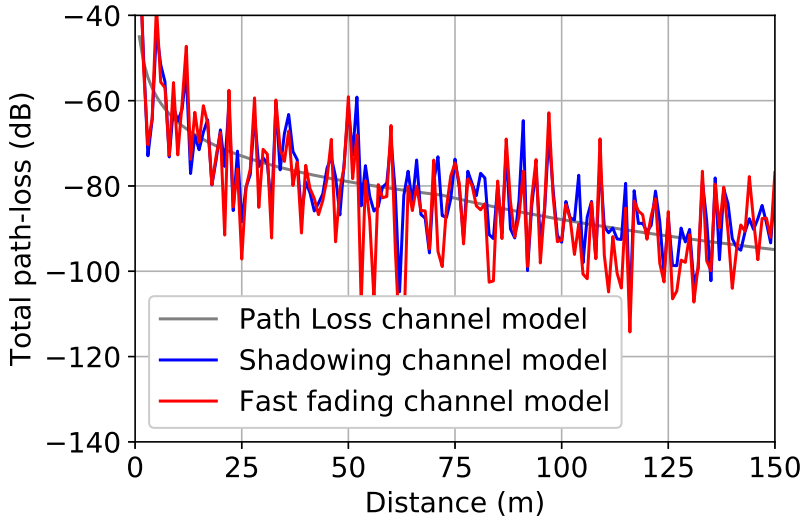


Figure 4.5: The D2D link channel fading model ($\sigma = 8$dB)

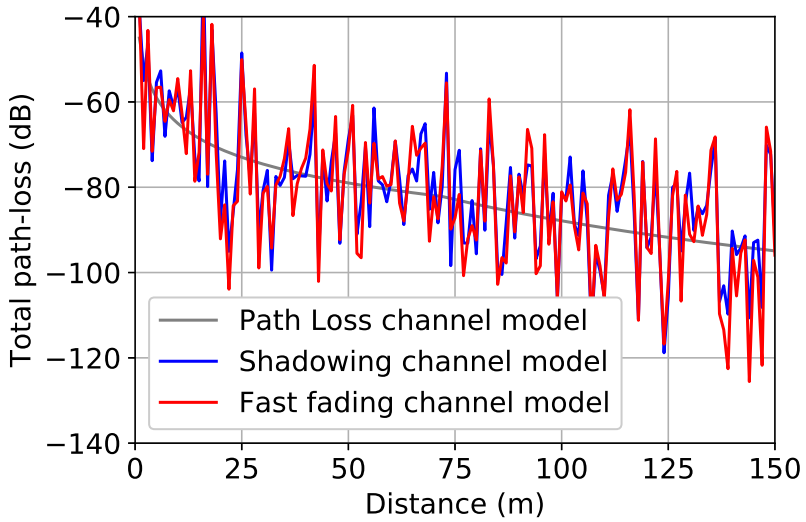


Figure 4.6: The D2D link channel fading model ($\sigma = 12$dB)

Table 4.1: Network Parameter Settings

Cell range	1km×1km
Cell frequency	2.4GHz
Bandwidth	5MHz
Distance	[5,65]m
Maximum transmit power	20 dBm
Noise power spectral density	-174 dBm/Hz
Antenna height	1.5m
Antenna gain per device	-2.5dB
Noise figure	7dB
Circuit static power	10dBm
Amplifier inefficiency	1

Table 4.2: Channel Models for D2D Links

Channel type	Fading component	Channel gain
Path Loss channel model	$\alpha_{ij} = 10^{-(L_{ij}-2G_a)/10}$, $g_{ij} = 1$	$h_{ij} = g_{ij}\alpha_{ij}$
Shadowing channel model	$\alpha_{ij} = 10^{-(L_{ij}+\xi-2G_a)/10}$, $g_{ij} = 1$	
Fast Fading channel model	$\alpha_{ij} = 10^{-(L_{ij}+\xi-2G_a)/10}$, $g_{ij} \sim \text{Exp}(1)$	

i7 Intel Core, one GeForce GTX 1080Ti graphic card, and 32GB memory. The number of training samples and testing samples are set as 250000 and 5000, respectively.

4.5.2 Data Generation

The data is generated in the following manner. First, the channel power gain $\{h_{ij}\}$ are generated following the Path Loss channel model, which only accounts for the impact of distance related path-loss. The corresponding optimized power vector p_i is generated by running the SPCA algorithm. To ensure the scalability of the NN, we normalize the corresponding device distance information d_{ij} as $\bar{d}_{ij} = d_{ij}/(\sqrt{2}R)$, where R is the square side length of the area. We also normalize the output power control as $\bar{p}_i = p_i/p_{\max}$. Then the normalized $\bar{d}_{ij}$ together with $\bar{p}_i$ form one entry of the training dataset. We repeat the process

Table 4.3: Averaged EE (kbps/Joule) for different types of fading channels

N	Methods	Path Loss channel model		Shadowing channel model		Fast Fading channel model	
		EE (kbps/Joule)	Percentage	EE (kbps/Joule)	Percentage	EE (kbps/Joule)	Percentage
5	DNN	0.6683	99.10%	0.6340	96.16%	0.5872	95.38%
	PowerNet	**0.6636**	**98.40%**	**0.6306**	**95.63%**	**0.5849**	**95.01%**
	SPCA	0.6744	100%	0.6594	100%	0.6157	100%
10	DNN	0.5577	95.69%	0.5136	91.10%	0.4796	90.12%
	PowerNet	**0.5518**	**94.69%**	**0.5087**	**90.24%**	**0.4757**	**89.39%**
	SPCA	0.5828	100%	0.5637	100%	0.5322	100%
20	DNN	0.4203	88.90%	0.3775	82.50%	0.3556	81.26%
	PowerNet	**0.4346**	**91.93%**	**0.3862**	**84.40%**	**0.3630**	**82.96%**
	SPCA	0.4728	100%	0.4576	100%	0.4356	100%
30	DNN	0.3378	83.46%	0.2984	75.76%	0.2822	74.27%
	PowerNet	**0.3603**	**89.09%**	**0.3158**	**80.18%**	**0.2973**	**78.22%**
	SPCA	0.4048	100%	0.3939	100%	0.3800	100%

for multiple times to generate the entire training data set. 10% of the training dataset is used for validation in the training process.

4.5.3 Training Process

Suppose for a training input $\bar{d}_{ij}$ and the desired training output $\{p_i\}$, $\{\hat{p}_i\}$ is the corresponding NN output. Then the learning process consists of minimizing the loss function $\mathcal{L} = \mathbb{E}[(p_i - \hat{p}_i)^2]$ We choose a batch size of 100 and the training epoch to be 300. The optimization problem is solved by the ADAM optimizer.

4.5.4 Testing Stage

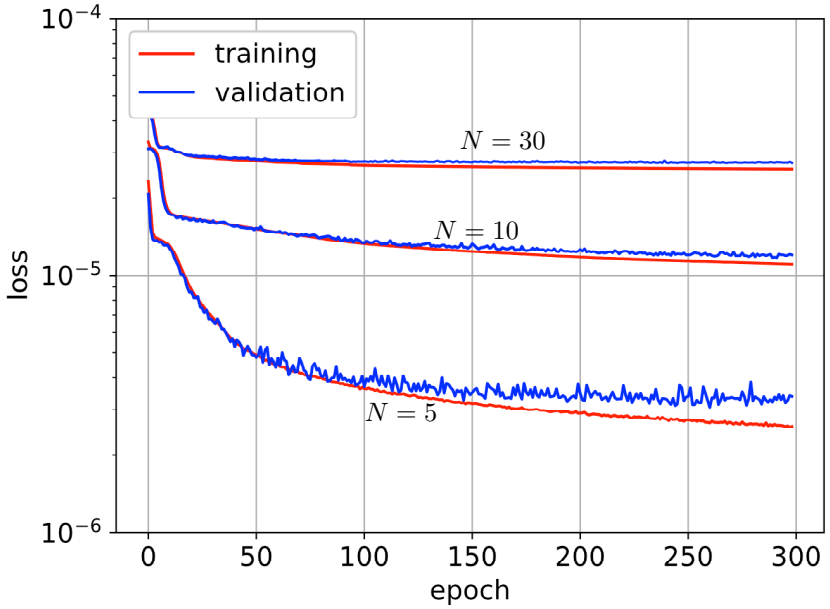
We generate the channels following the same distribution as in the training stage. Then we compute the resulted EE with the solution obtained by SPCA. We will test the robustness and generalization capabilities of the trained NN by generating channels that consider the impact of both shadowing and fast-fading.

4.6 Simulation Results

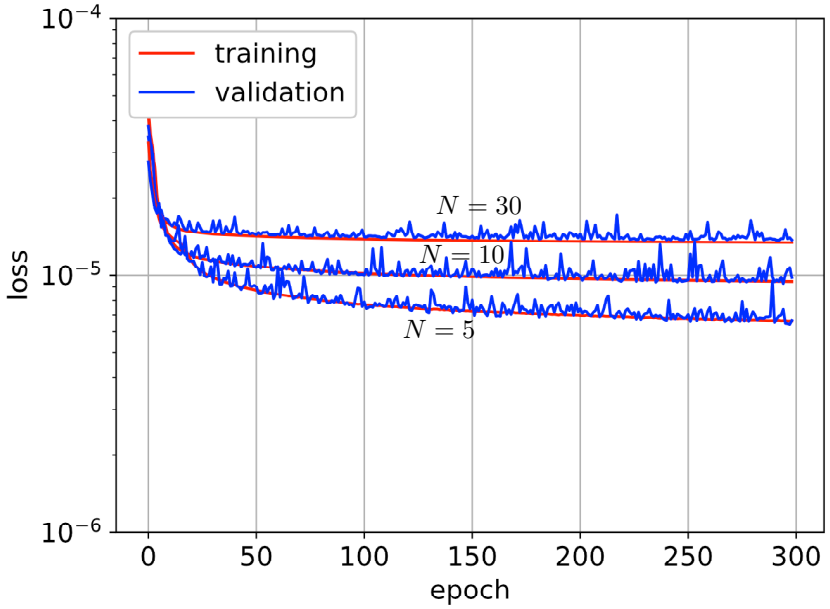
4.6.1 Training Loss and Validation Loss

The training and validation loss for the DNN is presented in Fig. 4.7(a). We change the number of D2D links while fix the number of neurons and network structure of the DNN. It can be seen that for a large-sized problem (e.g., $N = 30$), the training loss decreases at the first few epochs and after approximately 50 epochs, the training loss almost stays at a fixed level. Also, the training loss and validation loss match well, which suggests there is no overfitting problems or under fitting problems. While for a small-sized problem (e.g., $N = 5$), the training loss decreases gradually and it still keeps decreasing after even more than 200 epochs. As training goes on, there is a slight mismatch between the training loss and the validation loss. Hence there exists an under-fitting problem. Moreover, the training loss for a large-sized problem is generally greater than that for a small-sized problem. This is because we use the same DNN structure for problems of all sizes. DNN shows a greater learning ability for small-sized problems hence the corresponding training loss is much lower.

We also plot the corresponding training and validation loss for the proposed PowerNet in Fig. 4.7(b). Training and validation loss decreases rapidly at the beginning of the training. After approximately 200 epochs, the training loss and validation loss almost keeps at a fixed level. Hence, in our simulation, choosing the number of training epochs to be 300 is reasonable for this problem. Different from DNN, for both large-sized problems (e.g., $N = 30$) and small-sized problems (e.g., $N = 5$), the validation loss and the training loss matches very well. There is no overfitting problem here. Moreover, comparing Fig. 4.7(a) with Fig. 4.7(b), we find that for small-sized problems, DNN has a smaller training loss while for large-sized problems, PowerNet has a smaller training loss. This is because kernel maps of dimension 3×3 only works well for a moderate size of problems. If the problem size is too small, either max pooling or average pooling would incur some kinds of distortion. We



(a) DNN (see Fig. 4.2).



(b) PowerNet (see Fig. 4.3).

Figure 4.7: Training and validation loss.

can infer that DNN may perform better in small-sized problems, while the proposed PowerNet may be more suitable for medium-sized or large-sized problems. We will validate our conjecture in Section VI. Finally, we find that neither DNN nor PowerNet has an overfitting

problem. This is because the channel pattern comes from the location of devices (i.e., GLI). There is no unpredictable randomness. If we adopt practical channels as training data, the unpredictable randomness resulted from shadowing and fast fading may incur an overfitting problem.

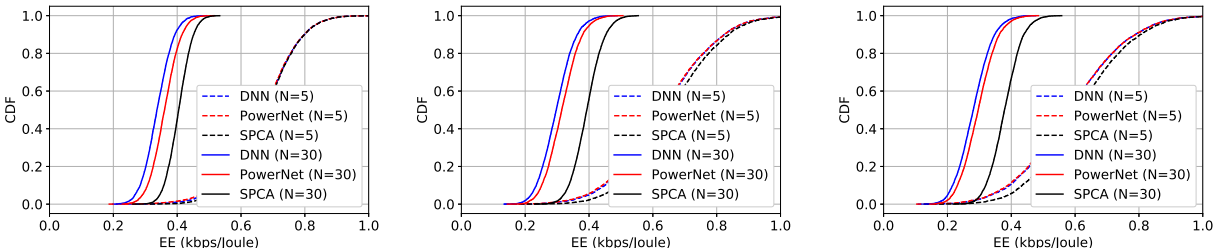
4.6.2 Generalization Performance

In this subsection, we investigate the EE performance of the DNN and PowerNet and test their generalization ability by changing the size of the problem and the type of fading channels.

Averaged EE of the Testing Samples

The averaged EE performance over all testing samples for different types of fading channels is presented in Table 4.3. The baseline method is the SPCA algorithm given in Algorithm 6. First of all, it can be seen that for small-sized problems where $N = 5$, under the Path Loss channel fading model, both the trained DNN and the PowerNet achieve a satisfactory performance. Specifically, DNN achieves 99.10% of the baseline performance and PowerNet achieves 98.40% of the baseline performance. DNN performs slightly better than PowerNet in this case, but their performance gap is almost negligible. As with the increase of the problem size, when $N = 30$, the performance of DNN degrades significantly and only 83.46% of the baseline performance can be achieved. On the other hand, the proposed PowerNet still achieves 89.09% of the baseline performance. Hence, the proposed PowerNet has a stronger generalization ability than the conventional DNN in terms of problem sizes. This is because PowerNet leverages the convolutional layer to better capture the interference patterns and the residual block makes the model more robust.

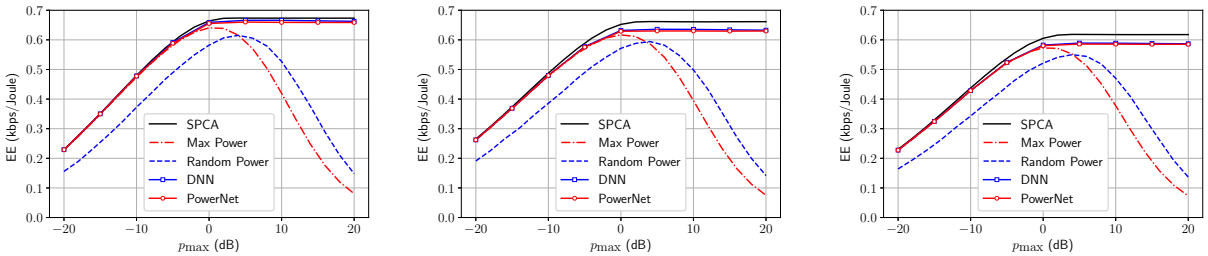
We also compare the achieved EE performance under different channel settings. The adopted NNs are trained with the GLI based on the Path Loss channel model. It can be seen that the performance of both the trained DNN and PowerNet does not degrade too



(a) Path Loss channel model (b) Shadowing channel model (c) Fast fading channel model

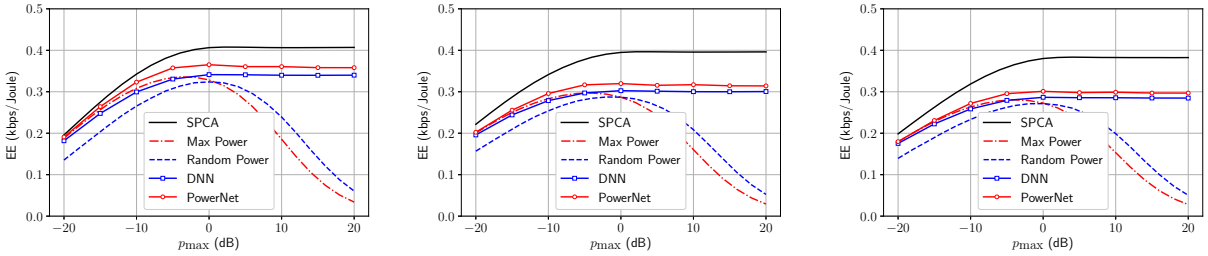
Figure 4.8: Empirical cumulative distribution function (cdf) for different types of fading channels

much when we apply the trained model on a different channel setting. Specifically, when $N = 5$, the trained DNN achieves a 96.16% performance for the Shadowing channel model and a 95.38% performance for the Fast Fading channel model. The proposed PowerNet also achieves a similar performance. Even when the problem size grows larger to $N = 30$, the proposed PowerNet still achieves an 80.18% performance under the Shadowing channel and a 78.22% performance under the Fast Fading channel model. This demonstrates that the distance based Path Loss channel model already captures the main channel characteristics. It is feasible to train the NN with the GLI. This way, the time-consuming channel estimation process can be avoided, which further reduces the response time. This is extremely important for delay sensitive D2D applications, e.g., high-speed vehicle-to-everything (V2X) communication scenarios [136]. When $N = 30$, due to the structure of convolutional layers, the proposed PowerNet outperforms DNN by 5.63% under Path Loss channel model, 4.42% under shadowing channel, and 3.95% under Fast Fading channel model in terms of the achieved averaged EE. Hence it is more suitable for large-sized problems to adopt PowerNet than DNN. In conclusion, the proposed PowerNet exhibits great generalization ability in terms of both problem sizes and channel fading types.



(a) Path loss channel model (b) Shadowing channel model (c) Fast Fading channel model

Figure 4.9: EE performance comparison for different types of fading channels ($N = 5$)



(a) Path Loss channel model (b) Shadowing channel model (c) Fast Fading channel model

Figure 4.10: EE performance comparison for different types of fading channels ($N = 30$)

Cumulative Distribution Function (CDF) of the Testing EE Samples

The empirical cumulative distribution functions (CDF) for different channel fading models are presented in Figs. 4.8. In all the cases, the CDFs of both PowerNet and DNN are obtained by feeding the GLI to a trained NN. For SPCA, the CDF for different types of fading channels comes by running Algorithm 6 with the corresponding channel realization as input.

Fig. 4.8(a) shows the CDF performance under the Path Loss channel model. It can be seen that for both the DNN and PowerNet, the performance gap from the optimal SPCA algorithm is almost negligible when $N = 5$. This shows that both PowerNet and DNN have great learning ability for small-sized problems. When $N = 30$, the performance gap from the optimal increases. However, the proposed PowerNet outperforms the DNN, which again validates that the proposed PowerNet is more suitable for large-sized problems.

When shadowing and fast fading effect are added, the performance gap from SPCA for both the DNN and PowerNet starts to increase, as shown in Fig. 4.8(b) and Fig. 4.8(c). This is because the DNN and PowerNet are trained with GLI, which only captures the distance-based path-loss, while the SPCA utilizes the real-time CSI to perform optimal channel control. Although SPCA achieves a better performance, the complex channel estimation process will waste a lot resources and the iterative nature causes real-time deployment issues. On the other hand, PowerNet and DNN do not rely on real-time CSI, but they still achieve a promising performance. For example, in Fig. 4.8(c), when $N = 5$, the performance gap is less than 0.03 kbps/Joule. Even for large-sized problems ($N = 30$), PowerNet still achieves nearly an average EE that is 80% that of the SPCA and outperforms DNN.

4.6.3 Impact of the Transmit Power Budget

To investigate the impact of the transmit power budget $p_{\max}$, we provide two benchmark algorithms here:

1. Random Power: each device randomly choose a transmitting power that is uniformly distributed between $[0, p_{\max}]$

2. Max Power: each device chooses its maximum power to transmit.

For a small-sized network ($N = 5$), The EE performance comparison for different algorithms is given in Fig. 4.9. It can be seen that when $p_{\max}$ is small, to achieve a high EE, each device is encouraged to transmit data with its maximum power. Hence Max Power transmission is near optimal. DNN, PowerNet and Max Power all share a similar performance as the optimal benchmark algorithm, SPCA. When $p_{\max}$ is large enough, the resulted power control will always satisfy the power budget constraints. Hence the optimal power control does not change any more. The EE will stay at a fixed level. However, in this case, Max Power transmission will cause severe interference to other links and the EE will decrease dramatically. As a comparison, Random Power transmission performs slightly better than

Max Power transmission, but the performance is still not satisfactory. On the contrary, the deep learning-based method can achieve a near optimal performance. In Fig. 4.9(a), the performance gap from optimal in the Path Loss channel model is almost zero. Even when shadowing and fast fading are added, the performance gap from SPCA is also small, as shown in Fig. 4.9(b) and Fig. 4.9(c). This experiment shows that deep learning-based method is quite suitable for small-sized problems. Compared with SPCA, the deep learning-based method does not require any instant CSI. The response time is greatly saved.

We also give a similar plot for a large-sized network ($N = 30$) in Fig. 4.10. Comparing Fig. 4.9 with Fig. 4.10, we find that with the increase of network sizes, the deep learning-based method shows a performance degradation. This is because we fix the number of neurons and the number of training samples to be the same. The same NN structure is used for all problem sizes. When $N = 5$, the input dimension is 25. When $N = 30$, the input dimension increases to 900. DNN exhibits great capability in learning such small-scale input-output relationship. Sometimes increasing the number of hidden layers or increasing the number of neurons in each hidden layer may help to improve the learning ability to a certain extent. However, this is not always true. In our experiment, when we adopt a DNN with 3 hidden layers and 200 neurons in each layer, the training loss failed to decrease when applied the training data with $N = 30$. In other words, if the parameters are not set properly, we may fail to train a DNN. On the contrary, due to the adoption of the convolutional layer and deep residual learning, the proposed PowerNet do not have such training problems. We also note that, for a large-sized problem, PowerNet always outperforms DNN regardless of the value of the power budget $p_{\max}$ and the channel fading types. This experiment again demonstrates the superiority of the proposed PowerNet.

4.6.4 Complexity Comparison

- *Computational Analysis.* Since both the DNN and PowerNet is trained based on the SPCA, their performance will not exceed that of the SPCA. The reason why we want to

adopt PowerNet is that the deep learning-based method has a lower online complexity and it is more suitable for real-time deployment.

The complexity of the NN based methods comes from two parts: the off-line training stage and the on-line computation stage. The off-line training stage complexity mainly comes from the training dataset generation. It does not have any impact on the algorithm's real-time on-line operations. In this paper, we generate the training dataset with the measurement of distance based large-scale fading. They can be obtained by existing channel modeling methods as well as ray-tracing approaches. Compared with other works that collect training data from the instant CSI, our methods significantly simplifies the training data preparation process. The on-line complexity comes from the linear combination of layer input and activation function operations, which is almost negligible.

As a comparison, the existing SPCA algorithms require one outer iterations and two inner iterations. Suppose the outer loop has a_1 iterations. Two inner loops which are used to solve (4.18) and (4.19) has b_1 and b_2 iterations, respectively. Then the total iterations will be $a_1(b_1 + b_2)$. Although the Dinkelbach's algorithm converges quickly, in each iteration a gradient has to be computed. In the successive step-size search loop, the step-size is narrowed down until a satisfactory result is found. In large networks, such computations will significantly slow down the real-time response.

- *Experiment Verification.* For a fair comparison, we write the algorithms in python and run them under the parameter setting introduced in Section 4.5.1. We present the computation time comparison in Table 4.4. First of all, look at the CPU time. For DNN, the average running time is almost 1000 times faster than SPCA under different problem sizes. PowerNet is 150 times faster than SPCA when $N = 5$ and 16 times faster when $N = 30$. Deep learning-based approach works fast due to the simple neuron network computations. In contrast, the iterations in SPCA significantly

slow down the algorithm. The reason why PowerNet performs a bit slower than DNN is that PowerNet performs convolutional operations and it is deeper than the DNN. However, the speed difference is not significant in small-sized problems.

Actually, for a large-sized problem, the running time of the NN can be further reduced if a GPU is enabled. This is because the implementation of NN is highly amenable for parallel processing. The benefit of the parallel computation power of GPU can be fully exploited. By running all the algorithms on GPU, the gap between DNN and PowerNet narrows down significantly in large-sized problems. For example, when $N = 30$, DNN is only 4.56 times faster than the proposed PowerNet with GPU (as a comparison, the DNN is 55 times faster than PowerNet when the CPU is used). PowerNet is almost 88 times faster than SPCA (as a comparison, it is 16 times faster than SPCA when the CPU is used). Hence, with the deployment of GPU, PowerNet achieves a significant saving in running time compared to SPCA and DNN. Moreover, SPCA requires considerable extra time to perform channel estimation, while PowerNet and DNN only use GLI which is much easier to obtain.

- *NN Size Comparison* When it comes to algorithm deployment, the NN size is also an important issue. when $N = 30$, the total trainable parameters for PowerNet is 57,378, while the total number of parameters for DNN is 206,880, which is about 3.6 times of that of PowerNet. Hence, the size of the proposed PowerNet is much smaller than that of the conventional DNN. PowerNet is more suitable to be deployed on devices with memory constraints. Considering that in many cases, the number of D2D links may change with time. Hence, pre-trained PowerNet models with different configuration N should be stored in the memory. Each time, when the configuration N is changed, the corresponding trained PowerNet must be restored. Hence NN size should also be taken good care of to ensure a promising generalization ability in terms of the number of D2D links.

Table 4.4: Computational Time Comparison

N	Methods	CPU time (ms)	percentage	GPU time (ms)	percentage
5	DNN	0.022	0.17%	0.025	0.33%
	PowerNet	0.091	0.69%	0.109	1.44%
	SPCA	13.268	100%	7.546	100%
10	DNN	0.008	0.07%	0.026	0.32%
	PowerNet	0.160	1.43%	0.107	1.33%
	SPCA	11.159	100%	8.066	100%
20	DNN	0.012	0.09%	0.026	0.27%
	PowerNet	0.483	3.75%	0.115	1.20%
	SPCA	12.887	100%	9.549	100%
30	DNN	0.019	0.11%	0.029	0.25%
	PowerNet	1.107	6.10%	0.131	1.14%
	SPCA	18.161	100%	11.541	100%

4.7 Conclusions

In this chapter, we explored the possibility of using deep learning method to accelerate the resource allocation decision process. The idea is to use a DNN to mimic the functionality of conventional optimization methods. In this way, the online computational cost is transferred to off-line training. Simulation results demonstrated that the proposed PowerNet could achieve a near-optimal EE performance at a much faster speed compared with the conventional optimization method. Moreover, different from conventional optimization algorithms, which require the knowledge of precise CSI, the proposed PowerNet could perform power control based on GLI which can be obtained by current positioning system. This way, the channel estimation process could be saved and the developed approach would be extremely suitable for the scenarios where devices change their location rapidly.

Chapter 5

Channel Block Resource Allocation In Downlink RAN With Deep Reinforcement Learning

5.1 Introduction

The envisioned applications of the 5G communication technologies, such as video streaming, virtual reality, and 360 degree videos, are all imposing stringent delay requirements [137]. Radio access network (RAN) has become a key enabling technology to address this issue. In RAN, radio resource management (RRM) [138, 139] plays a vital role in allocating the resources to user equipments (UEs) by jointly exploiting the advanced MAC layer functions, such as resource sharing, link adaptation, hybrid automatic retransmission request (HARQ), and channel quality indicator (CQI) reporting.

In conventional RAN, a packet scheduler is deployed at the base station (BS) and it is responsible for allocating the wireless spectrum resources to UEs based on their quality of service (QoS) requirements as well as their reported channel conditions. In each Transmission Time Interval (TTI), the packet scheduler needs to solve a decision-making problem to decide how the resources are allocated to the UEs. Due to the large number of UEs and the dynamic wireless environment, it is difficult to obtain an optimal solution with the existing optimization methods within the channel coherence time. As a result, most of the existing approaches rely on some predefined rules. For example, conventional packet scheduling strategies, such as the max-CQI scheduling approach and proportional fairness (PF) scheduling, are all rule-based scheduling policies which simply assign the resource block (RB) to the UEs' that have the best channel condition or the best relative channel condition. The problem is these methods are designed based on human's understanding of a network. Although simple and useful, it is hard to guarantee they are optimal and accurate.

In todays' highly complex wireless environment, variety of applications have emerged, for which rule-based scheduling methods have their limitations. For example, from a spectrum efficiency point view, max-CQI scheduling, which allocate radio resources to the UEs that is expected to have the best achieved rate, would be optimal. However, this is at a cost of sacrificing other important performance metrics, such as fairness, energy-saving, and complexity. In some delay-sensitive scenarios such as vehicle-to-vehicle (V2V) communications [136], ensuring timely delivery of packets is more important than bringing an additional increase to the system throughput. As UEs' QoS requirements may change, developing an intelligent scheduling method that can adaptively select the best scheduling policy is of vital importance.

Recently, deep reinforcement learning has made breakthroughs the field of networking and communications [32, 126, 140–143]. In time-varying and unpredictable networks, DRL has proved to be effective in tackling real-time decision-making problems. For example, the authors in [144] developed a decentralized resource allocation mechanism for V2V communications based on DRL. The vehicles can make decisions to find the optimal sub-channel allocation as well as the power level for transmission without global information. Vehicles can learn to satisfy their QoS requirement while also minimizing the interference to other vehicles. In [145], an online DRL-based algorithm is developed to optimally adapt task offloading decisions and wireless resource allocations to the channel conditions. Instead of solving the large combinatorial problem directly, the proposed agent learns the binary decisions from past experience. This method can achieve near optimal performance while significantly decreasing the computational time. In [140], the joint beam forming, power control, and interference coordination problem in a downlink multiple access OFDM cellular network is formulated as a combinatorial problem. The authors show that closed-form expression does not exist and finding the optimal solution requires an exhaustive search. The developed method leverages the power of DRL to avoid the exhaustive search and achieves a near-optimal performance. In our recent works, we have applied DRL to solve the resource

allocation problem at a wireless backhaul [143] and to develop a smart congestion control scheme [126].

In the field of cellular traffic scheduling, Ref. [146] investigates how DRL can help solve scheduling problems in cellular networks. It shows that by exploiting the expert knowledge of existing rule-based scheduling method in the training process of the DRL agent, the DRL's learning ability can get improved. In [147], a DRL-packet scheduler is developed to adapt to the dynamic scheduling conditions. Instead of using a single scheduling rule across the entire transmission, Ref. [147] proposes a framework to dynamically select the best scheduling rule at each TTI based on UEs' QoS requirements. Simulations show that the developed method outperforms a conventional scheduling method in terms of delay and package drop rate requirements.

It is envisioned that the application of AI/ML techniques to the design of 6G systems will be fundamental to improve the system performance [148]. In this chapter, we incorporate DRL to address the problem of delay-aware packet scheduling in the downlink of a cellular network. We show that delay-aware packet scheduling is a complex combinatorial problem, which is challenging to solve. By modeling the packet scheduling problem as a Markov decision process (MDP) problem, a deep Q-learning agent that is based on a recurrent neural network (RNN) is designed to learn a delay optimized scheduling solution through the interactions with the environment. Simulation results show that the DRL-based packet scheduler, designed to minimize the queueing delay of all UEs, outperforms several existing scheduling schemes. Meanwhile, the framework in this chapter can be easily generalized to other scenarios with various QoS requirements.

The chapter is organized as follows. In Section 5.2, the system model is introduced and the delay-aware scheduling problem is formulated. In Section 5.3, a DRL-based packet scheduling algorithm is proposed. Simulation results are presented in Section 5.4 to validate the superiority of the proposed method. Finally, Section 5.5 concludes this chapter.

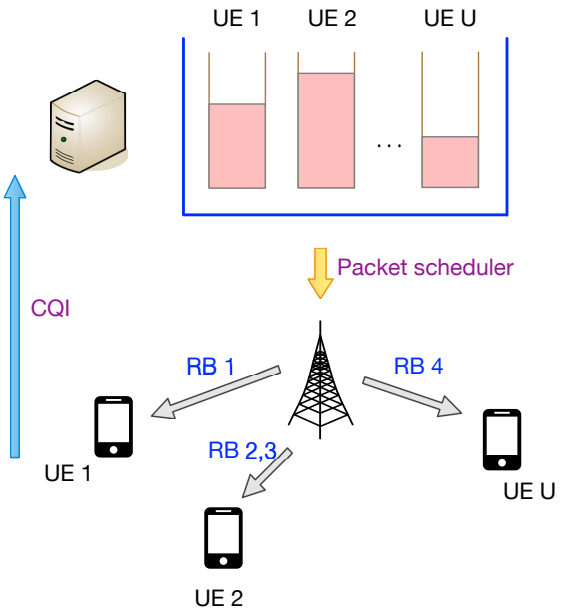


Figure 5.1: Delay-aware cellular downlink traffic scheduling system model.

5.2 System Model and Problem Statement

5.2.1 System Model

We consider the downlink transmissions of a cellular network, where UEs are served by a base station (BS). As shown in Fig. 5.1, multiple UEs request packets from the BS and the BS performs traffic scheduling with UE-specific queues. We consider an orthogonal frequency division multiplexing (OFDM) system, where the available wireless resources in time and frequency are divided into resource blocks (RBs). The RB scheduling is performed at each time slot (i.e., TTI).

Let $\mathcal{U} = \{1, 2, ..., U\}$ denote the set of active UEs and $\mathcal{B} = \{1, 2, ..., B\}$ the set of RBs, where U and B are the total number of UEs and RBs, respectively. We aim to develop a packet scheduler to allocate the set of RBs to the UEs so that the average delay of the UEs can be minimized. At a time slot t, we denote the maximum number of bits that could be sent for an RB $b \in \mathcal{B}$ to a UE $u \in \mathcal{U}$ as $C_{ub}[t]$. According to [149], the value of $C_{ub}[t]$ depends on the channel quality indicator (CQI) reported by UE u. Based on the CQI, a

proper modulation and coding scheme (MCS) is assigned to the allocated RB. In practice, there exists a mapping table between $C_{ub}[t]$ and the MCS.

5.2.2 Traffic Model

We consider the case where the UEs are characterized by data request pattern. Suppose that at time slot T, the requested data size of UE u is $A_u[t]$. The BS will assign RBs to UEs and transmit the requested data to the corresponding UEs. However, the wireless resources are limited. Not all the UEs can get allocated RBs immediately. The BS maintains a separate queue for each UE. The requested packets of the UE will be queued in the buffer. At time slot t, the queue length of UE u is denoted as $Q_u[t]$. The buffer state of a UE can be written as

$$Q_u[t] = \max(\min(Q_u[t-1] + A_u[t], Q_{\max}) - D_u[t], 0), \tag{5.1}$$

where $Q_{\max}$ is the maximum buffer size of all UEs, $D_u[t]$ is the scheduled, transmitted data for UE u in time slot t, which can be computed as

$$D_u[t] = \sum_{b=1}^{B} x_{ub}[t] \cdot C_{ub}[t], \tag{5.2}$$

where $x_{ub}[t]$ is the RB allocation indicator: $x_{ub}[t] = 1$ if UE u is assigned with RB b at time slot t; and $x_{ub}[t] = 0$ otherwise.

5.2.3 Traffic Delay

Packets of UEs are time stamped and queued in the buffers for transmission based on the first-in-first-out principle. For each packet, the difference between the current time and the arrival time, which we call the head of line (HoL) time, is used to measure the packet delay. At time slot t, the HoL packet delay of UE u is denoted by $d_u[t]$. During a time period

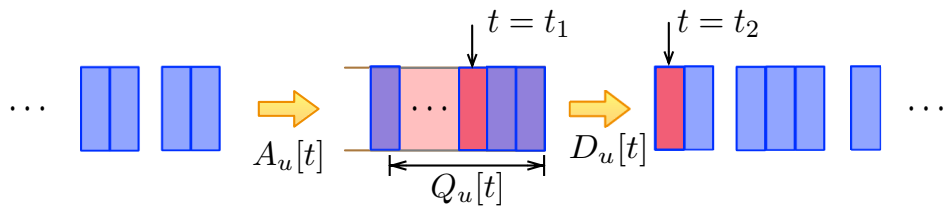


Figure 5.2: The traffic model considered in this chapter.

T, the average delay of UE u can be computed as

$$W_u = \frac{1}{T} \sum_{t=1}^{T} d_u[t] \tag{5.3}$$

As shown in Fig. 5.2, at a time t, the HoL time $d_u[t]$ can be computed by definition $d_u[t] = t_2 - t_1$, where t_1 is the time when the packet first enters the queue and t_2 is the time when the packet leaves the queue.

5.2.4 Problem Formulation

Our objective is to jointly optimize the RB allocation in the wireless links so that UE's average delay can be minimized. We formulate the problem as follows.

$$\min_{x_{ub}[t]} \quad \Gamma_{\text{Delay}} = \frac{1}{U} \sum_{u=1}^{U} W_u \tag{5.4}$$

$$\text{s.t.} \quad x_{ub}[t] = \{0, 1\}, \ \forall u, b, t \tag{5.5}$$

$$\sum_{u}^{U} x_{ub}[t] \leq 1, \ \forall b, \tag{5.6}$$

where constraint (5.5) means that the RB assignment variables are binary, and constraint (5.6) ensures that each RB can only be assigned to one UE. To solve Problem (5.4) is to find the best RB scheduling policy at each time slot for all UEs and RBs. This problem is difficult for the following reasons: (i) Constraints (5.5) and (5.6) makes the problem combinatorial; (ii) The objective function does not have a closed-form expression in terms of the scheduling policy $x_{ub}[t]$. A direct optimization may become hard; (iii) the number of RBs and the number

of UEs may be very large, which makes the optimization problem more challenging. In some delay-sensitive applications such as vehicle communications, a fast and efficient algorithm is needed [136, 144].

5.3 Resource Block Allocation with Deep Reinforcement learning

A direct optimization of problem (5.4) is hard. In this section, we show that problem (5.4) exhibits Markov decision process (MDP) property, based on which, a deep reinforcement learning (DRL) based packet scheduling policy is proposed.

5.3.1 MDP Problem

As in (5.1), the queue length of UE u at time slot t depends on the queue length at time slot $t-1$, the requested data $A_u[t]$, and the scheduled traffic $D_u[t]$. The BS observes the queue length of each UE. Based on the requested packet size of different UEs and their corresponding channel conditions, the BS allocates RBs to the UEs so that their averaged delay can be minimized. Mathematically, at time slot t, the decision of computing the current scheduled traffic for UE u is a function of the current channel condition (i.e., the maximum amount of traffic carried in an RB), the previous queue status, the current HoL delay, and the current request data size of all UEs, i.e.,

$$D_u[t] = f(C_{ub}[t], Q_u[t-1], d_u[t], A_u[t]). \tag{5.7}$$

In addition, as analyzed in Section 5.2, the current HoL packet delay depends on the queue status as well as the history data request rates $A_u[t]$ and data traffic $D_u[t]$, $t = 1, 2, ...$ Therefore, the buffer state (5.1) can be modeled as an MDP. MDP is a discrete time stochastic control process. It provides a mathematical framework for modeling the decision making where the outcomes are partly random and partly depending on the policy that is made by the controller. As shown in Fig. 5.3, the packet scheduler, which we call *agent*, observes the

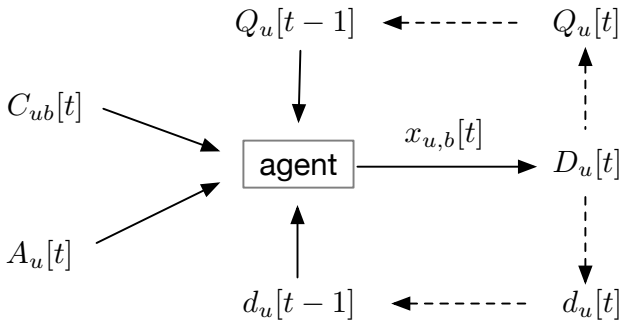


Figure 5.3: A Markov decision process (MDP) problem.

current state $\{C_{ub}[t], Q_u[t-1], d_u[t], A_u[t]\}$, and then makes decision on how each RBs are assigned to UEs. Then the traffic is scheduled, which in turn updates the queue length as well as the HoL packet delay for each UE. At the next time slot, the agent makes decision again, and so forth. The ultimate goal is to find a stable "policy" so that the averaged HoL delay can be minimized.

5.3.2 DRL-based Delay-aware Packet Scheduler

We next present the DRL-based RB scheduling algorithm. At each time step t, the agent performs a certain action a_t based on the current state s_t. The agent receives a reward and moves to the next state.

The action is defined as the choice of $x_{ub}[t]$. Note that we have U UEs and B RBs in total; so the dimension of the action space at each time slot would be $\mathcal{O}(2^{U\times B})$. We map the choice of $x_{ub}[t]$ to a real integer number that is between $[0, U^B - 1]$. Each integer number in the interval corresponds to a unique RB allocation to the UEs. We use the interval to denote the action space $\mathcal{A}$, i.e.,

$$\mathcal{A} = [0, U^B - 1]. \tag{5.8}$$

In practical systems, the number of RBs is huge. A direct use of RB would bring an extremely large action space. In OFDM, frequency selective wireless channels are transferred into multiple flat channels over different sub-bands. consecutive sub-bands can be grouped

together, which is called resource block group (RBG) in LTE [139]. This way, the action space can be greatly reduced and the algorithm can converge faster.

The state space include UE's packet arrival rate, buffer state, the transmission rate of different RBs, which is denoted by

$$\mathcal{S}[t] = \{S_1[t], S_2[t], ..., S_U[t]\}, \tag{5.9}$$

where $S_u[t]$ is the observed state of UE u, given by

$$\mathcal{S}_u[t] = \{A_u[t], Q_u[t-1], d_u[t-1], C_{ub}[t]\}. \tag{5.10}$$

As a result, the dimension of the state space is $UB + 3U$.

What makes DRL appealing is we can customize the reward function for a specific problem. In this chapter, we directly use the negative value of the objective function (5.4) as the reward function. Maximizing the reward function will be equivalent to minimizing the average delay of all UEs. Our developed method can be quite general and flexible, as the reward function can be extended by jointly consider the effect of throughput, QoS requirements, latency, and priority. We leave this topic for our future investigation.

We propose Algorithm 8 to solve Problem (5.4), which is a DRL-based approach. The algorithm perform RB scheduling so that the UEs' average delay can be minimized. An RNN is incorporated, which takes state as input and outputs a Q-value for each of the candidate actions. The main steps include

- Select an action at time slot t
- Reward the action based on the computed time delay
- Train the RNN based on the outcome

We call the period of time in which an interaction between the agent and the environment takes place as an *episode*. In the beginning of each episode, the environment initializes the

Algorithm 8 Delay-optimal cellular traffic scheduling algorithm with DRL

if training **then**
  Start environment simulator, generating arrival traffic and the status of the RBs ;
  Initialize the time, states, action and replay buffer $\mathcal{D}$;
  for each episode **do**
    for each time slot t **do**
      Observe state $S[t]$ as in (5.9) ;
      $\epsilon = \max(\epsilon \cdot d, \epsilon_{\min})$;
      Sample $r \sim \mathcal{N}(0,1)$;
      if $r \leq \epsilon$ **then**
        Select an action $A[t] \in \mathcal{A}$ randomly ;
      else
        Select an action such that $A[t] = \arg\max_{a'} Q_\pi(S[t], a'; \theta[t])$;
      end if
      Compute reward as $r[t] = -\frac{1}{U}\sum_{u=1}^{U} W_u$;
      Observe the next state S' ;
      Store the experience $(S[t], A[t], S', r[t])$ in $\mathcal{D}$;
      Minibatch sample from $\mathcal{D}$, $e_j \triangleq (S_j, A_j, r_j, S'_j)$;
      Set $y_j := r_j + \gamma \cdot \max_{a'} Q_\pi(s_{j+1}, a'; \theta[t])$;
      Perform gradient optimization method on $(y_j - Q_\pi(s_{j+1}, a'; \theta[t]))$ and obtain the optimal θ^* ;
      $\theta_t = \theta^*$;
      $t = t + 1$;
      $S[t] = S'$;
    end for
  end for
  Save θ_t and the agent ;
else
  Load the agent ;
  Based on the observed state, output the action ;
end if

state and then the agent interacts with the environment for several several training steps (or TTIs) during this episode. We perform several episodes until the accumulated reward converges.

5.4 Simulation Results and Discussions

5.4.1 Parameter Setting

We consider the case where two UEs request data from the BS. The data request pattern of UE 1 follows the Poisson process with parameter $\lambda = 8$, i.e.,

$$P(A_1[t] = k) = \frac{e^{-\lambda}\lambda^k}{k!},\ k = 0, 1, 2, ... \tag{5.11}$$

The data request pattern of UE 2 follows the uniform distribution, where

$$P(A_2[t] = k) = \frac{1}{5},\ \forall k \in \{6, 7, 8, 9, 10\}. \tag{5.12}$$

The maximum queue length $Q_{\max}$ is set as 100. We assume that the quality of each RB can be measured at each TTI, based on which the maximum number of bits that can be transmitted on the RB can be computed. In this simulation, we assume that the maximum number of bits that can be transmitted on the RB for a specific UE can only take discrete values from a set $\{2, 3, 4, 5\}$ uniformly. Although we use discrete values here, it is worth noting that the proposed algorithm also applies to continuous value settings and various distributions of the UE data requests.

We create a DQN agent with RNN. The structure of the critic network is shown in Fig. 5.4. There are three layers between the input and the output: two dense layers and one long-short-term memory (LSTM) layer. Each layer has 50 neurons. The training parameters are listed in Table 5.1. The number of episodes is 200. In each episode, the number of steps is 300. In the offline training phase, the value of the requested data size $A_u[t]$ and the channel capacity $C_{ub}[t]$ are generated according to a known distribution in each step. In the testing phase, we generate 1000 simulations and average the results.

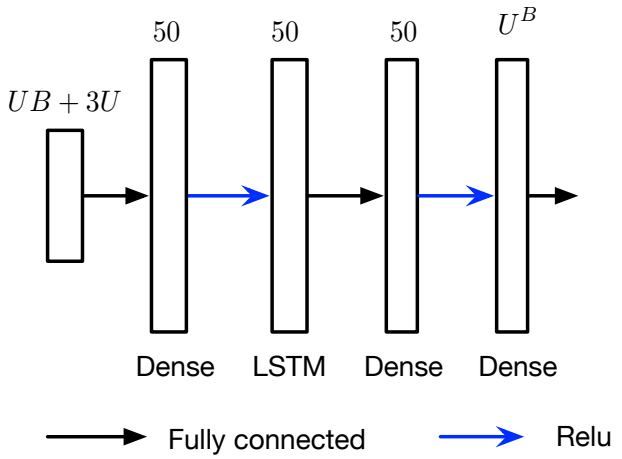


Figure 5.4: The recurrent neural network used in the agent.

5.4.2 Benchmark Algorithms

Round-robin

The round robin allocation strategy simply allocates all RBs to each user in turn. It is one of the simplest algorithms. Regardless of the channel condition and the traffic requirement, Round robin assumes all the UEs have equal priority, while different UEs may have different channel quality on the same radio resource. No optimization is performed. It is evident that round-robin scheduling will incur a poor performance.

Max-CQI

The Max-CQI scheduler only considers the channel condition while allocating RBs to the UEs. The Max CQI aims to maximize the system's capacity by allocating RBs to the UEs that have the best channel condition, i.e., at each time slot t, $x_{u^*,b}[t] = 1$ where

$$u^* = \arg\max_u \{C_{ub}[t]\}. \tag{5.13}$$

The max-CQI packet scheduler can maximize the transmitted traffic since the RBs are allocated greedily to the UEs that can achieve the highest transmission rate. However, the

Table 5.1: The DRL Hyperparameters

Parameter	*Value*
Number of episodes	200
Number of steps per episode	300
Discount factor	0.99
Experience replay length	1000,000
ϵ decay d	0.9999
Initial exploration rate ϵ	1
Maximum batch-training trajectory (RNN)	20

UEs with low channel quality would have little chance to get the transmission resources. As a result, the average delay would be bad. Extremely, a UE may never get RB allocated.

Proportional Fairness (PF)

Proportional fairness aims to maximize network capacity while also ensure fairness. The idea is to balance the average past throughput and the expected rate. In our simulations, we choose $x_{u^*,b}[t] = 1$ where

$$u^* = \arg\max_u \left\{ \frac{C_{ub}[t]}{Y_u}[t-1] \right\}, \tag{5.14}$$

where $Y_u[t]$ is the average scheduled traffic of UE u in period $[0, t-1]$ and $Y_u[t] = \frac{1}{t}\sum_{i=1}^{t} D_u[i]$. It is easy that if a UE does not get RBs for a long time, the value of $Y_u[t]$ will be low. Therefore, there will be a high chance that the UE gets RB allocated in the next time slot. Some prior works use a time window to average the allocated traffic.

5.4.3 Performance Metrics

The aim of this chapter is to minimize the packet delay for all UEs. Therefore, the performance metric will be the averaged delay, which is calculated as in (5.4). Meanwhile,

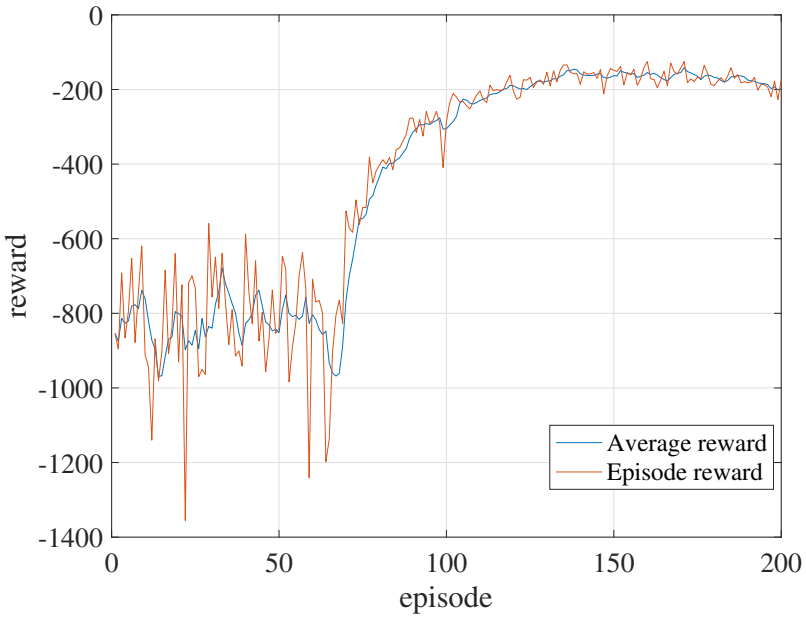


Figure 5.5: Training reward vs. episode for the proposed algorithm.

we also compare the scheduled traffic (i.e., throughput) of the UEs, which is computed as

$$\Gamma_{\text{throughput}} = \frac{1}{UT} \sum_{u=1}^{U} \sum_{t=1}^{T} D_u[t], \tag{5.15}$$

and the average queue length, which is defined as

$$\Gamma_{\text{QueueLength}} = \frac{1}{UT} \sum_{u=1}^{U} \sum_{t=1}^{T} Q_u[t]. \tag{5.16}$$

5.4.4 Results and Discussions

Convergence

The convergence plot of the training process is presented in Fig. 5.5. Both the average result of all the episode as well as the instant per episode reward are plotted. It can be seen that after around 100-episode training, the reward value begins to converge. Furthermore, the maximum reward is attained at around the training episode 150. The optimal number of training episodes can be set to be around 150 to achieve the best performance.

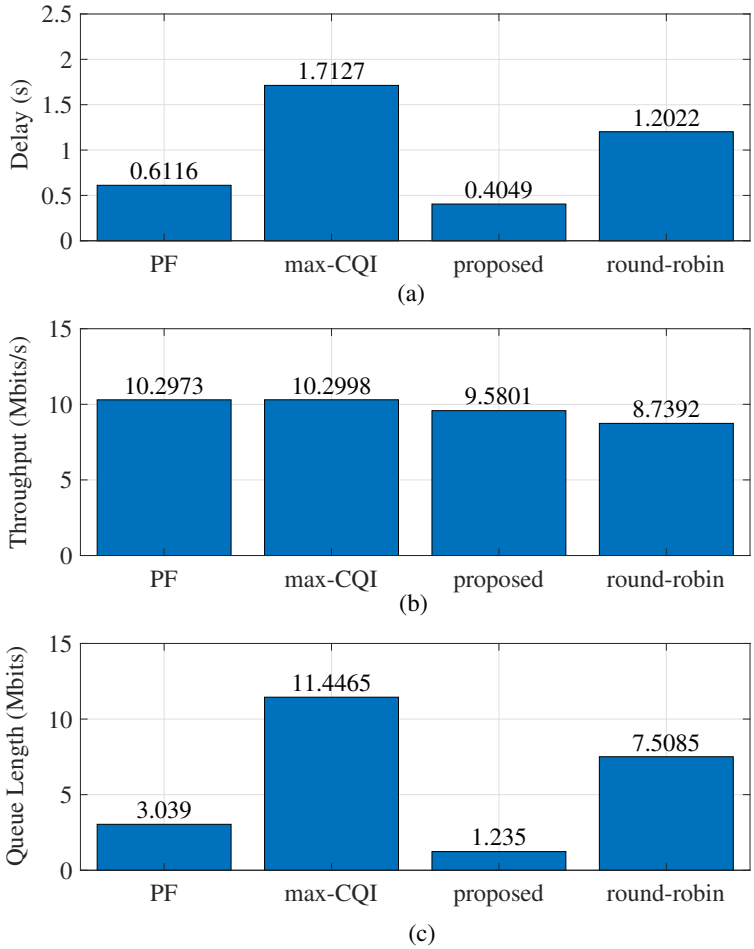


Figure 5.6: Performance comparison of four scheduling algorithms.

Delay

The delay performance comparison is shown in Fig. 5.6(a). The proposed DRL-based scheduling algorithm achieves the lowest delay, which is 0.4049s. Max-CQI, which aims to maximize the traffic throughput, has the largest delay, which is nearly 3.5 times larger than that of the proposed algorithm. By taking fairness into consideration, PF avoids the situation where UEs with bad channel condition do not get RB allocated. Therefore, the average queue delay can be reduced. Despite that, the proposed DRL scheduling algorithm achieves the best performance in terms of packet delay.

Throughput

The achieved throughput comparison is shown in Fig. 5.6(b). This performance metric is defined as the sum of the transmitted packets over a time period and reflects the system throughput. As analyzed before, max-CQI achieves the highest system throughput. The performance of PF is quite close to that of max-CQI. The proposed method also achieves a quite high throughput. Round-robin has the poorest throughput performance.

Queue Length

The average queue length is also an essential metric when we compare the delay performance. In addition, when the queue length is large, there is a high chance that the packet may be dropped, which will bring a high packet drop rate (PDR). As a result, the UE will request the packet again and the delay performance would be even worse. The average queue length is depicted in Fig. 5.6(c). Our proposed method has the lowest average queue length. As a comparison, the max-CQI method has the longest queue length, which is almost 9 times longer than our proposed method. Round-robin, which serves the UEs in turn at different time slots, can ensure that each UE get served "equally." However, it does not consider the channel condition diversity for different UEs and does not fully exploit the resources. As a result, the queue length performance of round-robin is also poor.

5.5 Conclusions

In this chapter, we aimed to minimize the average packet delay of a downlink multi-access OFDM cellular network, where the UEs have different packet request patterns and channel conditions. We developed a DRL-based packet scheduling method. We showed that our proposed method can achieve the lowest delay and the shortest average queue length. The proposed method has a great potential for delay-sensitive applications, such as vehicle-to-vehicle (V2V) communication in the 5G era. Moreover, the reward function can be

customized by jointly considering the dynamic traffic load, QoS parameters, and application requirements.

Chapter 6

Summary and Future directions

6.1 Summary

In this dissertation, we have studied resource allocation in next generation wireless networks. Specifically, we focus on two representative technologies in 6G: 1) IRS technology and 2) machine learning. This dissertation can be broadly organized into two parts under the theme of resource allocation. The first part (chapter 2 and chapter 3) studied resource allocation in IRS-assisted wireless networks with optimization methods. The convergence and effectiveness of the proposed approach are demonstrated via both simulations and analysis. The second part (chapter 4 and chapter 5) investigated resource allocation problems with machine learning approaches. It can be seen that conventional optimization methods still show advantages in the optimization of emerging network types, such as IRS. However, they generally require complex iterations and may be sub-optimal. Meanwhile, machine learning shows great potential in shifting the time-consuming computation to the off-line training, and tackling problems which is hard to model.

Based on the examples presented in the previous chapters, we proposed a workflow to deal with general resource allocation problems in figure 6.1. As can be seen, depending on the specific purpose, network type and performance criteria, a resource allocation problem can be formulated first. Based on the property of the problem, optimization method and machine learning approach can be used. In the following, we summarize the main contributions and discuss possible directions for our future work.

In Chapter 1, we presented a big picture of the evolvement of wireless networks over decades. The key technologies in 6G including IRS and machine learning are introduced. We then presented the basics of resource allocation, challenges and approaches.

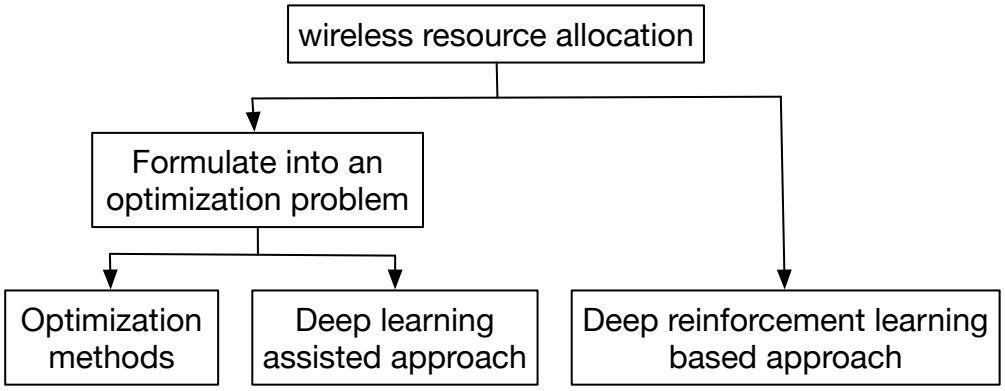


Figure 6.1: A workflow to address general wireless resource allocation problems

In Chapter 2, we studied the beamforming design problem in an IRS-assisted rate splitting system. We developed a BCD algorithm to jointly optimize the active beamforming at the BS as well as the passive beamforming at the IRS. For the first time, we considered the impact of imperfect CSI in the IRS-assisted RS system, which is well handled by our developed algorithm. We first analyzed and explained the performance gain when RS meets IRS. The robustness of RS in combating channel uncertainties in IRS-assisted wireless network has been demonstrated via simulations.

In Chapter 3, we studied the resource allocation problem in an IRS-assisted federated learning system. The IRS phase shift is optimized under both the single user case and the multiple user case. Our results show that the properly configured IRS can significantly save the total energy consumption, especially when the number of reflecting elements on the IRS is large. For future work, we consider incorporating the AirComp framework into our design and develop more efficient schemes for IRS-assisted federated learning.

In Chapter 4, we studied the power allocation problem in wireless networks with interfering links. Classic optimization methods have been developed to solve this problem. Motivated by the strong learning power of residual networks and the fast operating speed of deep learning, we developed a deep learning based power control. Moreover, we incorporate

the GLI information into our design and use convolutional neural network to capture the interference pattern of the interfering links. Our developed deep learning approach shows great generalizability and can achieve a near-optimal performance at a remarkably high speed.

In Chapter 5, we studied the channel resource block allocation problem with DRL approach. Different from the previous examples, in this example the formulated problem does not exhibit closed-form expressions in terms of the optimization variables, hence only heuristic algorithms can be used. We incorporate DRL into the design of packet scheduling. Moreover, we deployed our developed algorithm in a mmwave testbed. The performance of machine learning approach is verified via both simulations and real-world test.

6.2 Future Directions

The research on IRS and machine learning has just emerged. Both would be key candidate 6G communication technologies. There are still a lot of open problems in this area. Here we briefly list some future directions.

6.2.1 IRS Optimization with Machine Learning

In our dissertation, the design and optimization of IRS-assisted networks still relies on conventional optimization methods, which requires complex iterations. A good direction is to use machine learning in the design of IRS-assisted networks. In IRS-assisted network, it is challenging to obtain precise CSI. Deep learning approach, has the potential to map the received pilots to the direct and cascaded channels. As a result, the channel estimation performance can be improved. In terms of the optimization of the reflecting elements on the IRS, we can view it as a decision making process. The IRS learns how to reflect the incident signals in the best possible way with DRL. Compared with optimization method, this approach would no longer need precise system model and could be more adaptive in the dynamic wireless environment.

6.2.2 Training Data Acquisition and Real-world Deployment

The machine learning approaches faced many challenges in the training process since they require collecting a large number of high quality data along with the network profiles. For example, the supervised learning approaches require the accurate labeled data, which may be hard to obtain in the rapidly changing environment. Other approaches rely on simulated dataset, which is generated based on a specific type of network. Hence there may be an inconsistency between the developed method and the ideal method that works on a practical system. In practice, most of the data is obtained via the measurement of the wireless state or the local states reported by the devices. As future network becomes more and more dense, the communication cost and measurement cost will be a big challenge.

6.2.3 New Applications

Virtual reality (VR) and augmented reality (AR) have been regarded as one of the key applications in 6G. They are expected to revolutionize the interactions between humans and the perceived world by providing a highly immersive virtual world experience. As a transformative service, VR/AR has been used in a wide range of applications, ranging from personal entertainment, social interactions, automotive video streaming, to remote control and tactile internet. The stringent delay requirement (less than 20 ms) and large bandwidth consumption (350 Mbits/s) makes it challenging in optimizing the communication, computation and storage resources. We believe that the potential of IRS and machine learning in supporting these applications has not been fully unleashed. Efficient frameworks with edge intelligence are needed to support future new applications, such as wireless VR/AR.

6.2.4 Secure Communication

In the era of internet of things (IoT), machine and devices are connected for real-world sensing and interaction. Users expect that their devices are secure from attack. Many private information such as health-care data, bank account information should be protected

well from network security threats such as eavesdropping and jamming. The secure communications bring about new challenges for the resource allocation in next generation wireless networks. For example, current resource allocation may be invalid when the network faces potential threats from the eavesdroppers and pseudo base stations. A security-oriented resource allocation is needed to ensure the safe communication for next generation wireless networks.

Appendices

Appendix A

Proof for chapter 2

A.1 Proof of Theorem 1

Note that Problem (2.14) without constraint (2.14f) is convex. Therefore the Lagrangian function in terms of the active beamforming $\mathbf{W}_i (i \in \mathcal{M})$ can be expressed as

$$\begin{aligned}\mathcal{L} = \mathcal{T} + \alpha \left(P_{\max} - \sum_{k \in \mathcal{M}} \mathrm{Tr}\left(\mathbf{W}_k\right) \right) + \sum_{k \in \mathcal{K}} \beta_k \left(\log_2 \left(\sum_{i \in \mathcal{K}} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) + \sigma_0^2 \right) \right. \\ \left. - \frac{\Gamma_{\mathrm{p},k}(\mathbf{W}^t)}{\log(2)} \sum_{i \in \mathcal{K}, i \neq k} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) \right) + \sum_{k \in \mathcal{K}} \gamma_k \left(\log_2 \left(\sum_{i \in \mathcal{M}} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) + \sigma_0^2 \right) \right. \\ \left. - \frac{\Gamma_{\mathrm{c},k}(\mathbf{W}^t)}{\log(2)} \sum_{i \in \mathcal{K}} \mathrm{Tr}\left(\mathbf{h}_k \mathbf{h}_k^H \mathbf{W}_i\right) \right) + \sum_{i \in \mathcal{M}} \mathrm{Tr}(\mathbf{W}_i \mathbf{Y}_i),\end{aligned}$$

where α, β_k, γ_k, and $\mathbf{Y}_i$ are the Lagrange multipliers associated with the corresponding constraint (2.14c), (2.14d), (2.14e) and (2.14g), respectively, and $\mathcal{T}$ is the term irrelevant to $\mathbf{W}_i$. The KKT conditions for the optimal $\mathbf{W}_i^* (i \in \mathcal{M})$ can be expressed as

$$\begin{aligned}&\alpha^*, \beta_k^*, \lambda_k^* \geq 0, \quad k \in \mathcal{K} \\ &\mathbf{Y}_i^* \succ \mathbf{0}, \quad i \in \mathcal{M} \\ &\mathbf{Y}_i^* \mathbf{W}_i^* = \mathbf{0}, \quad i \in \mathcal{M} \\ &\nabla_{\mathbf{W}_i^*} \mathcal{L} = 0, \quad i \in \mathcal{M},\end{aligned}$$

where α^*, β_k^*, γ_k^*, and $\mathbf{Y}_i^*$ are the optimal Lagrange multipliers and $\nabla_{\mathbf{W}_i^*}\mathcal{L}$ is the gradient of $\mathcal{L}$ with respect to $\mathbf{W}_i^*$. In this case, for the private message, we have

$$\nabla_{\mathbf{W}_k^*}\mathcal{L} = -\alpha^*\mathbf{I} + \mathbf{Y}_k^* + \beta_k \frac{\Gamma_{\mathrm{p},k}(\mathbf{W}^*)}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H + \sum_{k\in\mathcal{K}} \gamma_k \frac{\Gamma_{\mathrm{c},k}(\mathbf{W}^*)}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H - \sum_{k\in\mathcal{K}} \gamma_k \frac{\Gamma_{\mathrm{c},k}(\mathbf{W}^t)}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H, k \in \mathcal{K}.$$

Note that when the algorithm converges, the value of $\mathbf{W}$ in the tth iteration will be close to the optimal value $\mathbf{W}^*$, and hence $\Gamma_{\mathrm{c},k}(\mathbf{W}^*) = \Gamma_{\mathrm{c},k}(\mathbf{W}^t)$. In this case, we have

$$\nabla_{\mathbf{W}_k^*}\mathcal{L} = -\alpha^*\mathbf{I} + \mathbf{Y}_k^* + \beta_k \frac{\Gamma_{\mathrm{p},k}(\mathbf{W}^*)}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H,\ k \in \mathcal{K}. \tag{A.1}$$

For the common part, we have

$$\nabla_{\mathbf{W}_\mathrm{c}^*}\mathcal{L} = -\alpha^*\mathbf{I} + \mathbf{Y}_\mathrm{c}^* + \sum_{k\in\mathcal{K}} \gamma_k \frac{\Gamma_{\mathrm{c},k}(\mathbf{W}^*)}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{I}. \tag{A.2}$$

Multiplying both sides of (A.1) by $\mathbf{W}_k^*$ and multiplying both sides of (A.2) by $\mathbf{W}_\mathrm{c}^*$, we have

$$\begin{aligned}
&-\alpha^*\mathbf{W}_k^* + \mathbf{Y}_k^*\mathbf{W}_k^* + \beta_k \frac{\Gamma_{\mathrm{p},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_k^* = \mathbf{0} \\
&-\alpha^*\mathbf{W}_\mathrm{c}^* + \mathbf{Y}_\mathrm{c}^*\mathbf{W}_\mathrm{c}^* + \sum_{k\in\mathcal{K}} \gamma_k \frac{\Gamma_{\mathrm{c},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_\mathrm{c}^* = \mathbf{0}.
\end{aligned}$$

Note that $\mathbf{Y}_i^*\mathbf{W}_i^* = \mathbf{0}, i \in \mathcal{M}$. Hjjence we have

$$\begin{aligned}
\alpha^*\mathbf{W}_k^* &= \beta_k \frac{\Gamma_{\mathrm{p},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_k^* \\
\alpha^*\mathbf{W}_\mathrm{c}^* &= \sum_{k\in\mathcal{K}} \gamma_k \frac{\Gamma_{\mathrm{c},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_\mathrm{c}^*.
\end{aligned}$$

With the basic rank inequalities, for the private covariance matrix $\mathbf{W}_k^*, k \in \mathcal{K}$, we have

$$\mathrm{rank}(\mathbf{W}_k^*) = \mathrm{rank}(\alpha^*\mathbf{W}_k^*) = \mathrm{rank}\left(\beta_k \frac{\Gamma_{\mathrm{p},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_k^*\right) \le \mathrm{rank}(\mathbf{h}_k\mathbf{h}_k^H) = 1. \tag{A.3}$$

However, for the common covariance matrix $\mathbf{W}_{\mathrm{c}}^*$, we have

$$\begin{aligned}\operatorname{rank}(\mathbf{W}_{\mathrm{c}}^*) = \operatorname{rank}(\alpha^*\mathbf{W}_{\mathrm{c}}^*) = \ & \operatorname{rank}\left(\sum_{k\in\mathcal{K}}\gamma_k\frac{\Gamma_{\mathrm{c},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\mathbf{W}_{\mathrm{c}}^*\right)\\ \leq \operatorname{rank}&\left(\sum_{k\in\mathcal{K}}\gamma_k\frac{\Gamma_{\mathrm{c},k}(\mathbf{W})}{\log(2)}\mathbf{h}_k\mathbf{h}_k^H\right) \leq \min\{M,K\}.\end{aligned}$$

A.2 Proof of Theorem 2

Suppose $(\mathbf{w}^*, \mathbf{G}^*, \mathbf{U}^*, s^*, \mathbf{R}_{\mathrm{c}}^*, \mathbf{R}_{\mathrm{p}}^*, \mathbf{v}^*)$ are the stationary solutions of problem (2.24), then $(\mathbf{w}^*, \mathbf{G}^*, \mathbf{U}^*, s^*, \mathbf{R}_{\mathrm{c}}^*, \mathbf{R}_{\mathrm{p}}^*, \mathbf{v}^*)$ satisfy the KKT condition of problem (2.24). As a result, we have

$$\frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{p},k}^g\nabla_{\mathbf{G}}\xi_{\mathrm{p},k} + \frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{c},k}^g\nabla_{\mathbf{G}}\xi_{\mathrm{c},k} = 0 \tag{A.4}$$

$$\frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{p},k}^u\nabla_{\mathbf{U}}\xi_{\mathrm{p},k} + \frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{c},k}^u\nabla_{\mathbf{U}}\xi_{\mathrm{c},k} = 0 \tag{A.5}$$

$$\frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{p},k}^w\nabla_{\mathbf{w}}\xi_{\mathrm{p},k} + \frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{c},k}^w\nabla_{\mathbf{w}}\xi_{\mathrm{c},k} + 2\sum_{k=1}^{K}\lambda_k^w\mathbf{w}_k = 0 \tag{A.6}$$

$$\frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{p},k}^v\nabla_{\mathbf{v}}\xi_{\mathrm{p},k} + \frac{1}{\ln 2}\sum_{k=1}^{K}\lambda_{\mathrm{c},k}^v\nabla_{\mathbf{v}}\xi_{\mathrm{c},k} + 2\lambda^v\mathbf{v} = 0 \tag{A.7}$$

where $\lambda_{\mathrm{p},k}^g$, $\lambda_{\mathrm{p},k}^u$, $\lambda_{\mathrm{p},k}^w$, $\lambda_{\mathrm{p},k}^v$ are nonnegative Lagrange multipliers associated with constraint (2.24c) and $\lambda_{\mathrm{c},k}^g$, $\lambda_{\mathrm{c},k}^u$, $\lambda_{\mathrm{c},k}^w$, $\lambda_{\mathrm{c},k}^v$ are nonnegative Lagrange multipliers associated with constraint (2.24b). Note suppose $(\mathbf{G}^*, \mathbf{U}^*)$ satisfy condition (2.18) and (2.22), then $g_{\mathrm{p},k} = g_{\mathrm{p},k}^{\mathrm{MMSE}}, g_{\mathrm{c},k} = g_{\mathrm{c},k}^{\mathrm{MMSE}}$, $u_{\mathrm{p},k} = u_{\mathrm{p},k}^{\mathrm{MMSE}}$ and $u_{\mathrm{c},k} = u_{\mathrm{c},k}^{\mathrm{MMSE}}$. In this case, we always have

$$\begin{aligned}\frac{\partial\xi_{\mathrm{p},k}}{\partial g_{\mathrm{p},k}} &= u_{\mathrm{p},k}\frac{\partial\epsilon_{\mathrm{p},k}}{\partial g_{\mathrm{p},k}} = 0\\ \frac{\partial\xi_{\mathrm{c},k}}{\partial g_{\mathrm{c},k}} &= u_{\mathrm{c},k}\frac{\partial\epsilon_{\mathrm{c},k}}{\partial g_{\mathrm{c},k}} = 0\end{aligned} \tag{A.8}$$

Hence, (A.8) is in line with (A.4)., which means $g_{\mathrm{p},k}^{\mathrm{MMSE}}$ and $g_{\mathrm{c},k}^{\mathrm{MMSE}}$ satisfy the KKT condition (A.4). Similarly, $u_{\mathrm{p},k}^{\mathrm{MMSE}}$ and $u_{\mathrm{c},k}^{\mathrm{MMSE}}$ satisfy the KKT condition (A.5), hence they are

stationary solutions of problem (2.24). When $g_{\mathrm{p},k} = g_{\mathrm{p},k}^{\mathrm{MMSE}}, g_{\mathrm{c},k} = g_{\mathrm{c},k}^{\mathrm{MMSE}}$, $u_{\mathrm{p},k} = u_{\mathrm{p},k}^{\mathrm{MMSE}}$ and $u_{\mathrm{c},k} = u_{\mathrm{c},k}^{\mathrm{MMSE}}$, we got the relationship (2.23). In this case, (A.6) and (A.7) is transformed to

$$-\sum_{k=1}^{K} \lambda_{\mathrm{p},k}^{w} \nabla_{\mathbf{w}} R_{\mathrm{p},k} - \sum_{k=1}^{K} \lambda_{\mathrm{c},k}^{w} \nabla_{\mathbf{w}} r_{\mathrm{c},k} + 2\sum_{k=1}^{K} \lambda_{k}^{w} \mathbf{w}_k = 0 \tag{A.9}$$

$$-\sum_{k=1}^{K} \lambda_{\mathrm{p},k}^{v} \nabla_{\mathbf{v}} R_{\mathrm{p},k} - \sum_{k=1}^{K} \lambda_{\mathrm{c},k}^{v} \nabla_{\mathbf{v}} r_{\mathrm{c},k} + 2\lambda^{v} \mathbf{v} = 0 \tag{A.10}$$

which are the KKT condition for problem (2.9). Also, the KKT condition is the same for problem (2.9) and problem (2.24) in terms of variable $s, R_{\mathrm{p}}, R_{\mathrm{c}}$. Hence, $(\mathbf{w}^*, s^*, \mathbf{R}_{\mathrm{c}}^*, \mathbf{R}_{\mathrm{p}}^*, \mathbf{v}^*)$ are stationary solutions for problem (2.9).

A.3 Proof of Theorem 3

Denote the objective function in (2.24) as $F(\mathbf{G}, \mathbf{U}, \mathbf{w}, \mathbf{v})$. Then we have

$$F(\mathbf{G}^{t-1}, \mathbf{U}^{t-1}, \mathbf{w}^{t}, \mathbf{v}^{t}) \leq F(\mathbf{G}^{t}, \mathbf{U}^{t}, \mathbf{w}^{t}, \mathbf{v}^{t}) \leq F(\mathbf{G}^{t}, \mathbf{U}^{t}, \mathbf{w}^{t+1}, \mathbf{v}^{t}) \leq F(\mathbf{G}^{t}, \mathbf{U}^{t}, \mathbf{w}^{t+1}, \mathbf{v}^{t+1}),$$

where the first inequality holds since $\mathbf{U}^t$ is the optimal weight and $\mathbf{G}$ is the optimal MMSE equalizer to minimize the MSE. The second inequality holds since $\mathbf{w}^{t+1}$ is the optimal solution to Problem (2.25) and the third inequality holds since $\mathbf{v}^{t+1}$ is the optimal solution to Problem (2.42).

Appendix B

Proof for chapter 3

B.1 Proof of Theorem 4

The objective function E_k in (3.12) is an increasing function in terms of f_k. The time constraint (C1) of Problem (P1) suggests that the IoT device should work on the lowest frequency f_k^* that is allowed by the delay constraint.

B.2 Proof of Theorem 5

If $p_{c,k} - A_k > 0$, then $p_k + p_{c,k} - A_k > 0$. Minimizing the energy consumption E_k is equivalent to maximizing the function $g(p_k) = \frac{R_k}{p_{c,k}+p_k-A_k}$. For simplicity of notation, we rewrite $g(p_k)$ as

$$g(p_k) = \frac{b_k}{\ln(2)} \frac{\ln(1 + a_k p_k/b_k)}{p_{c,k} + p_k - A_k}, \tag{B.1}$$

where $a_k = \frac{|\mathbf{w}_k^H \mathbf{h}_k|^2}{N_0 |\mathbf{w}_k^H|^2} > 0$. Then we have

$$g'(p_k) = \frac{b^k}{\ln(2)} \frac{\frac{a_k}{b_k + a_k p_k}(p_{c,k} + p_k - A_k) - \ln(1 + a_k p_k/b_k)}{(p_{c,k} + p_k - A_k)^2}. \tag{B.2}$$

Let the numerator be denote by $h(p_k)$ and we have

$$h'(p_k) = \frac{-a_k^2}{(b_k + a_k p_k)^2}(p_{c,k} + p_k - A_k) < 0, \tag{B.3}$$

which means $h(p_k)$ is a decreasing function on $[0, P_{\max}]$. Also note that $h(0) = a_k(p_{c,k} - A_k)/b_k > 0$ and $\lim_{p_k \to \infty} h(p_k) = -\infty$. Hence there exists a $p_k' \in [0, \infty]$ such that $h(p_k') = 0$.

As a result, $h(p_k) > 0$ on the interval $[0, p'_k]$ and $h(p_k) < 0$ on the interval $[p'_k, +\infty]$. Hence $g'(p_k) > 0$ on the interval $[0, p'_k]$ and $g'(p_k) < 0$ on the interval $[p'_k, +\infty]$. We then claim that $g(p_k)$ achieves its maximum value when $p_k = p'_k$.

It is not straightforward to obtain a closed-form expression of p_k^* by solving $h(p_k) = 0$. However, this is a one-dimensional search problem and function $h(p_k)$ has the monotone property. Hence, some simple algorithms (e.g., bisection search) can be used to obtain the solution [150].

B.3 Proof of Theorem ??

Let's examine the property of $f(\eta)$. Its first order derivative is

$$f'(\eta) = \frac{-v(1-\eta) + u\eta - v\eta\ln(\eta)}{\eta(1-\eta)^2}. \tag{B.4}$$

Letting $h(\eta) = -v(1-\eta) + u\eta - v\eta\ln(\eta)$, we have $\lim_{\eta\to 0^+} h(\eta) = -v < 0$ and $\lim_{\eta\to 1^-} h(\eta) = u > 0$. Moreover,

$$h'(\eta) = u - v\ln(\eta) > 0,\ \eta \in (0,1). \tag{B.5}$$

Hence, $h(\eta)$ is an increasing function in terms of η on $(0,1)$ and there exists only one point η' such that $h(\eta') = 0$. Then $h(\eta) < 0$ ($f'(\eta) < 0$) on the interval $(0, \eta']$, and $h(\eta) > 0$ ($f'(\eta) > 0$) on the interval $[\eta', 1)$. As a result, $f(\eta)$ will be decreasing on $(0, \eta']$ and increasing on $[\eta', 1)$. The optimal solution that minimizes the energy consumption E_k should be $\eta = \eta'$.

B.4 Proof of Theorem 7

The objective function in Problem (P3c) is in the form of sum-of-ratios. To prove the sum function is convex, we only need to show that each sub ratio function is convex.

First of all, we consider the phase shift matrix $\mathbf{V}$. It can be seen that the denominator is actually a logarithm function of $f_k(\mathbf{V})$, which is concave, and $f_k(\mathbf{V})$ is a linear function of $\mathbf{V}$. Hence, each individual ratio function is convex in terms of $\mathbf{V}$. Hence, their sum will also be convex in terms of $\mathbf{V}$.

Next, note that the denominator of each sub ratio function is the achievable rate $R_k(b_k)$ of each device. Since

$$\frac{\partial^2 R_k}{\partial b_k^2} = -\frac{p_k^2 f_k(\mathbf{V})^2}{\ln(2)(N_0 b_k + p_k f_k(\mathbf{V}))^2 b_k} < 0, \tag{B.6}$$

$R_k(b_k)$ is concave in terms of b_k and $(p_{c,k} + p_k)/R_k(b_k)$ is convex in terms of b_k.

B.5 Proof of Theorem 8

Theorem 7 also suggests that Problem (P3d') is a convex optimization problem. Introduce the Lagrange multipliers associated with the objective function in (P3d') and the bandwidth constraint. Thus the Lagrange function of the problem can be written as $\mathcal{L} = \sum_k \beta_k + \sum_k \lambda_k \left(p_{c,k} + p_k - A_k - \beta_k b_k \log_2(1 + \frac{p_k f_k(\mathbf{V})}{N_0 b_k}) \right) + \mu(\sum_k b_k - B)$. Then the optimal solution $\mathbf{V}^*$, $\{b_k^*\}$, and $\{\beta_k^*\}$ and the Lagrange multipliers λ_k^* and μ^* should satisfy the

following KKT conditions.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = -\frac{1}{\ln 2}\sum_k \lambda_k \beta_k b_k \frac{1}{1+\frac{p_k f_k(\mathbf{V})}{N_0 b_k}} f_k'(\mathbf{V}) = 0 \tag{B.7a}$$

$$\frac{\partial \mathcal{L}}{\partial b_k} = \mu + \frac{\lambda_k \beta_k}{\ln(2)(N_0 b_k + p_k f_k(\mathbf{V}))} \left[p_k f_k(\mathbf{V}) - (p_k f_k(\mathbf{V}) + N_0 b_k) \ln\left(1+\frac{p_k f_k(\mathbf{V})}{N_0 b_k}\right)\right] = 0 \tag{B.7b}$$

$$\frac{\partial \mathcal{L}}{\partial \beta_k} = 1 - \lambda_k b_k \log_2\left(1+\frac{p_k f_k(\mathbf{V})}{N_0 b_k}\right) = 0 \tag{B.7c}$$

$$\lambda_k \frac{\partial \mathcal{L}}{\partial \lambda_k} = \lambda_k \left(p_{c,k} + p_k - A_k - \beta_k b_k \log_2\left(1+\frac{p_k f_k(\mathbf{V})}{N_0 b_k}\right)\right) = 0 \tag{B.7d}$$

$$\mu \frac{\partial \mathcal{L}}{\partial \mu} = \mu \left(\sum_k b_k - B \right) = 0 \tag{B.7e}$$

$$\lambda_k \geq 0, \mu \geq 0. \tag{B.7f}$$

$$p_{c,k} + p_k - A_k - \beta_k b_k \log_2(1+\frac{p_k f_k(\mathbf{V})}{N_0 b_k}) \leq 0 \tag{B.7g}$$

$$\sum_k b_k - B \leq 0. \tag{B.7h}$$

From (B.7c), we can infer that $\lambda_k^* > 0$ and conclude that the equality in (3.36a) holds. Similarly, from (B.7d), we have (3.36b). Moreover, note that given $\lambda = \lambda_k^*$ and $\beta_k = \beta_k^*$, (B.7a), (B.7b), (B.7e), and (B.7f) are just the KKT conditions for Problem (P3). Since (P3) is convex programming for parameter $\lambda_k > 0$ and $\beta_k \geq 0$, the KKT conditions are also sufficient optimality conditions. This completes the proof of the theorem.

B.6 Proof of Theorem 9

Following Theorem (8), the optimal solution to (P3d') should satisfy the KKT conditions (B.7). We have from (B.7b)

$$\mu = \frac{\lambda_k \beta_k \left(-p_k f_k(\mathbf{V}) + (N_0 b_k + p_k f_k(\mathbf{V})) \ln(1+\frac{p_k f_k(\mathbf{V})}{N_0 b_k}) \right)}{(N_0 b_k + p_k f_k(\mathbf{V})) \ln(2)},$$

which can be written as

$$\frac{\lambda_k \beta_k}{\ln(2)} \left(\ln(1+x) - \frac{x}{1+x} \right) = \mu, \tag{B.8}$$

where $x = \frac{p_k f_k(\mathbf{V})}{N_0 b_k}$. The solution is found to be

$$\ln(1+x) + \frac{1}{1+x} = 1 + \frac{\mu \ln(2)}{\lambda_k \beta_k} \triangleq C_k. \tag{B.9}$$

Hence, we obtain (3.38) and (3.39). Note that the optimal solution of bandwidth b_k^* can be obtained numerically by substituting μ^* into (3.39) and (3.38). Again, the bisection algorithm can be applied to find the numerical solution of μ when solving $\sum_k b_k = B$, by leveraging the monotonicity of the *Lambert W* function.

B.7 Proof of Theorem 10

Algorithm 4 is a two-layer alternating optimization algorithm. In the outer layer, the auxiliary variable λ_k and β_k are updated with a Newton-like method, the convergence of which has been proved in [151]. We only need to show that the inner layer iteration (Algorithm 3) converges, where the variables $\mathbf{V}$ and b_k are optimized.

Denote the objective function of (P3d') as $f(b_k, \mathbf{V})$. In the sth iteration, we have

$$f(b_k^s, \mathbf{V}^s) \overset{(a)}{\leq} f(b_k^s, \mathbf{V}^{s+1}) \overset{(b)}{\leq} f(b_k^{s+1}, \mathbf{V}^{s+1}).$$

Note that the above inequalities (a)-(b) hold true because Problems $\mathbf{V}^s$ and b_k^s are both optimally solved in each iteration s. However, we have to mention that inequality (a) will not hold strictly since we deal with the non-convex rank one constraint with the Gaussian randomization method, which may violate the monotonic improvement property of the above equation. To tackle this issue, our solution is to perform a significant number of randomization processes and select the best solution that maximizes the objective function in (P3d'). In simulations, we perform 100 Gaussian randomization and select the best $\mathbf{v}$ that achieves

the maximum objective function. As a result, the inequality (a) will be guaranteed. Due to limited BS power and the finite number of IRS reflecting elements, the objective function in (P3d') is lower bounded and will converge after a finite number of steps.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *IEEE communications magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[4] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang, J. Wang *et al.*, "Towards 6g wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences*, vol. 64, no. 1, pp. 1–74, 2021.

[5] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[6] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. C. Zhang, "Artificial intelligence-enabled cellular networks: A critical path to beyond-5g and 6g," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 212–217, 2020.

[7] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond," *IEEE access*, vol. 7, pp. 78 729–78 757, 2019.

[8] J. Zhao, "A survey of intelligent reflecting surfaces (irss): Towards 6g wireless communication networks," *arXiv preprint arXiv:1907.04789*, 2019.

[9] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.

[10] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6g: A comprehensive survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.

[11] Y. Zhao, J. Zhao, W. Zhai, S. Sun, D. Niyato, and K.-Y. Lam, "A survey of 6g wireless communications: Emerging technologies," in *Future of Information and Communication Conference*. Springer, 2021, pp. 150–170.

[12] M. Sanmarti, "The battle for 6g has already started," *New Zealand International Review*, vol. 46, no. 4, pp. 18–20, 2021.

[13] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[14] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, 2019.

[15] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. de Rosny, and S. Tretyakov, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Nov. 2020.

[16] E. F. Kuester, M. A. Mohamed, M. Piket-May, and C. L. Holloway, "Averaged transition conditions for electromagnetic fields at a metafilm," *IEEE Trans. Antennas Propag.*, vol. 51, no. 10, pp. 2641–2651, Oct. 2003.

[17] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019.

[18] V. Arun and H. Balakrishnan, "RFocus: Beamforming using thousands of passive antennas," in *Proc. USENIX NSDI'20*, Santa Clara, CA, Feb. 2020, pp. 1047–1061.

[19] Y. Yang, B. Zheng, S. Zhang, and R. Zhang, "Intelligent reflecting surface meets OFDM: Protocol design and rate maximization," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4522–4535, July 2020.

[20] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Aug. 2019.

[21] X. Yu, D. Xu, Y. Sun, D. W. K. Ng, and R. Schober, "Robust and secure wireless communications via intelligent reflecting surfaces," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2637–2652, Nov. 2020.

[22] C. Pan, H. Ren, K. Wang, M. Elkashlan, A. Nallanathan, J. Wang, and L. Hanzo, "Intelligent reflecting surface aided MIMO broadcasting for simultaneous wireless information and power transfer," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1719–1734, Aug. 2020.

[23] Z. Chu, P. Xiao, M. Shojafar, D. Mi, J. Mao, and W. Hao, "Intelligent reflecting surface assisted mobile edge computing for Internet of Things," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 619–623, Mar. 2020.

[24] S. Ali, W. Saad, N. Rajatheva, K. Chang, D. Steinbach, B. Sliwa, C. Wietfeld, K. Mei, H. Shiri, H.-J. Zepernick *et al.*, "6g white paper on machine learning in wireless communication networks," *arXiv preprint arXiv:2004.13875*, 2020.

[25] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 341–346.

[26] X. Wang, L. Gao, S. Mao, and S. Pandey, "Deepfi: Deep learning for indoor fingerprinting using channel state information," in *2015 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2015, pp. 1666–1671.

[27] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[28] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1065–1069, 2019.

[29] K. Xiao, S. Mao, and J. K. Tugnait, "Tcp-drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access*, vol. 7, pp. 11 892–11 904, 2019.

[30] M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Leveraging uavs for coverage in cell-free vehicular networks: A deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2835–2847, 2020.

[31] C. Perfecto, M. S. Elbamby, J. Del Ser, and M. Bennis, "Taming the latency in multi-user vr 360°: A qoe-aware deep learning-aided multicast framework," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2491–2508, 2020.

[32] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2016.

[33] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5g," *IEEE network*, vol. 30, no. 1, pp. 44–51, 2016.

[34] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, Third Quarter 2019.

[35] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?" *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7331–7376, 2019.

[36] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE communications magazine*, vol. 58, no. 1, pp. 19–25, 2020.

[37] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural networks*, vol. 6, no. 6, pp. 861–867, 1993.

[38] M. Di Renzo *et al.*, "Smart radio environments empowered by reconfigurable AI meta-surfaces: An idea whose time has come," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, Article no.: 129, May 2019.

[39] Y. Xu, G. Yue, and S. Mao, "User grouping for massive MIMO in FDD systems: New design methods and analysis," *IEEE Access J.*, vol. 2, no. 1, pp. 947–959, Sept. 2014.

[40] M. Dai, B. Clerckx, D. Gesbert, and G. Caire, "A rate splitting strategy for massive MIMO with imperfect CSIT," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4611–4624, July 2016.

[41] B. Clerckx, H. Joudeh, C. Hao, M. Dai, and B. Rassouli, "Rate splitting for mimo wireless networks: A promising phy-layer strategy for lte evolution," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 98–105, 2016.

[42] H. Joudeh and B. Clerckx, "Robust transmission in downlink multiuser MISO systems: A rate-splitting approach," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6227–6242, Dec. 2016.

[43] Y. Mao, O. Dizdar, B. Clerckx, R. Schober, P. Popovski, and H. V. Poor, "Rate-splitting multiple access: Fundamentals, survey, and future research trends," *arXiv preprint arXiv:2201.03192*, 2022.

[44] Y. Mao, B. Clerckx, and V. O. Li, "Rate-splitting multiple access for downlink communication systems: Bridging, generalizing, and outperforming SDMA and NOMA," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, Article no.: 133, May 2018.

[45] H. Joudeh and B. Clerckx, "Sum-rate maximization for linearly precoded downlink multiuser MISO systems with partial CSIT: A rate-splitting approach," *IEEE Trans. Commun.*, vol. 64, no. 11, pp. 4847–4861, Nov. 2016.

[46] H. Fu, S. Feng, W. Tang, and D. W. K. Ng, "Robust secure beamforming design for two-user downlink MISO rate-splitting systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8351–8365, Dec. 2020.

[47] K. Weinberger, A. A. Ahmad, A. Sezgin, and A. Zappone, "Synergistic benefits in irs- and rs-enabled c-ran with energy-efficient clustering," *IEEE Transactions on Wireless Communications*, 2022.

[48] Z. Yang, J. Shi, Z. Li, M. Chen, W. Xu, and M. Shikh-Bahaei, "Energy efficient rate splitting multiple access (RSMA) with reconfigurable intelligent surface," in *Proc. IEEE ICC 2020 Workshops*, Virtual Conference, June 2020, pp. 1–6.

[49] T. Fang, Y. Mao, S. Shen, Z. Zhu, and B. Clerckx, "Fully connected reconfigurable intelligent surface aided rate-splitting multiple access for multi-user multi-antenna transmission," *arXiv preprint arXiv:2201.07048*, 2022.

[50] A. Bansal, K. Singh, B. Clerckx, C.-P. Li, and M.-S. Alouini, "Rate-splitting multiple access for intelligent reflecting surface aided multi-user communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9217–9229, 2021.

[51] H. Li, Y. Mao, O. Dizdar, and B. Clerckx, "Rate-splitting multiple access for 6g–part iii: Interplay with reconfigurable intelligent surfaces," *arXiv preprint arXiv:2205.02036*, 2022.

[52] Z. Wang, L. Liu, and S. Cui, "Channel estimation for intelligent reflecting surface assisted multiuser communications: Framework, algorithms, and analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6607–6620, 2020.

[53] B. Clerckx *et al.*, "Is NOMA efficient in multi-antenna networks? a critical look at next generation multiple access techniques," *arXiv preprint arXiv:2101.04802*, Jan. 2021. [Online]. Available: https://arxiv.org/abs/2101.04802

[54] A. S. de Sena, P. H. Nardelli, D. B. da Costa, P. Popovski, and C. B. Papadias, "Rate-splitting multiple access and its interplay with intelligent reflecting surfaces," *IEEE Communications Magazine*, 2022.

[55] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2181–2195, Oct. 2017.

[56] F. Fang, Y. Xu, Q.-V. Pham, and Z. Ding, "Energy-efficient design of irs-noma networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14 088–14 092, 2020.

[57] Y. Cheng, K. H. Li, Y. Liu, K. C. Teh, and H. V. Poor, "Downlink and uplink intelligent reflecting surface aided networks: NOMA and OMA," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3988–4000, June 2021.

[58] Z. Ding and H. V. Poor, "A simple design of IRS-NOMA transmission," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1119–1123, May 2020.

[59] Z. Li, C. Ye, Y. Cui, S. Yang, and S. Shamai, "Rate splitting for multi-antenna downlink: Precoder design and practical implementation," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1910–1924, 2020.

[60] H. Xia, Y. Mao, X. Zhou, B. Clerckx, S. Han, and C. Li, "Secure beamforming design for rate-splitting multiple access in multi-antenna broadcast channel with confidential messages," *arXiv preprint arXiv:2202.07328*, 2022.

[61] D. Yu, S.-H. Park, O. Simeone, and S. Shamai, "Robust design of rate-splitting multiple access with imperfect csi for cell-free mimo systems," *arXiv preprint arXiv:2203.03690*, 2022.

[62] B. Matthiesen, Y. Mao, P. Popovski, and B. Clerckx, "Globally optimal beamforming for rate splitting multiple access," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4775–4779.

[63] H. Fu, S. Feng, and D. W. K. Ng, "Resource allocation design for IRS-aided downlink MU-MISO RSMA systems," in *Proc. IEEE ICC 2021 Workshops*, Virtual Conference, June 2021, pp. 1–6.

[64] X. Mu, Y. Liu, L. Guo, J. Lin, and N. Al-Dhahir, "Exploiting intelligent reflecting surfaces in NOMA networks: Joint beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6884–6898, Oct. 2020.

[65] Z. Lin, M. Lin, T. de Cola, J.-B. Wang, W.-P. Zhu, and J. Cheng, "Supporting IoT with rate-splitting multiple access in satellite and aerial integrated networks," *IEEE Internet of Things J.*, vol. 8, no. 14, pp. 11 123–11 134, July 2021.

[66] R. Feng, Q. Li, Q. Zhang, and J. Qin, "Robust secure beamforming in MISO full-duplex two-way secure communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 408–414, Jan. 2015.

[67] Y. Nesterov and A. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM, 1994.

[68] Y. Omid, S. M. Shahabi, C. Pan, Y. Deng, and A. Nallanathan, "Low-complexity robust beamforming design for irs-aided miso systems with imperfect channels," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1697–1701, 2021.

[69] G. Zhou, C. Pan, H. Ren, K. Wang, and A. Nallanathan, "A framework of robust transmission design for irs-aided miso communications with imperfect cascaded channels," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5092–5106, 2020.

[70] S. Hong, C. Pan, H. Ren, K. Wang, K. K. Chai, and A. Nallanathan, "Robust transmission design for intelligent reflecting surface-aided secure communication systems with imperfect cascaded csi," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2487–2501, 2020.

[71] N. Jindal and A. Goldsmith, "Dirty-paper coding versus TDMA for MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1783–1794, May 2005.

[72] H. Joudeh and B. Clerckx, "Rate-splitting for max-min fair multigroup multicast beamforming in overloaded systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7276–7289, Nov. 2017.

[73] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intell. Stat.*, Fort Lauderdale, FL, Apr. 2017, pp. 1273–1282.

[74] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, Oct. 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[75] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process.*, vol. 37, no. 3, pp. 50–60, May 2020.

[76] S. Zargari, A. Khalili, and R. Zhang, "Energy efficiency maximization via joint active and passive beamforming design for multiuser MISO IRS-aided SWIPT," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 557–561, Mar. 2020.

[77] K. Yang, Y. Shi, Y. Zhou, Z. Yang, L. Fu, and W. Chen, "Federated machine learning for intelligent IoT via reconfigurable intelligent surface," *IEEE Network*, vol. 34, no. 5, pp. 16–22, Sept./Oct. 2020.

[78] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[79] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1387–1395.

[80] T. Bai, C. Pan, Y. Deng, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Latency minimization for intelligent reflecting surface aided mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2666–2682, Nov. 2020.

[81] Z. Wang, J. Qiu, Y. Zhou, Y. Shi, L. Fu, W. Chen, and K. B. Lataief, "Federated learning via intelligent reflecting surface," *arXiv preprint arXiv:2011.05051*, 2020.

[82] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive mimo for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[83] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, Mar. 2017.

[84] C. Ma, J. Konečnỳ, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, 2017.

[85] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[86] M. Grant, S. Boyd, and Y. Ye, "cvx users' guide," *online: http://www. stanford. edu/˜ boyd/software. html*, 2009.

[87] Y. Jong, "An efficient global optimization algorithm for nonlinear sum-of-ratios problem," *Optimization Online*, Aug. 2012.

[88] S. He, Y. Huang, L. Yang, and B. Ottersten, "Coordinated multicell multiuser precoding for maximizing weighted sum energy efficiency," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 741–751, Feb. 2013.

[89] I. CVX Research, "Using mosek with cvx," 2012. [Online]. Available: http://cvxr.com/cvx/doc/mosek.html

[90] I. Pólik and T. Terlaky, "Interior point methods for nonlinear optimization," in *Nonlinear optimization*. Springer, 2010, pp. 215–276.

[91] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2016.

[92] G. Zhou, C. Pan, H. Ren, K. Wang, and A. Nallanathan, "Intelligent reflecting surface aided multigroup multicast miso communication systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3236–3251, 2020.

[93] R. Jin, X. He, and H. Dai, "On the design of communication efficient federated learning over wireless networks," *arXiv preprint arXiv:2004.07351*, June 2020. [Online]. Available: https://arxiv.org/abs/2004.07351

[94] S. Buzzi, I. Chih-Lin, T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, "A survey of energy-efficient techniques for 5G networks and challenges ahead," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 697–709, Apr. 2016.

[95] S. Chen, S. Sun, Q. Gao, and X. Su, "Adaptive beamforming in TDD-based mobile communication systems: State of the art and 5G research directions," *IEEE Wireless Commun.*, vol. 23, no. 6, pp. 81–87, Dec. 2016.

[96] N. Alliance, "5G white paper," *Next generation mobile networks, white paper*, pp. 1–125, 2015.

[97] L. Ericsson, "More than 50 billion connected devices," *White Paper*, vol. 14, p. 124, 2011.

[98] E. Björnson, E. G. Larsson, and M. Debbah, "Massive MIMO for maximal spectral efficiency: How many users and pilots should be allocated?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1293–1308, Feb. 2016.

[99] G. Bacci, M. Luise, H. V. Poor, and A. M. Tulino, "Energy efficient power control in impulse radio UWB wireless networks," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 3, pp. 508–520, Oct. 2007.

[100] A. H. Sakr and E. Hossain, "Cognitive and energy harvesting-based D2D communication in cellular networks: Stochastic geometry modeling and analysis," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1867–1880, May. 2015.

[101] A. Zappone, E. A. Jorswieck, and S. Buzzi, "Energy efficiency and interference neutralization in two-hop MIMO interference channels," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6481–6495, Dec. 2014.

[102] C. Xiong, G. Y. Li, S. Zhang, Y. Chen, and S. Xu, "Energy-efficient resource allocation in OFDMA networks," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3767–3778, Dec. 2012.

[103] C. Isheden, Z. Chong, E. Jorswieck, and G. Fettweis, "Framework for link-level energy efficiency optimization with informed transmitter," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2946–2957, Aug. 2012.

[104] A. Zappone, E. Bjornson, L. Sanguinetti, and E. Jorswieck, "Globally optimal energy-efficient power control and receiver design in wireless networks," *IEEE Trans. Signal Process.*, vol. 65, pp. 2844–2859, Jun 2017.

[105] A. Zappone, L. Sanguinetti, G. Bacci, E. Jorswieck, and M. Debbah, "Energy-efficient power control: A look at 5G wireless technologies," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1668–1683, Apr. 2016.

[106] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in OFDMA systems with large numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3292–3304, Jun. 2012.

[107] Q. Xu, X. Li, H. Ji, and X. Du, "Energy-efficient resource allocation for heterogeneous services in OFDMA downlink networks: Systematic perspective," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2071–2082, Jun. 2014.

[108] L. Venturino, A. Zappone, C. Risi, and S. Buzzi, "Energy-efficient scheduling and power allocation in downlink OFDMA networks with base station coordination," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 1–14, Jan. 2015.

[109] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5289–5304, Oct. 2017.

[110] D. Nguyen, L.-N. Tran, P. Pirinen, and M. Latva-aho, "Precoding for full duplex multiuser MIMO systems: Spectral and energy efficiency maximization," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4038–4050, Aug. 2013.

[111] H. He, C. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmwave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.

[112] F. Liang, C. Shen, and F. Wu, "An iterative BP-CNN architecture for channel decoding," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 144–159, Feb. 2018.

[113] H. He, S. Jin, C. Wen, F. Gao, G. Ye Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, to be published.

[114] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, 2018.

[115] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[116] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–5.

[117] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of MIMO-OFDM system based on auto encoder and extreme learning machine," in *Proc. IEEE IJCNN 2017*, Anchorage, AK, May 2017, pp. 1602–1606.

[118] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[119] A. Klautau, P. Batista, N. Prelcic, Y. Wang, and R. Heath, "5G MIMO data for machine learning: Application to beam-selection using deep learning," in *Proc. Inf. Theory and Appl. Workshop (ITA)*, San Diego, CA, Feb. 2016, pp. 1–6.

[120] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *arXiv preprint arXiv:1901.00354*, Jan. 2019.

[121] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[122] S. Xu, P. Liu, R. Wang, and S. S. Panwar, "Realtime scheduling and power allocation using deep neural networks," *arXiv preprint arXiv:1811.07416*, Nov. 2018.

[123] B. Matthiesen, A. Zappone, E. A. Jorswieck, and M. Debbah, "Deep learning for optimal energy-efficient power control in wireless interference networks," *arXiv preprint arXiv:1812.06920*, Dec. 2018.

[124] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[125] W. Wang, X. Wang, and S. Mao, "Deep convolutional neural networks for indoor localization with CSI images," *IEEE Trans. Netw. Sci. Eng.*, vol. 5.

[126] K. Xiao, S. Mao, and J. Tugnait, "TCP-Drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access Journal*, vol. 7, no. 1, pp. 11 892–11 904, Jan. 2019.

[127] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *arXiv preprint arXiv:1807.10025*, Jul. 2018.

[128] R. W. Freund and F. Jarre, "Solving the sum-of-ratios problem by an interior-point method," *Springer Journal of Global Optimization*, vol. 19, no. 1, pp. 83–102, 2001.

[129] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *INFORMS Operations Research*, vol. 14, no. 4, pp. 699–719, July/Aug. 1966.

[130] Y. Yang and M. Pesavento, "A unified successive pseudoconvex approximation framework," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3313–3328, Jul. 2017.

[131] O. L. Mangasarian, *Nonlinear Programming*. Philadelphia, PA: SIAM, 1993, vol. 10.

[132] W. Dinkelbach, "On nonlinear fractional programming," *INFORMS Management Science*, vol. 13, no. 7, pp. 492–498, 1967.

[133] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Elsevier Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[134] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR 2016*, Las Vegas, NV, June 2016, pp. 770–778.

[135] X. Wang, X. Wang, and S. Mao, "ResLoc: Deep residual sharing learning for indoor localization with CSI tensors," in *Proc. IEEE PIMRC 2017*, Montreal, Canada, Oct. 2017, pp. 1–6.

[136] X. Wang, S. Mao, and M. Gong, "An overview of 3GPP cellular vehicle-to-everything standards," *GetMobile: Mobile Comp. and Comm.*, vol. 21, no. 3, pp. 19–25, Sep. 2017.

[137] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 46–53, June 2013.

[138] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management toward 5G radio access networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1656–1686, Third Quarter 2016.

[139] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 678–700, Second Quater 2012.

[140] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1581–1592, Mar. 2019.

[141] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, June 2019.

[142] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key technologies and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, Fourth Quarter 2019.

[143] M. Feng and S. Mao, "Dealing with limited backhaul capacity in millimeter wave systems: A deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 50–55, Mar. 2019.

[144] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[145] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, in press.

[146] J. Wang, C. Xu, Y. Huangfu, R. Li, Y. Ge, and J. Wang, "Deep reinforcement learning for scheduling in cellular networks," in *Proc. WCSP'19*, Xi'an, China, Oct. 2019, pp. 1–6.

[147] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, Y. Lu, R. Trestian, and G. Ghinea, "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1661–1675, Dec. 2018.

[148] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access J.*, vol. 8, no. 1, pp. 57 063–57 074, Mar. 2020.

[149] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498–513, Feb. 2010.

[150] D. La, T. M. Rocca, and D. La, "Numerical methods and applications," 2002.

[151] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.