Radio Resource Management for New Application Scenarios in 5G: Optimization and Deep Learning

Rui Dong

A thesis submitted to fulfil
requirements for the degree of
Doctor of Philosophy

Australian Centre of Excellence in Telecommunications
School of Electrical and Information Engineering
Faculty of Engineering
University of Sydney

2020

Declaration

I hereby declare that this submission is my own work carried out by myself, in collaboration with my supervisors, while enrolled in the School of Electrical and Information Engineering at the University of Sydney as a candidate for the Doctor of Philosophy. To the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Chapter 3 and 4 of this thesis are published as [J2] and [J1]. I designed the study, analyzed the data and wrote the drafts of the MS. In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Rui Dong

31 December 2019

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Yonghui Li

31 December 2019

Abstract

The fifth-generation (5G) New Radio (NR) systems are expected to support a wide range of emerging applications with diverse Quality-of-Service (QoS) requirements. New application scenarios in 5G NR include enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and ultra-reliable low-latency communications (URLLC). New wireless architectures, such as full-dimension (FD) massive multiple-input multiple-output (MIMO) and mobile edge computing (MEC) system, and new coding scheme, such as short block-length channel coding, are envisioned as enablers of QoS requirements for 5G NR applications. Resource management in these new wireless architectures is crucial in guaranteeing the QoS requirements of 5G NR systems. The traditional optimization problems, such as subcarriers and user association, are usually non-convex or Non-deterministic Polynomial-time (NP)-hard. It is time-consuming and computing-expensive to find the optimal solution, especially in a large-scale network. To solve these problems, one approach is to design a low-complexity algorithm with near optimal performance. In some cases, the low complexity algorithms are hard to obtain, deep learning can be used as an accurate approximator that maps environment parameters, such as the channel state information and traffic state, to the optimal solutions.

In this thesis, we design low-complexity optimization algorithms, and deep learning frameworks in different architectures of 5G NR to resolve optimization problems subject to QoS requirements. First, we propose a low-complexity algorithm for a joint cooperative beamforming and user association problem for eMBB in 5G NR to maximize the network capacity. Next, we propose a deep learning (DL) framework to optimize user association, resource allocation, and offloading probabilities for delay-tolerant services and URLLC in 5G NR. Finally, we address the issue of time-varying traffic and network conditions on resource management in 5G NR.

To my beloved ones,

Mama and J.R.

Acknowledgements

I would like to acknowledge the help and support of my two supervisors: Prof. Yonghui Li and Dr. Wibowo Hardjwanna. Since the day Prof. Yonghui granted me admission, he has been unwaveringly supportive, knowledgeable and very kind during my Ph.D. studies. I am grateful for the opportunity as well as for the time he always takes to give me advice on my research and career. Dr. Wibowo Hardjwanna has been a resourceful mentor and an open-minded supporter since day one. I would like to thank him immensely for his years of availability for research discussions.

Dr. Changyang She lit a lamp on my research path and guided me towards a brand new research direction. He has been very patient, very helpful, a trusted advisor, colleague, and friend. He has my endless gratitude for all that he has done for me. I would like to give special thanks to Dr. Ang Li for helping me achieve my very first publication. His confidence and heartfelt friendship helped me go through a hard time. Throughout my Ph.D. journey, my colleagues at the Centre of Excellence in Telecommunications have become my life-long friends, and have made the working environment pleasant and my life beautiful.

I would like to thank my various sources of financial support: the Australian Research Council for the Research Training Program Scholarship, the school of Electrical and Information Engineering at the University of Sydney for the Norman I Prize scholarship, and Prof. Yonghui Li and the PRSS scheme from the University of Sydney for the generous financial support for my conference trips.

I would also like to thank Dr. Floriana Badalotti for her careful proofreading of my thesis.

I am indebted to all my friends for those sincere wishes and endless encouragement. They spiritually supported me and dragged me away from misery, and will forever be my priceless treasure through life. To my husband and my soulmate J.R., he has been always there for me with unconditional love and support. He never doubted my capability and his confidence in me brought out the best in me. Finally, to my mother, she has been my role model forever. Without her, I am nobody, with her, I have the whole world. I simply could not achieve what I have without you all.

Contents

Declaration i

Abstract ii

Acknowledgements iv

Contents v

List of Figures x

List of Tables xii

List of Acronyms xiii

List of Publications xv

1 Introduction 1

1.1 5G New Radio 1

1.2 Wireless Communications in 5G New Radio 3

1.2.1 Full-Dimension Massive MIMO 3

1.2.2 Mobile Edge Computing 4

1.2.3 Short Block-length Channel Coding 5

1.3 Research Problems and Contributions 6

1.3.1 Design of Low-Complexity Algorithm in Full-Dimension Massive MIMO Networks 9

1.3.2 Design of Deep Learning in Mobile Edge Computing Systems . 10
1.3.3 Design of Deep Transfer Learning with Quality-of-Service Guarantee . . . 12
1.4 Thesis Outline . . . 13

2 Preliminaries **14**
2.1 Network Architectures . . . 14
2.1.1 Full Dimension Massive MIMO . . . 14
2.1.2 Mobile Edge Computing . . . 18
2.1.3 Shannon Capacity v.s. Short Block-Length Channel Coding . 22
2.2 Optimization Solvers . . . 24
2.2.1 Belief Propagation . . . 24
2.2.2 Deep Learning . . . 26
2.2.3 Deep Transfer Learning . . . 28

3 Design of Low Complexity Algorithm in Full-Dimension Massive MIMO Networks **30**
3.1 Introduction . . . 30
3.2 System Model . . . 32
3.2.1 Channel Model . . . 34
3.2.2 System Model . . . 35
3.3 Optimization Problem Formulation . . . 36
3.3.1 Asymptotic Analysis For Massive MIMO Channel . . . 37
3.3.2 SLNR Based FD Beamforming . . . 38
3.4 Proposed Gaussian Belief Propagation-based Algorithms . . . 41
3.4.1 Binary Relaxation and Linear Programming Formulation . . . 42
3.4.2 Gaussian Belief Propagation . . . 45
3.4.3 Binary Mapping . . . 49
3.5 Computational Complexity Analysis . . . 53
3.6 Simulation Results . . . 56
3.6.1 Convergence and Computational Complexity . . . 57
3.6.2 Capacity Performance Comparisons . . . 58
3.7 Chapter Summary . . . 63

4 Design of Deep Learning in Mobile Edge Computing Systems 64

4.1 Introduction 64

4.2 Related Work 66

4.3 System Model 67

4.3.1 MEC System 67

4.3.2 Computation Tasks and Communication Packets 70

4.3.3 Achievable Data Rate over Wireless Links 70

4.3.4 Offloading Policies 72

4.3.5 Queueing Model 72

4.3.6 Energy Consumption and Processing Rate at Local Server . . 73

4.4 Problem Formulation and Deep Learning Framework 74

4.4.1 QoS Constraints of URLLC Service 74

4.4.2 Stability of Delay-tolerant Services 76

4.4.3 Objective Function: Normalized Energy Consumption 77

4.4.4 Optimization Problem 78

4.4.5 Structure of Deep Learning 81

4.5 Algorithm to Solve Problem $\mathcal{P}_2$ 82

4.5.1 Outline of the Algorithm 82

4.5.2 Optimal Offloading Probability 84

4.5.3 Convergence of the Algorithm 87

4.5.4 Complexity Analysis 88

4.6 Deep Learning for User Association 89

4.6.1 Exploitation and Exploration of the DNN 90

4.6.2 The DNN Training 91

4.7 Simulation Results 91

4.7.1 Simulation Setup 91

4.7.2 Optimal Bandwidth Allocation and Offloading Probabilities . 93

4.7.3 DL Algorithm for User Association 94

4.8 Chapter Summary 98

5 Deep transfer learning for resource allocation for new applications with diverse QoS requirements 100

5.1 Introduction . . . 100

5.1.1 Background . . . 100

5.1.2 Related Works . . . 102

5.1.3 Chapter Outline . . . 103

5.2 System Model and Problem Formulation . . . 103

5.2.1 System Model . . . 103

5.2.2 Delay-Tolerant Services . . . 105

5.2.3 Delay-Sensitive Services . . . 105

5.2.4 URLLC Services . . . 107

5.2.5 Problem Formulation . . . 108

5.3 Supervised Deep Learning — Cascaded Neural Networks for QoS Guarantee . . . 108

5.3.1 Preliminary of Deep Learning . . . 109

5.3.2 Cascaded Neural Networks for QoS Guarantee . . . 110

5.3.3 Labeled Training Samples . . . 111

5.3.4 Train the Cascaded NNs . . . 118

5.4 Deep Transfer Learning in Non-Stationary Wireless Networks . . . 118

5.4.1 Preliminary of Deep Transfer Learning . . . 119

5.4.2 Transfer Learning with Non-Stationary Wireless Channels . . . 120

5.4.3 Transfer Learning with Different Types of Services . . . 120

5.5 Simulation and Numerical Results . . . 122

5.5.1 Validating the Properties of URLLC . . . 122

5.5.2 Performance Evaluation . . . 123

5.5.3 Performance with Transfer Learning . . . 128

5.6 Chapter Summary . . . 131

6 Conclusion 132

6.1 Summary of Content and Results . . . 132

6.2 Future Work . . . 134

A Proofs of Chapter 4 **136**

A.1 Proof of Proposition 4.1 136

A.2 Proof of Property 4.1 137

A.3 Proof of Property 4.2 137

B Proofs of Chapter 5 **139**

B.1 Proof of Optimality of the Algorithm in Table 5.2 139

B.2 Proof of Property 5.1 141

B.3 Proof of Optimality of the Algorithm in Table 5.3 142

Bibliography **144**

List of Figures

2.1 One ring model with uniform linear antenna array 15
2.2 Summary of MEC models 18
2.3 Queueing model at MEC servers 20
2.4 Factor graph for belief propagation 24
2.5 The illustration of deep learning with deep neural networks 27

3.1 Full Dimension Massive MIMO Deployment Scenario 34
3.2 Graphical Model for Gaussian Belief Propagation 46
3.3 Convergence of BP and GaBP 57
3.4 Capacity of GaBP Algorithm 58
3.5 The Number of Connected Users of GaBP Algorithm 59
3.6 Capacity of GaBP Algorithm with Different Number of Users 60
3.7 The Number of Connected Users of GaBP Algorithm with Different Number of Users 61
3.8 Capacity of GaBP Algorithm with Different Number of BSs 62
3.9 The Number of Connected Users of GaBP Algorithm with Different Number of BSs 63

4.1 System Model for Mobile Edge Computing 69
4.2 Queueing Model for Mobile Edge Computing 73
4.3 Digital twin enabled deep learning algorithm 81
4.4 Network topologies of deep learning algorithm in mobile edge computing system 92
4.5 Normalized energy consumption v.s. total number of users 94

4.6 Training loss function v.s. number of learning epoch 95

4.7 Normalized energy consumption v.s. user distribution ratio 96

4.8 Normalized energy consumption with uncertain user distribution ratios 98

5.1 Illustration of the cascaded NNs 110

5.2 Deep transfer learning with non-stationary wireless channels. 120

5.3 Deep transfer learning with multiple types of services. 121

5.4 Validating Convergence Condition for URLLC services. 125

5.5 Probability without QoS guarantee v.s. extra transmit power reserved to the users 126

5.6 Power consumption v.s. number of users. 127

5.7 Accuracy v.s. the number of training epochs when the number of antennas varies 128

5.8 Accuracy v.s. the number of training epochs, where the target task is resource allocation for delay-sensitive services 129

5.9 Accuracy v.s. the number of training epochs, where the target task is resource allocation for URLLC 129

5.10 Transfer knowledge from networks with a single type of services to networks with multiple types of services. 130

List of Tables

3.1 Notations 33
3.2 Complexity Comparison 54
3.3 The Complexity Analysis with Various System Configurations 58

4.1 Notations 68
4.2 Offloading and Subcarrier Allocation Algorithm 83
4.3 Parameters in Simulation 93
4.4 Performance Comparison When $M = 3$ and $K^{\mathrm{b}} = K^{\mathrm{u}} = 5$ 98

5.1 Notations 104
5.2 Bandwidth Allocation Algorithm for Solving Problem (5.18) 113
5.3 Bandwidth Allocation Algorithm for Solving Problem (5.14) 116
5.4 Deep Transfer Learning for Multiple Types of Services 123
5.5 Parameters in Simulation 124
5.6 Hyper-parameters of NNs 125

List of Acronyms

1-D	1-dimension
2-D	2-dimension
3GPP	Third Generation Partnership Project
5G	fifth-generation
ADMM	alternating direction method of multipliers
AP	access point
AR	augmented reality
AWGN	additive white Gaussian noise
BAF	binary beam association factors
BnB	branch-and-bound
BP	belief propagation
BS	base stations
CCDF	complementary cumulative distribution function
CDF	cumulative density function
CSI	channel state information
DFT	discrete Fourier transform
DL	deep learning
DNN	deep neural network
E2E	end-to-end
EE	energy efficiency
eMBB	enhanced mobile broadband
ES	exhaustive search
FCFS	first-come-first-served
FD	full-dimension
FNN	fully-connected neural network
GaBP	Gaussian belief propagation
ICI	inter-cell interference
i.i.d.	independent and identically distributed
IoT	Internet of Things
KKT	Karush-Kuhn-Tucker
LP	linear programming
MCC	mobile cloud computing
MEC	mobile edge computing

MIMO	multiple-input multiple-output
MINLP	Mixed-Integer Non-Linear Programming
MISO	multiple-input single-output
MME	mobility management entity
mMTC	massive machine-type communication
NN	neural network
NP	Non-deterministic Polynomial-time
NR	New Radio
OFDMA	orthogonal frequency division multiple access
PS	processor-sharing
QoS	Quality-of-Service
SINR	signal-to-interference-and-noise ratio
SLNR	signal-to-leakage-and-noise ratio
SNR	signal-to-noise ratio
TTI	transmission time interval
UAF	user association factor
URLLC	ultra-reliable low-latency communications
VR	virtual reality

List of Publications

The following is a list of submitted and published papers in refereed journals and conference proceedings produced during my Ph.D. candidature. In some cases, the journal papers contain material overlapping with the conference publications.

Journal Papers

[J1] Rui Dong, Changyang She, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic, "Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn from a Digital Twin," *IEEE Transactions on Wireless Communications,* vol. 18, no. 10, pp. 4692-4707, Oct. 2019.

[J2] Rui Dong, Ang Li, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic, "Joint Beamforming and User Association Scheme for Full-Dimension Massive MIMO Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7733-7746, Aug. 2019.

[J3] Rui Dong, Changyang She, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic, "Deep Learning for Radio Resource Allocation with Diverse Quality-of-Service Requirements in 5G," *submitted to IEEE Transactions on Wireless Communications,* 2019.

[J4] Changyang She, Rui Dong, Zhouyou Gu, et al., "Deep Learning for Ultra-Reliable and Low-Latency Communications in 6G Networks," *submitted to IEEE Network*.

[J5] Changyang She, Chengjian Sun, Rui Dong, Mahyar Shirvanimoghaddam, Yonghui Li, Chenyang Yang, and Branka Vucetic, "A Survey of Ultra-Reliable and Low-Latency Communications: Analysis Tools, Model-based Design, and Deep Learning," *submitted to IEEE Communications Surveys & Tutorials*.

Conference Papers

[C1] Changyang She, Rui Dong, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic, "Optimizing Resource Allocation for 5G Services with Diverse Quality-of-Service Requirements," IEEE Global Communications Conference, 2019.

[C2] Rui Dong, Changyang She, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic, "Improving Energy Efficiency of Ultra-Reliable Low-Latency and Delay-tolerant Services in Mobile Edge Computing Systems," IEEE International Conference on Communications Workshops, pp. 1–5, 2019.

[C3] Rui Dong, Wibowo Hardjawana, Ang Li, Yonghui Li, and Branka Vucetic, "Cooperative Beamforming for Multi-Cell Full Dimensional Massive MIMO Networks," IEEE International Conference on Communications, 2019.

[C4] Rui Dong, Wibowo Hardjawana, Yonghui Li, and Branka Vucetic, "Dynamic Sectoring with Elevation Optimization Technique in 5G Cellular Networks," IEEE International Conference on Communications Workshops, pp. 1–5, 2018.

Chapter 1

Introduction

1.1 5G New Radio

In the last few decades, wireless communications have become an inherent part of our daily life and experienced an exponential growth. Prior to 5G networks, most mobile communications served human-oriented connections, such as real-time access and information sharing including texts, images, audios and videos [1]. Next phase of telecommunications growths will be expanded greatly to machine applications which connect different types of machines and environments, equipped with sensors and actuators, and the systems operate in an automated fashion with little human interaction. Beyond the need for human-type communications, 5G networks are also expected to support orders of magnitude machine-type devices, which are extremely likely to drive an ever-increasing demand for wireless connectivity.

5G NR is expected to support various wireless communication needs and create ubiquitous connectivity for everyone and every device and machine. It will enable a wide range of application scenarios with diverse QoS requirements. According to QoS requirements, new application scenarios in 5G NR can be categorized into three types: eMBB with a significant increase in data rate requirements, URLLC with stringent latency and reliability requirements, and mMTC with a large number of connectivities [2, 3, 4]. The applications will be very common in the very near future, such as virtual

reality (VR)/augmented reality (AR), mission-critical Internet of Things (IoT) and autopilot vehicles [5, 6].

The QoS requirements of three new application scenarios mainly focus on the ever-increasing data rate, extra-low latency, and extra high reliability.

- *High Data Rate*: High data rate requirements are inevitable when the needs of users are not only limited to the content of text messages and voice calls but also expand to daily gaming and high-definition video streaming [7, 8]. The network deployment should be optimized to provide such high data rate services, estimated up to gigabytes per second, e.g., 10 Gb/s for eMBB services [9].

  A system with a massive number of antennas deployed at the BS in a 2-dimension (2-D) grid is envisioned for enlarged traffic volume networks. It is often referred to as an FD massive MIMO system [10, 11, 12].

- *Low Latency*: The latency in the literature is often measured by the end-to-end (E2E) delay of a packet, which is defined as the interval between the arrival time of a packet and the time when the processing of the packet is finished [13, 14]. In wireless communications, the possible delay components in E2E delay compose of transmission delay, queueing delay, computing delay and backhaul delay [15]. Overall, the required E2E delay for URLLC services should be bounded by 1 ms.

  One promising way of reducing E2E delay is to allow users to upload their packets to a nearby AP, such that the delays in backhauls and core networks, transmission and computing can be relieved. A system that enables such uploading and computing operations in a local area is the MEC system [16], which deploys APs equipped with computing capability servers.

- *High Reliability*: The reliability is a performance metric of how often a packet is lost, i.e., packet loss probability. It is particularly crucial for applications such as remote surgery, intelligent transportation and factory automation [17],

as the violation of packet loss probability will lead to incalculable consequences. Packet loss probability can be contributed by decoding errors and processing delay violations [18]. In traditional wireless communications, Shannon's capacity has been widely used to estimate the achievable data rate for an infinite block-length. However, Shannon's capacity cannot characterize decoding error probability [19].

To reduce the transmission delay of URLLC services, the block-length of channel codes is short, and the decoding error probability should be taken into account. To characterize the decoding error probability, the approximation of the achievable rate in the short block-length regime should be applied [20].

1.2 Wireless Communications in 5G New Radio

In this section, we will introduce current promising solutions to satisfy QoS requirements, including FD massive MIMO and MEC, as well as the development of short block-length channel coding.

1.2.1 Full-Dimension Massive MIMO

The idea of using a massive number of antennas was proposed for the first time in [21], in which a BS equipped with hundreds of antennas can simultaneously serve tens of users. It subsequently received growing research interests [22, 23, 24]. First, with an infinite number of antennas at the BS, the effect of small-scale channel fading is eliminated. Second, simple forms of user detection and precoding can achieve optimal performance, e.g., match filter and eigenbeamforming, as uncorrelated noise and intra-cell interference disappear. Third, the transmit power at the user during uplink transmission can be arbitrarily small. However, the implementation of massive MIMO in practice is challenging due to the limited physical BS tower space and operating carrier frequency. For example, a uniform linear array requires around

1.9m horizontal space to deploy 32 antenna elements with half of the wavelength when the carrier frequency is 2.5 GHz [25].

FD massive MIMO, which deploys a large number of antenna elements in a 2-D grid at the BS, has been proposed in Third Generation Partnership Project (3GPP) standard [26] to settle the issue of physical space limitation at the BS [27, 28, 29, 30, 31]. With an FD massive MIMO array at the base stations (BS), we can exploit the angles of arrival of the FD wireless channels in both horizontal and vertical directions to form a beam, while the conventional 1-dimension (1-D) massive MIMO can only generate beams in either the horizontal or the vertical direction. As a result, users in the same horizontal direction (related to the BS) but different in the vertical direction can be distinguished and served simultaneously.

1.2.2 Mobile Edge Computing

MEC is a fusion of techniques and theories in mobile computing and wireless communications [32]. The idea is to gather the idle computation resources distributed at the network's edges to accommodate the needs of computation-intensive and latency-critical tasks at mobile users [33]. Expected to welcome tens of billions of wireless devices requesting computing services [34], especially with the requirements of stringent QoS in latency, proximate access to a BS or an AP with computing servers becomes a key enabler in 5G NR.

One of the advantages of MEC over mobile cloud computing (MCC) is to shorten the E2E delay from the end user to the remote cloud center. The latency can be segmented into three components (propagation, computation and communication) in MEC, which are determined by propagation distance, computing capacity and the wireless transmission rate, respectively. In a typical dense small-cell network, the propagation distance for MEC is tens of meters, which dramatically shortens the delay in propagation. In terms of the computing capacity at a BS or an AP, the edge server normally has $10^2 \sim 10^4$ times higher capacity than a CPU with 3.3 GHz, 8 GB RAM, 70 GB storage space for running a popular game [32]. For the latency in

communication, advanced designs should be jointly considered with MEC such that latency-critical tasks can be guaranteed, such as short block-length coding techniques.

Another advantage of MEC is mobile energy savings. Most user devices, such as IoT, health monitoring and surveillance applications [35], have limited battery capacity. By offloading the computation-intensive tasks to MEC servers, the energy consumption of processing these tasks can be relieved from the user devices. It has been shown in [36] that the battery life can be extended by $30 \sim 50\%$ for different AR applications.

1.2.3 Short Block-length Channel Coding

Shannon's capacity has been adopted for decades to approximate an accurate error-free achievable rate when the block-length of channel codes is sufficiently large. However, Shannon's capacity model underestimates the delay for finite block-length packet transmissions and fails to sustain the desired error probability at a given fixed finite block-length. For some mission-critical applications, 3GPP outlines the general QoS requirement for one transmission of a packet as 99.999% (error probability of 10^5) for 32 bytes with a latency of 1 ms [26]. Since such applications, e.g. URLLC services, often have small packets to transmit and the transmission delay is very low, the transmission is not error-free in finite block-length channel codes. As a result, Shannon's capacity cannot characterize the maximal rate with a given error probability.

In the short block-length regime, Polyanskiy *et al.* derived the maximal achievable rate as a function of block-length and error probability [20]. Finite block-length information theory reveals the relationship between the desired decoding error rate and system bandwidth in low-latency short packet applications. Based on the short block-length channel model, Schiessl *et al.* computed probabilistic delay bounds in Rayleigh fading channels for low-latency applications [37]. They also pointed out that the design of finite block-length models should extend up to the application layer by taking queueing effects into account to characterize the latency of short packet applications.

1.3 Research Problems and Contributions

5G NR specifies the different types of application scenarios with their corresponding QoS requirements. In this work, we will focus on the designs, optimizations, and machine learning techniques over radio resources to achieve these QoS requirements, i.e. high data rate, low latency and high reliability.

The designs of resource management in wireless communications have attracted significant attention from both academia and industry and have been widely studied. In the literature, there are two kinds of approaches to find the optimal solution to the problem: optimization algorithms and machine learning algorithms. There are many challenges in designing and applying these algorithms, such as balancing the trade-off between complexity and performance, ensuring real-time implementation, adjusting to non-stationary wireless networks, and guaranteeing QoS. In the following, we will identify some research problems in algorithm designs and applications for wireless communications.

- *Low Computational Complexity*: The radio resources in wireless communications include continuous variables, such as transmit powers, beamforming vectors, and discrete variables; for example, the number of subcarriers, antennas and the user association decisions. For joint optimization problems over both continuous and discrete variables, they are non-convex and NP-hard problems [38, 39, 40]. Taking the example of joint problems over beamforming and user association for eMBB services, optimization algorithms can be classed into two groups. In [41, 42, 43, 44, 45, 46, 47], the joint beamforming and user association optimizations were solved via sequential optimization, where the beamforming vectors and user association decisions are optimized alternately. However, this method is inefficient for a large-scale network with a large number of BSs and users. The second group includes the distributed algorithms employed in [48, 49, 50, 51, 52, 53, 54, 55, 56]. In particular, a distributed alternating direction method of multipliers (ADMM) method was introduced to obtain the cooperative beamforming solutions in a parallel manner in [51].

However, Lagrangian multipliers need to be exchanged between the BSs at the central unit for the ADMM algorithm, which leads to severe computation delays, especially for a dense network. The belief propagation (BP) solver was proposed in [48, 49, 50, 52] to compute the optimal beamforming vectors at local factor nodes, where each factor node independently computes all possible beamforming vectors solutions, and removes the requirement of a central processing unit as needed in the ADMM-based method. However, the complexity of the distributed BP in [48, 49, 50, 52] becomes prohibitive when the number of factor nodes or the number of possible states increases. Moreover, the computation complexity of these schemes will significantly increase with the increase in the number of transmit antennas, which results in heavy computational burdens at the BSs. It is thus essential to develop distributed and low-complexity algorithms to solve joint NP-hard problems.

- *Real-time Implementation*: QoS constraints of some services, such as delay-sensitive and URLLC services, may not have closed-form expressions. To execute optimization algorithms, the system needs to evaluate the complicated non-closed form QoS constraints, and thus suffers from a long processing delay [57, 58]. Since optimization algorithms need to search the optimal solution when the network settings change, such as the number of users or different channel distributions, they are suitable for small-scale problems, such as resource allocation in a single AP scenario [59, 60]. When the scale of the problem grows, DL algorithms have the potential to find a near-optimal solution in a real-time manner [61]. Based on the universal approximation theorem of deep neural network (DNN) [62], a DNN can be used as an approximator of an optimal policy. The state-action pairs obtained from the optimal policy can be used as labeled samples to train the DNN. Once the training of the DNN is finished, the resource allocation can be computed from it with different channel realizations. The accuracy performance of DNNs is proportional to the number of training samples. As a result, a well-trained DNN requires a long period of training in general. How to enable DL algorithms to allocate resources in real time deserves

further study.

- *Adjustment to Time-varying Channel and Traffic Conditions*: It is worth noting that the application of DL in wireless networks is not straightforward. Even if we can obtain a large number of labeled training samples, the pre-trained DNN is not accurate when the wireless network is non-stationary. For example, the distribution of wireless channels and the types of services in the network may vary. These non-stationary parameters that are not included in the input of the DNN are referred to as hidden variables [63]. During the training phase, we assume that the hidden variables are fixed. However, in practical systems, these hidden variables drift over time. As discussed in [63], a DL algorithm can easily pick up on these hidden variations, which leads to inconsistent prediction performances over time.

- *QoS Guarantee*: According to the Universal Approximation Theory, if the optimal policy is a deterministic and continuous function, then the approximation can be arbitrarily accurate [62]. For some discrete optimization variables, such as the number of subcarriers and the user association decisions, the approximation of the DNN can be inaccurate due to the quantization of these discrete variables. As a result, the solution obtained from the DNN cannot fully guarantee the QoS requirements of different types of application scenarios.

Satisfying the QoS requirements in 5G NR for resource management cannot be addressed by only algorithm designs, but should take into account of enabling network architectures. Within different wireless scenarios targeted at guaranteeing QoS, algorithms should be analyzed specifically and designed to tackle the above-mentioned challenges.

1.3.1 Design of Low-Complexity Algorithm in Full-Dimension Massive MIMO Networks

To support application scenarios such as eMBB services, FD massive MIMO can boost the data rate and improve spectrum efficiency. Most of the existing works in FD massive MIMO beamforming designs have focused on a single-cell deployment only, and few works have been done in multi-cell networks. In the latter, the existence of inter-cell interference (ICI) leads to a significant signal-to-interference-and-noise ratio (SINR) and spectral efficiency degradation [64]. To tackle this problem, cooperative beamforming, which enables the cooperation between BSs to generate desired beams to each user in the network jointly, has been widely used to manage ICI and improve the spectrum efficiency of conventional MIMO cellular networks [65, 66, 67, 48, 49, 50, 51, 52]. In addition to cooperative beamforming approaches, the concept of user association has also been introduced in cellular massive MIMO networks to improve spectral efficiency [68]. User association brings additional degrees of freedom in communication between BSs and users, where users can be dynamically associated with the BS that provides a higher SINR and indirectly further suppresses ICI.

Such a joint optimization problem of cooperative beamforming and user association is a non-convex and NP-hard problem. As mentioned above, it is challenging to obtain a low-complexity algorithm, especially for large-scale massive MIMO networks. In our first research problem (Chapter 3), we propose a distributed Gaussian belief propagation (GaBP)-based algorithm to tackle the joint optimization problem in FD massive MIMO cellular networks [J2]. The main contributions of the chapter include:

- Incorporating the vertical direction in the joint design of beamforming and user association will require different channel models and thus different optimization variables. The channel models in our considered FD massive MIMO networks capture multiple channel paths in both horizontal and vertical directions. This is in contrast to the channel model used in the previous works on a conventional MIMO network with 1-D antenna arrays that capture only horizontal directions.

We introduce a binary indicator to represent the association between a user and a specific beam from a BS, defined as the binary beam association factors (BAF). This variable is in contrast to existing methods where the binary indicator represents the association between a user and a BS. By incorporating this new variable, i.e. BAF, and FD channel models, we can extract both horizontal and vertical beam directions and associate users with them, which has not been done before, leading to higher network capacity and a significantly lower computational cost.

- The above problem is formulated as a general non-convex and NP-hard problem, which is known to be unsolvable in linear time. To tackle this, we propose a novel low-complexity three-step GaBP-based distributed optimization solver to obtain feasible solutions efficiently. Compared with existing works that use alternating or sequential optimization approaches, the complexity of our proposed GaBP-based algorithm is significantly lower, which is linear to the size of the network. Note that the optimization solver can be used for other optimization problems that follow the same general optimization formulation structure in this chapter as well.

- Simulation results confirm that the proposed GaBP-based distributed solver that converges within only a few iterations significantly outperforms the non-cooperative scheme in [69] and the 1-D beamforming scheme in [44]. Moreover, the distributed computing methods based on GaBP can significantly reduce computation delays, which makes it favorable in practical cellular networks.

1.3.2 Design of Deep Learning in Mobile Edge Computing Systems

To enable ultra-low latency requirement of application scenarios, such as delay-sensitive and URLLC services, MEC can shorten the processing time of users' computational tasks in propagation delays and computation delays. How to implement the algorithm in real time so that the stringent latency requirement can be guaranteed

deserves further study. In Chapter 4, we propose a DL architecture, where a digital twin of the real network environment is used to train the DL algorithm off-line at a central server. From the pre-trained DNN, the mobility management entity (MME) can obtain user association schemes in real time. Specifically, we improve energy efficiency (EE) of users in an MEC system, subject to the delay and reliability constraints of URLLC services and the stability constraint of delay-tolerant services. To the authors' knowledge, this is the first research work that incorporates the concept of a digital twin with wireless networks [J1]. The main contributions of this chapter are summarized as follows.

- We propose a DL framework for improving EE of URLLC and delay-tolerant services in an MEC system with multiple APs. The normalized energy consumption, defined as the energy consumption per bit, is minimized by optimizing user association, resource allocation, and offloading probabilities. Within this framework, the optimal user association scheme is approximated by a DNN, which is first trained off-line at the central server and then sent to MME. For a given user association scheme, the resource allocation and task offloading policy is optimized at each AP.

- We establish the digital twin of the MEC system to explore labeled training samples, where the network topology, the channel and queueing models, and the fundamental rules are adopted in the digital twin to mirror the real system. The basic idea is to evaluate the normalized energy consumption, delay, and reliability of different user association schemes in the digital twin and save the best one in the memory as a training sample.

- Considering that the performance achieved by different user association schemes depends on the behavior of each AP, i.e., the resource allocation and task offloading policy, we propose an algorithm that can converge to the global optimal resource allocation with linear complexity. Then, the algorithm is used in the digital twin for performance evaluation.

Furthermore, simulation results show that the proposed DL framework can achieve a

lower normalized energy consumption with less computing complexity compared with an existing solution, and can approach the global optimal solution.

1.3.3 Design of Deep Transfer Learning with Quality-of-Service Guarantee

Our previous sections focus on how to design algorithms to guarantee QoS requirements of either one type of eMBB services or two types of services (delay-tolerant services and URLLC). In Chapter 5, we will extend our algorithm to support various types of services with QoS requirements on data rate, latency, and reliability [J3]. In addition, the proposed algorithm can also adjust to time-varying channel and traffic conditions with a high probability of QoS guarantee. To illustrate our approach, we consider an example problem that minimizes the total power consumption. The method can be easily extended to other kinds of problems, such as maximizing spectrum efficiency. Our main contributions are summarized below:

- We establish a deep learning framework that can obtain a near-optimal energy-efficient bandwidth and transmit power allocation scheme in 5G NR systems, where the QoS requirements of delay-tolerant, delay-sensitive, and URLLC services are satisfied. The optimization problem is a Mixed-Integer Non-Linear Programming (MINLP) since the number of subcarriers allocated to each user is an integer and the transmit power is a continuous variable.

- To obtain training samples, we develop an optimization algorithm to solve the MINLP, and analyze the convergence conditions, in which the algorithm converges to the global optimal solution of the MINLP. In addition, we prove that the conditions hold for delay-tolerant and delay-sensitive services. For URLLC, our analysis shows that the conditions hold in an asymptotic scenario, where the number of antennas is sufficiently large. Our numerical results validate that the conditions also hold in non-asymptotic scenarios.

- We observe that the output of a DNN cannot guarantee the QoS requirement of different types of services. To address this issue, we develop a cascaded structure of neural network (NN). The first NN obtains bandwidth allocation for multiple users. Given bandwidth allocation, the transmit power that is required to satisfy the QoS requirement of each user is obtained from the second NN.

- We adopt deep transfer learning to fine-tune pre-trained NNs in non-stationary wireless networks. The basic idea is to reuse the first several layers of the pre-trained NNs and train the last a few layers with a small number of new training samples. Numerical and simulation results show that the cascaded NNs can converge quickly in non-stationary wireless networks.

1.4 Thesis Outline

The rest of this thesis is organized as follows.

Chapter 2 starts by briefly introducing the concepts of network architectures, including full-dimension massive MIMO, mobile edge computing, and short block-length channel codes. Then, it presents the optimization solvers from optimization algorithms to deep learning. The main contributions of this thesis can be found in Chapter 3-5. Each chapter has its own separate notations which are tabulated and presented at the start for the readers' convenience. Chapter 3 focuses on the low-complexity algorithm design for eMBB services in FD massive MIMO cellular networks. In Chapter 4, we extend to hybrid 5G services (delay-tolerant services and URLLC) in MEC networks using deep learning to realize real-time resource management. In Chapter 5, a deep transfer learning framework is designed for the purposes of: adjusting to services in 5G NR with diverse QoS requirements and being flexible to non-stationary network parameters, and guaranteeing a high probability of QoS requirement. Finally, a summary of this thesis and its major findings are provided in Chapter 6 along with some concluding remarks and directions for future research.

Chapter 2

Preliminaries

In this chapter, we will introduce some network architectures and mathematical solvers used in the analyses and optimizations in the thesis.

2.1 Network Architectures

In this section, we provide some preliminary information on three enablers of wireless network architectures, which support the QoS requirements in 5G NR, i.e., high data rate, low latency, and high reliability.

2.1.1 Full Dimension Massive MIMO

Compared with the conventional MIMO system, FD massive MIMO is characterized by two main features: the number of antennas is asymptotically infinite and 2-D in angular domain. Before we analyze the 2-D antenna array, we first examine the asymptotic performance of a massive MIMO system with a 1-D antenna array.

Let us consider a downlink system in with a 1-D massive MIMO BS, having a uniform linear M antennas array serving an omni-directional single-antenna user. For simplicity, we first analyze this typical multiple-input single-output (MISO) system,

then we will extend to a multi-user multi-BS MIMO system. The antennas at the BS is uniformly distributed with an equal space of λD, where λ is the frequency carrier wavelength and $D = \frac{0.5}{\sqrt{(1-\cos(2\pi/M))^2+\sin(2\pi/M)^2}}$, resulting in a minimum distance between antenna elements being equal to $\lambda/2$.

Assuming no line-of-sight propagation, the channel vector between the BS to the user is $\mathbf{h} \in \mathbb{C}^M$ and we have $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{R})$, where $\mathbf{R}$ is the channel covariance matrix. By using Karhunen–Loeve representation, the channel vector can be expressed by

$$\mathbf{h} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{w}, \tag{2.1}$$

where $\mathbf{U} \in \mathbb{C}^{M\times r}$ is the unitary matrix of the eigenvectors of $\mathbf{R}$ corresponding to the nonzero eigenvalues, $\mathbf{\Lambda}$ is an $r \times r$ diagonal matrix with elements representing the nonzero eigenvalues of $\mathbf{R}$, and $\mathbf{w}$ is a random vector with each entry following $\mathcal{CN}(0, 1)$.

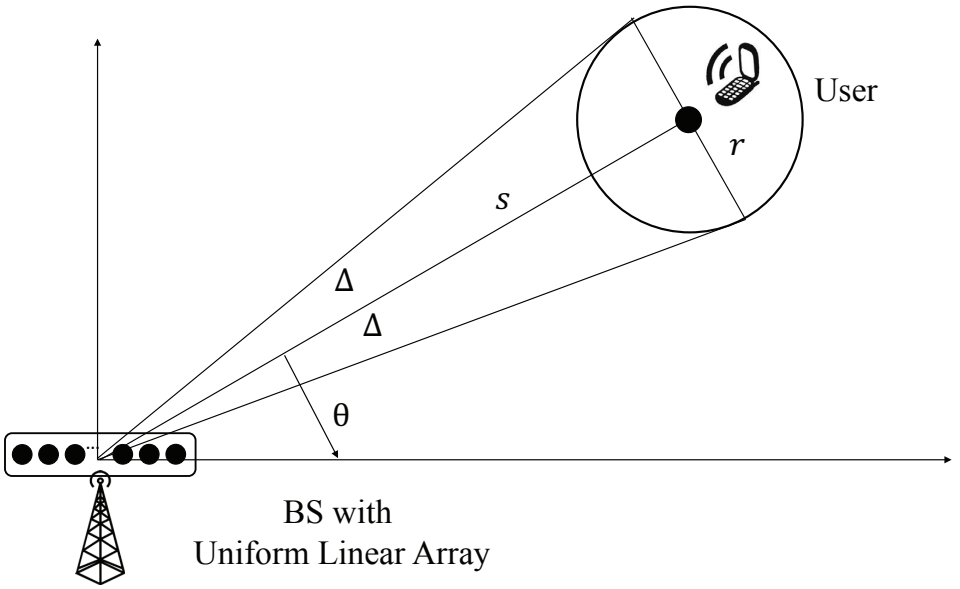

Figure 2.1 – One ring model with a uniform linear antenna array

For simplicity, we assume the one-ring model as in Fig. 2.1, where a user is located at a distance s and an azimuth angle of arrival (AoA) θ towards the BS, The user

is surrounded by a ring of scatterers with a radius r and the angular spread is $\Delta \approx \arctan(r/s)$. As in [70], the m,pth entry of correlation matrix, representing the correlation between the channel coefficients of antennas $1 \leq m,p \leq M$, is given by

$$[R]_{m,p} = \frac{1}{2\Delta}\int_{-\Delta+\theta}^{-\Delta+\theta} \exp^{-j2\pi D(m-p)\sin(\alpha)}\, d\alpha. \tag{2.2}$$

When the number of antennas M goes up to infinite, i.e., $M \to \infty$, the eigenvectors of a circulant matrix $\mathbf{C}$, i.e., $\mathbf{F}$, can approximate the eigenvectors of $\mathbf{R}$, i.e., $\mathbf{U}$, in the sense that

$$\lim_{M\to\infty} \frac{1}{M}||\mathbf{U}_{[a,b]}\mathbf{U}_{[a,b]}^H - \mathbf{F}_{[a,b]}\mathbf{F}_{[a,b]}^H||_F^2 = 0. \tag{2.3}$$

Here, $\mathbf{X}_{[a,b]}$ represents the vector for any interval $[a,b]$ such that the asymptotic eigenvalue cumulative density function (CDF) of the eigenvalues of $\mathbf{R}$ is continuous on $[a,b]$ and $\mathbf{X}^H$ is the conjugate-transpose operation for matrix $\mathbf{X}$. The detailed proofs can be found in Fact 1 and Fact 2 in [70].

As in [71], the eigenvectors of circulant matrix $\mathbf{C}$ form a unitary discrete Fourier transform (DFT) matrix with the m,pth entry of $[\mathbf{F}]_{m,p} = \frac{e^{-j2\pi(m-1)(p-1-M/2)/M}}{\sqrt{M}}$. This indicates that when the BS is equipped with a large number of antennas, the channel correlation matrix $\mathbf{R}$ can be well approximated by its circulant form $\mathbf{C}$. In other words, the corresponding matrix $\mathbf{U}$ of the channel covariance eigenvectors can be approximated by a unitary DFT matrix $\mathbf{F}$. The channel correlation $\mathbf{R}$ has the following property:

$$\lim_{M\to\infty} [\mathbf{R} - \mathbf{F}\mathbf{\Lambda}\mathbf{F}^H]_{m,p} = 0. \tag{2.4}$$

In this way, we can design the beamforming vectors by replacing $\mathbf{U}$ with its DFT approximation $\mathbf{F}$ in the regime of large M.

So far, we draw the conclusion based on the approximation of the DFT matrix for the channel correlation matrix in a 1-D antenna array for massive MIMO. Now we extend this approximation to a system where the antennas at the BS are deployed in

a 2-D grid and $M = M_V M_H$, where M_V and M_H are the number of antenna elements in the vertical and horizontal directions, respectively. With very large M_V and M_H, the horizontal and vertical channel correlations, i.e., $\mathbf{R}_H$ and $\mathbf{R}_V$, have the following properties:

$$\begin{aligned} \lim_{M_H \to \infty} [\mathbf{R}_H - \mathbf{F}_H \mathbf{\Lambda}_H \mathbf{F}_H^H]_{m,p} = 0, \\ \lim_{M_V \to \infty} [\mathbf{R}_V - \mathbf{F}_V \mathbf{\Lambda}_V \mathbf{F}_V^H]_{m,p} = 0, \end{aligned} \tag{2.5}$$

where $\mathbf{F}_H$ is the horizontal DFT matrix with

$$[\mathbf{F}_H]_{m,p} = \frac{1}{\sqrt{M_H}} e^{-j2\pi(m-1)(p-1-M_H/2)/M_H}, m, p = \{1, ..., M_H\} \tag{2.6}$$

and $\mathbf{F}_V$ is the vertical DFT matrix with

$$[\mathbf{F}_V]_{m,p} = \frac{1}{\sqrt{M_V}} e^{-j2\pi(m-1)(p-1-M_V/2)/M_V}, m, p = \{1, ..., M_V\}. \tag{2.7}$$

Substituting (2.5) into (2.1), we can extend the channel model from MISO system to K users N BSs MIMO system and transform it into the asymptotic approximation, given by

$$\mathbf{h}_{nk} = \mathbf{F} \mathbf{\Lambda}_{nk}^{\frac{1}{2}} \mathbf{w}_{nk}, k \in \{1, ..., K\}, n \in \{1, ..., N\}, \tag{2.8}$$

where $\mathbf{w}_{nk}$ is the vector form of matrix $\mathbf{W}_{nk} \in \mathbb{C}^{M_H \times M_V}$ with each entry following $\mathcal{CN}(0, 1)$, and

$$\mathbf{\Lambda}_{nk} = \mathbf{\Lambda}_{nk,H} \otimes \mathbf{\Lambda}_{nk,V} \tag{2.9}$$

is an $M \times M$ diagonal matrix with $\mathbf{\Lambda}_{nk} = \text{diag}\{\lambda_{nk}^1, \ldots, \lambda_{nk}^M\}$. In addition, the DFT matrix $\mathbf{F}$ is an $M \times M$ matrix obtained as

$$\mathbf{F} = \mathbf{F}_H \otimes \mathbf{F}_V. \tag{2.10}$$

With the help of the asymptotic approximation for FD massive MIMO as in (2.8), each specific orthogonal beam direction generated by nth BS can be represented by an eigenvector of $\mathbf{F}$. Moreover, the channel weights of kth user in these orthogonal

beam directions can be weighted by the eigenvalues of the diagonal matrix $\mathbf{\Lambda}_{nk}$.

2.1.2 Mobile Edge Computing

The initiative of MEC is to shorten the E2E delay from the end user to the remote cloud center by reducing delays in propagation and computation. The key components of a typical MEC system include mobile devices, such as users, remote sensors and autopilot vehicles, and MEC servers, normally equipped on BSs or APs. The MEC servers are normally deployed in close proximity with mobile devices and have small-scale data processing capability. The transmission between mobile devices and MEC servers is via wireless communications.

In the following, we will introduce the models of computation tasks, queueing models and computing models for MEC/local servers and energy consumption models for the the mobile devices in an MEC system, as illustrated in Fig. 2.2. The models of wireless communications will be introduced in the next Section 2.1.3.

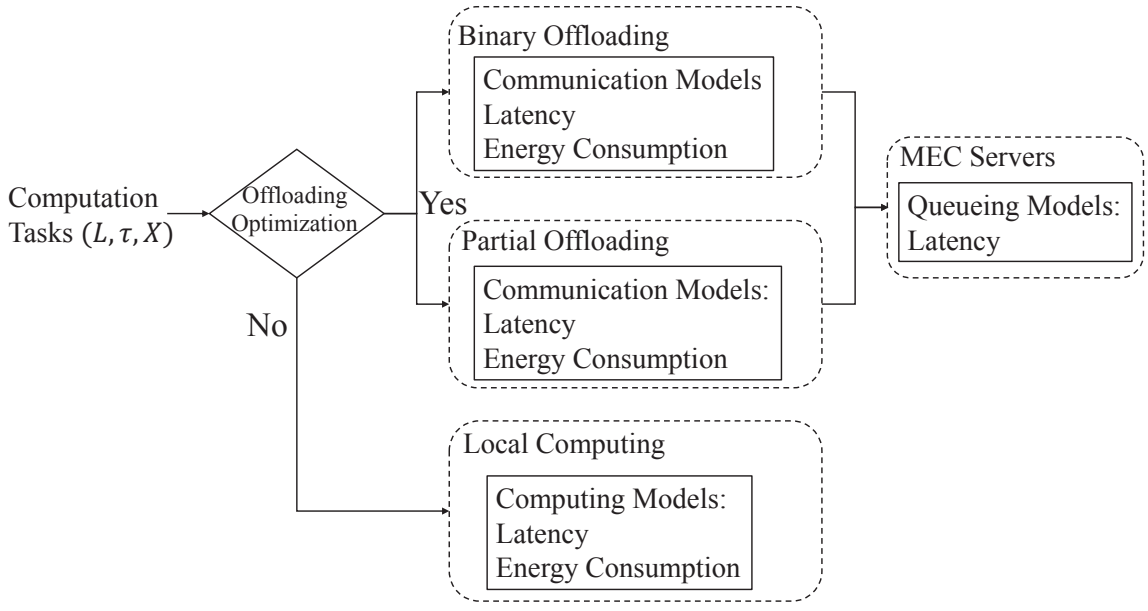

Figure 2.2 – Summary of MEC models

Computation Task Models

Mobile devices usually generate heavy computation tasks, which can be processed by either their local servers (an embedded processor equipped on a mobile device) or nearby MEC servers. Such tasks can be represented by three parameters (L, τ, X): the size of task L, the completion deadline τ and the computation workload X (CPU cycles/bit) [72].

The purpose of these parameters is to capture the information on the computation and communication demands of the task, QoS requirements (maximum latency) and potential energy consumption. For different scenarios in an MEC system, various parameters can be used to present similar information. For example, if queueing models are considered in wireless channels and computation processes, then the task arrival rate λ should replace the completion deadline τ to evaluate the queueing length and queueing delays [73].

When the computation tasks are processed at the MEC servers, there are two types of offloading methods depending on different applications: binary offloading and partial offloading. Binary offloading is suitable for highly integrated or small size tasks, which cannot be partitioned and need to be processed as a whole. For example, the size of a task generated by URLLC application is normally as small as 32 bytes [4]. Due to the small size of the task and stringent delay requirements, the entire URLLC task should be processed either by offloading to the MEC server or locally at the mobile device. On the other hand, some mobile applications include multiple components, (i.e., an AR application is composed of video process, location tracking, environment building, objects identifying and recognizer displaying [74]) which makes it possible for it to be partially processed at the MEC server as well as the local device.

Queueing Models

As indicated in [37], queueing effects should be taken into account in MEC systems to analyze latency components, especially when the system supports short packet

applications, i.e., URLLC. In an MEC system, a queueing model should be selected based on the supporting types of application scenarios.

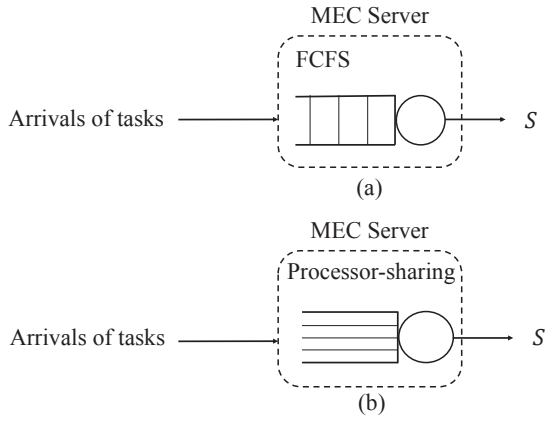

Figure 2.3 – Queueing model at MEC servers

As shown in Fig. 2.3(a), the packets of computation tasks from mobile devices are waiting in the queue of an MEC server and are served on a first-come-first-served (FCFS) basis. The service rate of the MEC server S is allocated to each task in turn, such that the processing delay is greatly shortened. However, the queueing delay bound can be violated if the distribution of the number of CPU circles required to process packets has a heavy tail. As a result, the total delay requirement (including processing delays and queueing delays) of users may not be satisfied.

Another type of queueing model is to let computation tasks arrive at the MEC server concurrently, as shown in Fig. 2.3(b). The service rate S is evenly allocated to all the computation tasks in the buffer, which is referred as to processor-sharing (PS). For example, when there are i tasks waiting in the queue, the service rate allocated to each task is S/i. With PS server, the application scenarios with short packets, such as URLLC, can bypass the queue of long packets in the MEC server.

Computing Models at Mobile Devices

The local computing is executed at the CPU of a mobile device. The performance of the CPU mainly depends on the CPU-cycle frequency f. In practice, the value

of f can adjust within the regime of $f \in [0, f^{\max}]$, with $f^{\max}$ being the maximum CPU-cycle frequency. For a task (L, τ, X), the execution latency at the local server (the mobile device) is

$$t_{\text{loc}} = \frac{LX}{f}. \tag{2.11}$$

Here, one can increase f to meet applications' latency requirements, but at the cost of higher CPU energy consumption.

The energy consumption of a CPU cycle is $k_0 f^2$ [75, 76], where k_0 is a coefficient depending on the chip architecture. As a result, for the task (L, τ, X), the energy consumption is given by

$$E_{\text{loc}} = k_0 LX f^2. \tag{2.12}$$

Computing Models at MEC Servers

The delay of a computation task offloading to its nearby MEC server consists of queueing delays in wireless transmissions and the server, transmission duration, and computation delays at the server. As indicated in [77, 78], the computation latency can be neglected compared with the transmission or queueing latency when the computation loads for MEC servers are much lower than their computation capacities. In the way, during the design of an MEC system, the computation loads should not exceed servers' computation capacities and the stability of the queue at the MEC servers should be guaranteed.

For a computation task L, τ, X, the transmission delay T_{d} can be calculated by

$$T_{\text{d}} = \frac{L}{R_{\text{mec}}}, \tag{2.13}$$

where R_{mec} is the achievable rate via wireless links and R_{mec} is proportional to the transmit power P_{mec}. One can increase the transmit power to reduce the transmission delay at the cost of higher energy consumption.

When mobile devices offload to MEC servers, the energy is consumed by transmitting

the computation task via wireless links, given by

$$E_{\text{mec}} = P_{\text{mec}} T_{\text{d}}. \tag{2.14}$$

2.1.3 Shannon Capacity v.s. Short Block-Length Channel Coding

The data rate gain has been treated as the main design objective for the previous generations of wireless communications. The way to accurately characterize the data rate depends on the information payload (data bits) required to transmit. In the channel coding process, the information payload is mapped into a continuous-time signal, which can be described by

$$N_{\text{bl}} \approx BT, \tag{2.15}$$

where B is the approximate bandwidth and T is the approximate duration. N_{bl} is the block-length, which represents the number of degrees of freedom required for the transmission of the information payload.

In information theory, when recovering the transmitted signal from distortion and noise over wireless channels, the receiver has higher probability of successfully decoding a signal when N_{bl} is large [79]. It is because by using the law of large numbers, i.e., $N_{\text{bl}} \to \infty$, the distortion and noise can be averaged out. However, when N_{bl} is small, the error in channel coding cannot be neglected. Next, we will examine the channel coding schemes for both long and short block-lengths.

Long Block-Length Channel Coding: Shannon Capacity

Long block-length refers to the fact that the information payload in a packet is much larger than its control information. Shannon capacity is used to asymptotically measure the achievable rate $R_{\max}(\epsilon_{\text{bl}}, N_{\text{bl}})$, when the packet error probability ϵ_{bl} can be arbitrarily small due to the infinite block-length [79].

For an additive white Gaussian noise (AWGN) channel, the Shannon capacity can be expressed as

$$\begin{aligned} C &= \lim_{\epsilon_{\text{bl}} \to 0, N_{\text{bl}} \to \infty} R_{\max}(\epsilon_{\text{bl}}, N_{\text{bl}}) \\ &= \log(1+\gamma), (\text{bits/second/Hz}), \end{aligned} \tag{2.16}$$

where γ is the signal-to-noise ratio (SNR). In wireless communications, the asymptotic achievable rate by Shannon capacity is given as

$$R_{\text{long}-\text{bl}} = B \log(1+\gamma), (\text{bits/second}). \tag{2.17}$$

Short Block-Length Channel Coding

In the short block-length regime, the control information in a short packet is no longer negligible in size compared to its information payload. Any inefficient encoding of control information may lead to the deterioration in the overall transmission [80].

To gain an accurate estimate of the achievable capacity in a short block-length regime, [20] derived $R_{\max}(\epsilon_{\text{bl}}, N_{\text{bl}})$ with ϵ_{bl} as

$$R_{\text{short}-\text{bl}} = B\left(C - \sqrt{\frac{V}{N_{\text{bl}}}} f_Q^{-1}(\epsilon_{\text{bl}})\right), (\text{bits/second}), \tag{2.18}$$

where $V = 1 - (1+\gamma)^{-2}$ is the channel dispersion, $f_Q^{-1}(\cdot)$ is the inverse function of the Q-function $f_Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$. Compared with (2.16), (2.18) indicates that a penalty on Shannon capacity is added to ensure a certain packet error probability ϵ_{lb} in the short block-length regime.

We denote the number of bits in a packet as b. With the given transmission duration T and $TR_{\text{short}-\text{bl}} = b$, the decoding error probability can be expressed by

$$\epsilon_{\text{bl}} = \int_0^\infty f_Q\left\{\sqrt{\frac{BT}{V}}\left[\ln(1+\gamma) - \frac{b\ln 2}{BT}\right]\right\} f_g(x) dx, \tag{2.19}$$

where $f_g(x)$ is the probability density function of the small-scale channel gain g.

2.2 Optimization Solvers

In this section, we will introduce the solvers used to obtain near-optimal solutions for NP-hard problems, ranging from optimization algorithms to machine learning ones.

2.2.1 Belief Propagation

BP is a graph-based message-passing algorithm which, by computing and propagating the marginal distribution of the variable node over the observations for the other variable nodes, is used to obtain a near-optimal solution for an objective function Q with discrete optimization variables $\boldsymbol{\beta} = [\beta_1, ..., \beta_K]$. Taking a maximization problem as an example, the objective function Q can be decomposed into K sub-objective functions, given by

$$Q(\boldsymbol{\beta}) = \max_{\boldsymbol{\beta}} \sum_{k \in K} Q_k(\boldsymbol{\beta}). \tag{2.20}$$

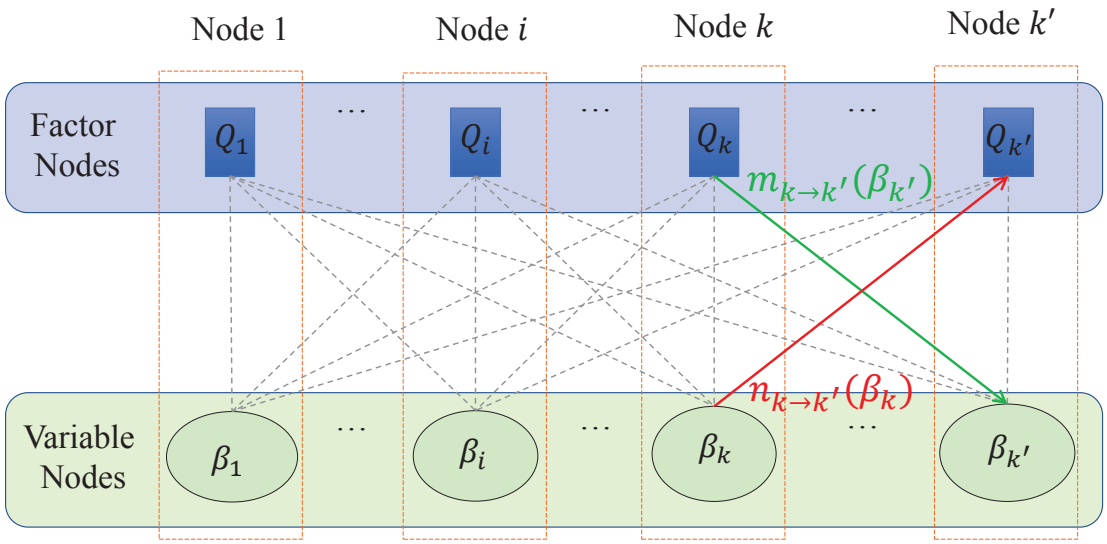

Figure 2.4 – Factor graph for belief propagation

We begin our introduction by building a factor graph, consisting of K variable nodes

and K factor nodes, as in Fig. 2.4. The edges connecting variable nodes and factor nodes, drawn as dotted lines, indicate the information communications between corresponding factor and variable nodes. The dashed boxes indicate that each node k consists of one variable node and one factor node, which represent optimization variable β_k and factor objective function $Q_k(\boldsymbol{\beta})$, respectively.

To explain the messages flowing in the factor graph, $n^{(t)}_{k\to k'}(\beta_k)$ and $m^{(t)}_{k'\to k}(\beta_k)$ denote messages from/to variable nodes to/from factor nodes in iteration t, shown as arrows in red and green in Fig. 2.4, respectively. These messages represent the estimates of the sub-objective function $Q_k(\boldsymbol{\beta})$ with the used states of variable β_k at factor and variable nodes. The message $n^{(t)}_{k\to k'}(\beta_k)$ is sent out by variable node k to factor node k' at iteration t. The equation (2.21) interprets the fact that the message depends on all messages coming into variable node k from neighboring factor node i, except for the one coming in from the target factor node k'.

$$n^{(t)}_{k\to k'}(\beta_k) = \delta(t-1)\sum\nolimits_{i\neq k'} m^{(t-1)}_{i\to k}(\beta_k). \tag{2.21}$$

Then, the message $n^{(t)}_{k\to k'}(\beta_k)$ is used as an input to calculate the message $m^{(t)}_{k'\to k}(\beta_k)$ sent by factor node k' to variable nodes in iteration t. The message $m^{(t)}_{k'\to k}(\beta_k)$ is obtained by summing all messages coming from variable nodes i into factor nodes k' (except for the factor node k), with the objective function $Q_{k'}(\boldsymbol{\beta})$ of factor node k'.

$$m^{(t)}_{k'\to k}(\beta_k) = \delta(t)\left(\sum\nolimits_{\boldsymbol{\beta}\setminus\beta_k} Q_{k'}(\boldsymbol{\beta}) + \sum\nolimits_{i\neq k} n^{(t)}_{i\to k'}(\beta_i)\right), \tag{2.22}$$

To prevent the computation overflow at each node, the messages flowing in factor graph at iteration t are normalized by the constant $\delta(t)$. We can calculate $\delta(t) = 1/\sqrt{\sum_{k=1}^{K}\bar{m}^{(t)}_k}$ at each variable node, where $\bar{m}^{(t)}_k = \sum_{\beta_k\in\boldsymbol{\beta}}\left(m^{(t)}_{k'\to k}(\beta_k)\right)^2$. The value of $\bar{m}^{(t)}_k$ is obtained from factor node k by summing all outgoing messages $m^{(t)}_{k'\to k}(\beta_k)$. The above steps are repeated in parallel until the following stopping criterion is met:

$$\|m^{(t)}_{k'\to k}(\beta_k) - m^{(t-1)}_{k'\to k}(\beta_k)\| \leq \varepsilon. \tag{2.23}$$

The belief of variable node can be obtained as $p^k(\beta_k) = \sum_{k' \in K} m^{(t)}_{k' \to k}(\beta_k)$, which represents the likelihood estimates of the optimization variables after the stopping criterion is met. The optimized $\hat{\beta}_k$ is then selected to achieve the maximum $p^k(\beta_k)$, that is $\hat{\beta}_k = \arg\max_{\beta_k} p^k(\beta_k)$.

We can use results from [81] to determine the convergence of BP algorithm.

Theorem 2.1. *[81, eq. (19)] With the independent variables in messages of BP algorithm, $\boldsymbol{\beta}$, the message $m^{(t)}_{k' \to k}$ converges to a fixed point ε when $t \to \infty$.*

To execute the BP algorithm, each node calculates its messages passing to the others simultaneously and computes the optimal solutions. Thus, the computation can be done in parallel at each node level. Considering the discrete variable β_k has M possible values, the message in (2.22) requires combinations of all the possible values. This computation grows exponentially in M.

2.2.2 Deep Learning

DL is a family of machine learning methods with the aid of DNN. The learning process can be categorized into supervised, unsupervised and reinforcement learning. In this thesis, we only focus on supervised deep learning.

In supervised DL, deep neural networks can be applied to infer a function from a set of labeled data, comprising inputs and desired outputs [82]. The deep neural networks can be tailored to the learning task by adjusting their weights and bias, with the purpose of minimizing a loss function representing the difference between the outputs of neural networks and the labeled outputs. As indicated in the Universal Approximation Theorem [62], any deterministic continuous function defined over a compact set can be approximated arbitrarily well with a neural network.

A DNN consists of multiple layers of neurons, as shown in Fig. 2.5. The set of all the weights and biases of the DNN is denoted as $\theta = \{\boldsymbol{W}^{[l]}, \boldsymbol{b}^{[l]}, l = 1, ..., L_{\text{DNN}}\}$, where L_{DNN} is the number of layers, and $\boldsymbol{W}^{[l]}$ and $\boldsymbol{b}^{[l]}$ are the weights and the biases in the

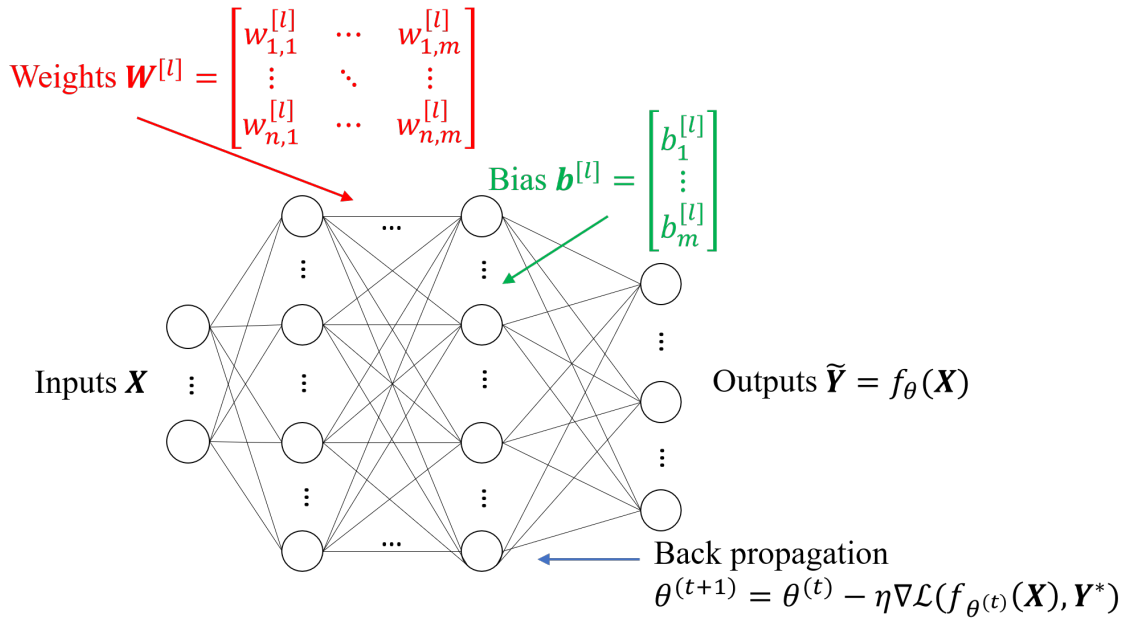

Figure 2.5 – The illustration of deep learning with deep neural networks

lth layer, respectively. The relation between the input and output of the lth layer can be expressed as

$$\boldsymbol{y}^{[l]} = f_\delta \left(\boldsymbol{W}^{[l]} \boldsymbol{x}^{[l]} + \boldsymbol{b}^{[l]} \right), \tag{2.24}$$

where $\boldsymbol{x}^{[l]}$ and $\boldsymbol{y}^{[l]}$ are the input and output of the lth layer, and the activation function, $f_\delta(x)$ is an element-wise operation of a vector.

Denoting the inputs of DNN are $\boldsymbol{X}$ and the labeled outputs are $\boldsymbol{Y}^*$; the DNN is used to approximate the function, mapping the relation $\boldsymbol{X} \rightarrow \boldsymbol{Y}^*$. The outputs of the DNN is $\tilde{\boldsymbol{Y}} = f_\theta(\boldsymbol{X})$, where $f_\theta(\cdot)$ is the approximator function with the DNN parameter θ.

The loss function is to measure the difference between $\boldsymbol{Y}^*$ and $\tilde{\boldsymbol{Y}}$, normally defined in the form of $\mathcal{L}(f_\theta(\boldsymbol{X}), \boldsymbol{Y}^*)$. For example, for discrete variables, a typical loss function is cross entropy $\mathcal{L} = (\boldsymbol{Y}^*)^T \log(\boldsymbol{Y}^*) + (1 - \tilde{\boldsymbol{Y}})^T \log(1 - \tilde{\boldsymbol{Y}})$ [83]. When the variables are continuous, quadratic loss $\mathcal{L} = \frac{1}{2}(\boldsymbol{Y}^* - \tilde{\boldsymbol{Y}})^2$ is used [84, 85].

By applying the chain rule of Calculus, the parameters of DNN θ at each layer is

updated through back propagation, given as

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla \mathcal{L}(f_{\theta^{(t)}}(\boldsymbol{X}), \boldsymbol{Y}^*), \tag{2.25}$$

where η is the learning rate to control the steps of parameter update.

2.2.3 Deep Transfer Learning

Authors in [86] pointed out one of the most serious problems in deep learning is insufficient data. It has been found that the scale and the accuracy performance of the model has a linear relationship to the size of the required amount of data. It is desirable to obtain as much data as possible to train a model. However, the data collection in some domains can be complex and expensive, thus it is difficult to build a large-scale data set.

In transfer learning, the requirements of training data and training time are relieved significantly, because the model in a target domain does not need to train from scratch, but transfers the knowledge learned from a source domain to the target domain. Before we introduce deep transfer learning, we will first clarify some notations and define transfer learning.

The learning process is to accomplish a learning *task* based on a data *domain*. According to the definitions in [87], a *domain* consists of a feature space and the corresponding marginal probability distribution, e.g., $\mathcal{D} = \{\chi, P(\boldsymbol{X})\}$, where $\boldsymbol{X} \in \chi$. A *task*, i.e., $\mathcal{T} = \{\boldsymbol{Y}^*, f(\boldsymbol{X})\}$, consists of a label space and an objective predictive function $f(\boldsymbol{X})$ that maps from $\boldsymbol{X}$ to $\boldsymbol{Y}^*$.

Definition 2.1. *(Transfer Learning)* [86, def. 1] For a learning task $\mathcal{T}_t$ and its data domain $\mathcal{D}_t$, a related learning task $\mathcal{T}_s$ with $\mathcal{D}_s$ can assist by discovering and transferring its latent knowledge, where $\mathcal{D}_s \neq \mathcal{D}_t$ and/or $\mathcal{T}_s \neq \mathcal{T}_t$. In this way, the performance of predictive function $f_{\mathcal{T}}(\cdot)$ for learning task $\mathcal{T}_t$ can be improved. In most cases, the size of $\mathcal{D}_s$ is much larger than that of $\mathcal{D}_t$.

Deep transfer learning is to apply deep neural networks to transfer the learned knowledge from a source domain to a target domain, for the purpose of saving the training time and training data. A formal definition is given by [86] as below.

Definition 2.2. *(Deep Transfer Learning)* [86, def. 2] For a transfer learning task $\{\mathcal{D}_s, \mathcal{T}_s, \mathcal{D}_t, \mathcal{T}_t, f_{\mathcal{T}}(\cdot)\}$, it is deep transfer learning when $f_{\mathcal{T}}(\cdot)$ is reflected by a deep neural network.

It is an interesting phenomenon that the first few layers of DNN can capture the general features of a domain and the last few layers map some more specific features. It motivates researchers to use the method of fine-tuning in deep transfer learning. By fixing the parameters of the first few layers of DNNs, the back propagation is activated only in the last few layers. In this way, only the parameters of DNN in the last few layers will be updated and fine-tuned to adjust to the target task $\mathcal{T}_t$ and domain $\mathcal{D}_t$.

Chapter 3

Design of Low Complexity Algorithm in Full-Dimension Massive MIMO Networks

3.1 Introduction

The cooperative beamforming designs in FD massive MIMO systems can be categorized into two groups based on the type of channel state information (CSI). In the first group, instantaneous CSI was assumed in [65, 66, 67, 48, 49, 50], where several cooperative beamforming designs were proposed to maximize the network capacity. In [65], authors for the first time proposed a novel optimization method and an alignment method to cancel the interference, which can guarantee the secure transmissions of users in cognitive radio networks. However, instantaneous CSI is challenging to acquire when large-scale antenna arrays are employed at the BSs. To alleviate the instantaneous CSI requirement, the second group of existing works exploits the statistical CSI (e.g., spatial channel correlation matrices between the users and BSs) for the cooperative beamforming design [51, 52].

In addition to cooperative beamforming approaches, the concept of user association has also been introduced in cellular massive MIMO networks to improve spectral

efficiency [68]. User association brings additional degrees of freedom in communication between BSs and users, where the latter can be dynamically associated with the BS, which provides a higher SINR and indirectly further suppresses ICI. The association between users and BSs is generally indicated by a binary user association factor (UAF) [53, 41, 42, 43, 54, 56, 88]. In [53], the user association is performed such that the data transmissions from BSs introduce the least interference in a multi-cell network, which therefore improves the network performance. In the current literature, a heuristic algorithm has been developed to solve user association problems in massive MIMO networks [44], where beamforming vectors are designed based on statistical CSI before user association decisions, i.e., a decoupled design is considered. The joint beamforming design and user association has also been widely studied to maximize the network capacity for conventional MIMO cellular networks [47, 45, 46, 55, 89]. Nevertheless, these existing works are designed for a 1-D antenna array and are incapable of capturing beams in both horizontal and vertical directions simultaneously, when FD MIMO arrays are employed at the BSs.

In this chapter, we focus on FD massive MIMO cellular networks and propose a joint cooperative beamforming and user association algorithm to maximize the downlink network capacity, subject to the SINR target of each user based only on statistical CSIs. The considered joint optimization problem is well known as a non-convex and NP-hard problem [90, 91, 92] due to the discrete association indicators. To obtain a feasible solution with a low computational cost, we first approximate statistical CSIs by using DFT matrices, where the eigenvectors represent the potential beam directions that each user can select and the eigenvalues indicate the channel weights in these beam directions, respectively. Building upon this model, we introduce a binary indicator to represent the association between a user and a specific beam from a BS, defined as the BAF. Accordingly, we transform the joint optimization problem into a problem on the BAFs only, such that the low-complexity distributed algorithm introduced in the following can be employed. The reformulated optimization problem is still non-convex due to the binary values of BAFs. To tackle this, we develop a three-step GaBP-based approach to obtain feasible and near-optimal BAFs in parallel

with a linear complexity. To be more specific, in the first step, by proving that maximizing the network capacity is approximately equivalent to maximizing the number of connected users in the network, we relax the binary BAFs into continuous values and further transform the optimization problem into a linear programming (LP) optimization. In the second step, the GaBP method is employed to obtain the solution of the LP in parallel with linear complexity. In the third step, we further propose two mapping algorithms to map these means back to binary forms such that the original user association constraint is satisfied. The central idea is to choose the largest continuous BAF by treating the continuous solutions from GaBP as incentive measures; i.e., a specific beam is more likely to be assigned to a user if the corresponding channel is in a better condition and the interference to the other users is smaller than in the other solutions. The first mapping scheme heuristically searches the largest BAFs that meet the constraints of the original optimization for all users, whose complexity is only linear to the number of users. The second mapping algorithm aims to directly find the binary BAFs that return the largest network capacity, whose complexity is polynomial to the number of BSs and users. Different performance-complexity trade-offs will be observed for the above two mapping algorithms.

The detailed notations and definitions used in this chapter are summarized in Table 3.1. The remainder of this chapter is organized as follows: Section 3.2 introduces the system model. The optimization problem of joint cooperative beamforming and user association is presented in Section 3.3. Section 3.4 demonstrates the three-step GaBP based distributed solver. In Section 3.5, the computational complexity is discussed. The simulation results are shown in Section 3.6, and Section 3.7 concludes this chapter.

3.2 System Model

We consider an FD multi-cell massive MIMO downlink in FDD mode with a total number of N BSs and K omni-directional single-antenna users, as shown in Fig. 3.1. We assume a 2D antenna array with a total number of A antenna elements at

Table 3.1 – Notations

Notation	Definition
x	scalar
$\mathbf{x}$	vector
$\mathbf{X}$	matrix
$\mathbf{I}_N$	$N \times N$ identity matrix
$(\cdot)^T$	transpose of a matrix
$(\cdot)^H$	conjugate-transpose of a matrix
$\succeq$	component-wise inequality between two vectors
$N(i)$	neighboring nodes of node i in Section 3.4
$\otimes$	Kronecker product of two matrices
$\odot$	Hadamard product of two matrices
$\lfloor \mathbf{X} \rfloor_a$	ath column of matrix $\mathbf{X}$
$[\mathbf{X}]_{p,q}$	(p,q)th element of matrix $\mathbf{X}$
$\text{vec}(\cdot)$	vectorization operation
$\text{diag}(\cdot)$	diagonal matrices
$\mathbb{E}$	expectation
$\mathbb{C}$	complex
$\mathbb{CN}$	complex Gaussian distribution
N	total number of BSs
K	total number of users
A	total number of antenna elements at each BS
N_v, N_h	number of antenna elements in the vertical , horizontal direction
$\mathbf{g}_{nk}$	channel vector between BS n and user k
α_{nk}	large-scale fading coefficient
$\mathbf{h}_{nk}$	small-scale channel coefficient vector
$\tilde{\mathbf{H}}_{nk}$	$N_h \times N_v$ random matrix with each entry following $\mathbb{CN}(0,1)$
$\mathbf{\Omega}_{nk,v}$, $\mathbf{\Omega}_{nk,h}$	vertical , horizontal channel correlation matrices
$\mathbf{U}_{nk,v}$, $\mathbf{U}_{nk,h}$	vertical , horizontal unitary matrix
$\mathbf{\Lambda}_{nk,h}$, $\mathbf{\Lambda}_{nk,v}$	vertical , horizontal eigenvalue matrix
λ_{nk}	eigenvalue of channel coefficient
$\mathbf{w}_{nk}$	unit-norm beamforming vector of user k from BS n
β_{nk}	user association factor between user k and BS n
Γ_k	SINR of user k
$\bar{\Gamma}_k$	minimum required SINR for user k
$\mathbf{F}_v$, $\mathbf{F}_h$	vertical , horizontal DFT matrix
γ_k	SLNR of γ_k
$\bar{\gamma}_k$	lower bound of the expected SLNR γ_k
θ_{nak}	beam association factor between user k and a-th beam of BS n

each BS and $A = N_v N_h$, where N_v and N_h are the number of antenna elements in the vertical and horizontal direction, respectively. To guarantee only one main lobe for each beamforming pattern, we assume half-wavelength antenna spacing for both horizontal and vertical directions [69]. The users are randomly distributed in both horizontal and vertical directions.

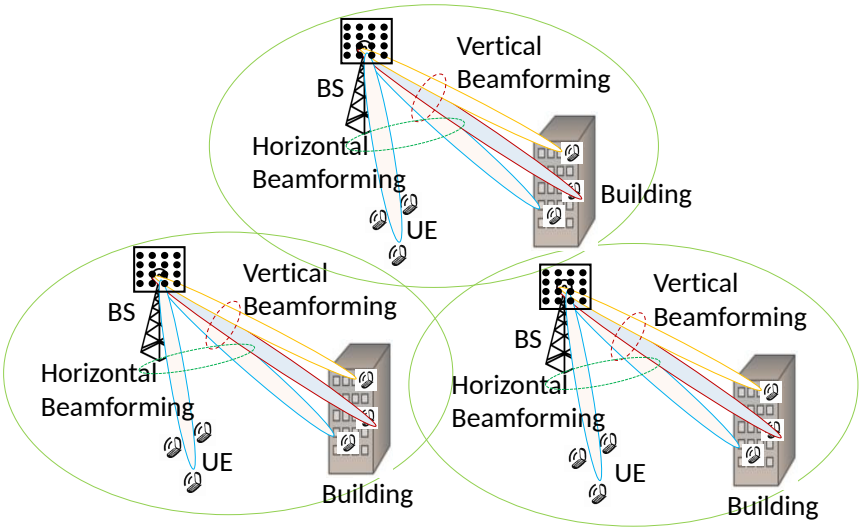

Figure 3.1 – Full Dimension Massive MIMO Deployment Scenario

3.2.1 Channel Model

Assuming no line-of-sight propagation, we express the channel vector between BS n and user k as

$$\mathbf{g}_{nk} = \sqrt{\alpha_{nk}}\mathbf{h}_{nk}, \tag{3.1}$$

where α_{nk} is the large-scale fading coefficient, and $\mathbf{h}_{nk} \in \mathbb{C}^{A\times 1}$ represents the small-scale channel coefficient vector between the antenna elements of BS antenna array and users. According to [69], $\mathbf{h}_{nk}$ can be written as

$$\mathbf{h}_{nk} = \mathrm{vec}(\mathbf{\Omega}_{nk,h}^{\frac{1}{2}}\tilde{\mathbf{H}}_{nk}\mathbf{\Omega}_{nk,v}^{\frac{1}{2}}). \tag{3.2}$$

In (3.2), $\tilde{\mathbf{H}}_{nk}$ is an $N_h \times N_v$ random matrix with each entry following $\mathbb{CN}(0,1)$. $\mathbf{\Omega}_{nk,h} \in \mathbb{C}^{N_h \times N_h}$ and $\mathbf{\Omega}_{nk,v} \in \mathbb{C}^{N_v \times N_v}$ are the horizontal and vertical channel correlation matrices between single-antenna user k and the antenna elements of BS n, respectively. Throughout this chapter, following [69, 93], we assume BSs have the information of $\mathbf{\Omega}_{nk,h}$ and $\mathbf{\Omega}_{nk,v}$. With the eigen decomposition, the horizontal and vertical channel correlation matrices can be written as

$$\begin{aligned} \mathbf{\Omega}_{nk,h} &= \mathbf{U}_{nk,h}\mathbf{\Lambda}_{nk,h}\mathbf{U}_{nk,h}^H, \\ \mathbf{\Omega}_{nk,v} &= \mathbf{U}_{nk,v}\mathbf{\Lambda}_{nk,v}\mathbf{U}_{nk,v}^H, \end{aligned} \tag{3.3}$$

where $\mathbf{U}_{nk,h}$ and $\mathbf{U}_{nk,v}$ are unitary matrices and $\mathbf{\Lambda}_{nk,h} = \text{diag}\{\lambda_{nk,h}^1, \ldots, \lambda_{nk,h}^{N_h}\}$, $\mathbf{\Lambda}_{nk,v} = \text{diag}\{\lambda_{nk,v}^1, \ldots, \lambda_{nk,v}^{N_v}\}$, This eigendecomposition reveals the fact that the channel vectors between BSs and users in (3.1) are composed of multiple horizontal and vertical directions, represented by unitary matrices $\mathbf{U}_{nk,h}$ and $\mathbf{U}_{nk,v}$, and the corresponding weights at these directions, indicated as $\mathbf{\Lambda}_{nk,h}$ and $\mathbf{\Lambda}_{nk,v}$, respectively.

3.2.2 System Model

In the considered multi-cell network, we assume BS n has only the statistical CSI of each user k, i.e., $\mathbf{\Omega}_{nk,h}$ and $\mathbf{\Omega}_{nk,v}$, whereas perfect CSI is assumed for each user [69]. We denote the unit-norm beamforming vector of user k from BS n as $\mathbf{w}_{nk} \in \mathbb{C}^{A\times 1}$ [69, 93] and the UAF as β_{nk}, respectively. $\beta_{nk} = 1$ if user k is connected to BS n, otherwise $\beta_{nk} = 0$. We assume that the data symbol s_k for user k is transmitted only by a single BS. Accordingly, the received signal at user k is expressed as

$$y_k = \sum_{n=1}^{N} \beta_{nk}\mathbf{g}_{nk}^H\mathbf{w}_{nk}s_k + \sum_{n=1}^{N}\sum_{j=1,j\neq k}^{K} \beta_{nj}\mathbf{g}_{nk}^H\mathbf{w}_{nj}s_j + z_k, \tag{3.4}$$

where z_k is the normalized complex additive white Gaussian noise with $\mathcal{CN}(0,\sigma_k^2)$. Based on (3.4) we obtain the expression of SINR of user k, given by

$$\Gamma_k = \frac{\sum_{n=1}^{N}\beta_{nk}\|\mathbf{g}_{nk}^H\mathbf{w}_{nk}\|^2}{\sigma_k^2 + \sum_{n=1}^{N}\sum_{j=1,j\neq k}^{K}\beta_{nj}\|\mathbf{g}_{nk}^H\mathbf{w}_{nj}\|^2}. \tag{3.5}$$

Subsequently, the ergodic achievable rate of user k can be expressed as

$$R_k = \mathbb{E}[\log_2(1 + \Gamma_k)], \tag{3.6}$$

and the ergodic sum rate of the network is

$$R = \sum_{k=1}^{K} R_k. \tag{3.7}$$

Accordingly, we construct the optimization problem that maximizes the network capacity on the UAF and the beamforming vectors as

$$\max_{\boldsymbol{\beta}, \mathbf{W}} R(\boldsymbol{\beta}, \mathbf{W}) \tag{3.8}$$

$$\text{s.t. } \mathbb{E}[\Gamma_k(\boldsymbol{\beta}, \mathbf{W})] \begin{cases} \geq \bar{\Gamma}_k, \text{if } \sum_{n=1}^{N} \beta_{nk} = 1, \\ = 0, \text{otherwise}, \end{cases} \tag{3.8.1}$$

$$\sum_{n=1}^{N} \beta_{nk} \leq 1, \tag{3.8.2}$$

$$\beta_{nk} \in \{0, 1\}, \tag{3.8.3}$$

where $\bar{\Gamma}_k$ is the minimum required SINR for user k, $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_K^T]^T$ represents a column vector with $\boldsymbol{\beta}_k = [\beta_{1k}, \ldots, \beta_{Nk}]^T$, and $\mathbf{W} = [\mathbf{w}_{11}, \ldots, \mathbf{w}_{1K}, \mathbf{w}_{21}, \ldots, \mathbf{w}_{NK}]$ represents the beamforming matrix. The constraint (3.8.1) is to guarantee that each associated user meets its minimum SINR requirement, and the constraint (3.8.2) indicates that each user k can be associated with at most one BS.

3.3 Optimization Problem Formulation

The formulated optimization problem (3.8) on the network capacity by jointly optimizing the beamforming vectors and the user association factors is well-known to be a non-convex and NP-hard problem [90, 91, 92]. Directly solving this non-convex

problem in our scenario with a large number of optimization variables is impossible in practice. To develop a practical solution for (3.8), we propose a distributed solver based on GaBP with low computational complexity and fast convergence speed. To enable the distributed GaBP algorithm, firstly the original optimization problem needs to be transformed into linear programming whose objective function will reflect and represent the network capacity, as detailed below. While the transformation into a linear problem inevitably leads to a sub-optimal result, this transformation allows us to implement the low-complexity and distributed GaBP-based algorithm for the considered scenario, which will be introduced in Section 3.4. The numerical results will also show that the proposed scheme is promising in terms of both performance and complexity, compared to the benchmark schemes in the literature.

In the following, we show how (3.8) is transformed into an optimization on UAF β_{nk} only. Firstly, we represent the wireless channel vector as DFT matrices by employing the asymptotic analysis. Then, we use the average signal-to-leakage-and-noise ratio (SLNR) metric to derive the optimal $\mathbf{w}_{nk}$ with respect to β_{nk}.

3.3.1 Asymptotic Analysis For Massive MIMO Channel

We assume a large number of antennas at BSs, i.e., $A \gg 1$, which is equivalent to $N_v \gg 1$ and $N_h \gg 1$. According to [70], the correlation matrices of a large-scale FD antenna array can be asymptotically approximated as DFT matrices. A similar result has also been adopted in [69, 93]. With very large N_v and N_h, the horizontal and vertical channel correlations of user k to BS n have the following properties:

$$\begin{aligned} \lim_{N_h \to \infty} [\mathbf{\Omega}_{nk,h} - \mathbf{F}_h \mathbf{\Lambda}_{nk,h} \mathbf{F}_h^H]_{pq} = 0, \\ \lim_{N_v \to \infty} [\mathbf{\Omega}_{nk,v} - \mathbf{F}_v \mathbf{\Lambda}_{nk,v} \mathbf{F}_v^H]_{pq} = 0, \end{aligned} \tag{3.9}$$

where $\mathbf{F}_h$ is the horizontal DFT matrix with

$$[\mathbf{F}_h]_{p,q} = \frac{1}{\sqrt{N_h}} e^{-j2\pi(p-1)(q-1-N_h/2)/N_h}, p, q = \{1, ..., N_h\} \tag{3.10}$$

and $\mathbf{F}_v$ is the vertical DFT matrix with

$$[\mathbf{F}_v]_{p,q} = \frac{1}{\sqrt{N_v}} e^{-j2\pi(p-1)(q-1-N_v/2)/N_v}, p, q = \{1, ..., N_v\}. \tag{3.11}$$

Substituting (3.9) into (3.1), the asymptotic approximation of the channel vector can be transformed into

$$\mathbf{g}_{nk} \approx \sqrt{\alpha_{nk}} \mathbf{F} \mathbf{\Lambda}_{nk}^{\frac{1}{2}} \tilde{\mathbf{h}}_{nk}, \tag{3.12}$$

where $\tilde{\mathbf{h}}_{nk} = \text{vec}(\tilde{\mathbf{H}}_{nk})$ and

$$\mathbf{\Lambda}_{nk} = \mathbf{\Lambda}_{nk,h} \otimes \mathbf{\Lambda}_{nk,v} \tag{3.13}$$

is an $A \times A$ diagonal matrix with $\mathbf{\Lambda}_{nk} = \text{diag}\{\lambda_{nk}^1, \ldots, \lambda_{nk}^A\}$. In (3.12), $\mathbf{F}$ is an $A \times A$ matrix obtained as

$$\mathbf{F} = \mathbf{F}_h \otimes \mathbf{F}_v. \tag{3.14}$$

Each column of $\mathbf{F}$ is an eigenvector corresponding to a specific orthogonal beam direction generated from the massive MIMO BSs. The elements of the diagonal matrix $\mathbf{\Lambda}_{nk}$ are eigenvalues representing the channel weights of user k in these orthogonal beam directions generated by BS n.

3.3.2 SLNR Based FD Beamforming

Based on the SINR expression in (3.5), it is observed that an update of the beamforming vector $\mathbf{w}_{nk}$ for one user k leads to the variations in the SINR for other users, which further results in variations in the beamforming designs $\mathbf{w}_{nj}$ for them. To avoid the above described interdependency between users' beamforming vectors $\mathbf{w}_{nk}$ and $\mathbf{w}_{nj}, j \neq k$, we adopt a SLNR metric instead, which is defined as

$$\gamma_k(\boldsymbol{\beta}, \mathbf{W}) = \frac{\sum_{n=1}^{N} \beta_{nk} \|\mathbf{g}_{nk}^H \mathbf{w}_{nk}\|^2}{\sigma_k^2 + \sum_{n=1}^{N} \sum_{j=1, j\neq k}^{K} \beta_{nj} \|\mathbf{g}_{nj}^H \mathbf{w}_{nk}\|^2}, \tag{3.15}$$

where the second term $\sum_{n=1}^{N} \sum_{j=1,j\neq k}^{K} \|\mathbf{g}_{nj}^H \mathbf{w}_{nk}\|^2$ in the denominator represents the total power leaked from user k's beamforming direction to other users' channel directions. By using the SLNR formulation (3.15) to approximate (3.5), the dependency

between users' beamforming vectors in the optimization problem is removed. However, changing the performance metric from SINR to SLNR results in loss of capacity. This is because the objective network capacity for the minimization of the interference is converted to that for minimizing signal leakage. Note that in previous beamforming design work [69, 70, 94, 93], the SLNR criterion has been used to achieve the trade-off between complexity and network capacity.

Based on the SLNR metric, we derive the beamforming vector of user k as a function of UAF $\boldsymbol{\beta}$. First, we obtain the lower bound of the expected SLNR γ_k defined in (3.15) as

$$\begin{aligned}
&\mathbb{E}[\gamma_k](\boldsymbol{\beta}, \mathbf{W}) \\
&= \mathbb{E}\left[\frac{\sum_{n=1}^{N} \beta_{nk} \|\mathbf{g}_{nk}^H \mathbf{w}_{nk}\|^2}{\sigma_k^2 + \sum_{n=1}^{N} \sum_{j=1, j\neq k}^{K} \beta_{nj} \|\mathbf{g}_{nj}^H \mathbf{w}_{nk}\|^2}\right], \\
&\overset{(a)}{\geq} \frac{\mathbb{E}\left[\sum_{n=1}^{N} \beta_{nk} \|\mathbf{g}_{nk}^H \mathbf{w}_{nk}\|^2\right]}{\mathbb{E}\left[\sigma_k^2 + \sum_{n=1}^{N} \sum_{j=1, j\neq k}^{K} \beta_{nj} \|\mathbf{g}_{nj}^H \mathbf{w}_{nk}\|^2\right]}, \\
&\overset{(b)}{\geq} \frac{\sum_{n=1}^{N} \beta_{nk} \alpha_{nk} \mathbf{w}_{nk}^H \mathbf{F} \boldsymbol{\Lambda}_{nk} \mathbf{F}^H \mathbf{w}_{nk}}{\sigma_k^2 + \sum_{n=1}^{N} \sum_{j=1, j\neq k}^{K} \beta_{nj} \alpha_{nj} \mathbf{w}_{nk}^H \mathbf{F} \boldsymbol{\Lambda}_{nj} \mathbf{F}^H \mathbf{w}_{nk}} \triangleq \bar{\gamma}_k,
\end{aligned} \tag{3.16}$$

where step (a) is derived by applying Mullens inequality [95], i.e., $\mathbb{E}[X/Y] \geq \mathbb{E}[X]/\mathbb{E}[Y]$ if X and Y are independent random variables. Step (b) is obtained by substituting (3.12) into (3.16.a), where $\mathbb{E}[\tilde{\mathbf{h}}_{nk} \tilde{\mathbf{h}}_{nk}^H] = \mathbf{I}_A$.

Without loss of generality, we denote the index of the BS chosen by user k for a given $\boldsymbol{\beta}$ by $\tilde{n}$, and we can rewrite $\bar{\gamma}_k$ in (3.16) with a given $\boldsymbol{\beta}$ as

$$\bar{\gamma}_k(\boldsymbol{\beta} | n = \tilde{n}) = \frac{\beta_{\tilde{n}k} \alpha_{\tilde{n}k} \mathbf{w}_{\tilde{n}k}^H \mathbf{F} \boldsymbol{\Lambda}_{\tilde{n}k} \mathbf{F}^H \mathbf{w}_{\tilde{n}k}}{\sigma_k^2 + \sum_{j=1, j\neq k}^{K} \beta_{\tilde{n}j} \alpha_{\tilde{n}j} \mathbf{w}_{\tilde{n}k}^H \mathbf{F} \boldsymbol{\Lambda}_{\tilde{n}j} \mathbf{F}^H \mathbf{w}_{\tilde{n}k}}. \tag{3.17}$$

For the condition of $\boldsymbol{\beta}_k = \mathbf{0}$, i.e., $\beta_{\tilde{n}k} = 0$, which means that user k does not connect to any BSs, (3.17) still holds. Based on the revised expression for $\bar{\gamma}_k$ in (3.17), we rewrite $\bar{\gamma}_k(\boldsymbol{\beta} | n = \tilde{n})$ in (3.17) in a matrix format as

$$\bar{\gamma}_k(\boldsymbol{\beta} | n = \tilde{n}) = \mathbf{w}_{\tilde{n}k}^H \mathbf{F} \mathrm{diag}(\mathbf{L}_k(\boldsymbol{\beta} | n = \tilde{n})) \mathbf{F}^H \mathbf{w}_{\tilde{n}k}, \tag{3.18}$$

where $\text{diag}(\mathbf{L}_k(\boldsymbol{\beta}|n=\tilde{n}))$ is a diagonal matrix with eigenvalue entries on the diagonal representing the weights for different eigenvectors. The entry of $\mathbf{L}_k(\boldsymbol{\beta}|n=\tilde{n}) = [l_k^1(\boldsymbol{\beta}|n=\tilde{n}), ..., l_k^a(\boldsymbol{\beta}|n=\tilde{n}), ..., l_k^A(\boldsymbol{\beta}|n=\tilde{n})]$ is given by

$$l_k^a(\boldsymbol{\beta}|n=\tilde{n}) = \frac{\beta_{\tilde{n}k}\alpha_{\tilde{n}k}\lambda_{\tilde{n}k}^a}{\sigma_k^2 + \sum_{j=1,j\neq k}^K \beta_{\tilde{n}j}\alpha_{\tilde{n}j}\lambda_{\tilde{n}j}^a}, \forall a \in \{1,\ldots,A\}. \tag{3.19}$$

Subsequently, in view of the Rayleigh-Ritz quotient result [96], the optimal beamforming vector $\mathbf{w}_{\tilde{n}k}$ that maximizes $\bar{\gamma}_k(\boldsymbol{\beta}|n=\tilde{n})$ is the eigenvector corresponding to the largest entry of $\mathbf{L}_k(\boldsymbol{\beta}|n=\tilde{n})$, i.e. λ_k^{max}. The index of λ_k^{max} is a function of $\boldsymbol{\beta}$, given as

$$x_k(\boldsymbol{\beta}) = \max_a(l_k^a(\boldsymbol{\beta}|n=\tilde{n})), \forall a \in \{1,\ldots,A\} \tag{3.20}$$

and the corresponding eigenvector is the $x_k(\boldsymbol{\beta})$-th column of $\mathbf{F}$. As a result, the optimal beamforming vector that leads to $\bar{\gamma}_k = \lambda_k^{max}$ is written as

$$\mathbf{w}_{\tilde{n}k}(\boldsymbol{\beta}|n=\tilde{n}) = \lfloor \mathbf{F} \rfloor_{x_k(\boldsymbol{\beta})}. \tag{3.21}$$

By substituting (3.21) and (3.17) into (3.16), the maximum lower bound of the expected SLNR can be updated as

$$\begin{aligned}\mathbb{E}[\gamma_k](\boldsymbol{\beta}) &\geq \frac{\beta_{\tilde{n}k}\alpha_{\tilde{n}k}\lfloor \mathbf{F} \rfloor_{x_k(\boldsymbol{\beta})}^H \mathbf{F}\boldsymbol{\Lambda}_{\tilde{n}k}\mathbf{F}^H \lfloor \mathbf{F} \rfloor_{x_k(\boldsymbol{\beta})}}{\sigma_k^2 + \sum_{j=1,j\neq k}^K \beta_{\tilde{n}j}\alpha_{\tilde{n}j}\lfloor \mathbf{F} \rfloor_{x_k(\boldsymbol{\beta})}^H \mathbf{F}\boldsymbol{\Lambda}_{\tilde{n}j}\mathbf{F}^H \lfloor \mathbf{F} \rfloor_{x_k(\boldsymbol{\beta})}} \\ &= \frac{\beta_{\tilde{n}k}\alpha_{\tilde{n}k}\lambda_{\tilde{n}k}^{x_k(\boldsymbol{\beta})}}{\sigma_k^2 + \sum_{j=1,j\neq k}^K \beta_{\tilde{n}j}\alpha_{\tilde{n}j}\lambda_{\tilde{n}j}^{x_k(\boldsymbol{\beta})}}.\end{aligned} \tag{3.22}$$

By substituting $\mathbf{w}_{\tilde{n}k}(\boldsymbol{\beta}|n=\tilde{n}) = \lfloor \mathbf{F} \rfloor_{x_k(\boldsymbol{\beta})}$ into (3.5) and (3.6), we can rewrite the ergodic achievable rate of user k as

$$\begin{aligned}R_k =& \mathbb{E}[\log_2(1+\Gamma_k(\boldsymbol{\beta}))] \\ \overset{(a)}{\leq}& \log_2(1+\mathbb{E}[\Gamma_k(\boldsymbol{\beta})]) \\ =& \log_2\left(1+\mathbb{E}\left[\frac{\beta_{\tilde{n}k}\alpha_{\tilde{n}k}\lambda_{\tilde{n}k}^{x_k(\boldsymbol{\beta})}}{\sigma_k^2 + \sum_{j=1,j\neq k}^K \beta_{\tilde{n}j}\alpha_{\tilde{n}k}\lambda_{\tilde{n}k}^{x_j(\boldsymbol{\beta})}}\right]\right),\end{aligned} \tag{3.23}$$

where step (a) uses the property of $\mathbb{E}[\log(x)] \leq \log(\mathbb{E}[x])$. With the above transformations, the optimization on both the beamforming weights in $\mathbf{W}$ and UAF $\boldsymbol{\beta}$ is transformed into an optimization problem on UAF $\boldsymbol{\beta}$ only, given as

$$\max_{\boldsymbol{\beta}} \sum_{k=1}^{K} R_k(\boldsymbol{\beta}) \tag{3.24}$$

$$\text{s.t. } \mathbb{E}[\Gamma_k(\boldsymbol{\beta})] \geq \begin{cases} \bar{\Gamma}_k, \text{if } \sum_{n=1}^{N} \beta_{nk} = 1, \\ 0, \text{otherwise}, \end{cases} \tag{3.24.1}$$

$$\sum_{n=1}^{N} \beta_{nk} \leq 1, \tag{3.24.2}$$

$$\beta_{nk} \in \{0, 1\}. \tag{3.24.3}$$

The optimization problem (3.24) can be solved in parallel by the standard BP method with discrete variables $\boldsymbol{\beta}$ as in [50]. To implement the BP algorithm, a factor graph needs to be developed with factor and variable nodes, where the BSs iteratively calculate the messages flowing between these nodes for each user k simultaneously and obtain the optimal solutions. However, the practical implementation of the above BP algorithm in a dense network can be challenging, due to the significant computational costs in calculating these messages, which grow exponentially in N and K.

3.4 Proposed Gaussian Belief Propagation-based Algorithms

To obtain a low-complexity solver, the BAF θ_{nak} is introduced to indicate the association between different beams and users. Subsequently, we propose a three-step GaBP based distributed optimization solver to solve non-convex NP-hard problems, where we first perform the binary relaxation and formulate a linear programming, followed by the GaBP algorithm to obtain the optimal continuous BAFs, and finally map these continuous BAFs into binary values. Two mapping algorithms are further

proposed to map the obtained BAFs into binary forms.

3.4.1 Binary Relaxation and Linear Programming Formulation

Based on the description in Section 3.3 that the beamforming vector can be represented by a specific beam generated by massive MIMO BSs, by adding one more dimension to UAF, we introduce the BAF, i.e., θ_{nak}, where $a \in A$ denotes the a-th beam as well as a-th column of $\mathbf{F}$. The BAF $\theta_{nak} = 1$ and $\theta_{nak} = 0$ represent whether user k is associated with a-th beam of BS n or not, respectively. With the introduced BAF, each $\boldsymbol{\theta}_k$ in $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T]^T$ is further expressed as $\boldsymbol{\theta}_k = [\boldsymbol{\theta}_{1k}^T, \ldots, \boldsymbol{\theta}_{Nk}^T]^T$ with $\boldsymbol{\theta}_{nk} = [\theta_{n1k}, \ldots, \theta_{nAk}]^T$, and the SINR can be expressed as

$$\mathbb{E}[\Gamma_k(\boldsymbol{\theta})] = \frac{\sum_{n=1}^{N} \alpha_{nk} \sum_{a=1}^{A} \theta_{nak} \lambda_{nk}^a}{\sigma_k^2 + \sum_{n=1}^{N} \sum_{j=1, j\neq k}^{K} \alpha_{nk} \sum_{b=1}^{A} \theta_{nbj} \lambda_{nk}^b}. \tag{3.25}$$

Subsequently, we approximate the maximization on the network capacity to that on the number of connected users. With the expression of the expected SINR in (3.25), the ergodic achievable rate of user k in (3.6) can be rewritten as

$$\begin{aligned} R_k =& \mathbb{E}[\log_2 (1 + \Gamma_k(\boldsymbol{\theta}))] \\ \overset{(a)}{\leq}& \log_2 (1 + \mathbb{E}[\Gamma_k(\boldsymbol{\theta})]) \\ \overset{(b)}{=}& \sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak} \log_2 (1 + \mathbb{E}[\Gamma_k(\boldsymbol{\theta})]) . \end{aligned} \tag{3.26}$$

In (3.26), step (a) uses the property of $\mathbb{E}[\log(x)] \leq \log(\mathbb{E}[x])$. The equivalence of step (b) of (3.26) holds with the summation term $\sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak}$ representing the binary association decision of user k, since each user can be associated with at most one BS

at a time. The ergodic achievable rate for the network can be written as

$$\begin{aligned} R &= \sum_{k=1}^{K} R_k \\ &\leq \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak} \log_2 \left(1 + \mathbb{E}[\Gamma_k(\boldsymbol{\theta})]\right). \end{aligned} \tag{3.27}$$

Observing (3.27), we can find that both the summation term $\sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak}$ and the logarithmic function $\log_2 \left(1 + \mathbb{E}[\Gamma_k(\boldsymbol{\theta})]\right)$ are functions of BAFs. By applying the monotonic increasing property of the logarithmic function, we observe that the decreasing speed of the logarithmic function is much slower than the increasing speed of the summation term outside the logarithmic function, which represents the number of connected users. Thus, the ergodic sum rate increase is mainly reflected by the increase in the number of connected users; i.e., for a given SINR target, the increase in $\sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak}$ leads to the increase in R. Therefore, we can approximate the optimization problem (3.24) as a maximization on the number of connected users subject to SINR constraints, given by

$$\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{a=1}^{A} \theta_{nak}, \tag{3.28}$$

$$\text{s.t. } \mathbb{E}[\Gamma_k(\boldsymbol{\theta})] \geq \begin{cases} \bar{\Gamma}_k, \text{if } \sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak} = 1, \\ 0, \text{otherwise}, \end{cases} \tag{3.28.1}$$

$$\sum_{n=1}^{N} \sum_{a=1}^{A} \theta_{nak} \leq 1, \tag{3.28.2}$$

$$\theta_{nak} \in \{0, 1\}. \tag{3.28.3}$$

It is worth mentioning that the proposed approximation from (3.27) to (3.28) changing the objective function from maximizing the network capacity to maximizing the number of connected users will lead to performance loss. This is because the new objective function pursues the maximum number of connected users. As a result, the solver tends not to select the user in a good channel condition if it interferes with

others.

However, optimization problem (3.28) is still difficult to solve due to the binary form of BAFs. To obtain a feasible solution, we propose the binary relaxation of (3.28.3) as $0 \leq \theta_{nak} \leq 1, \forall n \in N, \forall a \in A, \forall k \in K$. We now rewrite (3.28) as linear programming as

$$\min_{\boldsymbol{\theta}} -\mathbf{1}_{nak}^{T}\boldsymbol{\theta} \tag{3.29}$$

$$\text{s.t. } \boldsymbol{\rho}^{T}\boldsymbol{\theta} \succeq \bar{\boldsymbol{\Gamma}} \odot \mathbf{z}, \tag{3.29.1}$$

$$\sum_{n=1}^{N}\sum_{a=1}^{A}\theta_{nak} \leq 1, \tag{3.29.2}$$

$$\theta_{nak} \in [0,1], \tag{3.29.3}$$

where $\mathbf{1}_{nak} = [1,1,...,1]^{T}, \in \mathbb{C}^{NAK\times 1}$, $\bar{\boldsymbol{\Gamma}}$ is a $K \times 1$ vector of $\bar{\Gamma}_k, \forall k \in K$ and $\mathbf{z}$ is a $K \times 1$ vector with the entry of noise z_k in (3.4). The entry of the matrix $\boldsymbol{\rho} \in \mathbb{R}^{NAK\times K}$ is defined as

$$\boldsymbol{\rho}_{kj} = \begin{cases} -\bar{\Gamma}_k \left[\alpha_{1j}\lambda_{1j}^{1}, \ldots, \alpha_{nj}\lambda_{nj}^{b}, \ldots, \alpha_{Nj}\lambda_{Nj}^{A}\right]^{T}, \text{for } \theta_{nbj}, j \neq k, \\ \left[\alpha_{1k}\lambda_{1k}^{1}, \ldots, \alpha_{nk}\lambda_{nk}^{a}, \ldots, \alpha_{Nk}\lambda_{Nk}^{A}\right]^{T}, \text{for } \theta_{nak}, j = k. \end{cases} \tag{3.30}$$

From the optimization (3.29), we observe that optimality is achieved when all the users in the network are associated with different beams of BSs, i.e., $\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{a=1}^{A}\theta_{nak} = K$, with SINR constraints satisfied. The constraint in (3.29.1) with equality is to represent that the SINRs for all the users in the network are exactly equal to their SINR thresholds. Consequently, the user association decision will allocate the resources in beams and BSs until the balance in transmission SINRs for all users is achieved, which is equivalent to $\boldsymbol{\rho}^{T}\boldsymbol{\theta} = \bar{\boldsymbol{\Gamma}} \odot \mathbf{z}$.

3.4.2 Gaussian Belief Propagation

GaBP is widely used to solve the linear programming with continuous variables [97, 98]. Based on an undirected graphical model, GaBP consists of variable nodes and edges, where the variable nodes that represent the continuous BAFs are connected by edges. The implementation of GaBP does not involve the complicated direct matrix inversion but allows the parallel messages passing in a system. The linear programming, in a general form $\boldsymbol{\mathcal{A}}\mathbf{y} = \mathbf{b}$, is therefore shifted from an algebraic to a probabilistic domain, where $\boldsymbol{\mathcal{A}}$ is a covariance matrix, $\mathbf{b}$ is a shift vector and $\mathbf{y}$ is the variable vector. Instead of solving a vector-matrix linear problem, with GaBP an inference problem is solved efficiently and distributively by using a graphical model describing a Gaussian distribution function.

The formulation $\boldsymbol{\rho}^T\boldsymbol{\theta} = \bar{\boldsymbol{\Gamma}} \odot \mathbf{z}$ allows us to construct the covariance matrix and the shift vector as in

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \mathbf{I}_{NAK} & \boldsymbol{\rho} \\ \boldsymbol{\rho}^T & \boldsymbol{\Psi}_K \end{bmatrix} \in \mathbb{R}^{(NAK+K)\times(NAK+K)}, \tag{3.31}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{0}_{NAK} \\ \bar{\boldsymbol{\Gamma}} \odot \mathbf{z} \end{bmatrix} \in \mathbb{R}^{(NAK+K)\times 1}, \tag{3.32}$$

where $\mathbf{I}_{NAK}$ is a $NAK \times NAK$ identity matrix and $\boldsymbol{\Psi}_K$ is a $K \times K$ noise diagonal matrix to guarantee the convergence of GaBP. According to [99, Claim 4], if the matrix $\boldsymbol{\mathcal{A}}$ is strictly diagonally dominant, i.e., $|\mathcal{A}_{ii}| > \sum_{j\neq i} |\mathcal{A}_{ii}|, \forall i$, GaBP converges and the marginal means converge to the true means. Based on the observation in [98], the diagonal dominance of the matrix $\boldsymbol{\mathcal{A}}$ can be guaranteed by the diagonal noise matrix $\boldsymbol{\Psi}_K$, with each element satisfying $\Psi_k > NAK/\sqrt{K}$. Therefore, the convergence of proposed GaBP algorithm can be guaranteed.

By defining a new variable as $\mathbf{y} = [\boldsymbol{\theta}^T, \mathbf{n}^T]^T$, we can form the GaBP equation as

$$\boldsymbol{\mathcal{A}}\mathbf{y} = \mathbf{b}, \tag{3.33}$$

where $\mathbf{n} \in \mathbb{R}^{K\times 1}$ is an auxiliary hidden vector and $\boldsymbol{\theta}$ is the solution vector that we seek. As shown in [98], solving the problem $\boldsymbol{\mathcal{A}}\mathbf{y} = \mathbf{b}$ and taking the first NAK entries of the corresponding solution vector $\mathbf{y}$ is equivalent to solving $\boldsymbol{\rho}^T\boldsymbol{\theta} = \bar{\boldsymbol{\Gamma}} \odot \mathbf{z}$.

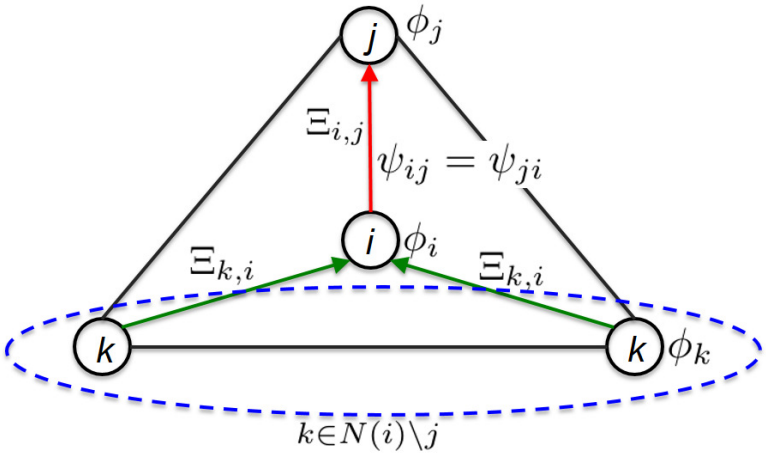

Figure 3.2 – Graphical Model for Gaussian Belief Propagation.

Given the covariance matrix $\boldsymbol{\mathcal{A}}$ and the shift vector $\mathbf{b}$, we can write the Gaussian density function as

$$p(\mathbf{y}) \sim \exp\left(-1/2\mathbf{y}^T\boldsymbol{\mathcal{A}}\mathbf{y} + \mathbf{b}^T\mathbf{y}\right), \tag{3.34}$$

and construct a corresponding undirected graphical model $\mathcal{G}$ in Fig. 3.2. We let $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X}$ is the set of nodes that each of which corresponds to the variable $\mathbf{y}$ and $\mathcal{E}$ is a set of undirected edges connecting between nodes. The graph $\mathcal{G}$ consists of edge potentials ψ_{ij} that represent the posterior probabilities between two nodes, and self potentials ϕ_i that represent the posterior probabilities of the nodes themselves, respectively. According to the following pairwise factorization of the Gaussian density function (3.34),

$$p(\mathbf{y}) \propto \prod_{i=1}^{K} \phi_i(y_i) \prod_{i=1,j=1}^{K} \psi_{ij}(y_i, y_j), \tag{3.35}$$

these graph potentials are determined by $\phi_i(y_i) = \exp(b_i y_i - A_{ii}y_i^2/2)$ and $\psi_{ij}(y_i, y_j) = \exp(-y_i A_{ij} y_j)$. The set of edges $\{i, j\}$ corresponds to the set of non-zero entries in $\boldsymbol{\mathcal{A}}$.

To solve the linear equation (3.33), we can infer the marginal densities, which must also be Gaussian, given by

$$p(y_i) \propto \mathcal{N}\left(\mu_i = \{\boldsymbol{\mathcal{A}}^{-1}\mathbf{b}\}_i, P_i^{-1} = \{\boldsymbol{\mathcal{A}}^{-1}\}_{ii}\right), \tag{3.36}$$

where μ_i and P_i are the marginal mean and inverse variance, respectively.

In Fig. 3.2, each node i is associated with a variable y_i and self potential ϕ_i, which is a function of this variable y_i. The edges between every two nodes are associated with the pairwise (symmetric) potentials $\psi_{i,j}$ between nodes i and j. Messages propagate along the edges on both directions, which are given as

$$\Xi_{i,j}(y_j) \propto \int_{y_i} \psi_{ij}(y_i, y_j)\phi_i(y_i) \prod_{k\in N(i)\backslash j} \Xi_{k,i}(y_i)dy_i, \tag{3.37}$$

where $N(i)$ represents the set of neighbor nodes of node i. According to (3.37), node i needs to first calculate the product of all incoming messages, except for the message coming from node j.

Since $p(\mathbf{y})$ in (3.35) is jointly Gaussian, the factorized self potentials $\phi_i(y_i) \propto \mathcal{N}(\mu_{ii}, P_{ii}^{-1})$ and all messages $\Xi_{ki}(y_i) \propto \mathcal{N}(\mu_{ki}, P_{ki}^{-1})$ are of Gaussian forms. Subsequently, as the product of the incoming messages $\prod_{k\in N(i)\backslash j} \Xi_{ki}(y_i)$ and the self potential $\phi_i(y_i)$ are both a function of the same variable y_i, then we can conclude that

$$\phi_i(y_i) \prod_{k\in N(i)\backslash j} \Xi_{k,i}(y_i) \propto \mathcal{N}(\mu_{i\backslash j}, P_{i\backslash j}^{-1}). \tag{3.38}$$

Within each iteration, the inverse variance $P_{i\backslash j}$ in (3.38) can be updated by

$$P_{i\backslash j} = P_{ii} + \sum_{k\in N(i)\backslash j} P_{ki}, \tag{3.39}$$

where $P_{ii} = \mathcal{A}_{ii}$ is the inverse variance a-priori in node i and P_{ki} is the inverse variance

of the message $\Xi_{ki}(y_i)$. The update rule for the mean $\mu_{i\setminus j}$ is given by

$$\mu_{i\setminus j} = P_{i\setminus j}^{-1}\left(P_{ii}\mu_{ii} + \sum_{k\in N(i)\setminus j} P_{ki}\mu_{ki}\right), \tag{3.40}$$

where $\mu_{ii} = b_i/\mathcal{A}_{ii}$ is the mean of the self potential and μ_{ki} is the mean of the incoming message.

Following [97], we observe the message $\Xi_{ij}(y_i)$ is proportional to normal distribution, i.e., $\Xi_{ij}(y_i) \propto \mathcal{N}\left(\mu_{ij}, P_{ij}^{-1}\right)$, where

$$\mu_{ij} = -P_{ij}^{-1}\mathcal{A}_{ij}\mu_{i\setminus j}, \tag{3.41}$$

$$P_{ij} = -\mathcal{A}_{ij}^2 P_{i\setminus j}. \tag{3.42}$$

These two scalars μ_{ij} and P_{ij} represent the messages propagated in the GaBP algorithm. The propagation of these messages is executed in parallel until the stopping criteria are met, which are defined as

$$\|P_{ij}[t] - P_{ij}[t-1]\| \leq \epsilon, \tag{3.43}$$

$$\|\mu_{ij}[t] - \mu_{ij}[t-1]\| \leq \epsilon. \tag{3.44}$$

In (3.43) and (3.44), t is the index of iteration.

Substituting μ_{ij} and P_{ij} in (3.41) and (3.42) into (3.36) the precision P_i and mean μ_i can be expressed as

$$P_i = \left(P_{ii} + \sum_{k\in N(i)} P_{ki}\right), \tag{3.45}$$

$$\mu_i = P_{i\setminus j}^{-1}\left(P_{ii}\mu_{ii} + \sum_{k\in N(i)} P_{ki}\mu_{ki}\right), \tag{3.46}$$

respectively. The inferred mean $\boldsymbol{\mu} = [\mu_1, ..., \mu_K]$ is identical to the optimal $\mathbf{y}^*$, and the corresponding $\boldsymbol{\theta}^*$ in $\mathbf{y}^*$ is the desired solution of (3.29). A summary of the proposed

process is given in **Algorithm 3.1**.

Algorithm 3.1: Gaussian Belief Propagation

Data: $\boldsymbol{\mathcal{A}}, \mathbf{b}$

Set the neighbor nodes as $N(i)$ to include $\forall k \neq i, k \in K$;

Set the scalar $P_{ii} = \mathcal{A}_{ii}$ and $\mu_{ii} = b_i / \mathcal{A}_{ii}$;

Set the initial messages from node k to node i, where $k \in N(i)$, $P_{ki} = 0$ and $\mu_{ki} = 0$;

Set a convergence threshold ϵ and initialize the iteration $t = 1$;

while $\|P_{ij}[t] - P_{ij}[t-1]\| > \epsilon$ *or* $\|\mu_{ij}[t] - \mu_{ij}[t-1]\| > \epsilon$ **do**

  Broadcast the aggregated sum messages P_i and μ_i in (3.45) and (3.46), $\forall k \in N(i), \forall i$;

  Compute μ_{ij} and P_{ij} by replacing (3.39) and (3.40) into (3.41) and (3.42) ;

  Compute the marginal means $\mu_i = y_i^*$ by (3.45) and (3.46) ;

  $t = t + 1$;

end

Result: Continuous $\mathbf{y}^* \Rightarrow \boldsymbol{\theta}^*$ as in $\mathbf{y} = [\boldsymbol{\theta}^T, \mathbf{n}^T]^T$

3.4.3 Binary Mapping

Once we obtain the continuous $\boldsymbol{\theta}^*$, we need to map them to binary values to satisfy the original user association constraint. In the literature, there are mainly two kinds of binary mapping solvers. The first one is to iteratively map the continuous solutions to discrete forms and feed them back as parts of the input until the solver reaches its convergence criteria, which achieves a near-optimal performance [100, 101]. However, this cannot be applied to our proposed GaBP method because the outputs of GaBP for each iteration cannot be fed back as the inputs at the next iteration.

The second group of mapping algorithms is first to obtain continuous solutions and then map these solutions iteratively via algorithms such as a branch-and-bound procedure until a sub-optimal feasible binary form is found [102, Chap9]. Such method exhibits prohibitive computational complexity during the mapping procedure, since

it partitions variables into feasible subdivisions sequentially. Due to the fact that the dimensions of the variable BAF in GaBP are largely based on the considered network configuration, it is essential to design feasible mapping schemes such that the mapping performance and the complexity are well maintained in an acceptable region.

Inspired by the inflation procedure in [101], we propose two mapping schemes for different performance-complexity trade-offs. In particular, we rely on the solution of **Algorithm 3.1** based on GaBP as an incentive measure to decide on the binary value of $\boldsymbol{\theta}^*$. Intuitively, the a-th beam from BS n is more likely to be assigned to user k if the corresponding channel is in a better condition and the interference to the other users is smaller than the other solutions, which is mathematically equivalent to finding a higher θ^*_{nak}. Based on the above intuitive observations, we propose iterative procedures to determine the set of selected users and their BS-beam-user association based on $\boldsymbol{\theta}^*$.

Binary Mapping Scheme 1

The mapping process starts with an initialization where there is no association between users and the beam of a BS, i.e., the set of connected pairs $\{n, a, k\}$ in $\mathcal{U}^t_{selected} = \emptyset$. Before the iteration starts, (3.29) is solved by GaBP in **Algorithm 3.1** with a continuous output $\boldsymbol{\theta}^*$, and we define a set $\mathcal{U}^t_{next}$ to store the potential solution pairs $\{n, a, k\} \in \boldsymbol{\theta}^*$. In each iteration, the BS-beam-user association with the largest θ^*_{nak} is activated and the set of connected pair $\{n, a, k\}$ is updated in $\mathcal{U}^t_{selected}$, which is given as

$$\mathcal{U}^t_{selected} = \mathcal{U}^{t-1}_{selected} \bigcup \{n^*, a^*, k^* =_{n,a,k} \boldsymbol{\theta}^*\}. \tag{3.47}$$

The feasibility of $\mathcal{U}^t_{selected}$ will be checked following constraints in (3.29). If $\mathcal{U}^t_{selected}$ meets the constraints, then the set $\mathcal{U}^{t+1}_{next}$ is updated by removing the connected user $k \in \mathcal{U}^t_{selected}$ along with the corresponding beam and BS pair $\{n, a\}$ such that the constraint (3.29.2) is met. If not, the iteration ends and $\mathcal{U}^{t-1}_{selected}$ is the optimal solution which meets the constraints and returns the maximum value of the objective

function. This mapping algorithm is presented in **Algorithm 3.2**.

Algorithm 3.2: Binary Mapping 1

Data: $\boldsymbol{\mathcal{A}}, \mathbf{b}$

Solve continuous $\boldsymbol{\theta}^*$ with GaBP in **Algorithm 3.1** ;

Initialize $t = 1$ and the sets $\mathcal{U}^t_{selected} = \emptyset$ and $\mathcal{U}^t_{next} = \{n, a, k \in \boldsymbol{\theta}^*\}$;

for $t = 1$ *to* K **do**

  Update $\mathcal{U}^t_{selected} = \mathcal{U}^{t-1}_{selected} \bigcup \{n^*, a^*, k^* = \arg\max_{n,a,k} \boldsymbol{\theta}^*\}$;

  if $\mathcal{U}^t_{selected}$ *meets the constraints in (3.24)* **then**

    Update $\mathcal{U}^{t+1}_{next} = \mathcal{U}^t_{next} \backslash \mathcal{U}^t_{selected}$;

  else

    $\mathcal{U}^{t-1}_{selected}$ is the optimal solution;

  end

end

Result: Binary output $\boldsymbol{\theta}^*$

Binary Mapping Scheme 2

Different from Mapping Scheme 1, Mapping Scheme 2 takes into account of the objective network capacity in (3.24) by searching for the optimal associating BS for each user that returns the largest network capacity. With the similar initiating steps, the optimal continuous $\boldsymbol{\theta}^*$ is obtained by GaBP, $\mathcal{U}^t_{selected} = \emptyset$ and $\mathcal{U}^t_{next} = \{n, a, k\} \in \boldsymbol{\theta}^*$. In each iteration, the BS-beam-user association with the largest $\{n, a^*, k^*\} = \arg\max_{a,k} \theta^*_{nak}, \forall n \in N$ is activated, and the set of connected pair $\{n, a^*, k^*\}$ is updated in $\mathcal{U}^t_{selected}(n)$, which is given as

$$\mathcal{U}^t_{selected}(n) = \mathcal{U}^{t-1}_{selected} \bigcup \{n, a^*, k^* =_{a,k} \boldsymbol{\theta}^*\}. \tag{3.48}$$

The index of the maximal $R^t(n)$ between BSs $n \in N$ based on (3.24) is computed as

$$n^* = \arg\max_n R^t(n), \tag{3.49}$$

and then compared within the iteration t, following the feasibility check for constraints in (3.29). If the set $\mathcal{U}^t_{selected}(n^*)$ meets the constraints, $\mathcal{U}^t_{selected}$ is updated to be

$$\mathcal{U}^t_{selected} = \mathcal{U}^{t-1}_{selected} \bigcup \mathcal{U}^t_{selected}(n^*). \tag{3.50}$$

We then compare $R^t(n^*)$ and $R^{t-1}(n^*)$ corresponding to the set $\mathcal{U}^t_{selected}$ and $\mathcal{U}^{t-1}_{selected}$, respectively. If $R^t(n^*) \geq R^{t-1}(n^*)$, we remove the connected user $k \in \mathcal{U}^t_{selected}$ along with the corresponding beam and BS pair $\{n, a\}$ from the set $\mathcal{U}^{t+1}_{next}$ and proceed to the next iteration. For the above cases where the set $\mathcal{U}^t_{selected}(n^*)$ does not meet the constraints or where $R^t(n^*) < R^{t-1}(n^*)$, the iteration ends and the $\mathcal{U}^{t-1}_{selected}$ is the

optimal solution. The overall algorithm is presented in **Algorithm 3.3**.

Algorithm 3.3: Binary Mapping 2

Data: $\boldsymbol{\mathcal{A}}, \mathbf{b}$

Solve continuous $\boldsymbol{\theta}^*$ with GaBP in **Algorithm 3.1** ;

Initialize $t = 1$, the sets $\mathcal{U}^t_{selected} = \emptyset$, $\mathcal{U}^t_{next} = \{n, a, k \in \boldsymbol{\theta}^*\}$ and $R^t = 0$;

for $t = 1$ *to* K **do**

  for $n = 1$ *to* N **do**

    Update $\mathcal{U}^t_{selected}(n) = \mathcal{U}^{t-1}_{selected} \bigcup \{n, a^*, k^* = \arg\max_{a,k} \boldsymbol{\theta}^*\}$ and compute the corresponding network capacity $R^t(n)$ based on (3.24)

  end

  Choose BS $n^* = \text{argmax}_n R^t(n)$;

  if *the set* $\mathcal{U}^t_{selected}(n^*)$ *meets the constraints in (3.24)* **then**

    Update $\mathcal{U}^t_{selected} = \mathcal{U}^{t-1}_{selected} \bigcup \mathcal{U}^t_{selected}(n^*)$;

    if $R^t(n^*) \geq R^{t-1}(n^*)$, *corresponding to the set* $\mathcal{U}^t_{selected}$ *and* $\mathcal{U}^{t-1}_{selected}$ *respectively* **then**

      Update $\mathcal{U}^{t+1}_{next} = \mathcal{U}^t_{next} \backslash \mathcal{U}^t_{selected}$;

    else

      $\mathcal{U}^{t-1}_{selected}$ is the optimal solution ;

    end

  else

    $\mathcal{U}^{t-1}_{selected}$ is the optimal solution;

  end

end

Result: Binary output $\boldsymbol{\theta}^*$

3.5 Computational Complexity Analysis

The computational complexity of our proposed algorithms is evaluated and compared with other algorithms in terms of numerical operations (including multiplications and additions) [103], which is summarized in Table 3.2 as follows:

Table 3.2 – Complexity Comparison

	Complexity	To N and K
GaBP+M1	$\tau_G[NAK + K] + K$	Linear
GaBP+M2	$\tau_G[NAK + K] + NK^2(NK + 3)$	Polynomial
BP	$\tau_B[(2^N - 1)^{K-1}(NK + 3) + 2^N + 2K]$	Exponential
ES	$(2^N - 1)^K K(NK + 3)$	Exponential

The exhaustive search (ES) method

The optimal result of ES is exhaustively searched over all possible states of $\boldsymbol{\beta}$ in the optimization function (3.24) and computed for each user at the BSs. The number of possible states of $\boldsymbol{\beta}$ for one user is $(2^N - 1)$ and subsequently the number of all possible states for all K users is $(2^N - 1)^K$. For each of these states, ES computes (3.24) which consists of $K(NK + 3)$ operations. Accordingly, the overall complexity of ES is $\mathcal{O}_{ES} = (2^N - 1)^K K(NK + 3)$.

The BP method with discrete variables in [50]

Within the BP method, the optimization problem of (3.24) is decomposed into sub-functions that represent the capacity of each user k. A local indicator function is defined to represent each sub-function such that the capacity of each user is estimated in parallel at user level only if the constraints in (3.24) hold. A factor graph is then developed to represent the sub-functions as factor nodes and their corresponding variables as variable nodes. The estimate of user capacity is the message iteratively exchanged between the variable node and its neighboring factor nodes, with each message corresponding to the used states of variable $\boldsymbol{\beta}_k$. This process is repeated until the message values at each factor node converge to a fixed value. The complexity of the BP algorithm mainly depends on the factor node computation. The complexity consists of the product of $(2^N - 1)^{K-1}$ and $(NK + 3)$ and $[(K - 2) + 1]$ operations. The outer normalization constant needs $[(K + 1) + (2^N - 1)]$ operations. Denoting τ_B as the number of iterations, the overall complexity of BP method is $\mathcal{O}_{BP} = \tau_B[(2^N - 1)^{K-1}(NK + 3) + 2^N + 2K]$.

The GaBP method in Algorithm 3.1

As the propagating messages of GaBP are the means and variances after continuous relaxation, we evaluate the complexity from the number of operations required to generate a propagating message P_{ij} and μ_{ij} at each iteration, which is $(NAK + K)$. There are in total $(NAK + K)$ messages propagating in the graph, broadcasted by $(NAK + K)$ nodes in the graph. Since each node only broadcasts its messages in parallel, the total complexity of the GaBP method is significantly reduced to $\mathcal{O}_{GaBP} = \tau_G[NAK + K]$, where τ_G is denoted as the number of iterations needed for convergence.

In addition, we need to take account of the complexity of the binary mapping schemes within the GaBP method. In **Algorithm 3.2**, only one user is selected to connect to a BS in Step 3-10 of each iteration. Accordingly, only K iterations are required for all K users. The complexity of Binary Mapping Scheme 1 is $\mathcal{O}_{M1} = K$. In **Algorithm 3.3**, one user is selected to connect to a BS in Step 3-18 of each iteration. Similar to Mapping Scheme 1, there are K iterations needed for the mapping scheme. Since only one BS is selected from N ones and the corresponding sum rate $R^t(n)$ is computed in each iteration, the complexity of Binary Mapping Scheme 2 is $\mathcal{O}_{M2} = NK(K(NK + 3)) = NK^2(NK + 3)$.

Table 3.2 shows the detailed comparisons on the computational complexities within the proposed algorithms and exhaustive search. It is worth mentioning that the value of A, corresponding to the number of antennas at massive MIMO BSs, is large but finite in practice. Compared to the exponential increase in N and K as in BP and exhaustive search, the complexity of GaBP increase linearly or polynomially in N and K. The numerical complexity with different system configurations will be shown in Section 3.6.

3.6 Simulation Results

We consider a network with $K = 6$ users and $N = 2$ BSs, unless mentioned otherwise. Each BS is equipped with $A = 64$ antennas, where $N_v = 8$ and $N_h = 8$. The position of 2 BSs is set on a 1×2 rectangular grid, as in [81], operating at frequency 2GHz and the system bandwidth is 10MHz. A BS is placed in the center of each grid, with a distance of 500 m between BSs. Users are uniformly distributed within the same area. The channel model is set as in (3.1) and the channel correlation matrix between BSs and users is set as $\mathbf{\Omega}_{nk} = \mathbf{\Omega}_{nk}^h \otimes \mathbf{\Omega}_{nk}^v$, where $\mathbf{\Omega}_{nk}^h$ and $\mathbf{\Omega}_{nk}^v$ are the horizontal and vertical correlation matrices that are generated according to [70, eq.(3)]. The large-scale fading coefficient is generated based on a distance-dependent path-loss model, i.e., $\alpha_{nk} = d_{nk}^{-\gamma}$, where d_{nk} is the distance in meters between user k and BS n and γ is set as 3.76. The noise level of K users is set the same, i.e., $\sigma_k^2 = \sigma^2 = -174$dBm. All results are tested over 1000 independent trials.

For simplicity, the following abbreviations are employed throughout this section:

- 'BP': The traditional BP method in [50] to directly solve (3.24).
- 'GaBP-w-Mapping1' ('GaBP-w-Mapping2'): The GaBP method in **Algorithm 3.1** with mapping scheme 1 in **Algorithm 3.2** (with mapping scheme 2 in **Algorithm 3.3**).
- 'FD-w/t-Coop': The scheme in [69] where the FD beamforming design for the massive MIMO network is designed for single-cell scenarios.
- '1D-Coop-w-UA': The scheme in [44] where 1-D cooperative beamforming vectors and user association factors are designed sequentially for the massive MIMO network.
- 'ES': The exhaustive search method that compares all possible solutions in (3.8).

3.6.1 Convergence and Computational Complexity

The required number of iterations and the network capacity with respect to the values of the stopping criteria are shown in Fig. 3.3, depicted in blue and red respectively. For the 'BP' algorithm, we set $\mu = 5$ to guarantee its performance following [81]. By increasing the accuracy of stopping criteria ϵ, the network capacities with both 'BP' and 'GaBP' methods increase while cost an increasing average number of iterations. Even though the average number of iterations grows almost linearly with respect to the logarithmic accuracy of ϵ, the network capacity improved slightly. Accordingly, we set $\epsilon = 0.01$ and $\epsilon = 0.001$ for 'BP' and 'GaBP' respectively in the subsequent simulations.

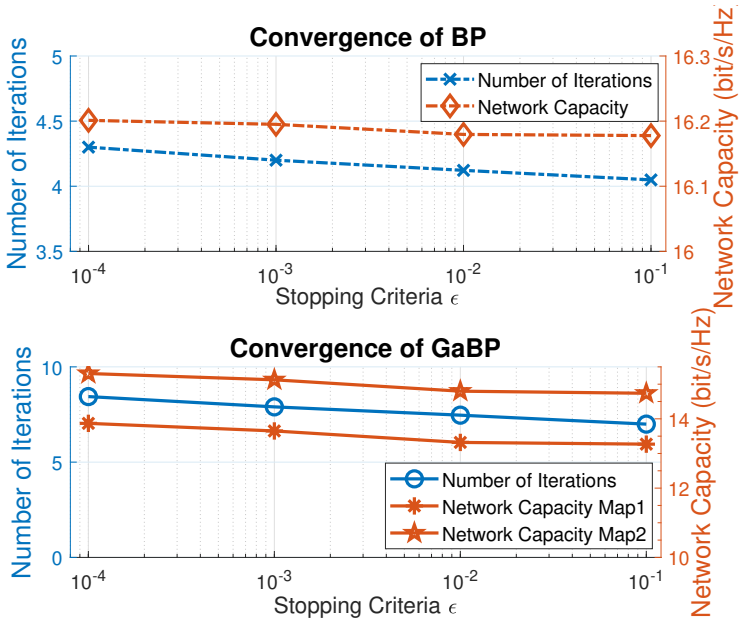

Figure 3.3 – Number of Iterations and Sum Capacity of BP and GaBP respect to ϵ with $N = 2$, $K = 6$ and SINR threshold of 5dB.

Table 3.3 reveals the computational costs for our proposed schemes for various system configurations, where we observe that the complexity of both 'ES' and 'BP' schemes present exponential increases in N and K. Based on the simulation results as in Fig. 3.3, it is observed that τ_B is around 4.2 for the considered system configuration and 'BP' costs 23.4% of the computational complexity of 'ES'. The complexity gain be-

Table 3.3 – The Complexity Analysis with Various System Configurations

No. of BSs N	2		3		4	
No. of users K	6	10	6	10	6	10
'GaBP-w-Mapping1'	$\tau_G \cdot 7.8 \times 10^2 + 6$	$\tau_G \cdot 1.3 \times 10^3 + 10$	$\tau_G \cdot 1.2 \times 10^3 + 6$	$\tau_G \cdot 1.9 \times 10^3 + 10$	$\tau_G \cdot 1.5 \times 10^3 + 6$	$\tau_G \cdot 2.6 \times 10^3 + 10$
'GaBP-w-Mapping2'	$\tau_G \cdot 7.8 \times 10^2$ + 1.1×10^3	$\tau_G \cdot 1.3 \times 10^3$ + 4.6×10^3	$\tau_G \cdot 1.2 \times 10^3$ + 2.3×10^3	$\tau_G \cdot 1.9 \times 10^3$ + 9.9×10^3	$\tau_G \cdot 1.5 \times 10^3$ + 3.9×10^3	$\tau_G \cdot 2.6 \times 10^3$ + 1.7×10^4
'BP'	$\tau_B \cdot 3.7 \times 10^3$	$\tau_B \cdot 4.5 \times 10^5$	$\tau_B \cdot 3.5 \times 10^5$	$\tau_B \cdot 1.3 \times 10^9$	$\tau_B \cdot 2.1 \times 10^7$	$\tau_B \cdot 1.7 \times 10^{12}$
'ES'	6.6×10^4	1.4×10^7	1.5×10^7	9.3×10^{10}	1.9×10^9	2.5×10^{14}

comes more significant with an increase in the number of BSs and users. In contrast, 'GaBP' methods illustrate a linear increase with respect to N and K, and accordingly exhibit significant complexity reduction compared to 'ES' and 'BP'. Taking into account $\tau_G = 8$ as in Fig. 3.3, 'GaBP-w-Mapping1' and 'GaBP-w-Mapping2' cost only $8.3 \times 10^{-9}\%$ and $1.5 \times 10^{-8}\%$ of the computational complexity of 'ES' when $N = 4$ and $K = 10$. The above observations validate the superiority of the proposed distributed 'GaBP' algorithm in terms of computation costs.

3.6.2 Capacity Performance Comparisons

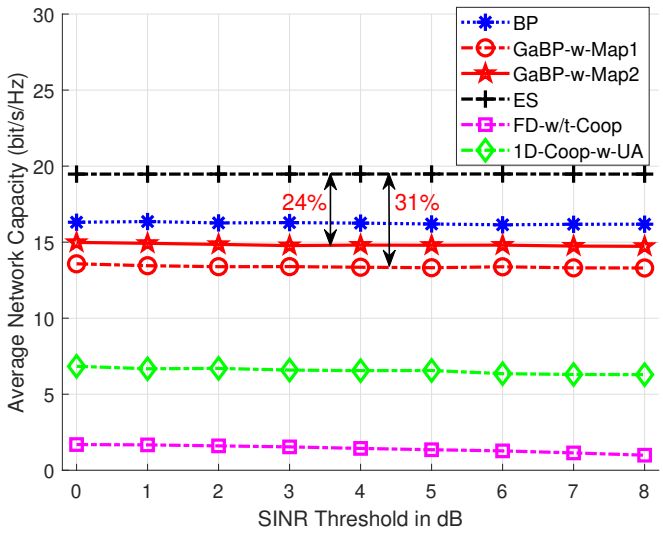

Figure 3.4 – The Average Network Capacity for $N = 2$ BSs and $K = 6$ users.

Fig. 3.4 shows the network capacity with different levels of SINR thresholds (varying from 0dB to 8dB). It is evident that 'ES' achieves the optimal network capacity with

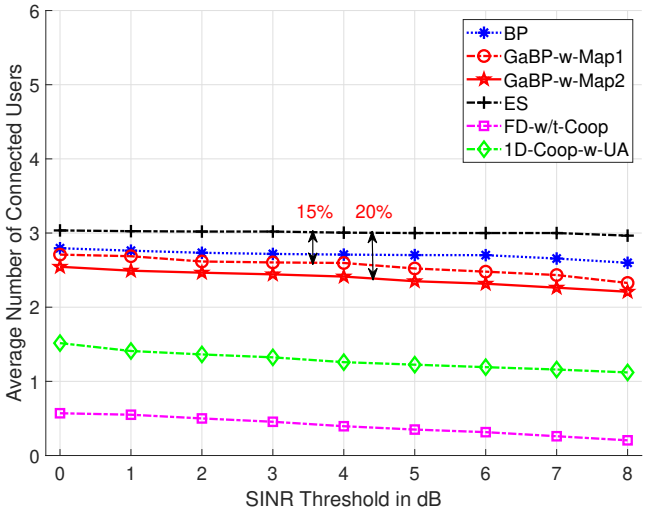

Figure 3.5 – The Average Number of Connected users for $N = 2$ BSs and $K = 6$ users.

around 19.5 bit/s/Hz, while the BP algorithm achieves around 83% of the capacity of 'ES'. Performance losses are observed for the proposed GaBP methods compared to the optimal 'ES' result, where 'GaBP-w-Mapping 1' and 'GaBP-w-Mapping 2' present 31% and 24% network capacity losses compared to the 'ES' method, respectively. The superiority of 'GaBP-w-Mapping 2' over 'GaBP-w-Mapping 1' results from the optimal capacity search between BSs in each iteration in **Algorithm 3.1**. The performance loss of GaBP-based methods compared to the BP-based scheme is caused by the binary relaxation in 'GaBP' and binary mapping errors. Overall, all proposed algorithms outperform 'FD-w/t-Coop' by 9.8 times and '1D-Coop-w-UA' by 2 times. The outperformance is the result of using joint cooperative FD beamforming and UAF to mitigate the inter-cell interference in both horizontal and vertical directions for a multi-cell network. This validates the importance of cooperation between BSs. We can also observe that with the increasing SINR threshold, the average network capacity of all schemes varies only slightly, which is due to the joint effect of a decrease in the average number of connected users and an increase in the received SINR for each user. As observed from (7), the network capacity depends on both the number of connected users and the received SINR of each connected user. For a given total transmit power available at the BS, when the SINR target increases, the total

number of connected users has to be decreased to reduce multi-user interference, so that the increasing SINR threshold can be met for each connected user. Meanwhile, the spectral efficiency for each connected user increases due to the increase in the received SINR. These two effects jointly lead to a relatively stable network capacity performance.

Fig. 3.5 shows the average number of connected users with the SINR threshold varying between $0 - 8$dB. When the SINR threshold increases, the number of connected users decreases with a different level of the gradient for all schemes. While the 'ES' method delivers the optimal performance, 'BP' presents a slightly outperformance over 'GaBP-w-Mapping 1' and 'GaBP-w-Mapping 2' by 6.2% in terms of the number of connected users. Meanwhile, 'GaBP-w-Mapping 1' and 'GaBP-w-Mapping 2' result in 15% and 20% performance loss compared to the 'ES' method in terms of the average number of connected users. Without surprises, the network can accommodate at least 1.9 / 6.5 times more users by our proposed algorithms over '1D-Coop-w-UA' and 'FD-w/t-Coop'.

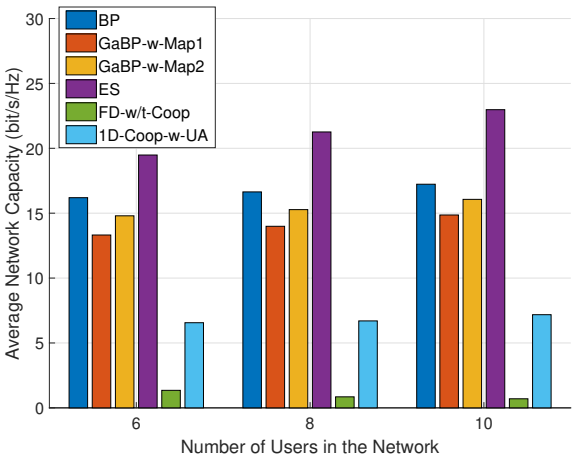

Figure 3.6 – The Average Network Capacity for $N = 2$ BSs and $K = 6/8/10$ users.

The effect of the number of users on the network capacity and the average connected users is shown in Fig. 3.6 and 3.7 when the SINR threshold is set to be 5dB, respectively. 'BP' has the closest performance to the optimal 'ES' in terms of the average network capacity, followed by 'GaBP-w-Mapping 1' and 'GaBP-w-Mapping

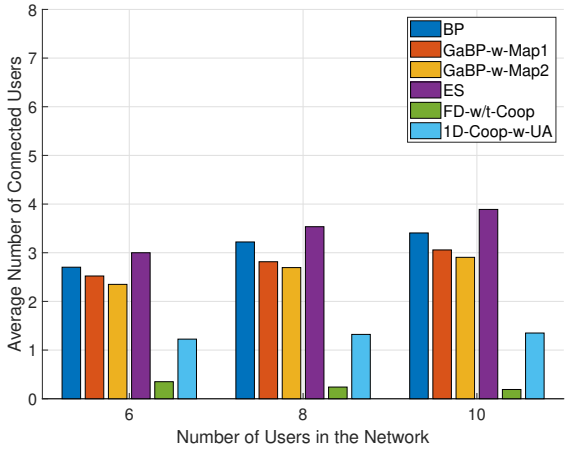

Figure 3.7 – The Average Number of Connected users for $N = 2$ BSs and $K = 6/8/10$ users.

2'. With the increase in the total number of users, the inter-cell interference in the network becomes more severe. Hence, without proper cooperative beamforming and user association scheme, the network capacity of 'FD-w/t-Coop' decreases due to the impact of inter-cell interference. For '1D-Coop-w-UA', even though the cooperation is performed between BSs, its network capacity is still worse than that with FD beamforming. It is because the beams on horizontal directions cannot differentiate users with the same location with different altitudes. We can also observe that the network capacity (in Fig. 3.6) and the numbers of connected users (in Fig. 3.7) of the proposed 'BP' and GaBP algorithms increase with the total number of users in the network. When the total number of users in the network increases, there is a higher probability that more users will experience better channel conditions. Accordingly, under the same total transmit power constraint at the BS as well as the same SINR threshold, this means that the probability that more users will satisfy the required SINR threshold increases, which then leads to an improved system performance. From Fig. 3.7, it is seen that both 'BP' and 'GaBP' with different mapping schemes can achieve more than 76% of the number of connected users of 'ES'. However, the computational complexity of 'ES' is much higher especially for a dense network. Thus, compared to 'ES', the proposed distributed 'GaBP' with mapping schemes algorithms are more

practical.

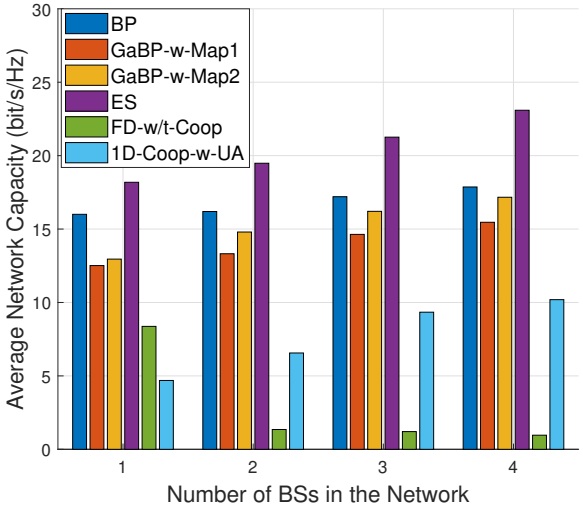

Figure 3.8 – The Average Sum Rate for $N = 1/2/3/4$ BSs and $K = 6$ users.

The effect of the number of BSs on the network capacity and the average connected users are plotted in Fig. 3.8 and Fig. 3.9, where the SINR threshold is set to be 5dB. With 'ES' achieving the optimal performance, there are growing gaps between 'BP' / 'GaBP-w-Mapping1' / 'GaBP-w-Mapping2' and 'ES' when the number of BSs increases for both performance metrics of average network capacity and the number of connected users. This implies a performance trade-off between the computation complexity and the network capacity. 'BP' costs 2.8% computation while achieving above 77% performance of 'ES' when $N = 4$ and $K = 6$. For GaBP algorithms, 'GaBP-w-Mapping1' / 'GaBP-w-Mapping2' can achieve 67% / 74% network capacity with only around 8.3×10^{-11} / 1.5×10^{-10} computation of 'ES' with respect to $K = 6$ and $N = 4$, respectively. With the increasing number of BSs, the performance of both network capacity and the number of connected users for all schemes but 'FD-w/t-Coop' have significant improvements. It again validates the importance of cooperation between BSs. With no doubt, '1D-Coop-w-UA' has inferior performance to the proposed schemes due to the lack of vertical interference cancellation.

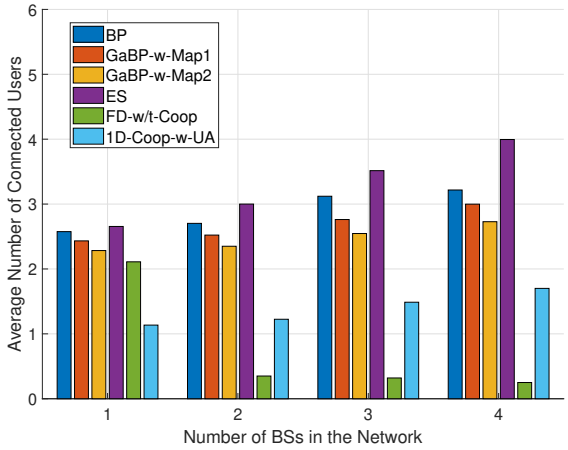

Figure 3.9 – The Average Number of Connected users for $N = 1/2/3/4$ BSs and $K = 6$ users.

3.7 Chapter Summary

In this chapter, we studied the cooperative full-dimension beamforming in massive MIMO network, and jointly designed the beamforming vectors and user association decisions to maximize the network capacity. By using the SLNR metric, the constructed problem was transformed into an optimization on UAFs only, followed by the introduction of beam association factor. A three-step GaBP-based algorithm was employed to obtain a feasible solution on BAF, which was shown to be convergent within only a few iterations and exhibit linear complexity in the size of network configuration. Simulation results showed that the proposed algorithms significantly outperformed the existing literature, and the distributed GaBP-based method achieved a similar performance to the traditional BP-based scheme with a much reduced computational cost.

Chapter 4

Design of Deep Learning in Mobile Edge Computing Systems

4.1 Introduction

In 5G communication systems, there are diverse applications ranging from high data rate delay-tolerant services to URLLC [4]. By achieving ultra-low E2E delay and ultra-high reliability, URLLC lies the foundation for emerging latency-critical applications, such as factory automation, autonomous vehicles, and virtual/augmented reality [3]. Devices in these applications will generate some tasks that require processing within a short time. To reduce processing time at the local server of each device and to avoid delays in backhauls and core networks, MEC is one promising solution [72]. However, when a task is packetized in a short packet and offloaded to a MEC server via a wireless link, the packet may be lost when the channel is in deep fading [104]. Besides, short block-length channel codes will cause the none-zero decoding error rate, even for an arbitrarily high SNR [105, 106]. Thus, achieving ultra-high reliability and ultra-low latency is very challenging in MEC systems.

On the other hand, mobile devices have only limited battery capacities. Improving the battery lifetime or EE of users is an urgent task [72, 107, 59]. By offloading tasks

to MEC servers, we can save energy consumption at the local servers (equipped at mobile devices), but extra energy is consumed for data transmissions. To minimize the total energy consumption of each user, we need to optimize the offloading probability. In a MEC network with multiple MEC servers and multiple users, the problem that optimizes user association [108], resource allocation, and offloading probabilities is non-convex and complicated. How to improve EE by solving a non-convex problem in the scenario with both URLLC services and delay-tolerant services remains an open problem.

To find the optimal solution to the problem, there are two kinds of approaches: optimization algorithms and machine learning algorithms. Since optimization algorithms need to search the optimal solution when channels change, they are suitable for small-scale problems, such as resource allocation in a single access point (AP) scenario [59, 60]. When the scale of the problem grows, DL algorithms have the potential to find a near-optimal solution in real time [61]. Based on the universal approximation theorem of DNN [62], a DNN can be used as an approximator of an optimal policy. The state-action pairs obtained from the optimal policy can be used as labeled samples to train the DNN. Once the training of the DNN is finished, we can compute the resource allocation from it with different channel realizations.

To train a DL algorithm, we first need to obtain optimal policies from simplified system models. However, optimal policies may not be available in practical systems. Thus, some other techniques are needed to enable DL algorithms. One approach that does not require labeled training samples is deep reinforcement learning [109]. By learning from the feedback of real-environment, deep reinforcement learning is widely used to maximize the long-term rewards of Markov decision processes. This approach is not suitable for URLLC services due to the following two reasons. First, maximizing the long-term reward cannot guarantee the delay and reliability requirements in each time slot. Second, to check whether the packet loss probability satisfies the reliability requirement from the feedback of real network environment, a user needs to transmit a large number of packets. If the required packet loss probability is 10^{-7}, a user needs to transmit more than 10^7 packets, which may be larger than the total number of

packets that will be generated within the service time of the user. To handle this issue, we need to compute the packet loss probability with the help of theoretical results that are obtained with model-based methods.

To merge the model-free deep learning algorithms with model-based theoretical results, we establish a digital twin of the real network environment. As shown in [110], a digital twin is a virtual digital model of the real network that consists of data from the real network (e.g., network topologies, schedulers, and channels) and fundamental rules from theoretical studies (e.g., trade-offs in information and queueing theories). With the help of a digital twin, we can compute the energy consumption, delay, and packet loss probability of a certain decision on user association and resource allocation. In addition, by monitoring the variations of the real network environment, the system can update the digital twin for training the DNN. As such, it is possible to implement deep learning algorithms in non-stationary environment. Nevertheless, how to apply a digital twin in the DL architecture for hybrid 5G services in MEC systems remains unclear.

Motivated by the above issues, we will answer the following questions in this chapter: 1) How to improve EE for URLLC and delay-tolerant services in MEC systems; 2) How to establish the digital twin that mirrors the real network environment; 3) How to train a DNN based on the digital twin when the optimal policy is not available.

4.2 Related Work

How to improve EE of mobile devices in MEC systems subject to the delay constraint has been widely studied in existing literature [60, 59, 107, 111, 112]. To study the trade-off between EE and latency, a weighted sum of energy consumption and latency was minimized in a single-AP scenario [59]. EE was maximized subject to the delay constraint in single-AP scenarios and multi-AP scenarios in [60] and [107], respectively. The authors of [111] analyzed the EE and latency with stochastic geometry and provided some useful guidelines to network provision and planning. The above studies mainly focused on one kind of services, and neglected the heterogeneities of

services. To address this issue, a game theory approach was proposed in [112], where resource management and user association were optimized in multi-access MEC systems.

How to apply machine learning algorithms for user association or task offloading in MEC systems was also studied in some recent works [113, 114, 115, 83]. Deep Q-learning was used to minimize the task execution cost by optimizing offloading decisions according to channel state information, queue state information, and energy queue state of the energy harvesting system [113]. A similar method was also applied for energy harvesting of IoT devices in [114]. The authors of [115] proposed an efficient reinforcement learning-based resource management algorithm to incorporate renewable energy into MEC systems. More recently, a deep reinforcement learning framework for task offloading was studied in a single-AP scenario [83].

The above studies provided useful insights and promising machine learning algorithms in MEC systems, but they did not consider 5G services. Supporting URLLC in MEC systems was studied in [116] and [117]. In [116], the long-term average power consumption of mobile devices is minimized subject to the latency and reliability constraints. The weighted sum of delay and reliability is minimized in [117] for a single-user. Nevertheless, how to serve hybrid 5G services in MEC systems remains unclear, and deserves further study.

4.3 System Model

4.3.1 MEC System

We consider a MEC system shown in Fig. 4.1, where M APs serve K^{u} URLLC services and K^{b} delay-tolerant services, which are indexed by $\mathcal{K}^{\mathrm{u}} = \{1, ..., K^{\mathrm{u}}\}$ and $\mathcal{K}^{\mathrm{b}} = \{K^{\mathrm{u}} + 1, ..., K^{\mathrm{u}} + K^{\mathrm{b}}\}$, respectively. For notational simplicity, we use a superscript $\xi = \{\mathrm{u}, \mathrm{b}\}$ to represent the types of services in this chapter. If $\xi = \mathrm{u}$, a parameter is used in URLLC services. Otherwise, it is used in delay-tolerant services. All the notations used in this chapter are listed in Table 4.1.

Table 4.1 – Notations

Notation	Definition	Notation	Definition
x	scalar	$\mathbf{x}$	vector
$\mathbb{E}$	expectation	$(\cdot)^{\mathrm{T}}$	transpose operator
M	number of APs	$\xi = \{\mathrm{u}, \mathrm{b}\}$	superscript representing URLLC and delay-tolerant services
K^{ξ}, $\mathcal{K}^{\xi}$	number of users and set of users	T_s	duration of each slot
S_m	service rate of the mth MEC server	C_k^{ξ}	service rate of the kth user
$C_k^{\max,\xi}$	maximum computing capacity of the kth user	λ_k^{ξ}	average task arrival rate generated by the kth user
b_k^{ξ}	number of bits of each task	c_k^{ξ}	number of CPU cycles required to process each task
$\beta_{m,k}^{\xi} = \{0, 1\}$	user association indicator	$N_{m,k}^{\xi}$	number of allocated subcarriers
W	bandwidth of each subcarrier	$\alpha_{m,k}^{\xi}$	large-scale channel gain
$g_{m,k}^{\xi}$	small-scale channel gain	$P_k^{\mathrm{t},\xi}$	transmit power
Φ	SNR loss coefficient	N_0	single-side noise spectral density
f_Q^{-1}	inverse of Q-function	$\varepsilon_k^{\mathrm{d,u}}$	decoding error probability
x_k^{ξ}	offloading probability	$e_k^{\mathrm{loc},\xi}$	energy consumption per CPU cycle
$E_k^{\mathrm{loc},\xi}$	energy consumption per packet at the local server	$D_k^{\mathrm{lc,u}}$	processing delay on local server
$D_k^{\mathrm{lq,u}}$	queueing delay on local server	$D^{\max,\mathrm{u}}$	maximum delay
$\epsilon_k^{\mathrm{lq,u}}$	queueing delay violation probability in local server	$\epsilon^{\max,\mathrm{u}}$	maximum queueing delay violation probability
$\epsilon_k^{\mathrm{mc,u}}$	processing delay violation probability in MEC server	ρ_m^{mc}	workload of the MEC server
$\bar{c}_k^{\mathrm{b}}$	average number of required CPU cycles	$\bar{b}_k^{\mathrm{b}}$	average number of bits in a packet
$D_k^{\mathrm{mc,u}}$	processing delay on MEC server	η_k^{ξ}	normalized energy consumption
$N^{\max}$	total number of subcarriers	$P_k^{\max,\xi}$	maximal transmit power
$\hat{\boldsymbol{\beta}}$	direct output of the DNN	$\tilde{\boldsymbol{\beta}}$	the best user association scheme obtained from the digital twin

The APs are connected to the MME that is in charge of user association. To establish the digital twin, the MME sends some parameters and models of the network to a central server, where the user association scheme is explored in the digital twin. We use a DNN to approximate the best user association scheme, where the DNN is trained in the digital twin off-line. After the training phase, a DNN for user association is sent to the MME. With a given user association scheme, the network can be decomposed into single-AP problems.[1] For each single-AP problem, the AP optimizes resource allocation and task offloading for users that are associated with it.

[1]Frequency reuse factor is less than one that different bandwidth is allocated to adjacent APs. As such, there is no strong interference, and weak interferences are considered as noise.

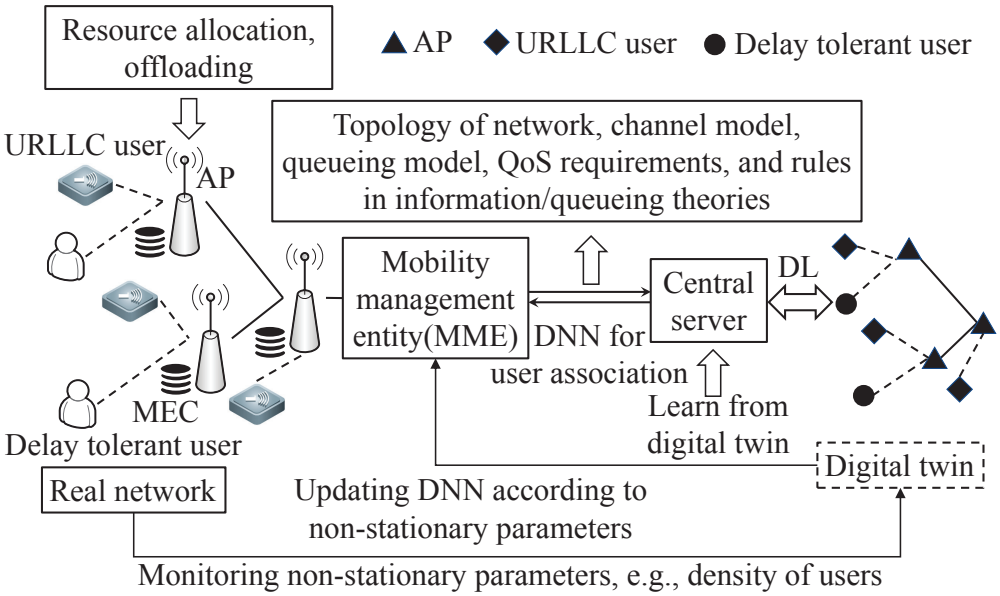

Figure 4.1 – System model

Each AP is equipped with a MEC server and each user has a local server. Time is discretized into slots. The duration of each slot is T_{s}. The service rates of the mth MEC server and the kth user are denoted as S_m (CPU cycles/slot) and C_k^{ξ} (CPU cycles/slot), respectively. The kth user can adjust C_k^{ξ} within the regime $[0, C_k^{\max,\xi}]$, where $C_k^{\max,\xi}$ is the maximum computing capacity of the user.

Without loss of the generality, non-stationary parameters in a system can be classified into two categories. The first category of parameters is highly dynamic, such as the large-scale channel gains and the average task arrival rates. The other category of parameters varies slowly, such as the density of users in a certain area. For the first category of parameters, we include them in the input of the DNN. For the second category of parameters, the system monitors their values and updates them in the digital twin. Then, the DNN learns from the updated digital twin. Rather than training a new DNN, the previous well-trained DNN will be used to initialize the new one. In this way, the output of the DNN changes with non-stationary parameters.

4.3.2 Computation Tasks and Communication Packets

The computation tasks of the kth user are characterized by $(\lambda_k^\xi, b_k^\xi, c_k^\xi)$, where λ_k^ξ (packets/slot) is the average task arrival rate generated by the kth user, b_k^ξ (bits/-packet) is the number of bits of each task (i.e., the size of a packet), and c_k^ξ (cycles/-packet) is the number of CPU cycles required to process each task. We assume that each task is conveyed in one packet, and the relation between b_k^ξ and c_k^ξ is given by $c_k^\xi = k_1 b_k^\xi$, where $k_1 > 0$ (cycles/bit) depends on the computational complexity of the task [76, 118, 119].

For URLLC services, we assume that the packet size and the number of CPU cycles required to process each packet are constant (e.g., 32 bytes [4]), and the packet arrival process follows a Bernoulli process. In each slot, a user either has a packet to transmit or stays silent. For delay-tolerant services, both the inter-arrival time between packets and the packet size may follow any general distributions. The only assumption is that the packet size of delay-tolerant services is much longer than that of URLLC services. In the rest of the chapter, the tasks of URLLC services and delay-tolerant services are referred to as short and long packets, respectively.

4.3.3 Achievable Data Rate over Wireless Links

The users can offload tasks to one of the MEC servers via wireless links. Let $\boldsymbol{\beta}$ be the user association vector with entry $\beta_{m,k}^\xi$ denoting whether the kth user is associated with the mth AP. If the kth user is associated with the mth AP, then $\beta_{m,k}^\xi = 1$. Otherwise, $\beta_{m,k}^\xi = 0$. We assume that each user can only offload packets to one of the APs, i.e., $\sum_{m\in\mathcal{M}} \beta_{m,k}^\xi = 1$, where $\mathcal{M} = 1, ..., M$ is the set of indices of APs.

Achievable Rate for URLLC

We consider orthogonal frequency division multiple access (OFDMA) systems. The number of subcarriers allocated to the kth user is denoted as $N_{m,k}^\xi$. Since the packet size of URLLC services is small, it is reasonable to assume that the bandwidth of

$N^{\mathrm{u}}_{m,k}$ subcarriers is smaller than the coherence bandwidth and the transmission time is smaller than channel coherence time as well. Thus, each packet is transmitted over a flat fading quasi-static channel. If the kth user is accessed to the mth AP, the achievable rate of the kth URLLC user, $k \in \mathcal{K}^{\mathrm{u}}$, can be approximated by [120]:

$$R^{\mathrm{u}}_k \approx \frac{N^{\mathrm{u}}_{m,k} W}{\ln 2} \left[\ln\left(1 + \frac{\alpha^{\mathrm{u}}_{m,k} g^{\mathrm{u}}_{m,k} P^{\mathrm{t,u}}_k}{\Phi N^{\mathrm{u}}_{m,k} W N_0}\right) - \sqrt{\frac{V^{\mathrm{u}}_k}{T_{\mathrm{s}} N^{\mathrm{u}}_{m,k} W}} f_Q^{-1}(\varepsilon^{\mathrm{d,u}}_k) \right] \text{(bits/s)}, \tag{4.1}$$

where W is the bandwidth of each subcarrier, $\alpha^{\mathrm{u}}_{m,k}$ is the large-scale channel gain, $g^{\mathrm{u}}_{m,k}$ is the small-scale channel gain, $P^{\mathrm{t,u}}_k$ is the transmit power, Φ is a SNR loss coefficient, which reflects the gap between the achievable rate of practical channel codes and the approximation, N_0 is the single-side noise spectral density, f_Q^{-1} is the inverse of Q-function, $\varepsilon^{\mathrm{d,u}}_k$ is the decoding error probability, and $V^{\mathrm{u}}_k = 1 - 1 \Big/ \left(1 + \frac{\alpha^{\mathrm{u}}_{m,k} g^{\mathrm{u}}_{m,k} P^{\mathrm{t,u}}_k}{\Phi N^{\mathrm{u}}_{m,k} W N_0}\right)^2$.

Data Rate for Delay-tolerant Services

For delay-tolerant services, the packet size is long, and Shannon's capacity is a good approximation of the achievable rate. If the kth user is accessed to the mth AP, the ergodic capacity of the kth user, $k \in \mathcal{K}^{\mathrm{b}}$, can be expressed as

$$\begin{aligned} &\mathbb{E}_{g^{\mathrm{b}}_{m,k}}\left(R^{\mathrm{b}}_k\right) \\ &= \mathbb{E}_{g^{\mathrm{b}}_{m,k}}\left[N^{\mathrm{b}}_{m,k} W \log_2\left(1 + \frac{\alpha^{\mathrm{b}}_{m,k} g^{\mathrm{b}}_{m,k} P^{\mathrm{t,b}}_k}{N^{\mathrm{b}}_{m,k} W N_0}\right)\right] \text{(bits/s)}, \end{aligned} \tag{4.2}$$

where $\alpha^{\mathrm{b}}_{m,k}$ is the large-scale channel gain, $g^{\mathrm{b}}_{m,k}$ is the small-scale channel gain, and $P^{\mathrm{t,b}}_k$ is the transmit power.

4.3.4 Offloading Policies

Offloading Policy of URLLC Services

Considering that feedback from receivers to transmitters may cause large overhead and extra delay, we assume that only 1 bit CSI is available at each transmitter, which indicates whether the small-scale channel gain is above a certain threshold, $g_k^{\mathrm{th,u}}$. If the small-scale channel gain is above the threshold, then the packets are offloaded to the MEC with probability one. Otherwise, the offloading probability is zero. Thus, the overall offloading probability, x_k^{u}, equals the probability that $g_{m,k}^{\mathrm{u}} \geq g_k^{\mathrm{th,u}}$, i.e.,

$$x_k^{\mathrm{u}} = \Pr\{g_{m,k}^{\mathrm{u}} \geq g_k^{\mathrm{th,u}}\} = \int_{g_k^{\mathrm{th,u}}}^{\infty} e^{-g} dg = e^{-g_k^{\mathrm{th,u}}}, \tag{4.3}$$

where Rayleigh fading is considered.

Offloading Policy of Delay-tolerant Services

For each long packet, the transmission duration may exceed the channel coherence time. We consider an offloading policy that does not depend on the current small-scale channel gain. When the kth user, $k \in \mathcal{K}^{\mathrm{b}}$, has a packet to process, the packet is offloaded to the MEC server with probability $x_k^{\mathrm{b}} \in [0, 1]$ and processed on the local server with probability $(1 - x_k^{\mathrm{b}})$.

4.3.5 Queueing Model

The queueing models of the local servers and the MEC servers are illustrated in Fig. 4.2. In the local servers, packets are served according to the FCFS order. The difference between URLLC and delay-tolerant services lies in the queueing model before uplink transmission. For URLLC services, each packet is transmitted in one slot. Since the packet arrival process follows a Bernoulli process, the peak arrival rate is one packet per slot, which is equal to the transmission rate of the wireless link. As a result, there is no queue before uplink transmission. For delay-tolerant services, the

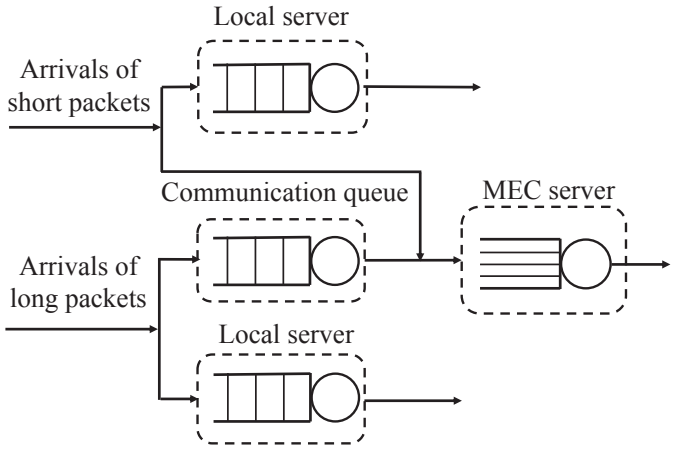

Figure 4.2 – Queueing model

peak arrival rate can be higher than the transmission rate, and hence some packets may wait in a communication queue before uplink transmission.

In the MEC servers, there are short and long packets. If the packets are served according to FCFS order, short packets arriving at the MEC servers after a long packet need to wait for the processing of the latter. To avoid long queueing delay, a PS server is adopted at each AP [121]. On the PS server, the service rate of the server is equally allocated to all the packets in the server. When there are i packets in the mth server, the service rate of each packet is S_m/i. As shown in [122], when there are short and long packets, the PS server outperforms the FCFS server.

4.3.6 Energy Consumption and Processing Rate at Local Server

Let $e_k^{\mathrm{loc},\xi}$ be the energy consumption per CPU cycle of the kth user. According to the measurements in [75, 76], $e_k^{\mathrm{loc},\xi} = k_0(C_k^{\xi})^2$ (J/cycle), where k_0 is a coefficient depending on the chip architecture. The typical value of k_0 is 10^{-15}. The energy consumption per packet at the local server is

$$E_k^{\mathrm{loc},\xi} = e_k^{\mathrm{loc},\xi} c_k^{\xi} = k_0 (C_k^{\xi})^2 c_k^{\xi}, (\mathrm{J/packet}), \tag{4.4}$$

which indicates that the energy consumption for processing one packet increases with the processing rate C_k^{ξ}.

4.4 Problem Formulation and Deep Learning Framework

In this section, we first analyze the QoS constraints of two different services. Then, we formulate an optimization problem to minimize the maximum energy consumption per bit of all the users by optimizing user association, resource allocation, and offloading probabilities subject to the QoS requirements. Finally, we introduce the deep learning framework.

4.4.1 QoS Constraints of URLLC Service

The E2E delay of a packet is defined as the interval between the arrival time of a packet and the time when the processing of the packet is finished. For URLLC service, we denote $D^{\max,\mathrm{u}}$ and $\epsilon^{\max,\mathrm{u}}$ as the required delay bound and the maximal threshold of the tolerable delay bound violation probability, respectively.

QoS Constraints on Local Servers

If a packet is executed locally, the processing delay is

$$D_k^{\mathrm{lc,u}} = \frac{c_k^{\mathrm{u}}}{C_k^{\mathrm{u}}} \text{ (slots)}. \tag{4.5}$$

When the channel is in deep fading, i.e., $g_{m,k}^{\mathrm{u}} < g_k^{\mathrm{th,u}}$, all the packets of a user are served by the local server and the arrival process is a Bernoulli process with average arrival rate λ_k^{u}. Given a constant service rate, C_k^{u}, the queueing model is a Geo/D/1/FCFS model. The complementary cumulative distribution function

(CCDF) of queueing delay, $D_k^{\mathrm{lq,u}}$, in the Geo/D/1/FCFS model is given by [123] as in (4.6).

$$\begin{aligned}\Pr\{D_k^{\mathrm{lq,u}} > i\} =& 1 - \frac{1-(1-x_k^{\mathrm{u}})\lambda_k^{\mathrm{u}} D_k^{\mathrm{lc,u}}}{(1-(1-x_k^{\mathrm{u}})\lambda_k^{\mathrm{u}})^{i+1}} \sum_{l=0}^{j} \left((1-x_k^{\mathrm{u}})\lambda_k^{\mathrm{u}} \left(1-(1-x_k^{\mathrm{u}})\lambda_k^{\mathrm{u}}\right)^{D_k^{\mathrm{lc,u}}-1}\right)^l \\ & (-1)^l \binom{i+l-lD_k^{\mathrm{lc,u}}}{l}, \text{if } jD_k^{\mathrm{lc,u}} \le i \le (j+1)D_k^{\mathrm{lc,u}} - 1.\end{aligned} \tag{4.6}$$

The constraint on E2E delay can be expressed as follows:

$$D_k^{\mathrm{lc,u}} + D_k^{\mathrm{lq,u}} \le D^{\mathrm{max,u}}. \tag{4.7}$$

The queueing delay violation probability should satisfy

$$\epsilon_k^{\mathrm{lq,u}} = \Pr\{D_k^{\mathrm{lq,u}} > (D^{\mathrm{max,u}} - D_k^{\mathrm{lc,u}})\} \le \epsilon^{\mathrm{max,u}}, \tag{4.8}$$

which can be computed according to (4.6).

QoS Constraints When Offloading to a MEC Server

When there are long and short packets in a PS server, an accurate approximation of the CCDF of the processing delay of short packets is given by [122],

$$\epsilon_k^{\mathrm{mc,u}} = (\rho_m^{\mathrm{mc}})^{\left(\frac{S_m D_k^{\mathrm{mc,u}}}{c_k^{\mathrm{u}}} - 1\right)}, \tag{4.9}$$

where ρ_m^{mc} is the workload of the mth MEC server, defined as follows,

$$\rho_m^{\mathrm{mc}} = \frac{\sum_{k\in\mathcal{K}^{\mathrm{u}}} x_k^{\mathrm{u}} \lambda_k^{\mathrm{u}} c_k^{\mathrm{u}} + \sum_{k\in\mathcal{K}^{\mathrm{b}}} x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} \bar{c}_k^{\mathrm{b}}}{S_m}, \tag{4.10}$$

where $\bar{c}_k^{\mathrm{b}}$ is the average number of CPU cycles required to process a packet of delay-tolerant services. The E2E delay of a packet when offloading to the MEC server

should satisfy the following constraint:

$$1 + D_k^{\mathrm{mc,u}} \leq D^{\mathrm{max,u}}, \tag{4.11}$$

where data transmission occupies one slot.

Due to decoding errors and processing delay violation, the overall packet loss probability can be expressed as $\epsilon_k^{\mathrm{u}} = 1 - (1 - \epsilon_k^{\mathrm{mc,u}})(1 - \epsilon_k^{\mathrm{d,u}}) \approx \epsilon_k^{\mathrm{mc,u}} + \epsilon_k^{\mathrm{d,u}}$, where the approximation is accurate since $\epsilon_k^{\mathrm{mc,u}}$ and $\epsilon_k^{\mathrm{d,u}}$ are extremely small. Then, the constraint on the reliability of the kth user can be expressed as, $\epsilon_k^{\mathrm{mc,u}} + \epsilon_k^{\mathrm{d,u}} \leq \epsilon^{\mathrm{max,u}}$. We set the upper bound of the decoding error probability and the upper bound of the processing delay violation probability to be equal, i.e.,

$$\epsilon_k^{\mathrm{mc,u}} \leq 0.5\epsilon^{\mathrm{max,u}}, \epsilon_k^{\mathrm{d,u}} \leq 0.5\epsilon^{\mathrm{max,u}}. \tag{4.12}$$

As shown in [104], setting different packet loss probabilities to be equal leads to minor power loss. By substituting the processing delay violation probability in (4.9) into constraint $\epsilon_k^{\mathrm{mc,u}} \leq 0.5\epsilon^{\mathrm{max,u}}$, we can derive the constraint on the workload as follows:

$$\rho_m^{\mathrm{mc}} \leq (0.5\epsilon^{\mathrm{max,u}})^{\left\lceil \frac{c_k^{\mathrm{u}}}{S_m(D^{\mathrm{max,u}}-1)-c_k^{\mathrm{u}}} \right\rceil} \triangleq \rho_{\mathrm{th}}. \tag{4.13}$$

4.4.2 Stability of Delay-tolerant Services

For delay-tolerant services, we only need to ensure the queueing system is stable, i.e., the average service rate is equal to or higher than the average arrival rate.

Rate Constraint of Local Servers

To ensure the stability of the queueing system on local servers, we need to guarantee that the processing rate is higher than the average data arrival rate:

$$C_k^{\mathrm{b}} \geq (1 - x_k^{\mathrm{b}})\lambda_k^{\mathrm{b}}\bar{c}_k^{\mathrm{b}}, (\mathrm{cycles/slot}). \tag{4.14}$$

Besides, the processing rate should not exceed the maximal computing capacity of the server, $C_k^{\mathrm{b}} \leq C_k^{\max,\mathrm{b}}$.

Rate Constraint of Wireless Link

To ensure the stability of the communication queue in Fig. 4.2, we need to guarantee that the average transmission rate of the wireless link is equal to or higher than the average data arrival rate, i.e.,

$$\mathbb{E}_{g_{m,k}^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right) \geq x_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} / T_{\mathrm{s}}, \tag{4.15}$$

where $\bar{b}_k^{\mathrm{b}}$ is the average number of bits in a long packet.

Workload Constraint on the MEC Server

In the case that only delay-tolerant services offload packets to the mth MEC server, $x_k^{\mathrm{u}} = 0, \forall k \in K^{\mathrm{u}}$, the stability of the PS server can be satisfied if the workload meets the following constraint:

$$\rho_m^{\mathrm{mc}} = \frac{\sum_{k \in \mathcal{K}^{\mathrm{b}}} x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} \bar{c}_k^{\mathrm{b}}}{S_m} \leq 1. \tag{4.16}$$

Otherwise, constraint (4.13) should be satisfied.

4.4.3 Objective Function: Normalized Energy Consumption

Our goal is to minimize the normalized energy consumption, defined as the energy consumption per bit.

URLLC Services

For URLLC services, the circuit power at the local server and the average transmit power for packets offloading are $\lambda_k^{\mathrm{u}} E_k^{\mathrm{loc,u}}$ and $\lambda_k^{\mathrm{u}} P_k^{\mathrm{t,u}} T_{\mathrm{s}}$ (J/slot), respectively. Since

the average data arrival rate is $\lambda_k^{\mathrm{u}} b_k^{\mathrm{u}}$ (bits/slot), the normalized energy consumption is

$$\begin{aligned}\eta_k^{\mathrm{u}} &= \frac{(1-x_k^{\mathrm{u}})\lambda_k^{\mathrm{u}} E_k^{\mathrm{loc,u}} + x_k^{\mathrm{u}}\lambda_k^{\mathrm{u}} P_k^{\mathrm{t,u}} T_{\mathrm{s}}}{\lambda_k^{\mathrm{u}} b_k^{\mathrm{u}}} \\ &= \frac{(1-x_k^{\mathrm{u}}) E_k^{\mathrm{loc,u}}}{b_k^{\mathrm{u}}} + \frac{x_k^{\mathrm{u}} P_k^{\mathrm{t,u}} T_{\mathrm{s}}}{b_k^{\mathrm{u}}} \text{ (J/bit)}. \end{aligned} \tag{4.17}$$

Delay-tolerant Services

If a packet is processed at the local server, the average energy consumption is $E_k^{\mathrm{loc,b}} = k_0 (C_k^{\mathrm{b}})^2 \bar{c}_k^{\mathrm{b}}$, which is obtained from (4.4). Then, the energy consumption per bit is $\eta_k^{\mathrm{loc,b}} = E_k^{\mathrm{loc,b}} / \bar{b}_k^{\mathrm{b}}$. If the packet is offloaded to a MEC server, the energy consumption and the average amount of data transmitted in each slot can be expressed as $P_k^{\mathrm{t,b}} T_{\mathrm{s}}$ and $x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}}$, respectively. Then, the energy consumption per bit is $\eta_k^{\mathrm{mec,b}} = P_k^{\mathrm{t,b}} T_{\mathrm{s}} / x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}}$. Therefore, the normalized energy consumption of user k, $k \in \mathcal{K}^{\mathrm{b}}$, can be expressed as follows:

$$\begin{aligned}\eta_k^{\mathrm{b}} &= (1-x_k^{\mathrm{b}})\eta_k^{\mathrm{loc,b}} + x_k^{\mathrm{b}} \eta_k^{\mathrm{mec,b}} \\ &= (1-x_k^{\mathrm{b}}) \frac{E_k^{\mathrm{loc,b}}}{\bar{b}_k^{\mathrm{b}}} + x_k^{\mathrm{b}} \frac{P_k^{\mathrm{t,b}} T_{\mathrm{s}}}{x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}}} \text{ (J/bit)}. \end{aligned} \tag{4.18}$$

4.4.4 Optimization Problem

To avoid users with bad channel conditions or high task arrival rates experiencing high energy consumption, we take fairness among all the users into consideration by minimizing the maximal normalized energy consumption of the $K^{\mathrm{u}} + K^{\mathrm{b}}$ users. If there is a central control plane that manages user association and resource allocation, the optimization problem can be formulated as follows:

$$\mathcal{P}_1 : \min_{\beta_{m,k}^{\xi}, P_k^{\mathrm{t},\xi}, N_{m,k}^{\xi}, x_k^{\xi}} \max_{k \in \mathcal{K}^{\xi}} \eta_k^{\xi} \tag{4.19}$$

$$\text{s.t. } x_k^{\xi} \in [0,1], \forall k \in \mathcal{K}^{\xi}, \tag{4.19a}$$

$$\sum_{k\in\mathcal{K}^{\xi}} N_{m,k}^{\xi} \leq N^{\max}, m = 1, ..., M \tag{4.19b}$$

$$\sum_{m\in\mathcal{M}} \beta_{m,k}^{\xi} = 1, \tag{4.19c}$$

$$\rho_m^{\mathrm{mc}} \leq \begin{cases} 1, \text{if } x_k^{\mathrm{u}} = 0, \forall k \in \mathcal{K}^{\mathrm{u}}; \\ \rho_{\mathrm{th}}, \text{otherwise}, \end{cases} \tag{4.19d}$$

$$C_k^{\xi} \leq C_k^{\max,\xi}, \forall k \in \mathcal{K}^{\xi}, \tag{4.19e}$$

$$P_k^{\mathrm{t},\xi} \leq P_k^{\max,\xi}, \forall k \in \mathcal{K}^{\xi}, \tag{4.19f}$$

$$(4.1), (4.2), (4.7), (4.8), (4.11), (4.12), (4.14) \text{ and } (4.15),$$

where $N^{\max}$ is the total number of subcarriers of each AP and $P_k^{\max,\xi}$ is the maximal transmit power of the kth user. Constraint (4.19d) is obtained from (4.13) and (4.16). Since the required transmit power is determined by the bandwidth allocation and the offloading probability, it can be removed from the optimization variables. The relation between the optimal solution and the inputs, i.e., large-scale channel gains and average task arrival rates, is denoted as $\pi_1 := \boldsymbol{\alpha}, \boldsymbol{\lambda} \rightarrow \boldsymbol{\beta}^*, \boldsymbol{N}^*, \boldsymbol{x}^*$, where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^{\mathrm{u}}, ..., \boldsymbol{\alpha}_{K^{\mathrm{u}}}^{\mathrm{u}}, \boldsymbol{\alpha}_1^{\mathrm{b}}, ..., \boldsymbol{\alpha}_{K^{\mathrm{b}}}^{\mathrm{b}})^{\mathrm{T}}$, $\boldsymbol{\alpha}_k^{\xi} = (\alpha_{1,k}^{\xi}, ..., \alpha_{M,k}^{\xi})^{\mathrm{T}}$, $\boldsymbol{\lambda} = (\lambda_1^{\mathrm{u}}, ..., \lambda_{K^{\mathrm{u}}}^{\mathrm{u}}, \lambda_1^{\mathrm{b}}, ..., \lambda_{K^{\mathrm{b}}}^{\mathrm{b}})^{\mathrm{T}}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{u}}, ..., \boldsymbol{\beta}_{K^{\mathrm{u}}}^{\mathrm{u}}, \boldsymbol{\beta}_1^{\mathrm{b}}, ..., \boldsymbol{\beta}_{K^{\mathrm{b}}}^{\mathrm{b}})^{\mathrm{T}}$, $\boldsymbol{\beta}_k^{\xi} = (\beta_{1,k}^{\xi}, ..., \beta_{M,k}^{\xi})^{\mathrm{T}}$, $\boldsymbol{N} = (\boldsymbol{N}_1^{\mathrm{u}}, ..., \boldsymbol{N}_{K^{\mathrm{u}}}^{\mathrm{u}}, \boldsymbol{N}_1^{\mathrm{b}}, ..., \boldsymbol{N}_{K^{\mathrm{b}}}^{\mathrm{b}})^{\mathrm{T}}$, $\boldsymbol{N}_k^{\xi} = (N_{1,k}^{\xi}, ..., N_{M,k}^{\xi})^{\mathrm{T}}$, $\boldsymbol{x} = (x_1^{\mathrm{u}}, ..., x_{K^{\mathrm{u}}}^{\mathrm{u}}, x_1^{\mathrm{b}}, ..., x_{K^{\mathrm{b}}}^{\mathrm{b}})^{\mathrm{T}}$, and $(\cdot)^{\mathrm{T}}$ denotes the transpose operator.

In practice, a user can subscribe to both kinds of services. If the kth user subscribes to both kinds of services, λ_k^{u} and λ_k^{b} are referred to as the average task arrival rates of URLLC and delay-tolerant services, respectively. The large-scale channel gains of the two kinds of services are the same, i.e., $\alpha_k^{\mathrm{u}} = \alpha_k^{\mathrm{b}}$. In the local server of the kth user, the packets from different services are waiting in two separated FCFS queues. The energy consumption per packet in (4.4) becomes $E_k^{\mathrm{loc}} = k_0(C_k)^2(c_k^{\mathrm{u}} + c_k^{\mathrm{b}})$. The transmit power constraint of the kth user becomes $P_k^{\mathrm{u}} + P_k^{\mathrm{d}} \leq P_k^{\max}$. The rest of the constraints remain the same.

Note that user association is managed by MME, but resource allocation and offloading probabilities are determined by each AP. The problem $\mathcal{P}_1$ is decomposed into two

subproblems that are solved in two timescales at MME and APs, respectively. In the first subproblem, each AP optimizes resource allocation and offloading probabilities with given user association scheme. In the second subproblem, the MME optimizes user association scheme with a DL algorithm, where the behavior of each AP (i.e., the optimal resource allocation and task offloading policy) is taken into account.

- Problem $\mathcal{P}_2$: The problem that optimizes subcarrier allocation and offloading probability can be formulated as follows:

$$\begin{aligned} \mathcal{P}_2 : & \min_{N_{m,k}^{\xi}, x_k^{\xi}} \max_{k \in \mathcal{K}^{\xi}} \eta_k^{\xi}, \\ \text{s.t. } & (4.19a), (4.19b), (4.19d), (4.19e), (4.19f), \\ & (4.1), (4.2), (4.7), (4.8), (4.11), (4.12), (4.14) \text{ and } (4.15). \end{aligned} \tag{4.20}$$

  The relation between the optimal $(\boldsymbol{N}^*, \boldsymbol{x}^*)$ and $(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\beta})$ is denoted as $\pi_2 := \boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\beta} \rightarrow \boldsymbol{N}^*, \boldsymbol{x}^*$. The minimal normalized energy consumption achieved with π_2 is denoted as $Q_2^*(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\beta} | \pi_2)$, which indicates that the normalized energy consumption depends on the user association.

- Problem $\mathcal{P}3$: The problem that optimizes user association scheme can be formulated as follows:

$$\begin{aligned} \mathcal{P}_3 : & \min_{\beta_{m,k}^{\xi}} Q_2^*(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\beta} | \pi_2), \\ \text{s.t. } & (4.19c) \end{aligned} \tag{4.21}$$

  The relation between the optimal $\boldsymbol{\beta}^*$ and $\boldsymbol{\alpha}, \boldsymbol{\lambda}$ is denoted as $\pi_3 := \boldsymbol{\alpha}, \boldsymbol{\lambda} \rightarrow \boldsymbol{\beta}^*$. The minimal normalized energy consumption achieved with π_3 is denoted as $Q_3^*(\boldsymbol{\alpha}, \boldsymbol{\lambda} | \pi_2, \pi_3)$, which also depends on π_2.

4.4.5 Structure of Deep Learning

It is worth noting that both $\mathcal{P}_2$ and $\mathcal{P}_3$ are non-convex. We will propose an optimization algorithm to solve problem $\mathcal{P}_2$ and apply the deep learning algorithm to solve problem $\mathcal{P}_3$.

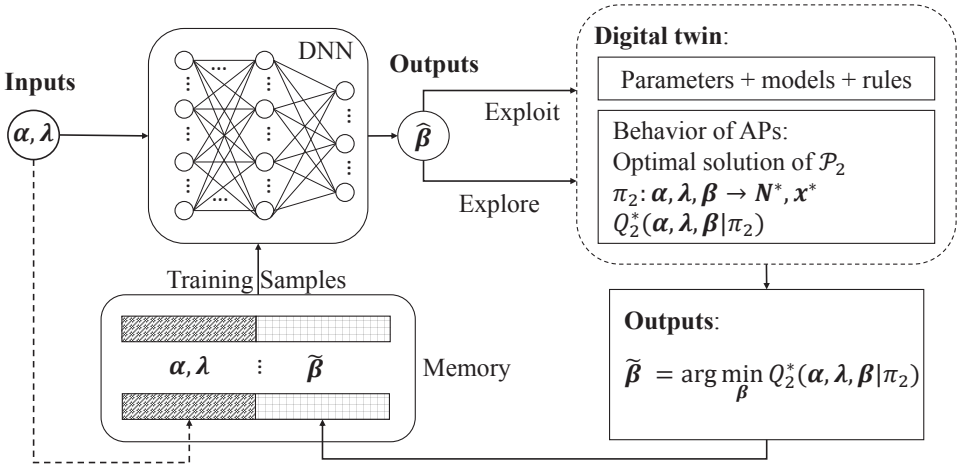

Figure 4.3 – Digital twin enabled DL algorithm

The framework of the digital twin-enabled DL algorithm is illustrated in Fig. 4.3. The inputs of the DNN are the large-scale channel gains from users to APs and the average task arrival rates of all the users, while the output of the DNN is the user association scheme. The direct output of the DNN is denoted as $\hat{\boldsymbol{\beta}}$, based on which we explore user association schemes. For a user association scheme, we can obtain the related normalized energy consumption from the digital twin. From its feedback we find the best user association scheme, $\tilde{\boldsymbol{\beta}}$, that minimizes the normalized energy consumption among the user association schemes randomly generated according to exploration policies. Finally, the inputs $\boldsymbol{\alpha}, \boldsymbol{\lambda}$ and the best output $\tilde{\boldsymbol{\beta}}$ are saved in the memory and will be used to train the DNN.

4.5 Algorithm to Solve Problem $\mathcal{P}_2$

In this section, we propose a method to find the optimal solution of problem $\mathcal{P}_2$. Note that when the user association scheme is given, problem $\mathcal{P}_2$ can be decomposed into multiple single-AP problems. In this section, we omit index m for notational simplicity.

4.5.1 Outline of the Algorithm

From problem $\mathcal{P}_2$ we can see that only constraints (4.19b) and (4.19d) depend on the optimization variables of all the users, and the other constraints only depend on the resource allocation and offloading probability of a single user. To solve problem $\mathcal{P}_2$, we first remove constraints (4.19b) and (4.19d) and decompose the problem into multiple single-user problems. After solving these, we check whether constraints (4.19b) and (4.19d) are satisfied or not. The algorithm is summarized in Table 4.2.

To remove constraint (4.19b), we first find the minimum of the maximal normalized energy consumption via binary search. For a given value of η^{th}, we minimize the total number of subcarriers that is required to guarantee $\max_{k\in\mathcal{K}^\xi} \eta_k^\xi \leq \eta^{\text{th}}$, i.e.,

$$\min_{N_k^\xi, x_k^\xi} \sum_{k\in\mathcal{K}^{\text{u}}} N_k^{\text{u}} + \sum_{k\in\mathcal{K}^{\text{d}}} N_k^{\text{d}}, \tag{4.22}$$

$$\text{s.t. } \max_{k\in\mathcal{K}^\xi} \eta_k^\xi \leq \eta^{\text{th}}, \tag{4.22a}$$

$$(4.19\text{a}), (4.19\text{d}), (4.19\text{e}), (4.19\text{f}),$$

$$(4.1), (4.2), (4.7), (4.8), (4.11), (4.12), (4.14) \text{ and } (4.15).$$

If the required bandwidth is larger than $N^{\max}$, η^{th} cannot be achieved, and the minimum of $\max_{k\in\mathcal{K}^\xi} \eta_k^\xi$ is higher than η^{th}. Otherwise, the minimum of $\max_{k\in\mathcal{K}^\xi} \eta_k^\xi$ is lower than η^{th}. Via binary search, η^{th} converges to the minimum of $\max_{k\in\mathcal{K}^\xi} \eta_k^\xi$, and the corresponding bandwidth allocation and offloading probabilities are the optimal solution of problem $\mathcal{P}_2$ (see proof in Subsection 4.5.3.).

Table 4.2 – Offloading and Subcarrier Allocation Algorithm

Require: Large-scale channel gains, $\boldsymbol{\alpha}$, the user association scheme, $\boldsymbol{\beta}$, and the required searching precisions of normalized energy consumption, number of subcarriers, and offloading probability, σ_η, σ_N, and σ_x.

Initialize $\eta^{\text{lb}} = 0$ and $\eta^{\text{ub}} = \mathcal{E}$, where $\mathcal{E}$ is the maximal normalized energy consumption when the equalities in constraints (4.19e) and (4.19f) hold, $\sigma_\eta = 10^{-10}$, $\sigma_N = 10^{-3}$.
while $\eta^{\text{ub}} - \eta^{\text{lb}} > \sigma_\eta$ **do**
  $\eta^{\text{th}} = (\eta^{\text{ub}} + \eta^{\text{lb}})/2$.
  Initialize $N_k^{\text{lb},\xi} = 0$, $N_k^{\text{ub,b}} = N^{\max}$ and $N_k^{\text{ub,u}} = \tilde{N}_k^{\text{u}}$.
  for $k \in K^\xi$ **do**
    while $N_k^{\text{b},\xi} - N_k^{\text{lb},\xi} > \sigma_N$ **do**
      $N_k^{\text{th},\xi} = (N_k^{\text{ub},\xi} + N_k^{\text{lb},\xi})/2$.
      Minimize η_k^ξ by optimizing $\hat{x}_k^\xi(N_k^{\text{th},\xi})$ according to the method in Section 4.5.2.
      if $\hat{x}_k^\xi(N_k^{\text{th},\xi}) == 0$ **then**
        $N_k^{\text{th},\xi} = 0$; Break;
      else
        if $\eta_k^\xi < \eta^{\text{th}}$ **then**
          $N_k^{\text{ub},\xi} = N_k^{\text{th},\xi}$;
        else
          $N_k^{\text{lb},\xi} = N_k^{\text{th},\xi}$.
        end if
      end if
    end while
    if $\eta_k > \eta^{\text{th}}$ **then**
      $\eta^{\text{lb}} = \eta^{\text{th}}$; Break. (problem $\mathcal{P}_2$ is infeasible)
    end if
  end for
  if $\sum_{k\in\mathcal{K}^\xi} N_k^{\text{th},\xi} \le N^{\max}$ and $\rho_m^{\text{mc}} \le \begin{cases} 1, & \text{if } \sum_{k\in\mathcal{K}^{\text{u}}} N_k^{\text{u}} = 0. \\ \rho_{\text{th}}, & \text{if } \sum_{k\in\mathcal{K}^{\text{u}}} N_k^{\text{u}} \neq 0. \end{cases}$ **then**
    $\eta^{\text{ub}} = \eta^{\text{th}}$; (problem $\mathcal{P}_2$ is feasible)
  else
    $\eta^{\text{lb}} = \eta^{\text{th}}$. (problem $\mathcal{P}_2$ is infeasible)
  end if
end while
return If $\eta^{\text{th}} = \mathcal{E}$, the problem is infeasible. Otherwise, $\eta^* := \eta^{\text{th}}$, $N_k^{\xi*} := N_k^{\text{th},\xi}$ and $x_k^{\xi*} := \hat{x}_k^\xi(N_k^{\text{th},\xi})$

In the second step, we remove constraint (4.19d), and decompose problem (4.22) into multiple single-user problems. For each single-user problem, we search the minimum number of subcarriers allocated to each user via binary search. For a given value of $N_k^{\text{th},\xi}$, we minimize η_k^ξ subject to $N_k^\xi = N_k^{\text{th},\xi}$, i.e.,

$$\min_{x_k^\xi} \eta_k^\xi, \tag{4.23}$$

$$\text{s.t. } N_k^\xi = N_k^{\text{th},\xi}, \tag{4.23a}$$

$$(4.19a), (4.19e), (4.19f),$$

$$(4.1), (4.2), (4.7), (4.8), (4.11), (4.12), (4.14) \text{ and } (4.15),$$

If $\eta_k^\xi \leq \eta^{\text{th}}$, then $N_k^{\xi*} \leq N_k^{\text{th},\xi}$. Otherwise, $N_k^{\xi*} \geq N_k^{\text{th},\xi}$ (see proof in Subsection 4.5.3). Thus, $N_k^{\text{th},\xi}$ either converges the minimum of N_k^ξ or $N^{\max}$ (i.e., $\eta_k^\xi > \eta^{\text{th}}$ even with $N_k^\xi = N^{\max}$).

After obtaining the solutions of the single-user problems, we have to check whether constraints (4.19b) and (4.19d) are satisfied or not in Line 23 of the algorithm in Table 4.2.

If constraints (4.19b) and (4.19d) cannot be satisfied by the end of the binary search, problem $\mathcal{P}_2$ is infeasible and the AP cannot guarantee the QoS requirements of all the users associated with it. In this case, the normalized energy consumption of the user association scheme will be set as infinite in the learning framework in Fig. 4.3. This user association scheme will not be used to train the DNN since the QoS requirements cannot be satisfied.

4.5.2 Optimal Offloading Probability

In this subsection, we show how to solve problem (4.23). Since the offloading probability depends on bandwidth allocation, we denote the optimal offloading probability as $\hat{x}_k^\xi(N_k^{\text{th},\xi})$.

URLLC Services

For URLLC services, the offloading probability is determined by the threshold of small-scale channel gain $g_k^{\mathrm{th,u}}$. To find the optimal offloading probability, we optimize $g_k^{\mathrm{th,u}}$ by the following three steps to meet all the constraints in problem (4.23).

In the first step, we find the minimal energy consumption per packet at the local server. Since the normalized energy consumption increases with the service rate, we first find the minimal service rate that is required to satisfy the constraints on the E2E delay and the queueing delay violation probability, i.e., (4.7) and (4.8). By substituting $D_k^{\mathrm{lc,u}} = \frac{c_k^{\mathrm{u}}}{C_k^{\mathrm{u}}}$ into (4.8), we have $\epsilon_k^{\mathrm{lq,u}} = \Pr\{D_k^{\mathrm{lq,u}} > D^{\mathrm{max,u}} - \frac{c_k^{\mathrm{u}}}{C_k^{\mathrm{u}}}\}$. From the CCDF of the queueing delay in (4.6), the minimal service rate can be obtained when $\epsilon_k^{\mathrm{lq,u}} = \epsilon^{\mathrm{max,u}}$. We denote the minimal service rate that is required to satisfy $D_k^{\mathrm{lq,u}}$ and $\epsilon_k^{\mathrm{lq,u}}$ as $C_k^{\mathrm{u}*}$. According to (4.4), the minimal energy consumption per packet at the local servers is $E_k^{\mathrm{loc,u}*} = k_0 (C_k^{\mathrm{u}*})^2 c_k^{\mathrm{u}}$ (J/packet).

In the second step, we find the minimal value of $g_k^{\mathrm{th,u}}$ that can satisfy the constraints on decoding error probability and maximal transmit power, i.e., $\epsilon_k^{\mathrm{d,u}} \leq 0.5\epsilon^{\mathrm{max,u}}$ in (4.12) and (4.19f). The decoding error probability can be obtained from (4.1) by setting $T_{\mathrm{s}} R_k^{\mathrm{u}} = b_k^{\mathrm{u}}$. Then, the required transmit power that satisfies $\epsilon_k^{\mathrm{d,u}} = 0.5\epsilon^{\mathrm{max,u}}$ can be expressed as follows:

$$P_k^{\mathrm{t,u}} = \frac{1}{g_k^{\mathrm{th,u}}} \varrho, \tag{4.24}$$

where

$$\varrho = \frac{N_k^{\mathrm{th,u}} W N_0}{\alpha_k^{\mathrm{u}}} \times \left[\exp\left(\sqrt{\frac{1}{T_{\mathrm{s}} N_k^{\mathrm{th,u}} W}} f_Q^{-1}(0.5\epsilon^{\mathrm{max,u}}) + \frac{b_k^{\mathrm{u}} \ln 2}{T_{\mathrm{s}} N_k^{\mathrm{th,u}} W} \right) - 1 \right], \tag{4.25}$$

and the approximation $V_k^{\mathrm{u}} \approx 1$ is applied, which is accurate when the receive SNR is higher than 5 dB [124, 125]. To satisfy the maximal transmit power constraint, we

can obtain the minimal $g_k^{\mathrm{th,u}}$ by substituting (4.24) into $P_k^{\mathrm{t,u}} = P^{\mathrm{max,u}}$, i.e.,

$$g_k^{\mathrm{min,u}} = \frac{1}{P^{\mathrm{max,u}}}\varrho. \tag{4.26}$$

In the third step, we derive the closed-form expression of the optimal threshold, $\hat{g}_k^{\mathrm{th,u}}$, that minimizes the normalized energy consumption. Substituting $x_k^{\mathrm{u}} = e^{-g_k^{\mathrm{th,u}}}$ and $P_k^{\mathrm{t,u}} = \frac{1}{g_k^{\mathrm{th,u}}}\varrho$ into the expression of η_k^{u} in (4.17), we can derive the derivative of η_k^{u} on x_k^{u} as follows:

$$\eta_k^{\mathrm{u}'} = \frac{e^{-g_k^{\mathrm{th,u}}}}{b_k^{\mathrm{u}}}\left(E_k^{\mathrm{loc,u*}} - \frac{\varrho T_{\mathrm{s}}}{g_k^{\mathrm{th,u}}} - \frac{\varrho T_{\mathrm{s}}}{\left(g_k^{\mathrm{th,u}}\right)^2}\right). \tag{4.27}$$

From (4.27), we can see that the sign of $\eta_k^{\mathrm{u}'}$ is the same as $f(g_k^{\mathrm{th,u}}) \triangleq E_k^{\mathrm{loc,u*}} - \frac{\varrho T_s}{g_k^{\mathrm{th,u}}} - \frac{\varrho T_s}{\left(g_k^{\mathrm{th,u}}\right)^2}$. When $g_k^{\mathrm{th,u}} \to 0$, $f(g_k^{\mathrm{th,u}}) < 0$. When $g_k^{\mathrm{th,u}} \to \infty$, $f(g_k^{\mathrm{th,u}}) > 0$. Moreover, $f(g_k^{\mathrm{th,u}})$ strictly increases with $g_k^{\mathrm{th,u}}$. Therefore, η_k^{u} first strictly decreases and then strictly increases with $g_k^{\mathrm{th,u}}$, and there is a unique solution of $g_k^{\mathrm{th,u}}$ that minimizes η_k^{u} (i.e., $f(g_k^{\mathrm{th,u}}) = 0$). The solution of $f(g_k^{\mathrm{th,u}}) = 0$ can be derived as follows:

$$\tilde{g}_k^{\mathrm{th,u}} = \frac{1}{2}\left(\frac{\varrho T_{\mathrm{s}}}{E_k^{\mathrm{loc,u*}}} + \sqrt{\left(\frac{\varrho T_{\mathrm{s}}}{E_k^{\mathrm{loc,u*}}}\right)^2 + 4\frac{\varrho T_{\mathrm{s}}}{E_k^{\mathrm{loc,u*}}}}\right). \tag{4.28}$$

If $g_k^{\mathrm{min,u}} \leq \tilde{g}_k^{\mathrm{th,u}}$, then $\tilde{g}_k^{\mathrm{th,u}}$ is the optimal threshold that minimizes η_k^{u} subject to the transmit power constraint. Otherwise, since η_k^{u} increases with $g_k^{\mathrm{th,u}}$ in the region $(\tilde{g}_k^{\mathrm{th,u}}, \infty)$, $g_k^{\mathrm{min,u}}$ is the optimal threshold. Thus, we have

$$\hat{g}_k^{\mathrm{th,u}} = \max\left\{g_k^{\mathrm{min,u}}, \tilde{g}_k^{\mathrm{th,u}}\right\}. \tag{4.29}$$

By substituting $\hat{g}_k^{\mathrm{th,u}}$ into (4.3), we can obtain the optimal offloading probability, $\hat{x}_k^{\mathrm{u}}(N_k^{\mathrm{th,u}}) = e^{-\hat{g}_k^{\mathrm{th,u}}}$.

Delay-tolerant Services

We apply the binary search to find the optimal offloading probabilities of delay-tolerant services that meet the constraints of problem (4.23). Given $N_k^{\mathrm{th,b}}$, the upper bound of the offloading probability that satisfies the constraints on average data rate and maximal transmit power in (4.15) and (4.19f) can be obtained by substituting $P_k^{\mathrm{t,b}} = P_k^{\max,\mathrm{b}}$ and $\mathbb{E}_{g_k^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right)$ in (4.2) into $\mathbb{E}_{g_k^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right) = x_k^{\mathrm{b}}\bar{b}_k^{\mathrm{b}}\lambda_k^{\mathrm{b}}/T_{\mathrm{s}}$. The lower bound of the offloading probability that satisfies the service rate constraint at the local server in (4.14) can be obtained by substituting $C_k^{\mathrm{b}} = C_k^{\max,\mathrm{b}}$ into $C_k^{\mathrm{b}} = (1-x_k^{\mathrm{b}})\lambda_k^{\mathrm{b}}\bar{c}_k^{\mathrm{b}}$. Let $x_k^{\mathrm{ub,b}}$ and $x_k^{\mathrm{lb,b}}$ be the upper and lower bounds of the offloading probability, respectively. If $x_k^{\mathrm{lb,b}} > x_k^{\mathrm{ub,b}}$, the problem is infeasible, which may happen when the average packet arrival rate λ_k^{b} is large. When the problem is feasible, to find the optimal offloading probability, $\hat{x}_k^{\mathrm{b}}(N_k^{\mathrm{th,b}})$ in $[x_k^{\mathrm{lb,b}}, x_k^{\mathrm{ub,b}}]$, we need the following proposition:

Proposition 4.1. *η_k^{b} in (4.18) is convex in x_k^{b}.*

Proof. See proof in Appendix A.1. □

Then, the optimal offloading probability $\hat{x}_k^{\mathrm{b}}(N_k^{\mathrm{th,b}})$ that minimizes η_k^{b} can be obtained via binary search.

4.5.3 Convergence of the Algorithm

In this subsection, we first prove that for a given threshold of the normalized energy consumption, η^{th}, the algorithm in Table 4.2 can find the minimum bandwidth that is required to achieve the threshold (from Line 4 to Line 22 in Table 4.2). To prove it, we only need to prove that the normalized energy consumption decreases with $N_k^{\mathrm{th},\xi}$.

Property 4.1. *The minimum of the objective function (4.23) decreases with $N_k^{\mathrm{th},\xi}$ in the region $[N_k^{\mathrm{ub},\xi}, N_k^{\mathrm{lb},\xi}]$.*

Proof. See proof in Appendix B.2. □

The above property indicates that the binary search converges to the minimal $N_k^{\text{th},\xi}$ that can guarantee $\eta_k \leq \eta^{\text{th}}$, unless it is infeasible (as shown in Line 20 of Table 4.2).

To find out whether problem $\mathcal{P}_2$ is feasible or not, we minimize the total number of subcarriers and see whether it is less than the total number of subcarriers of an AP. Besides, we also need to minimize the total offloading probability and see whether it satisfies the constraint in (4.19d).[2] The following property indicates that minimizing the offloading probability of the kth user is equivalent to minimizing the number of subcarriers allocated to it.

Property 4.2. *The optimal offloading probability $\hat{x}_k^{\xi}(N_k^{\text{th},\xi})$ increases with $N_k^{\text{th},\xi}$.*

Proof. See proof in Appendix B.3. □

Therefore, by minimizing the sum of the numbers of subcarriers, we also obtained the minimum of the sum of the offloading probabilities. In other words, both the sum of the numbers of subcarriers and the workload at the MEC server are minimized with the algorithm from Line 4 to Line 22 in Table 4.2. As a result, problem $\mathcal{P}_2$ is feasible if and only if constraints (4.19b) and (4.19d) are satisfied with $N_k^{\text{th},\xi}$ and $\hat{x}_k^{\xi}(N_k^{\text{th},\xi})$, i.e., the condition in Line 23 in Table 4.2.

If problem $\mathcal{P}_2$ is feasible when the normalized energy consumption equals η^{th}, then η^{th} is achievable and the minimal normalized energy consumption $\eta^* \leq \eta^{\text{th}}$. Otherwise, $\eta^* > \eta^{\text{th}}$. Therefore, with the binary search (i.e., Lines 2,3 and Lines 23 to 27), η^{th} converges to η^*. The corresponding $N_k^{\text{th},\xi}$ and $\hat{x}_k^{\xi}(N_k^{\text{th},\xi})$ converge to the optimal solution $N_k^{\xi*}$ and $x_k^{\xi*}$.

4.5.4 Complexity Analysis

Given the required searching precision of the normalized energy consumption σ_η, it takes $\mathcal{O}\left(\log_2(\frac{\eta^{\text{ub}}}{\sigma_\eta})\right)$ steps to obtain the minimum of the maximal normalized energy consumption of all the users. To achieve a target η^{th}, $(K^{\text{u}}+K^{\text{b}})\mathcal{O}\left(\log_2(\frac{N_k^{\text{ub},\xi}}{\sigma_N})\right)$ steps

[2]The rest of the constraints are satisfied with the solution of problem (4.23).

are needed to obtain the required numbers of subcarriers of $K^{\mathrm{u}}+K^{\mathrm{b}}$ users, where σ_N is the required searching precision of the number of subcarriers. For a given number of subcarriers, it takes $\mathcal{O}\left(\log_2(\frac{1}{\sigma_x})\right)$ steps to obtain the optimal offloading probability that minimizes the normalized energy consumption of the delay-tolerant user in the region $[0,1]$, where σ_x is the required searching precision of offloading probability. For URLLC services, the optimal offloading probability can be obtained in the closed-form expression in (4.29). Therefore, the complexity of the algorithm can be expressed as $(K^{\mathrm{u}}+K^{\mathrm{b}})\mathcal{O}\left(\log_2(\frac{\eta^{\mathrm{ub}}}{\sigma_\eta})\log_2(\frac{N_k^{\mathrm{ub},\xi}}{\sigma_N})\log_2(\frac{1}{\sigma_x})\right)$, which increases linearly with $(K^{\mathrm{u}}+K^{\mathrm{b}})$.

4.6 Deep Learning for User Association

In this section, we discuss how to explore user association schemes and how to train the DNN.

The set of all the weights and biases of the DNN is denoted as $\Theta = \{\boldsymbol{W}^{[l]}, \boldsymbol{b}^{[l]}, l = 1, ..., L_{\mathrm{dnn}}\}$, where L_{dnn} is the number of layers, $\boldsymbol{W}^{[l]}$ and $\boldsymbol{b}^{[l]}$ are the weights and the biases in the lth layer, respectively. The relation between the input and output of the lth layer can be expressed as

$$\boldsymbol{Y}^{[l]} = f_\delta\left(\boldsymbol{W}^{[l]}\boldsymbol{X}^{[l]} + \boldsymbol{b}^{[l]}\right), \tag{4.30}$$

where $\boldsymbol{X}^{[l]}$ and $\boldsymbol{Y}^{[l]}$ are the input and output of the lth layer, and the activation function, $f_\delta(x)$, is an element-wise operation of a vector. In this chapter, we use ReLU function as the activation function, i.e., $f_\delta(x) = \max(0, x)$.

In each learning epoch, the large-scale channel gains, $\boldsymbol{\alpha}$, and the average task arrival rates, $\boldsymbol{\lambda}$, are estimated by the system, and are used to calculate $\hat{\boldsymbol{\beta}}$ from the DNN with parameters Θ. With the output $\hat{\boldsymbol{\beta}}$, user association schemes are generated according to the exploration policies. Then, we find the best user association scheme that minimizes the normalized energy consumption. The pair of inputs $\boldsymbol{\alpha}, \boldsymbol{\lambda}$ and the best user association scheme, denoted as $\tilde{\boldsymbol{\beta}}$, are saved in the memory, and will be used to train the DNN. By the end of the epoch, N_{t} training samples, $(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \tilde{\boldsymbol{\beta}})$, are randomly

selected from the memory to train the DNN. After the training, Θ is updated for the next epoch.

4.6.1 Exploitation and Exploration of the DNN

With ReLU function, the outputs of the DNN are continuous variables, i.e., $\hat{\boldsymbol{\beta}}_k^{\xi} = (\hat{\beta}_{1,k}^{\xi}, ..., \hat{\beta}_{M,k}^{\xi})^{\mathrm{T}}$. We first discuss how to explore user association schemes based on the outputs, and validate the impacts of exploration policies on the normalized energy consumption with simulation.

Highest Value (Exploitation)

For the kth user, a direct way to map the continuous variables $\hat{\boldsymbol{\beta}}_k^{\xi}$ to a discrete user association scheme is to access to the AP with the highest output. We denote the index of the AP with the highest output as $m_k^* = \arg\max_{m \in \mathcal{M}} \hat{\beta}_{m,k}^{\xi}$. Then, $\beta_{m_k^*,k}^{\xi}(0) = 1$ and $\hat{\beta}_{m,k}^{\xi}(0) = 0, \forall m \neq m_k^*$. The user association scheme is denoted as $\boldsymbol{\beta}(0)$.

One Step Exploration

Based on $\boldsymbol{\beta}(0)$, we change the association scheme of one of the $K^{\mathrm{u}} + K^{\mathrm{b}}$ users, while that of the other users remains the same as $\boldsymbol{\beta}(0)$. Since only one user changes the scheme, this method is referred to as one step exploration. With this exploration policy, each user may access to $M-1$ APs, and hence there are $\mu_{\mathrm{OS}} = (K^{\mathrm{u}} + K^{\mathrm{b}})(M-1)$ possible user association schemes, which are denoted as $\boldsymbol{\beta}(1), ..., \boldsymbol{\beta}(\mu_{\mathrm{OS}})$. Different from $\boldsymbol{\beta}(0)$, the $[(k-1)(M-1)+m_k^*]$th of element of $\boldsymbol{\beta}[(k-1)(M-1)+m]$ is zero. Besides, if $m < m_k^*$, $\beta_{m,k}^{\xi}[(k-1)(M-1)+m] = 1$. If $m > m_k^*$, $\beta_{m+1,k}^{\xi}[(k-1)(M-1)+m] = 1$.

Random Exploration

With the random exploration policy, each user randomly selects one of M APs with probability $1/M$. The user association schemes generated with this method are denoted as $\boldsymbol{\beta}(\mu_{\mathrm{OS}}+1), ..., \boldsymbol{\beta}(\mu_{\mathrm{OS}}+\mu_{\mathrm{RE}})$, where μ_{RE} is the number of schemes generated with the method.

4.6.2 The DNN Training

From the $1+\mu_{\mathrm{OS}}+\mu_{\mathrm{RE}}$ user association schemes, we choose the one that minimizes the normalized energy consumption, $\tilde{\boldsymbol{\beta}} = \arg\min_{i=0,1,...,\mu_{\mathrm{OS}}+\mu_{\mathrm{RE}}} Q_2^*(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\beta}(i)|\pi_2)$, and save it in the memory. The memory is empty at the beginning of the first epoch, and the initial values of parameters in Θ follow a zero-mean normal distribution. When the memory is full, the newly obtained training set, $(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \hat{\boldsymbol{\beta}})$, replaces the oldest one.

We adopt the experience replay technique in [126] to train the DNN using N_{t} training samples. The parameters in Θ are updated by using the Adam algorithm [127] to reduce a training loss function, defined as $L(\Theta) = -\frac{1}{N_{\mathrm{t}}}\sum_{n_{\mathrm{t}}=1}^{N_{\mathrm{t}}}[(\tilde{\boldsymbol{\beta}}_{n_{\mathrm{t}}})^{\mathrm{T}}\log(\hat{\boldsymbol{\beta}}_{n_{\mathrm{t}}}) + (1-\tilde{\boldsymbol{\beta}}_{n_{\mathrm{t}}})^{\mathrm{T}}\log(1-\hat{\boldsymbol{\beta}}_{n_{\mathrm{t}}})]$. When the value of $L(\Theta)$ is below a required threshold, σ_L, the training phase is finished. After the training phase, MME can use the DNN to calculate user association scheme for any $\boldsymbol{\alpha}, \boldsymbol{\lambda}$.

4.7 Simulation Results

4.7.1 Simulation Setup

The real network topology that will be used in our simulation is illustrated in Fig. 4.4. We vary the user distribution ratio, defined as the user density in region 1 to the user density in region 2, to see the impacts of user distribution on normalized energy consumption. The path loss model is $35.3 + 37.6\log_{10}(d)$, where d is the distance (meters) between an AP and a user [128]. The shadowing is lognormal distributed

with 8 dB standard deviation. The small-scale channel fading follows Rayleigh fading. The packet arrival rate of delay-tolerant users, λ_k^{b}, is uniformly distributed between 5 and 10 packets/s. The packet arrival rate of URLLC users, λ_k^{u}, is 500 packets/s [129]. Simulation parameters are summarized in Table 4.3, unless mentioned otherwise.

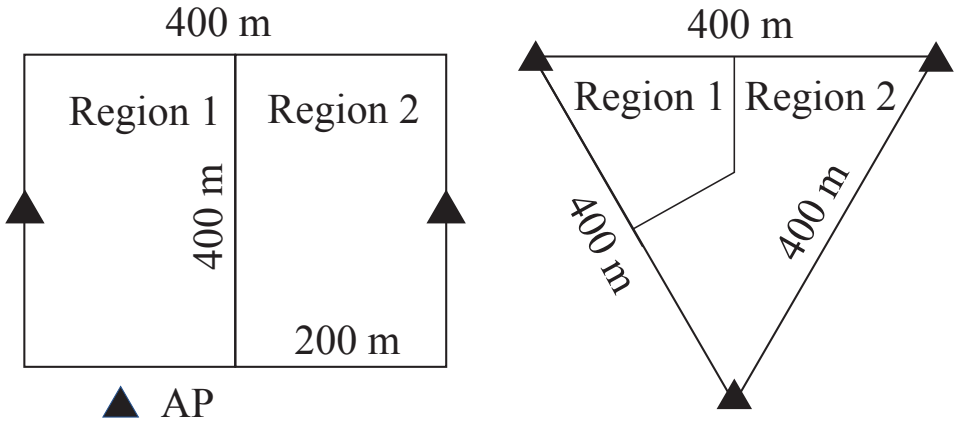

Figure 4.4 – Network topologies in our simulation

When using a single-layer neural network to approximate a certain policy, the required number of neurons increases exponentially with the required accuracy [130]. By increasing the number of layers, a smaller number of neurons is required to achieve the target accuracy. During simulation, we tried different numbers of hidden layers and neurons in each hidden layer, and found that by using 4 hidden layers with 100 neurons in each of them, a good performance could be achieved by the DNN. To achieve a better performance of the DL algorithm, we do not use $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ as the input of the DNN. Instead, the vector $[10\log(\frac{e^{\lambda_1^\xi}-1}{\alpha_{1,1}^\xi}+1),...,10\log(\frac{e^{\lambda_1^\xi}-1}{\alpha_{M,1}^\xi}+1),...,10\log(\frac{e^{\lambda_K^\xi}-1}{\alpha_{1,K}^\xi}+1),...,10\log(\frac{e^{\lambda_K^\xi}-1}{\alpha_{M,K}^\xi}+1)]^T$ with the size of $M(K^{\mathrm{u}}+K^{\mathrm{b}})\times 1$ is used as the input. The element $10\log(\frac{e^{\lambda_k^\xi}-1}{\alpha_{m,k}^\xi}+1)$ reflects the impacts of $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ on each user's transmit power (dB), which is dominant in the objective function of the normalized energy consumption. The numbers of neurons in the input and output layers are equal to $M(K^{\mathrm{u}}+K^{\mathrm{b}})$ and the dimension of $\boldsymbol{\beta}$, respectively. We set the learning rates of the DNN as 0.001. The number of training samples in each epoch is $N_{\mathrm{t}}=128$ and the memory can save up to 1024 training samples [83]. The DL algorithm is implemented

in Python with TensorFlow 1.11.

4.7.2 Optimal Bandwidth Allocation and Offloading Probabilities

In this subsection, we show the normalized energy consumption achieved by the optimal bandwidth allocation and offloading probability with a given user association scheme. In this case, we only need to consider single-AP scenarios. The users are randomly distributed around the AP. Since there is no existing method that optimizes bandwidth allocation and offloading probability for both URLLC and delay-tolerant services, we compare the proposed method (with legend 'Proposed') with two baselines. In the first baseline, the bandwidth allocation is the same as the optimal solution, but all the packets are offloaded to the MEC (with legend 'MEC'). In the second baseline, all the packets are processed at the local servers (with legend 'Local'). The normalized energy consumption depends on the location of users and shadowing. In this subsection, we generate 200 inputs, $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$, and calculate the average normalized energy consumption.

The normalized energy consumption is shown in Fig. 4.5, where the number of URLLC users equals the number of delay-tolerant users $K^{\mathrm{b}} = K^{\mathrm{u}}$. The total number of users increases from 10 to 26. The results show that the normalized energy consumption with the 'MEC' scheme increases rapidly as the total number of users

Table 4.3 – Parameters in Simulation

Notation	Description	Value
S_m	Computation capability of the mth MEC server	1.6 GHz
$P_k^{\max,\xi}$	Maximal transmit power of each user	23 dBm
T_{s}	Duration of one time slot	0.125ms
W	Bandwidth of each subcarrier	120 KHz
$N^{\max}$	Maximal number of subcarriers of each AP	128
N_0	Single-sided noise spectral density	-174 dBm/Hz
$C_k^{\max,\mathrm{b}}$	Computation capability of a local server	5000 cycles/slot
b_k^{b}	Number of bytes in a long packet	$[50, 100]$ KB
b_k^{u}	Number of bytes in a short packet	32 bytes
k_1	Number of CPU cycles required to process one byte of information [76]	330 cycles/byte
$D_k^{\max,\mathrm{u}}$	Delay requirement of URLLC services	1 ms
$\epsilon_k^{\max,\mathrm{u}}$	Maximal tolerable packet loss probability of URLLC services	10^{-7}

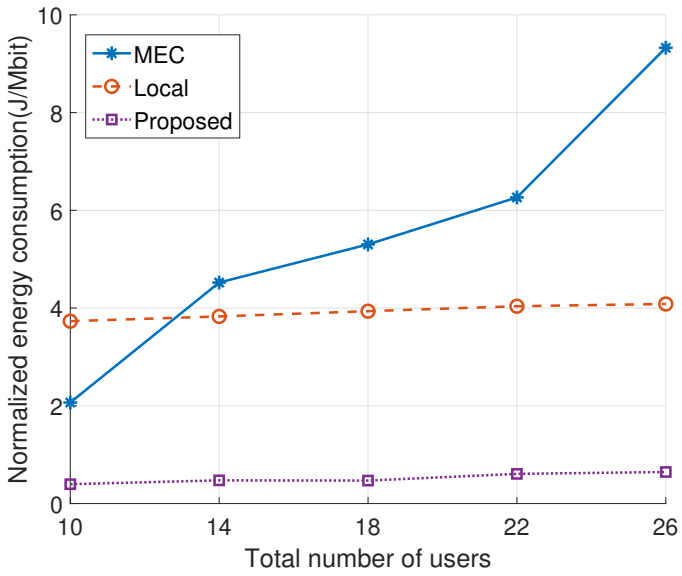

Figure 4.5 – Normalized energy consumption v.s. total number of users

increases. The normalized energy consumption of 'Local' scheme, however, remains the same as expected. Our proposed scheme can save around 89% of normalized energy consumption compared with the 'MEC' scheme and 87% compared with the 'Local' scheme.

4.7.3 DL Algorithm for User Association

In this subsection, we show the normalized energy consumptions of different user association schemes, where $N^{\max} = 48$, $S = 0.4$ GHz, and $K^{\mathrm{b}} = K^{\mathrm{u}} = 5$. We compared our DL algorithm (with legend 'DL') with the optimal user association scheme (with legend 'Optimal') obtained by exhaustively searching for all possible user association schemes. To show the impacts of non-stationary environment on the performance of the DL algorithm, we also provide the performance of a well-trained DNN that will not be updated when the user density varies (with legend 'DL Fixed DNN'). With this scheme, there is no exploration and the output of the DNN will be used as the user association scheme. Some similar studies focused on offloading and resource allocation with a single AP [60, 83]. The implicit assumption on the

user association is that the users are served by the nearest AP or the one with the highest large-scale channel gain. In addition, a game theory approach was proposed to optimize resource management and user association in [112]. Thus, we compared the proposed method with three baselines: With the first baseline method, users are served by the nearest AP (with legend 'Nearest AP'). With the second method, users are connected to the AP with the highest large-scale channel gain (with legend 'Highest α'). With the third baseline, the game theory approach based on a coalition game in [112] is used to iteratively optimize user association (with legend 'Game'). Following [112], we set the number of the coalitions as M (the number of APs) and the users are randomly chosen to perform Merge, Split and Exchange operations by preferring a smaller objective function as in (4.21).

We provide the simulation results in scenarios with different numbers of APs: $M = 2$ and $M = 3$. When $M = 2$, the optimal scheme can be obtained with the exhaustive searching method. However, when $M = 3$, the complexity of the exhaustive searching method is too high, and we cannot obtain the optimal scheme.

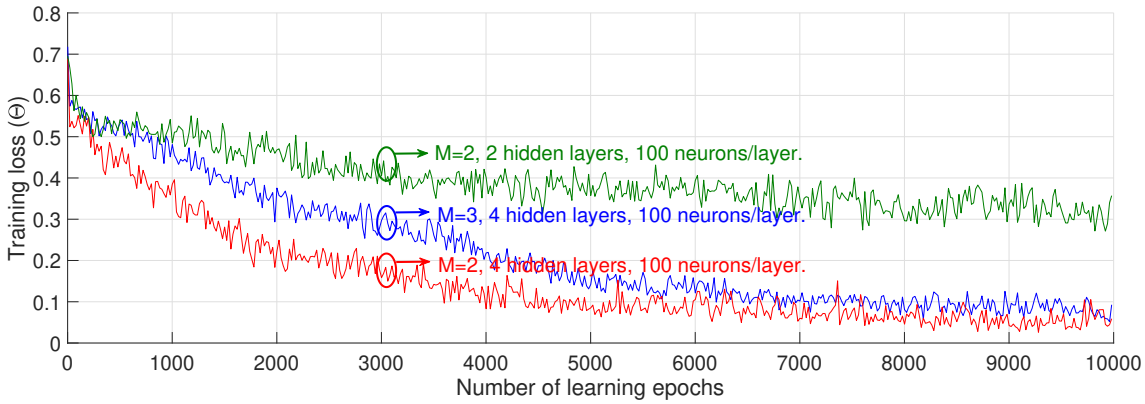

Figure 4.6 – Training loss function v.s. number of learning epoch

To show the convergence of the deep learning algorithm, we provide the values of the training loss function, $L(\Theta)$, as the number of learning epochs increases in Fig. 4.6, where the user distribution ratio is set to be $6 : 4$. The results with two hidden layers indicate that the DNN does not converge if the number of layers is too small. To find the proper structure of the DNN, we start from the case with one hidden layer and

increase the number of layers until the DNN can converge, i.e., four hidden layers. When $M = 2$, $L(\Theta)$ is around 0.1 after 4000 epochs. When $M = 3$, $L(\Theta)$ decreases slower than the scenario $M = 2$ since the algorithm needs to explore a larger feasible region when $M = 3$. For both scenarios, $L(\Theta)$ decreases gradually and approaches zero.

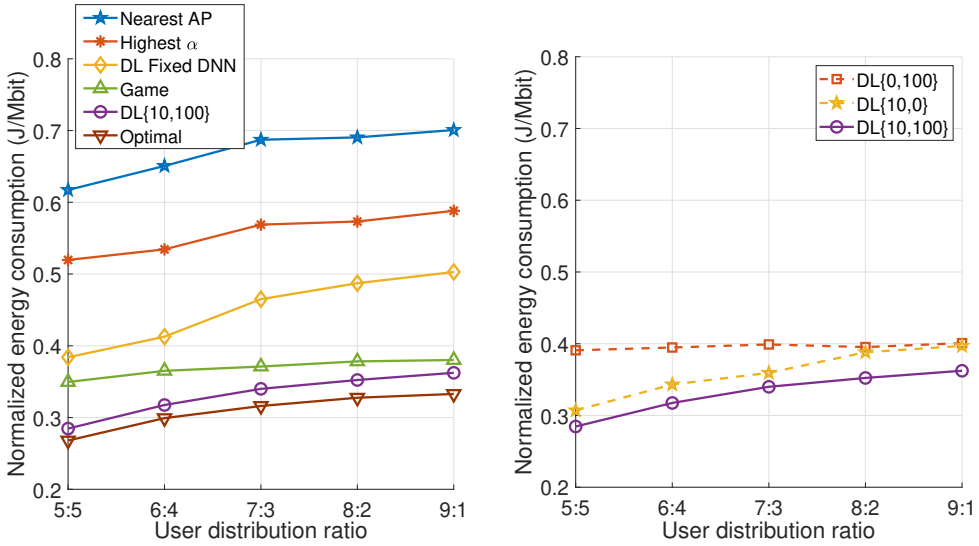

Figure 4.7 – Normalized energy consumption v.s. user distribution ratio, where $M = 2$ and $K^{\mathrm{b}} = K^{\mathrm{u}} = 5$

The averages of the normalized energy consumption in the last 1000 epochs are shown in Fig. 4.7, where the numbers of user association schemes generated by the two exploration policies are shown in the legends; e.g., 'DL$\{10, 100\}$' means $\mu_{\mathrm{OS}} = 10, \mu_{\mathrm{RE}} = 100$. The results in Fig. 4.7 show that our proposed method can achieve much smaller normalized energy consumption than the three baseline methods, and perform close to the optimal scheme. For the 'Game' scheme, it converges after 100 iterations on average. As indicated in [112], this scheme needs to evaluate the objective function twice in each iteration, which means around 200 times. However, our proposed algorithm 'DL$\{10, 100\}$' only needs to explore $(10 + 100)$ user association schemes, i.e., evaluating the objective function 110 times, which is less than the 'Game' scheme. Therefore, our proposed algorithm can achieve a lower normalized energy consumption with less computation complexity. We can also observe

that directly exploiting the output of DNN without any exploration can save around 30 % normalized energy consumption compared with 'Highest α'. With a few more explorations, the performance can be further improved as shown by 'DL$\{10, 100\}$'. Moreover, the one step exploration policy 'DL$\{10, 0\}$' can achieve lower normalized energy consumption with less explorations compared with the random exploration policy 'DL$\{0, 100\}$'. These results indicate that the output of the DNN can help improve the efficiency of the exploration policy.

In Fig. 4.8, we study the impacts of the variation of the user density on the proposed DL algorithm. In the digital twin, the user distribution ratio is set to be $5 : 5$. After 1000 tests with $5 : 5$ user distribution, the user distribution ratio in the real network becomes different, i.e., $9 : 1$. The MME needs to update DNN according to the variation of the user distribution. The legends of DL algorithms with the user distribution variation are followed by $(5 : 5 \rightarrow 9 : 1)$. For the other curves, the user distribution ratio is constant. The normalized energy consumptions are the average ones over 500 tests. In each test, the large-scale channel gains and the average task arrival rates of the users are generated randomly. From the results in Fig. 4.8, we can observe that our proposed DL algorithm 'DL$\{10, 100\}(5 : 5 \rightarrow 9 : 1)$' can adjust the DNN once the network user distribution ratio changes (after 2000 tests) and obtain a satisfactory performance compared to the 'Optimal$(5 : 5)$' and 'Optimal$(9 : 1)$'. To further evaluate the importance of the digital twin, we include the legend 'DL Fixed DNN' representing a well-trained DNN being used to make user association decisions without updating its parameters Θ. The results show that when the density of users varies, a fixed DNN can be worse than the baseline method 'Highest α'. These results indicate that updating DNN according to the non-stationary environment is necessary.

The normalized energy consumptions achieved with different schemes in the scenario with 3 APs are provided in Table 4.4. We compare the average of the normalized energy consumptions in the last 1000 epochs with the 'Nearest AP', the 'Highest α' and the 'DL' schemes. The results in Table. 4.4 show that the 'DL' scheme can save around 72 % and 59 % normalized energy consumption compared with the 'Nearest AP' and the 'Highest α' schemes, respectively. This observation indicates that our

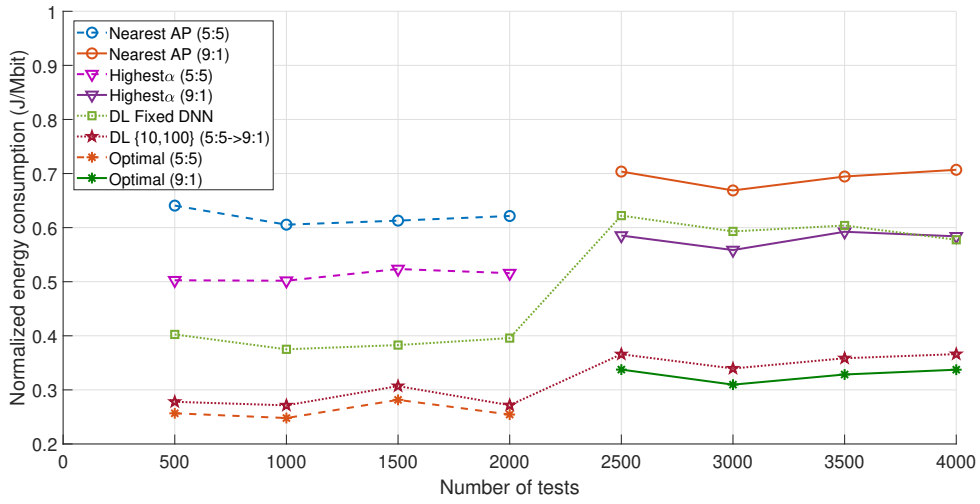

Figure 4.8 – Normalized energy consumption with uncertain user distribution ratios, where $M = 2$ and $K^{\mathrm{b}} = K^{\mathrm{u}} = 5$

proposed framework can find an efficient user association scheme when $M = 3$.

Table 4.4 – Performance Comparison When $M = 3$ and $K^{\mathrm{b}} = K^{\mathrm{u}} = 5$

Schemes	Nearest AP	Highest α	DL
Average normalized energy efficiency (J/Mbit)	0.50	0.34	0.14
Normalized energy consumption compared with 'Nearest AP'	100%	68%	28%

4.8 Chapter Summary

In this chapter, we studied how to reduce the normalized energy consumption of users with URLLC and delay-tolerant services in a MEC system. We proposed a DL architecture for user association, where a digital twin of network environment was established at the central server for training the algorithm off-line. After the training phase, the DNN was sent to the MME that manages user association. With a given user association scheme, we proposed a low-complexity optimization algorithm that optimized resource allocation and offloading probabilities at each AP. Simulation results indicated that by optimizing resource allocation and offloading probability, our low-complexity algorithm can save more than 87 % energy compared with the baselines. Besides, with the DL algorithm, our user association scheme can achieve

lower normalized energy consumption with less computing complexity compared with an existing method and approach the global optimal solution.

Chapter 5

Deep transfer learning for resource allocation for new applications with diverse QoS requirements

5.1 Introduction

5.1.1 Background

The 5G cellular networks are expected to support various emerging applications with diverse QoS requirements, such as enhanced mobile broadband services, massive machine-type communications, and URLLC [4]. To guarantee the QoS requirements of different types of services, existing optimization algorithms for radio resource allocation are designed to maximize spectrum efficiency or energy efficiency by optimizing scarce radio resources, such as time-frequency resource blocks and transmit power, subject to QoS constraints [131, 132, 133, 134, 135, 136, 57].

There are two major challenges for implementing existing optimization algorithms in practical 5G networks. First, QoS constraints of some services, such as delay-sensitive and URLLC services, may not have closed-form expressions. To execute an

optimization algorithm, the system needs to evaluate the complicated non-closed form QoS constraints, and thus suffers from a long processing delay [57, 58]. Second, even if the closed-form expressions of QoS constraints can be obtained in some scenarios, the optimization problems are non-convex in general [58, 137, 136]. The system also needs to update resource allocation by solving non-convex problems to accommodate the time-varying channel and traffic conditions, leading to very high computing overhead. Even for some convex optimization problems that can be solved by well-developed methods, like the interior-point method, the computing complexity is still too high for real-time implementation [138].

Deep learning is a promising approach to find the optimal resource allocation in real-time [139, 140, 141, 142, 143]. The basic idea is to use an artificial NN to approximate the optimal resource allocation policy that maps the system states to the optimal resource allocation. The system first trains the NN off-line with a large number of labeled samples. After the training phase, the optimal resource allocation can be obtained from the output of the NN for any given input. According to the Universal Approximation Theory, if the optimal policy is a deterministic and continuous function, then the approximation can be arbitrarily accurate [62].

It is worth noting that the application of deep learning in wireless networks is not straightforward. For some discrete optimization variables, such as the number of subcarriers, antennas and the user association decisions, the approximation of the NN can be inaccurate due to the quantization of these discrete variables. As a result, the solution obtained from the NN cannot fully guarantee the QoS requirements of different types of services. In addition, deep learning requires a large number of labeled training samples. To obtain labeled training samples, we should first design an optimization algorithm to solve the formulated optimization problem. Even if a large number of labeled training samples are obtained with the optimization algorithm, the pre-trained NN is not accurate when the wireless network is non-stationary. For example, the distribution of wireless channel and the types of services in the network may vary. These non-stationary parameters that are not included in the input of the NN are referred to as hidden variables [63]. During the training phase, we assume that

the hidden variables are fixed. However, in practical systems, these hidden variables drift over time. As discussed in [63], the dynamic hidden variables can be pernicious in deep learning.

5.1.2 Related Works

Improving resource utilization efficiency for different kinds of services has been extensively studied in the literature. For delay-tolerant services, the QoS requirement is formulated as an average data rate requirement in OFDMA systems [144], where the subcarrier and transmit power allocation and antenna configuration were optimized. To guarantee the queueing delay bound and the delay bound violation probability of real-time services, effective capacity was adopted in [145, 146] to optimize bandwidth allocation and power control schemes. In URLLC, to reduce transmission delay, the block-length of channel codes is short, and the fundamental relation between decoding error probability and block-length was derived in [120]. This relation was used to optimize resource allocation for short packet transmissions in URLLC [135, 147, 136]. For most of these problems, the QoS constraints do not have closed-form expressions and the optimization algorithms cannot be executed in real time.

Approximating optimal resource allocation policies with NNs has been studied in [139, 148, 140]. The authors of [139] proved that an iterative algorithm for power control in wireless networks can be accurately approximated by a fully-connected neural network (FNN). In [148] and [143], convolutional neural networks were used to approximate the power control policy and the content delivery policy, respectively. To improve energy efficiency, [140] proposed an online deep learning approach to approximate the energy-efficient power control scheme obtained from the monotonic fractional programming framework in [149]. When the optimal optimization algorithm is not available, unsupervised deep learning can be applied [150, 151]. In [150, 151], the parameters of a NN are trained to satisfy the Karush-Kuhn-Tucker (KKT) conditions of the optimization problem. However, for problems with integer variables that are not defined over a compact set, the KKT conditions do not exist.

Considering that wireless networks are highly dynamic, NNs trained offline may not achieve good performance in non-stationary networks. To handle this issue, deep transfer learning was used in some existing works. For example, when data arrival processes [152], traffic patterns [153], or the size of the network [154, 155] changes, deep transfer learning can be used to fine-tune the pre-trained NNs.

5.1.3 Chapter Outline

Motivated by the above issues, we will answer the following questions in this chapter: 1) How to design an optimization algorithm that can find the optimal resource allocation subject to diverse QoS requirements; 2) How to improve the approximation accuracy of the NN when there are quantization errors of discrete optimization variables; and 3) How to adapt the pre-trained NN according to non-stationary wireless networks.

The rest of the chapter is organized as follows. In Section 5.2, we formulate the system models. The cascaded NNs for ensuring the QoS requirement are presented in Section 5.3. In Section 5.4, we apply deep transfer learning in non-stationary wireless networks. We provide simulation results in Section 5.5 and conclude the work in Section 5.6. All the notations used in this chapter are listed in Table 5.1.

5.2 System Model and Problem Formulation

5.2.1 System Model

We consider an OFDMA system, where one multi-antenna BS serves K single-antenna users that request different kinds of services, including delay-tolerant, delay-sensitive and URLLC services. The corresponding sets of users are denoted by $\mathcal{K}^{\mathrm{t}}$, $\mathcal{K}^{\mathrm{s}}$, and $\mathcal{K}^{\mathrm{u}}$, respectively. For notational simplicity, we use a superscript $\xi \in \{\mathrm{t}, \mathrm{s}, \mathrm{u}\}$ to represent delay-tolerant, delay-sensitive and URLLC services. The bandwidth of each

Table 5.1 – Notations

Notation	Definition	Notation	Definition
x	scalar	$\mathbf{x}$	vector
$\mathbf{X}$	matrix	$\mathbb{E}$	expectation
$(\cdot)^{\mathrm{T}}$	transpose operator	K	total number of users
$\xi = \{\mathrm{t}, \mathrm{s}, \mathrm{u}\}$	superscript representing delay-tolerant, delay-sensitive and URLLC services	$\mathcal{K}^{\xi}$	set of users
W	bandwidth of each subcarrier	T_s	duration of each slot
T_c	channel coherence time	$D_k^{\mathrm{q,s}}$	delay bound
N_{T}	number of antennas at the BS	$\epsilon_k^{\mathrm{q,s}}$	maximal tolerable delay bound violation probability
$\epsilon^{\max,\mathrm{u}}$	threshold of decoding error probability	α_k^{ξ}	large-scale channel gain
$g_{k,n}^{\xi}$	small-scale channel gain on the n-th subchannel	P_k^{ξ}	transmit power
N_0	single-side noise spectral density	$\bar{a}_k$	average data arrival rate
N_k^{ξ}	number of allocated subcarriers	ν^{a}	inter-arrival time between packets
ν^{s}	size of each packet	θ_k^{s}	QoS exponent
$\epsilon_k^{\mathrm{d,u}}$	decoding error probability	V_k^{u}	channel dispersion
b_k^{u}	number of bits in one packet	$\bar{\epsilon}_k^{\mathrm{d,u}}$	average decoding error probability
ρ	power amplifier efficiency	P^{ca}	power consumption by each antennas
P_0^{c}	fixed circuit power consumption	c_k^{ξ}	feature of the packet arrival process

subcarrier and the duration of one transmission time interval (TTI) in the OFDMA system are denoted by W and T_{s}, respectively.

Channel Model

We assume that channels are block fading in both time and frequency domains. Channel gains on different blocks are independent and identically distributed (i.i.d.). Channel coherence time is denoted by T_{c}, which is much longer than the duration of a TTI T_{s}. We consider downlink transmissions and assume that CSI is only available at users to avoid the high overhead for channel estimations at the BS.

Queueing Model

For all kinds of services, packets in the buffer of the BS are served according to a first-come-first-served basis. For delay-tolerant services, we only need to ensure the stability of the queueing system. For delay-sensitive services, a delay bound, $D_k^{\mathrm{q,s}}$, and

a maximal tolerable delay bound violation probability, $\epsilon_k^{\mathrm{q,s}}$, should be satisfied. To avoid queueing delay for URLLC, packets should be served immediately after arriving at the BS. The decoding error probability of packets should not exceed a required threshold, $\epsilon^{\mathrm{max,u}}$.

5.2.2 Delay-Tolerant Services

For delay-tolerant services, the block-length of channel code can be sufficiently long, and the average data rate of each user approaches Shannon's capacity, i.e.,

$$\bar{R}_k^{\mathrm{t}} = N_k^{\mathrm{t}} \mathbb{E}_{g_{k,n}^{\mathrm{t}}} \left[W \ln \left(1 + \frac{\alpha_k^{\mathrm{t}} g_{k,n}^{\mathrm{t}} P_k^{\mathrm{t}}}{N_0 N_{\mathrm{T}} N_k^{\mathrm{t}} W} \right) \right] \text{ (bits/s)}, \tag{5.1}$$

where α_k^{t} is the large-scale channel gain, $g_{k,n}^{\mathrm{t}}$ is the small-scale channel gain on the n-th subchannel, P_k^{t} is the transmit power, N_0 is the single-side noise spectral density, N_{T} is the number of antennas at the BS, and N_k^{t} is the number of subcarriers allocated to the k-th delay-tolerant user. Since CSI is not available at the BS, the transmit power is equally allocated on different antennas and subcarriers.

To ensure the stability of the queueing system, the average service rate should be equal to or higher than the average data arrival rate of the user, i.e.,

$$\bar{R}_k^{\mathrm{t}} \geq \bar{a}_k, \tag{5.2}$$

where $\bar{a}_k$ is the average data arrival rate of the k-th delay-tolerant user.

5.2.3 Delay-Sensitive Services

For delay-sensitive services, the block-length of channel codes is finite. We denote Φ as the SNR gap between the channel capacity and a practical modulation and coding scheme as in [156]. The achievable rate of the k-th delay-sensitive user can be

expressed as

$$R_k^{\mathrm{s}} = \sum_{n=1}^{N_k^{\mathrm{s}}} W \ln\left(1 + \frac{\alpha_k^{\mathrm{s}} g_{k,n}^{\mathrm{s}} P_k^{\mathrm{s}}}{\Phi N_0 N_{\mathrm{T}} N_k^{\mathrm{s}} W}\right), \text{ (bits/s)}, \tag{5.3}$$

where the definitions of α_k^{s}, $g_{k,n}^{\mathrm{s}}$, P_k^{s}, and N_k^{s} are similar to that in (5.1).

To guarantee $D_k^{\mathrm{q,s}}$ and $\epsilon_k^{\mathrm{q,s}}$ for delay-sensitive services, effective bandwidth and effective capacity are widely used [157, 158]. We assume that the packet arrival process of each delay-sensitive user is a compound Poisson process[1]. The inter-arrival time between packets and the size of each packet follow exponential distributions with parameters ν^{a} and ν^{s}, respectively. Then, the effective bandwidth of the k-th delay-sensitive user can be written as follows [159]:

$$E_k^{\mathrm{B,s}} = \frac{\nu^{\mathrm{a}}}{\nu^{\mathrm{s}} - \theta_k^{\mathrm{s}}}, \text{ (bits/s)}, \tag{5.4}$$

where θ_k^{s} is the QoS exponent determining the service rate of a queue, which can be obtained from

$$\exp\left[-\theta_k^{\mathrm{s}} E_k^{\mathrm{B,s}}(\theta_k^{\mathrm{s}}) D_k^{\mathrm{q,s}}\right] \approx \epsilon_k^{\mathrm{q,s}}. \tag{5.5}$$

Substituting (5.4) into (5.5), we can derive that

$$\theta_k^{\mathrm{s}} = \frac{\nu^{\mathrm{s}} \ln(\epsilon_k^{\mathrm{q,s}})}{\ln \epsilon_k^{\mathrm{q,s}} - \nu^{\mathrm{a}} D_k^{\mathrm{q,s}}}. \tag{5.6}$$

Since $g_{k,n}^{\mathrm{s}}$ are i.i.d., the effective capacity can be simplified as follows [158]:

$$E_k^{\mathrm{C,s}} = -\frac{1}{\theta_k^{\mathrm{s}} T_{\mathrm{c}}} \ln \mathbb{E}_{g_{k,n}^{\mathrm{s}}}\left[\exp\left(-\theta_k^{\mathrm{s}} T_{\mathrm{c}} R_k^{\mathrm{s}}\right)\right] \tag{5.7}$$

$$= -\frac{N_k^{\mathrm{s}}}{\theta_k^{\mathrm{s}} T_{\mathrm{c}}} \ln\left[\mathbb{E}_{g_{k,n}^{\mathrm{s}}}\left(1 + \frac{\alpha_k^{\mathrm{s}} g_{k,n}^{\mathrm{s}} P_k^{\mathrm{s}}}{\Phi N_0 N_{\mathrm{T}} N_k^{\mathrm{s}} W}\right)^{-\varpi_k}\right], \text{ (bits/s)}, \tag{5.8}$$

where $\varpi_k = \frac{\theta_k^{\mathrm{s}} T_{\mathrm{c}} W}{\ln 2}$, and (5.8) is obtained by substituting R_k^{s} in (5.3) into (5.7). To

[1]For some other kinds of packet arrival processes, the method to compute effective bandwidth can be found in [159, 160].

guarantee $D_k^{\mathrm{q,s}}$ and $\epsilon_k^{\mathrm{q,s}}$, the following constraint should be satisfied:

$$E_k^{\mathrm{C,s}} \geq E_k^{\mathrm{B,s}}. \tag{5.9}$$

5.2.4 URLLC Services

When transmitting short packets of URLLC, the block-length of channel codes is much shorter than the previous services. According to the Normal Approximation of the achievable rate in [120] and the analysis in Appendix E of [104], the achievable rate over the frequency-selective channel can be approximated by

$$R_k^{\mathrm{u}} \approx \frac{W}{\ln 2}\left\{\left[\sum_{n=1}^{N_k^{\mathrm{u}}} \ln\left(1+\frac{\alpha_k^{\mathrm{u}} g_{k,n}^{\mathrm{u}} P_k^{\mathrm{u}}}{N_0 N_{\mathrm{T}} N_k^{\mathrm{u}} W}\right)\right] - \sqrt{\frac{V_k^{\mathrm{u}}}{T_{\mathrm{s}} W}} f_Q^{-1}\left(\epsilon_k^{\mathrm{d,u}}\right)\right\}, \text{ (bits/s)}, \tag{5.10}$$

where f_Q^{-1} is the inverse of Q-function, V_k^{u} is the channel dispersion, which is given by $V_k^{\mathrm{u}} = N_k^{\mathrm{u}} - \sum_{n=1}^{N_k^{\mathrm{u}}} \frac{1}{\left(1+\frac{\alpha_k^{\mathrm{u}} g_{k,n}^{\mathrm{u}} P_k^{\mathrm{u}}}{N_0 N_{\mathrm{T}} N_k^{\mathrm{u}} W}\right)^2}$ [104], $\epsilon_k^{\mathrm{d,u}}$ is the decoding error probability, and the definitions of α_k^{u}, $g_{k,n}^{\mathrm{u}}$, P_k^{u} and N_k^{u} are similar to that in (5.1).

In each TTI, a URLLC user either generates a packet or stays silent [161]. Thus, the packet arrival process can be modeled as a Bernoulli process. To avoid queueing delay, the transmission duration of a packet should be one TTI. We denote the number of bits in one packet as b_k^{u}. From $T_{\mathrm{s}} R_k^{\mathrm{u}} = b_k^{\mathrm{u}}$, we can derive the average decoding error probability, i.e.,

$$\bar{\epsilon}_k^{\mathrm{d,u}} \approx \mathbb{E}_{g_{k,n}^{\mathrm{u}}}\left\{f_Q\left(\sqrt{\frac{T_{\mathrm{s}} W}{N_k^{\mathrm{u}}}}\left\{\left[\sum_{n=1}^{N_k^{\mathrm{u}}} \ln\left(1+\frac{\alpha_k^{\mathrm{u}} g_{k,n}^{\mathrm{u}} P_k^{\mathrm{u}}}{N_0 N_{\mathrm{T}} N_k^{\mathrm{u}} W}\right)\right] - \frac{b_k^{\mathrm{u}} \ln 2}{T_{\mathrm{s}} W}\right\}\right)\right\}, \tag{5.11}$$

where $V_k^{\mathrm{u}} \approx N_k^{\mathrm{u}}$ is applied, which is accurate when the signal-to-noise ratio is higher than 10 dB.

To guarantee the reliability requirement of URLLC, the following constraint should be satisfied:

$$\bar{\epsilon}_k^{\mathrm{d,u}} \leq \epsilon^{\max,\mathrm{u}}. \tag{5.12}$$

5.2.5 Problem Formulation

The total power consumption of a BS consists of the transmit power and the circuit power, given by [162]:

$$P_{\text{tot}} = \frac{1}{\rho} \sum_{k \in \mathcal{K}^{\xi}} P_k^{\xi} + P^{\text{ca}} N_{\text{T}} \sum_{k \in \mathcal{K}^{\xi}} N_k^{\xi} + P_0^{\text{c}}, \tag{5.13}$$

where $\rho \in (0, 1]$ is the power amplifier efficiency, P^{ca} is the power consumption by each antenna for signal processing on each subcarrier, and P_0^{c} is the fixed circuit power consumption.

To save the power consumption of the BS, we minimize P_{tot} subject to QoS constraints, i.e.,

$$\min_{P_k^{\xi}, N_k^{\xi}} \; P_{\text{tot}}, \tag{5.14}$$

$$\text{s.t.} \sum_{k \in \mathcal{K}^{\xi}} N_k^{\xi} \leq N^{\max}, \tag{5.14a}$$

$$\sum_{k \in \mathcal{K}^{\xi}} P_k^{\xi} \leq P^{\max}, \tag{5.14b}$$

$$(5.2), (5.9), \text{ and } (5.12).$$

where (5.14a) and (5.14b) are the constraints on the total number of subcarriers and the maximum transmit power of the BS. Problem (5.14) is an MINLP problem, which is non-convex. Finding the global optimal solution is very challenging.

5.3 Supervised Deep Learning — Cascaded Neural Networks for QoS Guarantee

In this section, we develop cascaded NNs to approximate the optimal resource allocation policy that maps the system states to the optimal solution. Training the cascaded NNs requires a large number of labeled training samples. We first develop

an optimization algorithm to find the global optimal solutions of the problem. Then, we illustrate how to train the cascaded NNs.

5.3.1 Preliminary of Deep Learning

A feed-forward NN consists of multiple layers of neurons. Each neuron includes a non-linear activation function and some parameters to be optimized in the training phase [163]. We denote the input and output vectors of the l-th layer as $\boldsymbol{x}^{[l]}$ and $\boldsymbol{y}^{[l]}$ respectively. Then, from the activation function and parameters in the l-th layer, the output vector can be expressed as follows:

$$\boldsymbol{y}^{[l]} = \delta_{\mathrm{a}}(\boldsymbol{W}^{[l]}\boldsymbol{x}^{[l]} + \boldsymbol{b}^{[l]}), \tag{5.15}$$

where $\Lambda = \{\boldsymbol{W}^{[l]}, \boldsymbol{b}^{[l]}, l = 1, ..., L\}$ are the parameters and L is the number of layers. $\delta_{\mathrm{a}}(\cdot)$ is the activation function. We will use $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ as the activation function in the rest of this chapter unless otherwise specified.

In problem (5.14), the resource allocation policy depends on the large-scale channel gains, $\boldsymbol{\alpha} = [\alpha_1^{\xi}, ..., \alpha_K^{\xi}]^T$, and the packet arrival processes of different kinds of services, where $(\cdot)^T$ denotes the transpose operator. More specifically, for delay-tolerant services, the average service rate requirements are determined by the average arrival rates, $\bar{\boldsymbol{a}} = [\bar{a}_1, ..., \bar{a}_K]^T$. For delay-sensitive services, the effective capacities of the service processes should be equal to or higher than the effective bandwidth of the arrival processes $\boldsymbol{E}^{\mathrm{B,s}} = [E_1^{B,\mathrm{s}}, ..., E_K^{B,\mathrm{s}}]^T$. For URLLC services, the numbers of bits to be transmitted in each TTI depend on the packet sizes of different users $\boldsymbol{b}^{\mathrm{u}} = [b_1^{\mathrm{u}}, ..., b_K^{\mathrm{u}}]^T$. To unify the notations, we use $\boldsymbol{c} = [c_1^{\xi}, ..., c_K^{\xi}]^T$ to represent the features of the packet arrival processes of all kinds of services. The optimal policy of problem (5.14) maps the features of channels and packet arrival processes to the optimal resource allocation and is denoted by π^*:

$$\pi^* : \boldsymbol{X} \rightarrow \boldsymbol{Y}^*, \tag{5.16}$$

where $\boldsymbol{X} = [\boldsymbol{\alpha}^T, \boldsymbol{c}^T]^T$, $\boldsymbol{Y}^* = [\boldsymbol{P}^{*T}, \boldsymbol{N}^{*T}]^T$, $\boldsymbol{P}^* = [P_1^{\xi*}, ..., P_K^{\xi*}]^T$, and $\boldsymbol{N}^* = [N_1^{\xi*}, ..., N_K^{\xi*}]^T$.

As indicated in the universal approximation theorem of NNs [62], any deterministic continuous function defined over a compact set can be approximated arbitrarily well with an FNN. For our problem, the output of the FNN, denoted by $\tilde{\boldsymbol{Y}}$, includes transmit power and bandwidth allocation, $\tilde{\boldsymbol{P}}$ and $\tilde{\boldsymbol{N}}$, i.e., $\tilde{\boldsymbol{Y}} \triangleq [\tilde{\boldsymbol{P}}^T, \tilde{\boldsymbol{N}}^T]^T$. For our problem, there are two kinds of errors that will deteriorate the QoS. First, approximation errors are inevitable since $\tilde{\boldsymbol{Y}}$ will not be the same as $\boldsymbol{Y}^*$ with probability one. Second, the output of an FNN is continuous, but the numbers of subcarriers are integers. The quantization errors will be introduced when mapping the continuous output to integers.

5.3.2 Cascaded Neural Networks for QoS Guarantee

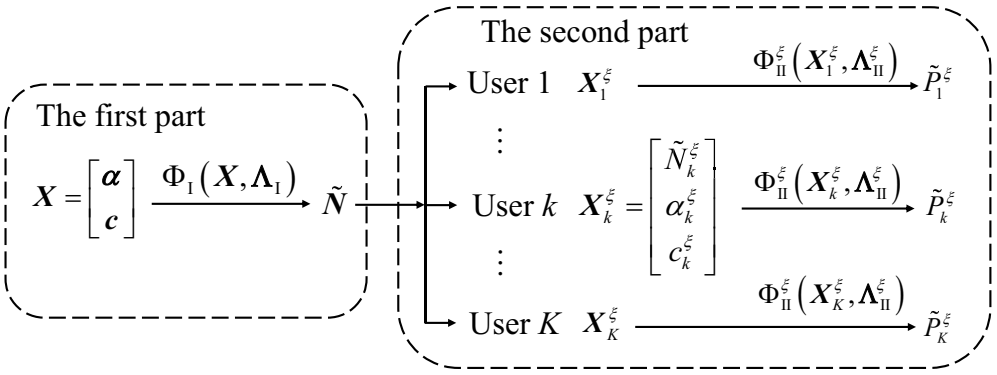

Figure 5.1 – Illustration of the cascaded NNs

To improve the accuracy of the approximation and to ensure the QoS requirements of an MINLP, we propose a cascaded structure consisting of two parts of NNs in Fig. 5.1. The first NN maps the system states to the discrete variables, i.e., $\tilde{\boldsymbol{N}} = \Phi_{\mathrm{I}}(\boldsymbol{X}, \Lambda_{\mathrm{I}})$, where Λ_{I} is the parameters of the NN. Like an FNN, the first NN will introduce quantization errors. To alleviate the effect of quantization errors on the QoS, we train another NN that maps the obtained discrete variables to the continuous variables that are required to guarantee the QoS constraint of each user. Specifically, in a system with K users, the second part of the cascaded structure consists of K NNs. Each of

them approximates the power allocation policy, $\tilde{P}_k^{\xi} = \Phi_{\mathrm{II}}^{\xi}(\boldsymbol{X}_k^{\xi}, \Lambda_{\mathrm{II}}^{\xi})$, where the input of the k-th NN is defined as $\boldsymbol{X}_k^{\xi} \triangleq [\tilde{N}_k^{\xi}, \alpha_k^{\xi}, c_k^{\xi}]^T$. To obtain labeled training samples, binary search is applied to find the minimum transmit power that can guarantee the QoS constraints in (5.2), (5.9) and (5.12). Since the parameters of the second NN, $\Lambda_{\mathrm{II}}^{\xi}$, depend on the type of services and do not rely on the values of α_k^{ξ} and c_k^{ξ}, the system only needs to train a NN for each type of services.

Denote the accuracy of the approximated power allocation policy as Δ_P, which is defined as a threshold that satisfies the following requirement:

$$\Pr\{|\Phi_{\mathrm{II}}^{\xi}(\boldsymbol{X}_k^{\xi}, \Lambda_{\mathrm{II}}^{\xi}) - P_k^{\xi}(\tilde{N}_k^{\xi})| \leq \Delta_P\} \geq P_{\mathrm{req}}, \tag{5.17}$$

where P_{req} is the required probability with QoS guarantee and $P_k^{\xi}(\tilde{N}_k^{\xi})$ is the minimum transmit power that is needed to satisfy the constraint in (5.2), (5.9) or (5.12). If the BS allocates $\tilde{P}_k^{\xi} + \Delta_P$ transmit power to the kth user, its QoS requirement can be satisfied with probability P_{req}.

The accuracy of the approximated power allocation policy is determined by Φ_{II}^{ξ}. The quantization errors and the approximation accuracy of Φ_{I} has little impact on Δ_P. If an FNN is used to approximate the subcarrier and transmit power allocation policy, the quantization errors of $\tilde{\boldsymbol{N}}$ and the accuracy of $\tilde{\boldsymbol{P}}$ are intertwined. Therefore, compared with the FNN, the cascaded NNs can guarantee the QoS of different services with a higher probability. We will validate their performance via simulation.

5.3.3 Labeled Training Samples

To obtain a large number of labeled training samples, the optimal solutions of an MINLP problem can be found through some well-known algorithms, such as branch-and-bound (BnB) [164]. However, BnB requires a very high computational complexity, possibly approaching the exhaustive search for some worst cases [165].

To obtain a large number of training samples, we develop an optimization algorithm that converges to the global optimal solution of problem (5.14) with acceptable com-

plexity. First, we validate the feasibility of problem (5.14), i.e., whether the radio resources, $N^{\max}$ and $P^{\max}$, can guarantee the QoS requirements of all the K users. If the problem is feasible, then we find the optimal solution of problem (5.14).

Feasibility of Problem (5.14)

To find out whether problem (5.14) is feasible, we minimize the required total transmit power $\sum_{k\in\mathcal{K}^\xi} P_k^\xi$ subject to the other constraints; if the required total transmit power is less than $P^{\max}$, then the problem is feasible. Otherwise, it is infeasible. The required minimum total transmit power can be found by solving the following problem:

$$\begin{aligned} &\min_{P_k^\xi, N_k^\xi} \sum_{k\in\mathcal{K}^\xi} P_k^\xi, \\ &\text{s.t. (5.14a), (5.2), (5.9), and (5.12),} \end{aligned} \tag{5.18}$$

From (5.1), (5.8) and (5.11), we can see that the left-hand side of constraints (5.2), (5.9), and (5.12) are monotonous with respect to P_k^ξ. Thus, for a given bandwidth allocation, N_k^ξ, the minimum transmit power that is required to ensure constraints (5.2), (5.9), and (5.12) can be obtained via binary search, and is denoted by $P_k^\xi(N_k^\xi)$. In the sequel, we propose a bandwidth allocation algorithm.

Step 1: Initialize bandwidth allocation with $N_k^\xi = 1, \forall k$, and compute $\Delta P_k^\xi(N_k^\xi) = P_k^\xi(N_k^\xi) - P_k^\xi(N_k^\xi + 1), \forall k \in \mathcal{K}^\xi$.

Step 2: Assign one more subcarrier to the user with the highest power saving, i.e.,

$$k^* = \arg\max_{k\in\mathcal{K}^\xi} \Delta P_k^\xi(N_k^\xi).$$

Step 3: Update $N_{k^*}^\xi$ and $\Delta P_{k^*}^\xi(N_{k^*}^\xi)$.

Finally, we execute Step 2 and Step 3 iteratively until $\sum_{k\in\mathcal{K}^\xi} N_k^\xi = N^{\max}$.

The proposed algorithm for problem (5.18) is summarized in Table 5.2. With this algorithm, we need to compute $\Delta P_k^\xi(N_k^\xi)$ for K users in Step 1. After that, we only

Table 5.2 – Bandwidth Allocation Algorithm for Solving Problem (5.18)

Require: Large-scale channel gains α_k^{ξ} and QoS constraints c_k^{ξ}.
1: Initialize $N_k^{\xi} = 1, \forall k \in \mathcal{K}^{\xi}$.
2: Compute $\Delta P_k^{\xi}(N_k^{\xi}) = P_k^{\xi}(N_k^{\xi}) - P_k^{\xi}(N_k^{\xi}+1), \forall k \in \mathcal{K}^{\xi}$.
3: **while** $\sum_{k\in\mathcal{K}^{\xi}} N_k^{\xi} \leq N^{\max}$ and $\Delta P_{k^*}^{\xi}(N_{k^*}^{\xi}) > 0$ **do**
4: $k^* := \arg\max_{k\in\mathcal{K}^{\xi}} \Delta P_k^{\xi}(N_k^{\xi})$.
5: $N_{k^*}^{\xi} := N_{k^*}^{\xi} + 1$.
6: Update $P_{k^*}^{\xi}(N_{k^*}^{\xi})$ and $P_{k^*}^{\xi}(N_{k^*}^{\xi}+1)$ according to (5.2), (5.9), and (5.12).
7: $\Delta P_{k^*}^{\xi}(N_{k^*}^{\xi}) := P_{k^*}^{\xi}(N_{k^*}^{\xi}) - P_{k^*}^{\xi}(N_{k^*}^{\xi}+1)$.
8: **end while**
9: **return** $\hat{N}_k^{\xi} := N_k^{\xi}$ and $\hat{P}_k^{\xi}(\hat{N}_k^{\xi}) := P_k^{\xi}(N_k^{\xi})$, $k = 1, ..., K$.

---

need to update $\Delta P_{k*}^{\xi}(N_{k*}^{\xi})$ for the k^*-th user in the following $N^{\max} - K$ iterations. Thus, the complexity of the algorithm is $\mathcal{O}(N^{\max}\Omega_P)$, where Ω_P denotes the operations needed to compute $\Delta P_{k^*}^{\xi}(N_{k^*}^{\xi})$. In other words, the complexity of the algorithm linearly increases with $N^{\max}$ and does not change with the total number of users.

To update the values of $P_{k^*}^{\xi}(N_{k^*}^{\xi})$ and $P_{k^*}^{\xi}(N_{k^*}^{\xi}+1)$ for $\Delta P_{k^*}^{\xi}(N_{k^*}^{\xi})$ in (5.1), (5.8) and (5.11), one approach is to compute the multiple integrals and then apply binary search. Such an approach is time-consuming since the system needs to compute the multiple integrals in each step of the binary search. To avoid computing the integrals, we adopt the stochastic gradient descent method to find the minimum transmit power subject to constraints (5.2), (5.9), and (5.12), respectively.

Let $x(\tau)$ be a variable obtained in the τ-th iteration. For delay-tolerant services, by substituting (5.1) into (5.2), the optimal transmit power with a given bandwidth allocation can be found through the following iterations:

$$P_k^{\mathrm{t}}(\tau+1) = \left[P_k^{\mathrm{t}}(\tau) + \phi(\tau)\left(\frac{\bar{a}_k}{N_k^{\mathrm{t}}} - W\log_2\left(1 + \frac{\alpha_k^{\mathrm{t}} g_{k,n}^{\mathrm{t}} P_k^{\mathrm{t}}(\tau)}{N_0 N_{\mathrm{T}} N_k^{\mathrm{t}} W}\right)\right)\right]^+. \tag{5.19}$$

where $[x]^+ = \max\{x, 0\}$ to ensure a positive result, $\phi(\tau) > 0$ is the step size.

For delay-sensitive services, we first transform constraint (5.9) into an equivalent form

that is linear to the expectation, i.e.:

$$\mathbb{E}_{g^{\mathrm{s}}_{k,n}}\left[\exp\left(-\theta^{\mathrm{s}}_k T_{\mathrm{c}} R^{\mathrm{s}}_k\right)\right] - \exp(-\theta^{\mathrm{s}}_k T_{\mathrm{c}} E^{B,\mathrm{s}}_k) \leq 0. \tag{5.20}$$

Then, the optimal transmit power with a given bandwidth allocation can be found through the following iterations:

$$P^{\mathrm{s}}_k(\tau+1) = \left[P^{\mathrm{s}}_k(\tau) + \phi(\tau)\left(\exp(-\theta^{\mathrm{s}}_k T_{\mathrm{c}} R^{\mathrm{s}}_k(\tau)) - \exp(-\theta^{\mathrm{s}}_k T_{\mathrm{c}} E^{B,\mathrm{s}}_k)\right)\right]^+. \tag{5.21}$$

where $R^{\mathrm{s}}_k(\tau)$ is the achievable rate in (5.3), which is computed from a set of realizations of small-scale channel gains on the N^{s}_k subcarriers, $\{g^{\mathrm{s}}_{k,1}, ..., g^{\mathrm{s}}_{k,n}, ..., g^{\mathrm{s}}_{k,N^{\mathrm{s}}_k}\}$.

For URLLC services, by substituting (5.11) into (5.12), the optimal transmit power with a given bandwidth allocation can be obtained from the following iterations:

$$\begin{aligned} &P^{\mathrm{u}}_k(\tau+1) \\ &= \left[P^{\mathrm{u}}_k(\tau) + \phi(\tau)\left(f_Q\left(\sqrt{\frac{T_{\mathrm{s}}W}{N^{\mathrm{u}}_k}}\left\{\left[\sum_{n=1}^{N^{\mathrm{u}}_k}\ln\left(1+\frac{\alpha^{\mathrm{u}}_k g^{\mathrm{u}}_{k,n} P^{\mathrm{u}}_k(\tau)}{N_0 N_{\mathrm{T}} N^{\mathrm{u}}_k W}\right)\right] - \frac{b^{\mathrm{u}}_k \ln 2}{T_{\mathrm{s}}W}\right\}\right) - \epsilon^{\max,\mathrm{u}}\right)\right]^+. \end{aligned} \tag{5.22}$$

As indicated in [166], the stochastic gradient descent method can converge to the unique optimal transmit power with $\phi(\tau) \sim \mathcal{O}(1/\tau)$.

Algorithm for Solving Problem (5.14)

If problem (5.14) is feasible, we first find the optimal subcarrier and transmit power allocation that minimizes P_{tot} without the total transmit power constraint in (5.14b). The algorithm is summarized in Table 5.3.

We first replace $\Delta P^{\xi}_k(N^{\xi}_k)$ in Table 5.2 with

$$\begin{aligned} \Delta P^{\xi}_{\mathrm{tot},k}(N^{\xi}_k) &\triangleq P_{\mathrm{tot}}([N^{\xi}_1, ..., N^{\xi}_k, ..., N^{\xi}_K]) - P_{\mathrm{tot}}([N^{\xi}_1, ..., N^{\xi}_k + 1, ..., N^{\xi}_K]) \\ &= \Delta P^{\xi}_k(N^{\xi}_k) - P^{\mathrm{ca}} N_{\mathrm{T}}. \end{aligned} \tag{5.23}$$

Like the algorithm in Table 5.2, each subcarrier is allocated to the user with the highest $\Delta P^{\xi}_{\text{tot},k}(N^{\xi}_k)$. The solution to this step is denoted by $\check{N}^{\xi}_k$ and $\check{P}^{\xi}_k(\check{N}^{\xi}_k)$.

If the equality in constraint (5.14a) holds with this solution, i.e., $\sum_{k\in\mathcal{K}^{\xi}} \check{N}^{\xi}_k = N^{\max}$, then minimizing $\sum_{k\in\mathcal{K}^{\xi}} P^{\xi}_k$ is equivalent to minimizing P_{tot} since the second term in (5.13) is fixed. In other words, the solutions obtained from the algorithms in Tables 5.2 and 5.3 are the same.

If $\sum_{k\in\mathcal{K}^{\xi}} \check{N}^{\xi}_k < N^{\max}$, then we check whether constraint (5.14b) is satisfied or not. If it is satisfied, the outputs of the algorithm, $\dot{N}^{\xi}_k$ and $\dot{P}^{\xi}_k(\dot{N}^{\xi}_k)$, are the same as $\check{N}^{\xi}_k$ and $\check{P}^{\xi}_k(\check{N}^{\xi}_k)$, respectively. Otherwise, more subcarriers will be assigned to the users until the transmit power constraint is satisfied (Lines 13-21 in Table 5.3).

Since the second term in (5.23) is constant, to compute the value of $\Delta P^{\xi}_{\text{tot},k}(N^{\xi}_k)$ we only need to compute $\Delta P^{\xi}_k(N^{\xi}_k)$. Therefore, the complexity of the algorithm in Table 5.3 is the same as that of the algorithm in Table 5.2.

Optimality of Algorithm for Solving Problem (5.14)

In this subsection, we first discuss the optimality conditions of the algorithm in Table 5.2, and prove that the conditions are satisfied with all the three kinds of services. Then, we prove the optimality of the algorithm in Table 5.3.

The algorithm in Table 5.2 can find the global optimal solution for problem (5.18) if the following two conditions hold (see proof in Appendix B.1).

Condition 5.1. $P^{\xi}_k(N^{\xi}_k) > P^{\xi}_k(N^{\xi}_k + 1)$, $\forall N^{\xi}_k = 1, ..., N^{\max} - 1$.

Condition 5.1 means the required transmit power decreases with the number of subcarriers.

Condition 5.2. $\Delta P^{\xi}_k(N^{\xi}_k) \geq \Delta P^{\xi}_k(N^{\xi}_k + 1), \forall N^{\xi}_k = 1, ..., N^{\max} - 1$.

To validate whether the two conditions hold with different kinds of services, we relax the numbers of subcarriers as continuous variables, and then prove the following proposition.

Table 5.3 – Bandwidth Allocation Algorithm for Solving Problem (5.14)

Require: Large-scale channel gains α_k^ξ and QoS constraints c_k^ξ.
1: Check whether problem (5.14) is feasible or not with the algorithm in Table 5.2.
2: Initialize $N_k^\xi := 1, \forall k \in \mathcal{K}^\xi$, and $P_{\text{tot}}([1, ..., 1])$.
3: Compute $\Delta P_{\text{tot},k}^\xi(N_k^\xi) := P_{\text{tot}}([N_1^\xi, ..., N_k^\xi, ..., N_K^\xi]) - P_{\text{tot}}([N_1^\xi, ..., N_k^\xi + 1, ..., N_K^\xi]), \forall k \in \mathcal{K}^\xi$.
4: **while** $\sum_{k\in\mathcal{K}^\xi} N_k^\xi \leq N^{\max}$ and $\Delta P_{\text{tot},k^*}^\xi(N_{k^*}^\xi) > 0$ **do**
5: $k^* := \arg\max_{k\in\mathcal{K}^\xi} \Delta P_{\text{tot},k}^\xi(N_k^\xi)$.
6: $N_{k^*}^\xi := N_{k^*}^\xi + 1$.
7: Update $P_{\text{tot}}([N_1^\xi, ..., N_{k^*}^\xi, ..., N_K^\xi])$ and $P_{\text{tot}}([N_1^\xi, ..., N_{k^*}^\xi + 1, ..., N_K^\xi])$ according to (5.2), (5.9), (5.12), and (5.13).
8: $\Delta P_{\text{tot},k^*}^\xi(N_{k^*}^\xi) := P_{\text{tot}}([N_1^\xi, ..., N_{k^*}^\xi, ..., N_K^\xi]) - P_{\text{tot}}([N_1^\xi, ..., N_{k^*}^\xi + 1, ..., N_K^\xi])$.
9: **end while**
10: $\check{N}_k^\xi := N_k^\xi$ and $\check{P}_k^\xi(\check{N}_k^\xi) := P_k^\xi(N_k^\xi), \forall k = 1, ..., K$.
11: **if** $\sum_{k\in\mathcal{K}^\xi} \check{P}_k^\xi(\check{N}_k^\xi) \leq P^{\max}$ **then**
12: **return** $\dot{N}_k^\xi := \check{N}_k^\xi, \dot{P}_k^\xi(\dot{N}_k^\xi) := \check{P}_k^\xi(\check{N}_k^\xi)$.
13: **else**
14: $\dot{N}_k^\xi := \check{N}_k^\xi, \dot{P}_k^\xi(\dot{N}_k^\xi) := \check{P}_k^\xi(\check{N}_k^\xi)$.
15: **while** $\sum_{k\in\mathcal{K}^\xi} \dot{P}_k^\xi(\dot{N}_k^\xi) \leq P^{\max}$ **do**
16: $k^* := \arg\max_{k\in\mathcal{K}^\xi} \Delta\dot{P}_k^\xi(\dot{N}_k^\xi)$.
17: $\dot{N}_{k^*}^\xi := \dot{N}_{k^*}^\xi + 1$.
18: Update $\dot{P}_{k^*}^\xi(\dot{N}_{k^*}^\xi)$ and $\dot{P}_{k^*}^\xi(\dot{N}_{k^*}^\xi + 1)$ according to (5.2), (5.9), and (5.12).
19: $\Delta\dot{P}_{k^*}^\xi(N_{k^*}^\xi) := \dot{P}_{k^*}^\xi(\dot{N}_{k^*}^\xi) - \dot{P}_{k^*}^\xi(\dot{N}_{k^*}^\xi + 1)$.
20: **end while**
21: **return** $\dot{N}_k^\xi, \dot{P}_k^\xi(\dot{N}_k^\xi)$.
22: **end if**

Proposition 5.1. For a constraint $f_k^\xi(P_k^\xi, N_k^\xi) = C_k, P_k^\xi \in \mathbb{R}^+, N_k^\xi \in \mathbb{R}^+$, if $f_k^\xi(P_k^\xi, N_k^\xi)$ is jointly concave (or convex) in P_k^ξ and N_k^ξ and increases (or decreases) with P_k^ξ and N_k^ξ, then Condition 5.1 and 5.2 hold.

Proof. See proof in Appendix B.2. □

Delay-Tolerant Services: As proved in [144], (5.1) is strictly concave in P_k^{t}. If $f(x)$ is concave, then $yf(x/y)$ is joint concave in x and y [138]. Thus, (5.1) is jointly concave in P_k^{t} and N_k^{t}. In addition, the Shannon's capacity increases with transmit power and the number of subcarriers. Therefore, Condition 5.1 and 5.2 hold for delay-tolerant

services.

Delay-Sensitive Services: According to the results in [145], we know that effective capacity is jointly concave in P_k^{s} and N_k^{s} and increases with P_k^{s} and N_k^{s}. Therefore, Condition 5.1 and 5.2 also hold for delay-tolerant services.

URLLC Services: Unlike the above two types of services, constraint (5.12) for URLLC is not convex in P_k^{u} and N_k^{u} in general. To study whether the proposed algorithm can find the optimal solution, we first consider an asymptotic scenario: N_{T} is large. When N_{T} is sufficiently large, due to channel hardening, we have [167]

$$\ln\left(1+\frac{\alpha_k^{\mathrm{u}} g_{k,n}^{\mathrm{u}} P_k^{\mathrm{u}}}{N_0 N_{\mathrm{T}} N_k^{\mathrm{u}} W}\right) \rightarrow \ln\left(1+\frac{\alpha_k^{\mathrm{u}} P_k^{\mathrm{u}}}{N_0 N_k^{\mathrm{u}} W}\right). \tag{5.24}$$

Then, the minimum transmit power that can satisfy constraint (5.12) can be derived as follows:

$$P_k^{\mathrm{u}}(N_k^{\mathrm{u}}) = \frac{N_0 N_k^{\mathrm{u}} W}{\alpha_k^{\mathrm{u}}}\left\{\exp\left[\frac{b_k^{\mathrm{u}}\ln 2}{T_{\mathrm{s}} N_k^{\mathrm{u}} W}+\frac{f_Q^{-1}\left(\bar{\varepsilon}^{\max,\mathrm{u}}\right)}{\sqrt{T_{\mathrm{s}} N_k^{\mathrm{u}} W}}\right]-1\right\}. \tag{5.25}$$

According to the analysis in [136], $P_k^{\mathrm{u}}(N_k^{\mathrm{u}})$ first decreases with N_k^{u} and then increases with $P_k^{\mathrm{u}}(N_k^{\mathrm{u}})$. We denote $\acute{N}_k^{\mathrm{u}}$ as the optimal number of subcarriers that minimizes $P_k^{\mathrm{u}}(N_k^{\mathrm{u}})$. Since $\Delta P_k^{\mathrm{u}}(\acute{N}_k^{\mathrm{u}}) < 0$, the number of subcarriers assigned to the k-th URLLC user will not exceed $\acute{N}_k^{\mathrm{u}}$. Moreover, $P_k^{\mathrm{u}}(N_k^{\mathrm{u}})$ is convex and decreases with N_k^{u} in the region $[1, \acute{N}_k^{\mathrm{u}}]$ [136]. Therefore, Condition 5.1 and 5.2 hold for URLLC services in the asymptotic scenario. For non-asymptotic scenarios, we will validate Condition 5.1 and 5.2 via numerical results.

The above analysis indicates that the algorithm in Table 5.2 can find the optimal solution of problem (5.18). In addition, by solving problem (5.18), we know whether problem (5.14) is feasible or not. If the problem is feasible, the following proposition shows that the algorithm in Table 5.3 can find the optimal solution to the problem.

Proposition 5.2. The algorithm in Table 5.3 can find the global optimal solution for problem (5.14) if Conditions 5.1 and 5.2 hold.

Proof. See proof in Appendix B.3. □

5.3.4 Train the Cascaded NNs

With the algorithm in Table 5.3, we can obtain a labeled training sample, $\boldsymbol{N}^*$ and $\boldsymbol{P}^*$, for any given input $\boldsymbol{X}$. To obtain enough labeled training samples, we randomly generate a large number of inputs and find the corresponding optimal solutions. One part of the data is used to train the NNs, and the other part of the data is used to test the performance of the NNs.

The parameters of the NNs are initialized with Gaussian distributed random variables with zero mean and unit variance. In each training epoch, a batch of training samples is randomly selected from all the training samples to train the NNs. The parameters of Φ_{I} are optimized with the Adam algorithm [127] to minimize a loss function, defined as $\mathcal{L}_{\mathrm{I}}(\Lambda_{\mathrm{I}}) = \frac{1}{M_t}\sum_{m_t=1}^{M_t}(\log(\boldsymbol{N}^*_{m_t}+1) - \log(\tilde{\boldsymbol{N}}_{m_t}+1))^2$, where M_t is the number of training samples in each batch. Similarly, we optimize the parameters of Φ_{I}^{ξ} to minimize $\mathcal{L}_{\mathrm{II}}(\Lambda_{\mathrm{II}}^{\xi}) = \frac{1}{M_t}\sum_{m_t=1}^{M_t}(\log(\boldsymbol{P}^*_{m_t}+1) - \log(\tilde{\boldsymbol{P}}_{m_t}+1))^2$. When the value of a loss function is below a required threshold, the difference between the outputs of the NNs and the optimal resource allocation is small enough, and the outputs of the NNs are near-optimal.

5.4 Deep Transfer Learning in Non-Stationary Wireless Networks

Since the cascaded NNs is trained offline, it only works well in stationary wireless networks. However, real-world wireless networks are highly dynamic and non-stationary. There are a lot of hidden variables that are not included in the input of the cascaded NNs but have significant impacts on the optimal solution. For example, the optimal resource allocation for delay-sensitive and URLLC services depends on distributions of small-scale channel gains as well as the types of services in the network. If these

distributions and parameters change, a NN trained offline is no longer a good approximation of the optimal resource allocation policy in the new scenario [63]. Such an issue is known as the task mismatch problem [87].

A straightforward approach is to train a new NN from scratch in a new scenario. When the hidden variables change, the system can hardly obtain a large number of training samples in the new scenario. This is because the algorithm in Table 5.3 cannot be executed in real-time[2]. To update the NN with a few training samples, we apply deep transfer learning.

5.4.1 Preliminary of Deep Transfer Learning

The learning process is to accomplish a learning *task* based on a data *domain*. According to the definitions in [87], a *domain* consists of a feature space and the corresponding marginal probability distribution, e.g., $\boldsymbol{X}$ and its distribution. A *task* consists of a label space and an objective predictive function that maps from $\boldsymbol{X}$ to $\boldsymbol{Y}^*$. The function is not observed but learned from the training samples, i.e., $\{\boldsymbol{X}, \boldsymbol{Y}^*\}$. The basic idea of transfer learning is to exploit the knowledge from a well-trained source task to a new target task [86].

Fine-tuning is the most widely used method in deep transfer learning [168]. The basic idea is to fix the parameters in the first a few layers and update the parameters in the last a few layers. In deep transfer learning, parts of the well-trained NN of the source task are reused in the NN of the target task. In this way, the number of labeled training samples are needed to fine-tune the new NN is much lower than that needed to train a new NN with randomly initialized parameters (i.e., learning from scratch).

[2]To execute Lines 7 and 18 of the algorithm in Table 5.3, the system needs to compute $\Delta P_{k^*}^{\xi}(N_{k^*}^{\xi})$ with an iterative algorithm.

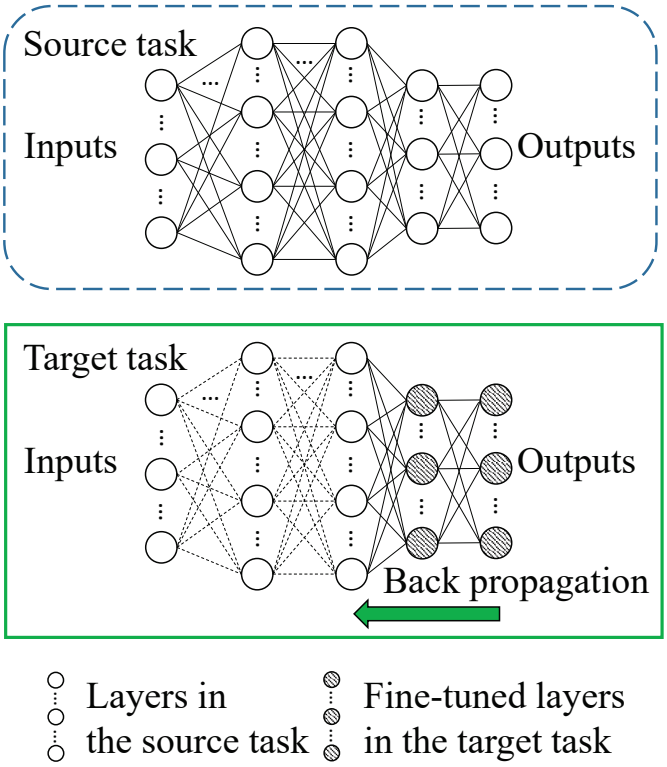

Figure 5.2 – Deep transfer learning with non-stationary wireless channels.

5.4.2 Transfer Learning with Non-Stationary Wireless Channels

In wireless communications, the distribution of small-scale channel gains may change over time. For example, the BS may switch ON/OFF some antennas. When the number of active antennas changes, the distribution of g_k becomes different. For the cascaded NNs proposed in the previous section, the system fine-tunes the last few layers of Φ_{I} as illustrated in Fig. 5.2. Since the power allocation policy depends on the distribution of wireless channels, the system fine-tunes all the layers of Φ_{II}^{ξ}.

5.4.3 Transfer Learning with Different Types of Services

In a wireless network, the service requests are highly dynamic. In other words, the number of different types of services in the wireless networks varies significantly over

time. Thus, the system needs to update the NN according to the QoS requirements of different types of services.

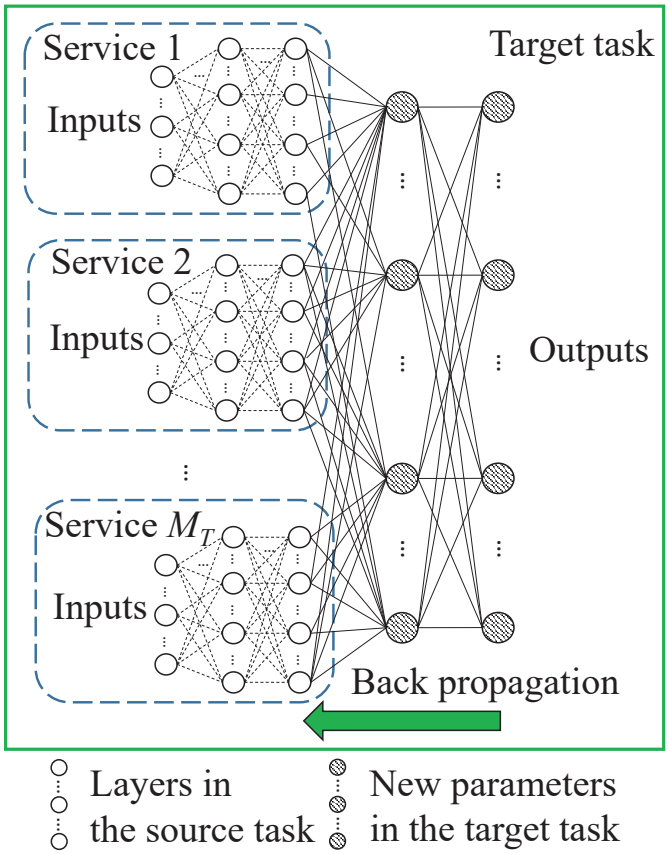

Figure 5.3 – Deep transfer learning with multiple types of services.

Transfer Learning from Delay-tolerant Services to Another Type of Services

For both delay-sensitive and URLLC services, the delay and reliability requirements depend on specific applications. Training NNs for all kinds of applications is not possible in practice. To overcome this difficulty, we first train a NN to approximate the optimal resource allocation policy of delay-tolerant services. Then, we fine-tune the NN for delay-sensitive and URLLC services. With the cascaded NNs in the previous section, the system needs to fine-tune the last few layers of Φ_{I} with the method in

Fig. 5.2. Since the power allocation policy depends on the QoS requirement of each service, all the layers of Φ_{II}^{ξ} should be updated.

Transfer Learning from a Single Type of Services to Multiple Types of Services

If the total number of different types of services is M_T, then there are 2^{M_T} possible combinations with different types of services. In 5G networks, M_T will be large, and it is impossible to train a NN for each combination. If we have a well-trained NN for each type of services (i.e., source task), then by replacing the last few layers of each NN, we can construct a new NN as that in Fig. 5.3. With the cascaded NNs, we only need to update Φ_{I} for bandwidth allocation. The power allocation for each user is determined by Φ_{II}^{ξ}, which is the same as that in the source task. The algorithm is summarized in Table 5.4.

5.5 Simulation and Numerical Results

In the considered scenario, the coverage of the BS is 200 meters. Users are uniformly distributed around the BS. The path loss model is $35.3 + 37.6 \log_{10}(d)$, where d is the distance (meters) between the BS and a user. The shadowing is lognormal distributed with 8 dB standard deviation. The small-scale channels are Rayleigh fading and the distribution of the small-scale channel gains follows $f_g(x) = \frac{1}{(N_{\mathrm{T}}-1)!} x^{N_{\mathrm{T}}-1} e^{-x}$. The rest of the simulation parameters are summarized in Table 5.5, unless specified otherwise.

5.5.1 Validating the Properties of URLLC

In this subsection, we first validate that Conditions 5.1 and 5.2 hold in non-asymptotic scenarios of URLLC. In Fig 5.4, we randomly select a user and illustrate the monotonicity of $P_k^{\mathrm{u}}(N_k^{\mathrm{u}})$ and $\Delta P_k^{\mathrm{u}}(N_k^{\mathrm{u}})$. The results show that even when the number of

Table 5.4 – Deep Transfer Learning for Multiple Types of Services

---

Require: Large-scale channel gains α_k^{ξ} and QoS constraints c_k^{ξ}.
1: Train a NN for delay-tolerant services.
2: Initialize an empty set $\mathcal{S}_T$
3: **for** the mth type of services, $m \in \{1, ..., M_T\}$ **do**
4: **if** the mth type of services is requested by some users **then**
5: $\mathcal{S}_T := \mathcal{S}_T \cup \{m\}$
6: Collect a few training samples for the mth type of services with the algorithm in Table 5.3.
7: Initialize parameters of a new NN with the parameters in the well-trained NN.
8: Fine-tuning the last a few layers of the new NN with the method in Section 5.4.3.
9: **end if**
10: Stack the NNs for all $m \in \mathcal{S}_T$ according to Fig. 5.3.
11: **end for**
12: Collect a few training samples for multiple types of services with the algorithm in Table 5.3.
13: Fine-tune the stacked NNs with the method in Section 5.4.3.
14: **return** the parameters of the fine-tuned NNs.

---

antennas is not large, such as $N_{\mathrm{T}} = 4, 8, 16$, Conditions 5.1 and 5.2 hold. The results indicate that the algorithms in Tables 5.2 and 5.3 can converge to the optimal solutions.

5.5.2 Performance Evaluation

In this subsection, we evaluate the performance achieved by the deep learning method in Section 5.3.2. In the scenarios with multiple types of services, the ratio of the number of users requesting the three types of services is set at $1 : 1 : 1$. The algorithm in Table 5.3 is used to find the optimal solutions of problem (5.14) with 10000 inputs, where $N^{\max} = 256$, $N_T = 64$ and $K^{\xi} = 20$. The first 9000 samples are used to train the NNs and the last 1000 samples are used to test their performance. In each epoch, $M_t = 128$ training samples are randomly selected from 9000 training samples, and

Table 5.5 – Parameters in Simulation

Maximal transmit power of the BS $P^{\max}$	46 dBm
Duration of one TTI T_{s}	0.125 ms
Bandwidth of each subcarrier W	120 kHz
Channel coherence time T_{c}	5 ms
Single-sided noise spectral density N_0	-174 dBm/Hz
Number of bytes in a packet b_k^{u}	$[20, 64]$ bytes
Average data arrival rate of delay-tolerant users $\bar{a}_k$	$[50, 100]$ KB/s
Packet loss probability of URLLC $\epsilon^{\max,\mathrm{u}}$	5×10^{-8}
Circuit power consumption per antenna $N^{\max} P^{\mathrm{ca}}$	50 mW
Fixed circuit power P_0^{c}	50 mW
Power amplifier efficiency ρ	0.5
Average packet arrival rate of delay-sensitive services ν^{a}	$[100, 1000]$ packets/s
Average packet size of delay-sensitive services $1/\nu^{\mathrm{s}}$	$[1, 20]$ kbits
Delay bound of delay-sensitive user $D_k^{\mathrm{q,s}}$	50 ms
Maximal tolerable delay bound violation probability of delay-sensitive user $\epsilon_k^{\mathrm{q,s}}$	10^{-2}

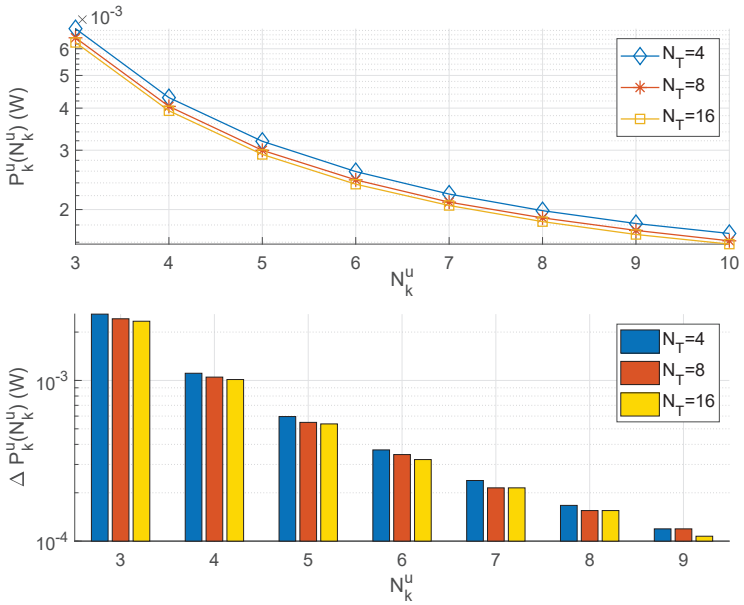

Figure 5.4 – Validating Condition 5.1 and 5.2 for URLLC services.

the learning rate is set to be 0.001. The DL algorithm is implemented in Python with TensorFlow 1.11.

Each neural network consists of one input layer, one output layer, and L^{ξ}_{hidden} hidden layers, where each hidden layer has N^{ξ}_{neurons} neurons. The input and output layers of the FNN are defined after (5.16). The input and output layers of the cascaded NNs are defined in Fig. 5.1. The hyper-parameters (i.e., L^{ξ}_{hidden} and N^{ξ}_{neurons}) for different types of services can be found in Table 5.6. We tried different values of hyper-parameters, and the values in Table 5.6 can achieve the best performance according to our experience.

Table 5.6 – Hyper-parameters of NNs

Service type	FNN		Cascaded NNs			
			The 1st part Φ_{I}		The 2nd part Φ^{ξ}_{II}	
	L^{ξ}_{hidden}	N^{ξ}_{neurons}	L^{ξ}_{hidden}	N^{ξ}_{neurons}	L^{ξ}_{hidden}	N^{ξ}_{neurons}
Delay-tolerant	4	800	4	800	4	20
Delay-sensitive	5	600	5	600	4	20
URLLC	4	600	4	600	4	20

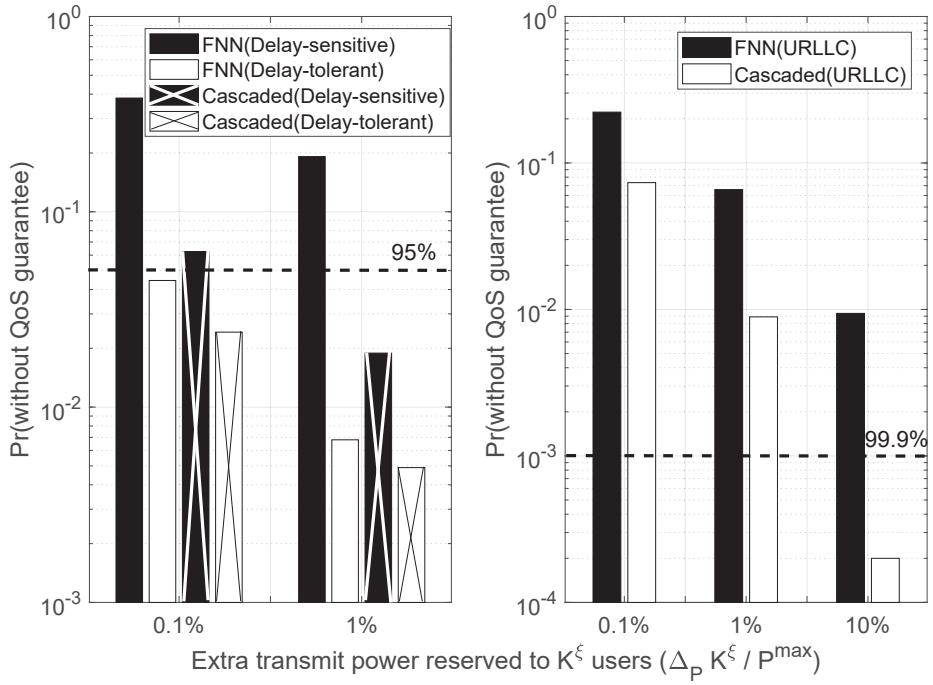

Figure 5.5 – Probability without QoS guarantee v.s. extra transmit power reserved to the users

In Fig. 5.5, we show the QoS achieved by the FNN and the cascaded NNs. Specifically, the relation between the probability without QoS guarantee and the extra transmit power reserved to all the K^{ξ} users, $\Delta_P K^{\xi}$, is provided. For each type of service, we set $K^{\xi} = 20$. The results are evaluated with 1000 testing samples.

From Fig. 5.5, we can observe that the cascaded NNs can achieve much better QoS compared with the FNN. For example, the system should guarantee the QoS requirement of URLLC with a probability higher than 99.9%. By reserving 10% of $P^{\max}$ extra transmit power to the 20 users that request URLLC services, the output of cascaded NNs can satisfy the QoS requirement with the probability of 99.98%. However, the output of the FNN can only satisfy the QoS requirement with the probability of 99.2%. For other types of services, the cascaded NNs also outperform the FNN in terms of achieving better QoS. This validates that the cascaded NNs can improve the approximation accuracy for all types of services.

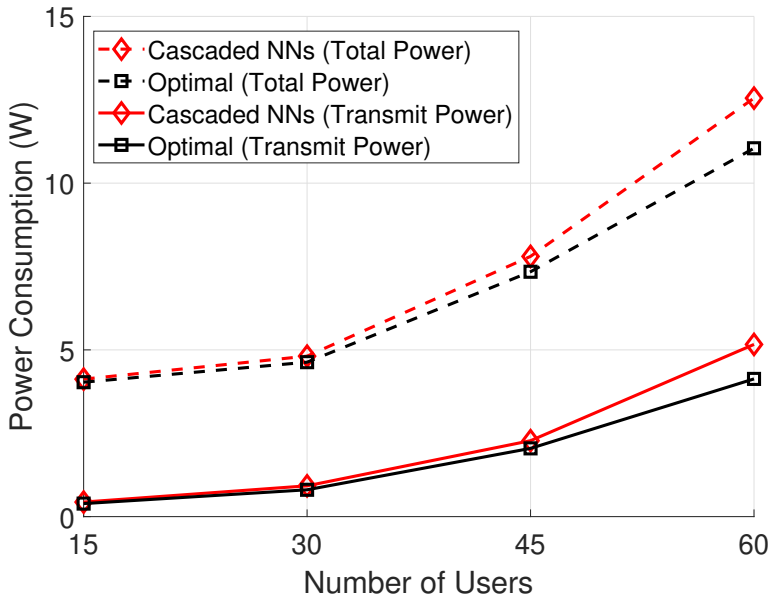

Figure 5.6 – Power consumption v.s. number of users when $N_T = 64$ and $N^{\max} = 256$.

The total power consumption and transmit power achieved with different schemes are illustrated in Fig. 5.6. We compare the performance of the cascaded NNs with the optimal solutions obtained with the algorithm in Tables 5.2 and 5.3 (with legend 'Optimal'). For the deep learning method, we train the cascaded NNs when $K^{\mathrm{t}} = K^{\mathrm{s}} = K^{\mathrm{u}} = 20$, which is close to the maximal number of users that can be served with the given radio resources[3]. In practical systems, the number of users is dynamic. When the number of users is less than 60, we do not change the dimension of the input, but set $c_k^{\xi} = 0$. It means that the required data rates, effective bandwidth or packet sizes of some users are zero. In this case, no resource will be assigned to them. The performance with the cascaded NNs is close to the optimal solutions. This implies the cascaded NNs are a good approximation of the optimal policy.

[3]The maximal number of users that can be served by a BS depends on the distribution of the users. We set the user-BS distance equals to the radium of the cell to calculate the maximal number of users.

5.5.3 Performance with Transfer Learning

Since NNs are used to approximate the optimal resource allocation policy, the accuracy is defined as follows:

$$\eta(\%) = 1 - \text{Error}(\%) = 1 - \frac{P_{\text{tot}}(\tilde{\boldsymbol{N}}, \tilde{\boldsymbol{P}}) - P_{\text{tot}}(\boldsymbol{N}^*, \boldsymbol{P}^*)}{P_{\text{tot}}(\boldsymbol{N}^*, \boldsymbol{P}^*)}, \tag{5.26}$$

which reflects the gap between the outputs of NNs and the optimal solutions.

To show that convergence time of different methods, we provide the relation between the numbers of training epochs and the accuracy. The transfer learning methods that fine-tune the well-trained NNs in the source domain and task are compared with the benchmark that trains new NNs with randomly initialized parameters (with legend 'Random initialization' and initializing each parameter with a zero mean and unit variance Gaussian variable). In this subsection, we only consider the cascaded NNs since this structure can guarantee the QoS constraints with a high probability.

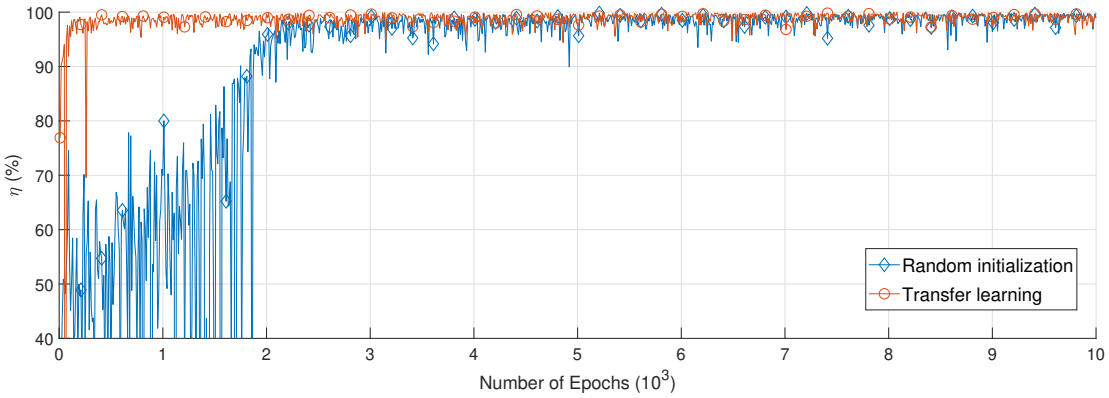

Figure 5.7 – Accuracy v.s. the number of training epochs when N_T varies, where $N^{\max} = 256$, $K^{\text{u}} = 20$

Transfer learning with non-stationary wireless channels

The training samples in the source domain and task are obtained when $N_{\text{T}} = 16$. The training samples in the target domain and task are obtained when $N_{\text{T}} = 64$. With

different numbers of antennas, the distribution of small-scale channel gains varies. With transfer learning, the first 3 layers of Φ_{I} are fixed. The last layer of Φ_{I} and Φ_{II}^{ξ} are fine-tuned. The results in Fig. 5.7 show that with transfer learning, only 400 epochs are needed to achieve around 98% accuracy, while 2000 epochs are needed to achieve the same accuracy with random initialization.

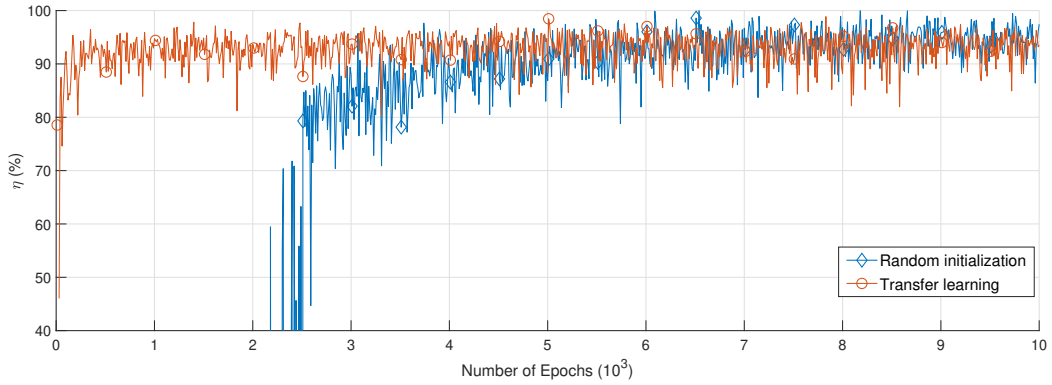

Figure 5.8 – Accuracy v.s. the number of training epochs, where the target task is resource allocation for delay-sensitive services, $N^{\max} = 256$, $N_T = 64$ and $K^{\xi} = 20$.

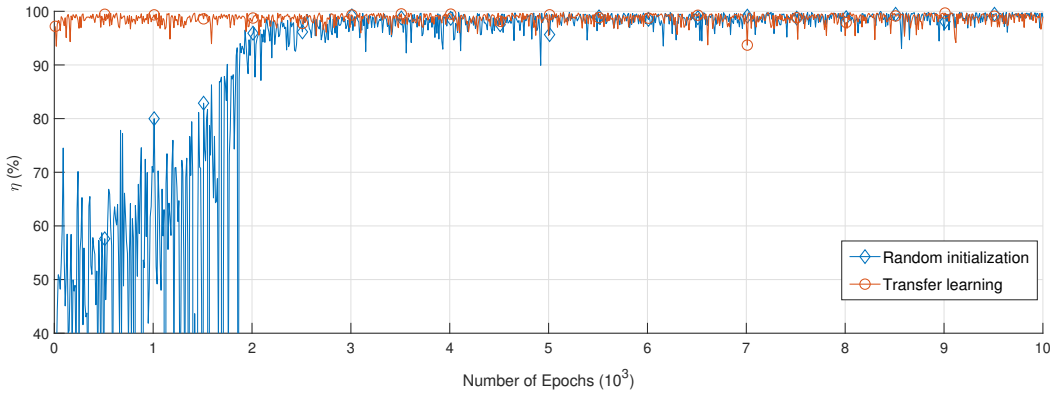

Figure 5.9 – Accuracy v.s. the number of training epochs, where the target task is resource allocation for URLLC, $N^{\max} = 256$, $N_T = 64$ and $K^{\xi} = 20$.

Transfer learning from delay-tolerant services to another type of services

We first train cascaded NNs for delay-tolerant services with 9000 labeled training samples. Then, we fine-tune the well-trained cascaded NNs with new labeled training samples of another type of services. Specifically, the first 3 layers of Φ_{I} are fixed and the NNs in the second part, $\Phi_{\mathrm{II}}^{\mathrm{t}}$, are replaced with NNs for delay-sensitive services, $\Phi_{\mathrm{II}}^{\mathrm{s}}$.

The results in Fig. 5.8 show that the transfer learning method can achieve 90% accuracy with around 150 epochs, while it takes 2500 epochs for the random initialization method to achieve the same accuracy. A similar conclusion can be observed from the results in Fig. 5.9. By comparing the results in Figs. 5.8 and 5.9, we can see that the accuracy of transfer learning for URLLC is higher than that for delay-sensitive services. As shown in Table 5.6, to achieve a good performance for delay-sensitive services, we need 5 hidden-layers in Φ_{I}. However, for delay-tolerant and URLLC services, only 4 hidden-layers are needed in Φ_{I}. Since we use the same hyper-parameter in the source task and target tasks, deep transfer learning achieves higher accuracy for URLLC compared with delay-sensitive services.

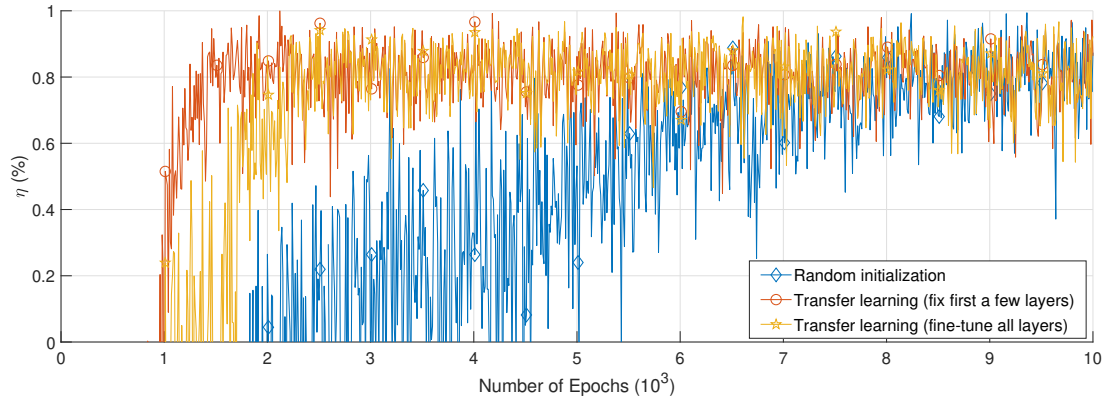

Figure 5.10 – Transfer knowledge from networks with a single type of services to networks with multiple types of services, where the target task is resource allocation for URLLC, $N^{\max} = 256$, $N_T = 64$ and $K^{\xi} = 20$.

Transfer knowledge from a single type of services to multiple types of services

To apply transfer learning in bandwidth allocation, the structure in Fig. 5.3 is adopted. Specifically, the output layers of the three NNs for the three types of services are replaced with an output layer with $(K^{\mathrm{t}} + K^{\mathrm{s}} + K^{\mathrm{u}})$ neurons. With deep transfer learning, we can either fix the first few layers or fine-tune all the layers, i.e., the curves with legends 'Transfer learning (fix first a few layers)' and 'Transfer learning (fine-tune all layers)', respectively. Their performance is compared with a benchmark that trains a NN with randomly initialized parameters (with legend 'Random initialization'), where the NN includes 4 hidden layers, each with 800 neurons. The results in Fig. 5.10 show that 'Transfer learning (fix first a few layers)' outperforms 'Transfer learning (fine-tune all layers)' in the first 2500 epochs, and they achieve the same performance after the 2500 epochs. This indicates that there is no need to fine-tune all the layers of the NN. Compared with the benchmark, transfer learning can achieve a higher accuracy in the first 8000 training epochs. Even after the 8000 epochs, their performance is almost the same.

5.6 Chapter Summary

In this chapter, we studied how to use deep learning in resource allocation with diverse QoS requirements in 5G networks. Specifically, we proposed an optimization algorithm that can converge to the optimal solution of an optimization problem that minimizes the total power consumption for delay-tolerant, delay-sensitive, and URLLC services. The obtained optimal solutions were used as labeled training samples to train NNs that approximate the optimal policy. To guarantee the diverse QoS requirements in non-stationary wireless networks, we designed cascaded NNs and fine-tuned their parameters with deep transfer learning. Our simulation results validated that the proposed deep transfer learning framework converges quickly when the wireless channels or the service requests are non-stationary.

Chapter 6

Conclusion

In this thesis, we focused on resource management by designing low-complexity optimization algorithms, and deep learning architectures in different settings for 5G NR to resolve the optimization problems subject to QoS requirements. This chapter provides a summary of the contributions and results and the potential future developments.

6.1 Summary of Content and Results

In Chapter 3, we first studied the joint cooperative beamforming and user association problem in the FD massive MIMO system, which is well-known for its high data rate capability, one of the QoS requirements in 5G NR networks. We proposed to maximize the network capacity by jointly optimizing the beamforming vectors and a binary vector that represents user association decisions. The binary vectors were further transformed into binary BAF as indicators of both the beamforming and the user association decisions. A three-step Gaussian belief propagation-based distributed solver was proposed to solve this non-convex NP-hard problem with low computational complexity. By relaxing the binary BAFs, we transformed the optimization problem into a linear programming. Subsequently, GaBP was used to efficiently obtain a feasible solution on BAFs in parallel, and two mapping algorithms were

proposed to achieve different performance-complexity trade-offs. Simulation results showed that the proposed cooperative beamforming methods significantly outperform the benchmarks in the literature in terms of network capacity with a low computation complexity.

In Chapter 4, we considered a MEC system to accommodate the QoS requirements on low latency and high reliability. Hybrid user types were considered with both URLLC and delay-tolerant services. We aimed to minimize the normalized energy consumption, defined as the energy consumption per bit, by optimizing user association, resource allocation, and offloading probabilities subject to the quality-of-service requirements. The user association was managed by the MME, while resource allocation and offloading probabilities were determined by each AP. Due to the ultra-low E2E requirement of URLLC, real-time implementation for optimization problems is crucial. We proposed a DL architecture, where a digital twin of the real network environment was used to train the DL algorithm off-line at a central server. From the pre-trained DNN, the MME could obtain a user association scheme in real time. Considering that real networks were not static, the digital twin monitored the variation of real networks and updated the DNN accordingly. For a given user association scheme, we proposed an optimization algorithm to find the optimal resource allocation and offloading probabilities at each AP. Simulation results showed that our method could achieve lower normalized energy consumption with less computation complexity compared with the existing baselines considered.

In Chapter 5, we extended the scenario of two types of services into multiple types of services with diverse QoS requirements. BS needs real-time optimization of wireless network resources in time-varying network conditions in 5G cellular networks. This brings high computational overheads and long processing delays. To tackle the problems of adjustment of time-varying channel and traffic conditions, and stringent QoS guarantees, we developed a deep learning framework to obtain the optimal resource allocation policy. The algorithm aimed to minimize the total power consumption of a base station by optimizing subcarrier and transmit power allocation. We proposed a cascaded structure of NNs, where the first NN approximated the optimal subcarrier

allocation, and the second NN obtained the transmit power required to ensure the QoS requirement with given subcarrier allocation. Considering that the distribution of wireless channels and the types of services in wireless networks are non-stationary, we applied deep transfer learning to update NNs in these. Simulation results confirmed that the cascaded NNs outperform the fully connected NN in terms of QoS guarantee. In addition, deep transfer learning could significantly reduce the number of training samples required to train the NNs.

6.2 Future Work

To conclude this thesis, we list a number of promising research directions that follow from the work conducted herein.

In Chapter 3, cooperative beamforming was designed with an assumption that users' location does not change when their large-scale channel gains remain unchanged. This work can be extended to a scenario including user mobility. Taking into account the mobility of users over a 2-D space can be beneficial for both urban and rural areas. However, it can also be challenging to achieve as the speed of mobility can be high. Although a low-complexity GaBP-based algorithm was proposed to solve such a non-convex NP-hard problem, the optimization method may be too slow to react to the time-varying channel conditions on time. In this case, the prediction of user locations can be performed based on their historical trajectories by using deep learning methods.

In Chapter 4, we decomposed the optimization problems into two stages. The first sub-problem was to solve a resource allocation problem with a given user association scheme at the BS through a conventional optimizing tool. The second sub-problem was to optimize user association decisions at the central server by using DL. Although the performance of the proposed framework was close to the optimal performance, a two-stage problem solving inevitably produces computational delays and information delays. It would be interesting to replace the two-stage method with a direct solver based on DL, such that optimization delays are shortened. In return, there

might be a trade-off between the solver's performance and computational complexity. Considering the problem was over both discrete and continuous multiple optimization variables, the outputs of a DNN can be less accurate compared with the optimal solution, as quantization errors are introduced by discrete variables.

The objective considered in Chapter 5 was to minimize the total power consumption of a BS. Alternatively, the maximum achievable capacity can be taken into account as an objective function in 5G cellular networks. Unlike the previous considered MEC system, where power consumption is a major issue for users' battery-powered devices, high data rate or high spectrum efficiency can be more critical in 5G wireless networks. Moreover, the same deep transfer learning framework could be extended to cellular networks, where user association can be considered to further increase the network capacity. At a cost, it may make the formulated problem even more challenging because different association schemes could lead to completely different resource allocation schemes for individual BSs. The difference between the outputs of NNs and the optimal solutions for user association decisions may be amplified to single BS resource allocation optimizations. In this case, extra care with user association designs by DL should be taken.

Appendix A

Proofs of Chapter 4

A.1 Proof of Proposition 4.1

Proof. Substituting the equality in (4.14) into (4.4), we can obtain that $E_k^{\mathrm{loc,b}} = k_0(\lambda_k^{\mathrm{b}})^2(\bar{c}_k^{\mathrm{b}})^3$. Further substituting $E_k^{\mathrm{loc,b}}$ into the first term in (4.18), we can derive the normalized energy consumption at the local server, i.e.,

$$\eta_k^{\mathrm{loc,b}} = (1 - x_k^{\mathrm{b}})\frac{E_k^{\mathrm{loc,b}}}{\bar{b}_k^{\mathrm{b}}} = \frac{k_0(\lambda_k^{\mathrm{b}})^2(\bar{c}_k^{\mathrm{b}})^3}{\bar{b}_k^{\mathrm{b}}}(1 - x_k^{\mathrm{b}})^3. \tag{A.1.1}$$

From (A.1.1), we can derive that $\frac{\partial^2 \eta_k^{\mathrm{loc,b}}}{\partial (x_k^{\mathrm{b}})^2} = \frac{6k_0(\lambda_k^{\mathrm{b}})^2(\bar{c}_k^{\mathrm{b}})^3}{\bar{b}_k^{\mathrm{b}}}(1 - x_k^{\mathrm{b}}) > 0$, and hence the first term in (4.18) is convex in x_k^{b}.

To prove the second term in (4.18) is convex in x_k^{b}, we only need to prove $P_k^{\mathrm{t,b}}$ is convex in x_k^{b}. From (4.15), the required average data rate linearly increases with x_k^{b}. Thus, we only need to prove $P_k^{\mathrm{t,b}}$ is convex in $\mathbb{E}\left(R_k^{\mathrm{b}}\right)$. From (4.2) we can see that R_k^{b} increases with $P_k^{\mathrm{t,b}}$, and it is concave in $P_k^{\mathrm{t,b}}$. Since the expectation does not change the monotonicity and convexity of the function, $\mathbb{E}\left(R_k^{\mathrm{b}}\right)$ is an increasing and concave function with respect to $P_k^{\mathrm{t,b}}$. According to [138], if the original function is an increasing and concave function, then the inverse function is a convex function. Therefore, $P_k^{\mathrm{t,b}}$ is convex in $\mathbb{E}\left(R_k^{\mathrm{b}}\right)$.

Since the two terms in (4.18) are convex in x_k^{b}, η_k^{b} is convex in x_k^{b}. The proof follows. □

A.2 Proof of Property 4.1

Proof. For URLLC services, according to [125], we know that the required transmit power $P_k^{\mathrm{t,u}}$ to achieve a certain service rate decreases with $N_k^{\mathrm{th,u}}$ when $N_k^{\mathrm{th,u}} \leq \tilde{N}_k^{\mathrm{u}}$, where $\tilde{N}_k^{\mathrm{th,u}}$ can be obtained from $\frac{\partial P_k^{\mathrm{t,u}}}{\partial N_k^{\mathrm{th,u}}} = 0$. As a result, $\eta_k^{\mathrm{u}} = \frac{x_k^{\mathrm{u}} P_k^{\mathrm{t,u}} T_{\mathrm{s}}}{b_k^{\mathrm{u}}} + \frac{(1-x_k^{\mathrm{u}}) E_k^{\mathrm{loc,u}}}{b_k^{\mathrm{u}}}$ decreases with $N_k^{\mathrm{th,u}}$ when the offloading probability $x_k^{\mathrm{u}} \neq 0$ and $N_k^{\mathrm{th,u}} \leq \tilde{N}_k^{\mathrm{u}}$.

For delay tolerant services, we need to guarantee the rate constraint of the wireless link in (4.15). We substitute $\mathbb{E}_{g_k^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right)$ in (4.2) into (4.15), i.e., $\mathbb{E}_{g_k^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right) \geq x_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}} \lambda_k^{\mathrm{b}} / T_{\mathrm{s}}$, and obtain the relationship between $\mathbb{E}_{g_k^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right)$ and N_k^{b} when the offloading probability $x_k^{\mathrm{b}} \neq 0$ as follows

$$\mathbb{E}_{g_k^{\mathrm{b}}}\left[N_k^{\mathrm{b}} W \log_2\left(1 + \frac{\alpha_k^{\mathrm{b}} g_k^{\mathrm{b}} P_k^{\mathrm{t,b}}}{N_0 N_k^{\mathrm{b}} W}\right)\right] \frac{T_{\mathrm{s}}}{\bar{b}_k^{\mathrm{b}}} \geq x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}}. \tag{A.2.1}$$

That is for a given average offloading packet rate $x_k^{\mathrm{b}} \lambda_k^{\mathrm{b}}$, the transmitting power $P_k^{\mathrm{t,b}}$ decreases with N_k^{b} when the offloading probability $x_k^{\mathrm{b}} \neq 0$. As a result, $\eta_k^{\mathrm{b}} = P_k^{\mathrm{t,b}} T_{\mathrm{s}} / \lambda_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}} + \eta_k^{\mathrm{loc,b}}$ also decreases with N_k^{b} when the offloading probability $x_k^{\mathrm{b}} \neq 0$. This completes the proof. □

A.3 Proof of Property 4.2

Proof. For the URLLC services, ϱ in (4.25) decreases with N_k^{u} in the region $[0, \tilde{N}_k^{\mathrm{u}}]$ [125]. Substituting ϱ in (4.25) into the close-form of $\tilde{g}_k^{\mathrm{th,u}}$ in (4.28), we can see that $\tilde{g}_k^{\mathrm{th,u}}$ also decreases with N_k^{u}. From $x_k^{\mathrm{u}} = e^{-\tilde{g}_k^{\mathrm{th,u}}}$ in (4.3), we know that x_k^{u} decreases with $\tilde{g}_k^{\mathrm{th,u}}$. Therefore, $\hat{x}_k^{\mathrm{u}}(N_k^{\mathrm{th,u}}) = e^{-\hat{g}_k^{\mathrm{th,u}}}$ increases with $N_k^{\mathrm{th,u}}$, where $\hat{g}_k^{\mathrm{th,u}}$ in (4.29) increases with $\tilde{g}_k^{\mathrm{th,u}}$.

For the delay tolerant services, to prove the property, we only need to prove that $\eta_k^{\mathrm{b}'}$ increases with $\hat{x}_k^{\mathrm{b}}$ and decreases with N_k^{b}. Hence, when $\eta_k^{\mathrm{b}'} = 0$, $\hat{x}_k^{\mathrm{b}}$ increases with N_k^{b}.

From Proposition 4.1, we know that η_k^{b} is a convex function in x_k^{b}, and hence $\eta_k^{\mathrm{b}'}$ increases with $\hat{x}_k^{\mathrm{b}}$. To prove that $\eta_k^{\mathrm{b}'}$ decreases with N_k^{b}, we first derive the expression of $\eta_k^{\mathrm{b}'}$ as follows,

$$\eta_k^{\mathrm{b}'} = -\frac{E_k^{\mathrm{loc,b}}}{\bar{b}_k^{\mathrm{b}}} + \frac{T_{\mathrm{s}}}{\lambda_k^{\mathrm{b}} \bar{b}_k^{\mathrm{b}}} \frac{\partial P_k^{\mathrm{t,b}}}{\partial x_k^{\mathrm{b}}}, \tag{A.3.1}$$

where the expression of η_k^{b} in (4.18) is applied. (A.3.1) indicates that to prove $\eta_k^{\mathrm{b}'}$ decreases with N_k^{b}, we only need to prove that $\frac{\partial P_k^{\mathrm{t,b}}}{\partial x_k^{\mathrm{b}}}$ decreases with N_k^{b}. By substituting $\mathbb{E}_{g_k^{\mathrm{b}}}\left(R_k^{\mathrm{b}}\right)$ in (4.2) into the rate constraint of the the wireless link in (4.15), we can derive that

$$\frac{\partial x_k^{\mathrm{b}}}{\partial P_k^{\mathrm{t,b}}} = \frac{T_s}{\ln 2 \bar{b}_k^{\mathrm{b}} \lambda_k^{\mathrm{b}}} \int_0^{\infty} \frac{\alpha_k^{\mathrm{b}}}{N_0} g_k^{\mathrm{b}} e^{-g_k^{\mathrm{b}}} \frac{1}{1 + \frac{\alpha_k^{\mathrm{b}} g_k^{\mathrm{b}} P_k^{\mathrm{t,b}}}{N_0 W N_k^{\mathrm{b}}}} dg. \tag{A.3.2}$$

Based on (A.3.2), we can see that $\frac{\partial x_k^{\mathrm{b}}}{\partial P_k^{\mathrm{t,b}}}$ is an increasing function of N_k^{b}. According to the characteristic of inverse function (i.e., $\frac{\partial x_k^{\mathrm{b}}}{\partial P_k^{\mathrm{t,b}}} \times \frac{\partial P_k^{\mathrm{t,b}}}{\partial x_k^{\mathrm{b}}} = 1$ at any point $(x_k^{\mathrm{b}}, P_k^{\mathrm{t,b}})$), we can obtain that $\frac{\partial P_k^{\mathrm{t,b}}}{\partial x_k^{\mathrm{b}}}$ decreases with N_k^{b}. As a result, $\eta_k^{\mathrm{b}'}$ decreases with N_k^{b}. The proof follows. □

Appendix B

Proofs of Chapter 5

B.1 Proof of Optimality of the Algorithm in Table 5.2

Proof. We denote the objective function in (5.18) as $f(\boldsymbol{N}) = \sum_{k\in\mathcal{K}^\xi} P_k^\xi(N_k^\xi)$ in this Appendix, where $\boldsymbol{N} = [N_1^\xi, ..., N_K^\xi]^T$. The outcome of the algorithm in Table 5.2 is denoted by $\boldsymbol{N}^* = [N_1^{\xi*}, ..., N_K^{\xi*}]^T$. To prove the optimality of the proposed algorithm, we only need to prove that for any bandwidth allocation scheme $\boldsymbol{N'} = [N_1^{\xi'}, ..., N_K^{\xi'}]^T$, $f(\boldsymbol{N}^*) \leq f(\boldsymbol{N'})$ holds.

The difference between $\boldsymbol{N}^*$ and $\boldsymbol{N'}$ is denoted by

$$\Delta\boldsymbol{N} = \boldsymbol{N'} - \boldsymbol{N}^* = [\Delta N_1, \Delta N_2, ..., \Delta N_K]^T. \tag{B.1.1}$$

We further denote that

$$\boldsymbol{N}^+ = [\max(0, \Delta N_1), ..., \max(0, \Delta N_K)],$$
$$\boldsymbol{N}^- = [-\min(0, \Delta N_1), ..., -\min(0, \Delta N_K)].$$

Then, we can obtain a bandwidth allocation policy $\boldsymbol{N}^0 = \boldsymbol{N}^* - \boldsymbol{N}^- = \boldsymbol{N'} - \boldsymbol{N}^+$.

The required transmit power with policy $\boldsymbol{N}^0 = [N_1^0, ..., N_K^0]$ is $f(\boldsymbol{N}^0)$. Based on $\boldsymbol{N}^0$, if we allocate $(N^{\max} - \sum_{k=1}^{K} N_k^0)$ extra subcarriers to the users according to $\boldsymbol{N}^+$, the amount of power saving is $f(\boldsymbol{N}^0) - f(\boldsymbol{N}')$. If we allocate $(N^{\max} - \sum_{k=1}^{K} N_k^0)$ subcarriers according to $\boldsymbol{N}^-$, then the amount of power saving is $f(\boldsymbol{N}^0) - f(\boldsymbol{N}^*)$.

The amount of power saving with the above two approaches can be expressed as the sum of $(N^{\max} - \sum_{k=1}^{K} N_k^0)$ terms, i.e.,

$$f(\boldsymbol{N}^0) - f(\boldsymbol{N}') = \sum_{k^+ \in \mathcal{K}^+} \sum_{n=N_{k^+}^0}^{N_{k^+}^{'}-1} \Delta P_{k^+}(n), \tag{B.1.2}$$

$$f(\boldsymbol{N}^0) - f(\boldsymbol{N}^*) = \sum_{k^- \in \mathcal{K}^-} \sum_{n=N_{k^-}^0}^{N_{k^-}^{*}-1} \Delta P_{k^-}(n), \tag{B.1.3}$$

where $\mathcal{K}^+ = \{k | \Delta N_k > 0\}$ and $\mathcal{K}^- = \{k | \Delta N_k < 0\}$.

According to Condition 5.1 and 5.2, we have

$$\Delta P_{k^+}(N_{k^+}^0) \geq \Delta P_{k^+}(N_{k^+}^0 + 1) \geq ... \geq \Delta P_{k^+}(N_{k^+}^{'} - 1), \forall k^+ \in \mathcal{K}^+. \tag{B.1.4}$$

$$\Delta P_{k^-}(N_{k^-}^0) \geq \Delta P_{k^-}(N_{k^-}^0 + 1) \geq ... \geq \Delta P_{k^-}(N_{k^-}^{*} - 1), \forall k^- \in \mathcal{K}^-. \tag{B.1.5}$$

With the proposed algorithm, a subcarrier will be assigned to the user with the highest power saving. Thus, we have

$$\Delta P_{k^-}(N_{k^-}^{*} - 1) \geq \Delta P_{k^+}(N_{k^+}^0), \forall k^+ \in \mathcal{K}^+, \forall k^- \in \mathcal{K}^-. \tag{B.1.6}$$

Since $\Delta P_{k^-}(N_{k^-}^{*} - 1)$ is the last term in (B.1.5) and $P_{k^+}(N_{k^+}^0)$ is the first term in (B.1.4), we can obtain that each term in the right-hand side of (B.1.2) is smaller than any term in the right-hand side of (B.1.3). Therefore, we have $f(\boldsymbol{N}^0) - f(\boldsymbol{N}') \leq f(\boldsymbol{N}^0) - f(\boldsymbol{N}^*)$, and hence $f(\boldsymbol{N}^*) \leq f(\boldsymbol{N}')$. This completes the proof. □

B.2 Proof of Property 5.1

Proof. We first derive the first-order derivatives of the right-hand and the left-hand sides of $f_k^\xi(P_k^\xi, N_k^\xi) = C$, i.e.,

$$\frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial N_k^\xi} + \frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial P_k^\xi}\frac{\partial P_k^\xi}{\partial N_k^\xi} = 0. \tag{B.2.1}$$

Since $f_k^\xi(P_k^\xi, N_k^\xi)$ increases with both N_k^ξ and P_k^ξ, we have $\frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial N_k^\xi} > 0$ and $\frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial P_k^\xi} > 0$. According to (B.2.1), we can see that $\frac{\partial P_k^\xi}{\partial N_k^\xi} < 0$, i.e., P_k^ξ decreases with N_k^ξ. Therefore, Condition 5.1 holds.

From (B.2.1), we further derive the second-order derivative, i.e.,

$$\begin{aligned}&\underbrace{\frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial (P_k^\xi)^2}}_{a}\underbrace{\left(\frac{\partial P_k^\xi}{\partial N_k^\xi}\right)^2}_{x^2} + 2\underbrace{\frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial N_k^\xi \partial P_k^\xi}}_{b}\underbrace{\frac{\partial P_k^\xi}{\partial N_k^\xi}}_{x} + \\ &\underbrace{\frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial (N_k^\xi)^2}}_{c} + \underbrace{\frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial P_k^\xi}\frac{\partial^2 P_k^\xi}{\partial (N_k^\xi)^2}}_{d} = 0.\end{aligned} \tag{B.2.2}$$

For notational simplicity, we can simplify (B.2.2) as $ax^2 + 2bx + c + d = 0$, which can be re-expressed as follows,

$$a\left[\left(x + \frac{b}{a}\right)^2 + \frac{ac - b^2}{a^2}\right] + d = 0. \tag{B.2.3}$$

Since $f_k^\xi(P_k^\xi, N_k^\xi)$ is jointly concave in P_k^ξ and N_k^ξ, we have $a = \frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial (P_k^\xi)^2} \leq 0$ and $\frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial (P_k^\xi)^2}\frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial (N_k^\xi)^2} - \left(\frac{\partial^2 f_k^\xi(P_k^\xi, N_k^\xi)}{\partial N_k^\xi \partial P_k^\xi}\right)^2 \geq 0$, i.e., $ac - b^2 \geq 0$. Thus, from (B.2.3), we can see that $d = \frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial P_k^\xi}\frac{\partial^2 P_k^\xi}{\partial (N_k^\xi)^2} \geq 0$. Further considering that $\frac{\partial f_k^\xi(P_k^\xi, N_k^\xi)}{\partial P_k^\xi} > 0$, we can conclude that $\frac{\partial^2 P_k^\xi}{\partial (N_k^\xi)^2} \geq 0$, i.e., P_k^ξ is convex in N_k^ξ. Therefore, Condition 5.2 holds. The proof follows. □

B.3 Proof of Optimality of the Algorithm in Table 5.3

Proof. We denote the objective function in (5.14) as $f(\boldsymbol{N}) = \frac{1}{\rho}\sum_{k\in\mathcal{K}^\xi} P_k^\xi(N_k^\xi) + P^{\mathrm{ca}} N_{\mathrm{T}} \sum_{k\in\mathcal{K}^\xi} N_k^\xi + P_0^{\mathrm{c}}$ in this Appendix, where $\boldsymbol{N} = [N_1^\xi, ..., N_K^\xi]^T$. The bandwidth allocation obtained in Line 10 and Line 12 (or 21) in Table 5.3 are denoted by $\check{\boldsymbol{N}} = [\check{N}_1^\xi, ..., \check{N}_K^\xi]^T$ and $\dot{\boldsymbol{N}} = [\dot{N}_1^\xi, ..., \dot{N}_K^\xi]^T$, respectively.

Since the algorithm in Lines 2-10 in Table 5.3 is similar to the algorithm in Table 5.2, with the method in Appendix B.1, we can prove that $\check{\boldsymbol{N}} = [\check{N}_1^\xi, ..., \check{N}_K^\xi]^T$ minimizes $f(\boldsymbol{N})$ when Conditions 5.1 and 5.2 hold.

If $\sum_{k\in\mathcal{K}^\xi} \check{P}_k^\xi(\check{N}_k^\xi) \leq P^{\max}$, then the resource allocation satisfies the transmit power constraint, and $\check{N}_k^\xi$ and $\check{P}_k^\xi(\check{N}_k^\xi)$, $k = 1, ..., K$, are the optimal solution of problem (5.14).

If $\sum_{k\in\mathcal{K}^\xi} \check{P}_k^\xi(\check{N}_k^\xi) > P^{\max}$, the resource allocation that minimizes the total power consumption does not satisfies the maximal transmit power constraint. From the algorithm in Table 5.2, we know whether problem (5.14) is feasible or not. In the cases that the problem is feasible, we have $\sum_{k\in\mathcal{K}^\xi} \check{N}_k^\xi < N^{\max}$ when $\sum_{k\in\mathcal{K}^\xi} \check{P}_k^\xi(\check{N}_k^\xi) > P^{\max}$. From the condition in Line 4 in Table 5.3, we know that $\Delta P_{\mathrm{tot},k}^\xi(\check{N}_k^\xi) > 0, \forall k = 1, ..., K$. Thus, the total power consumption increases with N_k^ξ when $N_k^\xi \geq \check{N}_k^\xi$. Minimizing the total power consumption is equivalent to minimizing the number of subcarriers that can guarantee the maximal transmit power constraint. In the rest part of this appendix, we prove that the algorithm in Table 5.3 can find the minimal number of subcarriers.

The algorithm from Lines 15-20 in Table 5.3 is the same as that in Table 5.2. Thus, the bandwidth allocation obtained in each iteration minimizes the sum of the required transmit power. According to the condition in Line 15 in Table 5.3, if the total number of occupied subcarriers is less than $\sum_{k\in\mathcal{K}^\xi} \dot{N}_k^\xi$, then the maximal transmit power constraint cannot be satisfied. Therefore, $\sum_{k\in\mathcal{K}^\xi} \dot{N}_k^\xi$ is the minimum number of subcarriers that is required to satisfy the maximal transmit power constraint. This

completes the proof.

□

List of References

[1] G. Fettweis and S. Alamouti, "5g: Personal mobile internet beyond what cellular did to telephony," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 140–145, Feb. 2014.

[2] A. Prasad, A. Benjebbour, O. Bulakci, K. I. Pedersen, N. K. Pratas, and M. Mezzavilla, "Agile Radio Resource Management Techniques for 5G New Radio," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 62–63, June 2017.

[3] P. Schulz, M. Matthé, H. Klessig, *et al.*, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, Feb. 2017.

[4] 3GPP TSG RAN TR38.913 R14, "Study on scenarios and requirements for next generation access technologies," Jun. 2017.

[5] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-Enabled Tactile Internet," *IEEE J. Select. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[6] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Toward haptic communications over the 5G tactile internet," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3034–3059, Fourthquarter 2018.

[7] M. Mu, M. Broadbent, A. Farshad, N. Hart, D. Hutchison, Q. Ni, and N. Race, "A scalable user fairness model for adaptive video streaming over SDN-assisted future networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2168–2184, Aug. 2016.

[8] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 65–74, Jun. 2018.

[9] V. Tikhvinskiy and G. Bochechka, "Prospects and QoS requirements in 5G networks," vol. 2015, pp. 23–26, Jan. 2015.

[10] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[11] Y. Nam, B. L. Ng, K. Sayana, Y. Li, J. Zhang, Y. Kim, and J. Lee, "Full-dimension MIMO (FD-MIMO) for next generation cellular technology," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 172–179, Jun. 2013.

[12] S. M. Razavizadeh, M. Ahn, and I. Lee, "Three-Dimensional Beamforming: A new enabling technology for 5G wireless networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 94–101, Nov. 2014.

[13] G. Pocovi, H. Shariatmadari, G. Berardinelli, K. Pedersen, J. Steiner, and Z. Li, "Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements," *IEEE Network*, vol. 32, no. 2, pp. 8–15, Mar. 2018.

[14] P. Popovski, Stefanović, J. J. Nielsen, E. de Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A. Bana, "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5783–5801, Aug. 2019.

[15] C. She, Z. Chen, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Improving network availability of ultra-reliable and low-latency communications with multi-connectivity," *CoRR*, vol. abs/1806.06336, 2018. [Online]. Available: http://arxiv.org/abs/1806.06336

[16] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[17] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: Use cases, challenges and approaches," *CoRR*, vol. abs/1709.00560, 2017. [Online]. Available: http://arxiv.org/abs/1709.00560

[18] C. She, C. Yang, and T. Q. Quek, "Cross-layer optimization for ultra-reliable and low-latency radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, pp. 127–141, 01 2018.

[19] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[20] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[21] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[22] J. Hoydis, S. ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of Cellular Networks: How Many Antennas Do We Need?" *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[23] ——, "Massive MIMO: How many antennas do we need?" in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2011, pp. 545–550.

[24] 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.

[25] X. Li, S. Jin, X. Gao, and R. W. Heath, "Three-dimensional beamforming for large-scale FD-MIMO systems exploiting statistical channel state information," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8992–9005, Nov. 2016.

[26] "Study on 3D-channel model for elevation beamforming and FD-MIMO studies for LTE," *3GPP TR 36.873 V2.0.0*, Mar. 2014.

[27] E. Hossain and M. Hasan, "5G cellular: key enabling technologies and research challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 3, pp. 11–21, Jun. 2015.

[28] Y. Zhou, Z. M. Fadlullah, B. Mao, and N. Kato, "A deep-learning-based radio resource assignment technique for 5G ultra dense networks," *IEEE Netw.*, vol. 32, no. 6, pp. 28–34, Nov. 2018.

[29] R. Dong, W. Hardjawana, A. Li, Y. Li, and B. Vucetic, "Cooperative beamforming for multi-cell full dimensional massive MIMO networks," in *IEEE ICC, Shanghai, China*, May 2019.

[30] A. Li and C. Masouros, "Hybrid Analog-Digital Millimeter-Wave MU-MIMO Transmission with Virtual Path Selection," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 438–441, Feb. 2017.

[31] A. Li and C. Masouros, "Hybrid precoding and combining design for millimeter-wave multi-user MIMO based on SVD," in *2017 IEEE International Conference on Communications (ICC), Paris, France*, May 2017, pp. 1–6.

[32] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[33] "Mobile-edge computing—introductory technical white paper," *White Paper, ETSI, Sophia Antipolis, France*, Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

[34] "The internet of things: How the next evolution of the internet is changing everything," *White Paper, Cisco, San Jose, CA, USA*, Apr. 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[35] A. Somov and R. Giaffreda, "Powering iot devices: technologies and opportunities," *IEEE IoT Newslett.*, 11 2015.

[36] B. Shi, J. Yang, Z. Huang, and P. Hui, "Offloading guidelines for augmented reality applications on wearable devices," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 1271–1274. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806402

[37] S. Schiessl, J. Gross, and H. Al-Zubaidy, "Delay analysis for wireless fading channels with finite blocklength channel coding," in *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '15. New York, NY, USA: ACM, 2015, pp. 13–22.

[38] D. Li, J. Xu, X. Wang, and X. Tao, "Joint optimization for cell association and vertical downtilts adjustment in 3D MIMO enabled hetnets," in *IEEE WCNC*, Apr. 2018, pp. 1–6.

[39] S. Luo, R. Zhang, and T. J. Lim, "Downlink and uplink energy minimization through user association and beamforming in c-ran," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 494–508, Jan. 2015.

[40] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, "Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 221–232, Feb. 2018.

[41] M. Razaviyayn, M. Hong, and Z. Q. Luo, "Linear transceiver design for a mimo interfering broadcast channel achieving max-min fairness," in *ASILOMAR*, Nov. 2011, pp. 1309–1313.

[42] M. Sanjabi, M. Razaviyayn, and Z. Q. Luo, "Optimal joint base station assignment and downlink beamforming for heterogeneous networks," in *IEEE ICASSP*, Mar. 2012, pp. 2821–2824.

[43] T. Huynh, K. Kuroda, and M. Hasegawa, "User association for massive MIMO cellular networks with small cell wireless backhaul," in *WPMC*, Nov. 2016, pp. 8–13.

[44] J. Ma, S. Zhang, H. Li, N. Zhao, and V. C. M. Leung, "Base station selection for massive MIMO networks with two-stage precoding," *IEEE Wireless Commun. Lett.*, vol. 6, no. 5, pp. 598–601, Oct. 2017.

[45] M. Xie and T. Lok, "SINR balancing via base station association, beamforming, and power control in downlink multicell MISO systems," *IEEE Trans. Wireless Commun*, vol. 17, no. 3, pp. 1811–1821, Mar. 2018.

[46] H. D. Thang, L. Boukhatem, M. KaneW, and S. Martin, "Performance-cost trade-off of joint beamforming and user clustering in cloud radio access networks," in *IEEE PIMRC*, Oct. 2017, pp. 1–5.

[47] D. H. N. Nguyen, L. B. Le, and T. Le-Ngoc, "Optimal joint base station association and beamforming design for downlink transmission," in *IEEE ICC, London, UK*, Jun. 2015, pp. 4966–4971.

[48] I. Sohn, S. H. Lee, and J. G. Andrews, "Belief propagation for distributed downlink beamforming in cooperative MIMO cellular networks," *IEEE Trans. Wireless Commun*, vol. 10, no. 12, pp. 4140–4149, Dec. 2011.

[49] S. Rangan and R. Madan, "Belief propagation methods for intercell interference coordination in femtocell networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 631–640, Apr. 2012.

[50] R. Dong, W. Hardjawana, Y. Li, and B. Vucetic, "Dynamic sectoring with elevation optimization technique in 5G cellular networks," in *IEEE ICC Workshops, Kansas, United States*, May 2018, pp. 1–5.

[51] S. Hu, C. Xu, X. Wang, Y. Huang, and S. Zhang, "A stochastic admm approach to distributed coordinated multicell beamforming for renewables powered wireless cellular networks," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2018.

[52] C. K. Wen, J. C. Chen, K. K. Wong, and P. Ting, "Message passing algorithm for distributed downlink regularized zero-forcing beamforming with cooperative base stations," *IEEE Trans. Wireless Commun*, vol. 13, no. 5, pp. 2920–2930, May 2014.

[53] M. Feng and S. Mao, "Interference management and user association for nested array-based massive MIMO hetnets," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 454–466, Jan. 2018.

[54] Y. Xu and S. Mao, "User association in massive MIMO hetnets," *IEEE Syst. J.*, vol. 11, no. 1, pp. 7–19, Mar. 2017.

[55] K. Shen and W. Yu, "Distributed pricing-based user association for downlink heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1100–1113, Jun. 2014.

[56] D. Liu, L. Wang, Y. Chen, T. Zhang, K. K. Chai, and M. Elkashlan, "Distributed energy efficient fair user association in massive MIMO enabled hetnets," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1770–1773, Oct. 2015.

[57] M. Amjad, L. Musavian, and M. H. Rehmani, "Effective capacity in wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts, early access*, 2019.

[58] Y. Hu, M. Ozmen, M. C. Gursoy, and A. Schmeink, "Optimal power allocation for QoS-constrained downlink multi-user networks in the finite blocklength regime," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5827–5840, Sep. 2018.

[59] J. Zhang, X. Hu, Z. Ning, E. C. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[60] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous mobile-edge computation offloading: energy-efficient resource management," 2018. [Online]. Available: https://arxiv.org/abs/1801.03668

[61] J. Guo, C. Yang, and C.-L. I, "Exploiting future radio resources with end-to-end prediction by deep learning," *IEEE Access*, vol. 6, pp. 75 729–75 747, Nov. 2018.

[62] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359 – 366, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0893608089900208

[63] P. Riley, "Three pitfalls to avoid in machine learning," 2019.

[64] X. Ge, K. Huang, C. Wang, X. Hong, and X. Yang, "Capacity analysis of a multi-cell multi-antenna cooperative cellular network with co-channel interference," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3298–3309, Oct. 2011.

[65] Y. Cao, N. Zhao, F. R. Yu, M. Jin, Y. Chen, J. Tang, and V. C. M. Leung, "Optimization or alignment: Secure primary transmission assisted by secondary networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 905–917, Apr. 2018.

[66] J. Ma, S. Zhang, H. Li, N. Zhao, and V. C. M. Leung, "Interference-alignment and soft-space-reuse based cooperative transmission for multi-cell massive MIMO networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1907–1922, Mar. 2018.

[67] J. Choi, "Distributed beamforming for macro diversity and power control with large arrays in spatial correlated channels," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 1871–1881, Apr. 2015.

[68] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K. K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1018–1044, Secondquarter 2016.

[69] X. Li, S. Jin, X. Gao, and R. W. Heath, "Three-dimensional beamforming for large-scale FD-MIMO systems exploiting statistical channel state information," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8992–9005, Nov. 2016.

[70] A. Adhikary, J. Nam, J. Y. Ahn, and G. Caire, "Joint spatial division and multiplexing the large-scale array regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6441–6463, Oct. 2013.

[71] R. M. Gray, "Toeplitz and circulant matrices: A review," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006. [Online]. Available: http://dx.doi.org/10.1561/0100000006

[72] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[73] C. She, Y. Duan, G. Zhao, T. Q. S. Quek, Y. Li, and B. Vucetic, "Cross-layer Design for Mission-Critical IoT in Mobile Edge Computing Systems," *CoRR*, vol. abs/1907.05210, 2019. [Online]. Available: http://arxiv.org/abs/1907.05210

[74] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Leveraging cloudlets for immersive collaborative applications," *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 30–38, Oct. 2013.

[75] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sept. 2013.

[76] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing." *HotCloud*, vol. 10, 2010.

[77] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[78] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[79] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[80] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[81] W. Hardjawana, N. I. A. Apandi, and B. Vucetic, "Parallel optimization framework for cloud-based small cell networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7286–7298, Nov. 2016.

[82] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning radio resource management in 5g networks: Framework, opportunities and challenges," *IEEE Commun. Mag.*, vol. 56, 09 2018.

[83] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online offloading in wireless powered mobile-edge computing networks," Sept. 2018. [Online]. Available: http://arxiv.org/abs/1808.01977

[84] R. Li, Z. Zhao, Q. Sun, C. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.

[85] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, "Deep reinforcement learning based resource allocation in low latency edge computing networks," in *2018 15th ISWCS*, Aug. 2018, pp. 1–5.

[86] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *ICANN*, 2018.

[87] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[88] D. Liu, Y. Chen, K. K. Chai, T. Zhang, and M. Elkashlan, "Two-dimensional optimization on user association and green energy allocation for hetnets with hybrid energy sources," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4111–4124, Nov. 2015.

[89] X. Ge, J. Ye, Y. Yang, and Q. Li, "User mobility evaluation for 5G small cell networks based on individual mobility model," *IEEE J. Sel. Areas Commun*, vol. 34, no. 3, pp. 528–541, Mar. 2016.

[90] D. Li, J. Xu, X. Wang, and X. Tao, "Joint optimization for cell association and vertical downtilts adjustment in 3D MIMO enabled hetnets," in *IEEE WCNC*, Apr. 2018, pp. 1–6.

[91] S. Luo, R. Zhang, and T. J. Lim, "Downlink and uplink energy minimization through user association and beamforming in c-ran," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 494–508, Jan. 2015.

[92] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, "Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 221–232, Feb. 2018.

[93] C. Zhang, Y. Huang, Y. Jing, S. Jin, and L. Yang, "Sum-rate analysis for massive MIMO downlink with joint statistical beamforming and user scheduling," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2181–2194, Apr. 2017.

[94] T. M. Berhane, W. X. Meng, L. Chen, G. D. Jobir, and C. Li, "SLNR-based precoding for single cell full-duplex MU-MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7877–7887, Sep. 2017.

[95] K. Mullen, "A note on the ratio of two independent random variables," in *Amer. Stat.*, vol. 21, no. 3, Jun. 1967, pp. 30–31.

[96] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[97] D. Bickson, "Gaussian belief propagation: Theory and aplication," *arXiv preprint arXiv: 0811.2518*, 2008.

[98] D. Bickson, D. Dolev, O. Shental, P. H. Siegel, and J. K. Wolf, "Gaussian belief propagation based multiuser detection," in *2008 IEEE International Symposium on Information Theory*, July 2008, pp. 1878–1882.

[99] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, no. 10, pp. 2173–2200, Oct. 2001.

[100] P. Luong, F. Gagnon, C. Despins, and L. Tran, "Joint virtual computing and radio resource allocation in limited fronthaul green C-RANs," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2602–2617, Apr. 2018.

[101] P. Luong, C. Despins, F. Gagnon, and L. Tran, "Designing green C-RAN with limited fronthaul via mixed-integer second order cone programming," in *IEEE ICC, Paris, France*, May 2017, pp. 1–6.

[102] S. Bradley, A. Hax, and T. Magnanti, *Applied Mathematical Programming*. Addison-Wesley Publishing Company, 1977. [Online]. Available: https://books.google.com.au/books?id=MSWdWv3Gn5cC

[103] A. Li, C. Masouros, F. Liu, and A. L. Swindlehurst, "Massive mimo 1-bit dac transmission: A low-complexity symbol scaling approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7559–7575, Nov. 2018.

[104] C. She, C. Yang, and T. Q. S. Quek, "Cross-layer optimization for ultra-reliable and low-latency radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 127–141, Jan. 2018.

[105] Y. Hu, M. Ozmen, M. C. Gursoy, and A. Schmeink, "Optimal power allocation for qos-constrained downlink multi-user networks in the finite blocklength regime," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5827–5840, Sep. 2018.

[106] C. She, C. Yang, and T. Q. S. Quek, "Radio resource management for ultra-reliable and low-latency communications," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 72–78, Jun. 2017.

[107] K. Cheng, Y. Teng, W. Sun, A. Liu, and X. Wang, "Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems," in *2018 IEEE ICC, Kansas, United States*, May 2018.

[108] R. Dong, A. Li, W. Hardjawana, Y. Li, X. Ge, and B. Vucetic, "Joint beamforming and user association scheme for full-dimension massive MIMO networks," *IEEE Trans. Veh. Technol.*, 2019.

[109] N. C. Luong, D. T. Hoang, S. Gong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *submitted to IEEE Commun. Surveys Tuts.*, 2018. [Online]. Available: https://arxiv.org/pdf/1810.07862.pdf

[110] M. Wise, "APM: Driving value with the digital twin," in *GE Digital*, 2017.

[111] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modelling and latency analysis," *IEEE Trans. on Wireless Commun.*, vol. 17, no. 8, pp. 5225–5240, Aug. 2018.

[112] J. Zhou, X. Zhang, and W. Wang, "Joint resource allocation and user association for heterogeneous services in multi-access edge computing networks," *IEEE Access*, vol. 7, pp. 12 272–12 282, Jan. 2019.

[113] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance optimization in mobile-edge computing via deep reinforcement learning," 2018. [Online]. Available: https://arxiv.org/abs/1804.00514v1

[114] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-Based Computation Offloading for IoT Devices With Energy Harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[115] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sept. 2017.

[116] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Globecom, Singapore*, 2017.

[117] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12 825–12 837, 2018.

[118] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[119] 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.

[120] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4232–4264, Jul. 2014.

[121] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[122] Y. Duan, C. She, G. Zhao, and T. Q. S. Quek, "Delay analysis and computing offloading of urllc in mobile edge computing systems," in *Proc. WCSP*, 2018.

[123] A. Gravey, J.-R. Louvion, and P. Boyer, "On the geo/d/1 and geo/d/1/n queues," *Perform. Eval.*, vol. 11, no. 2, pp. 117–125, Jul. 1990. [Online]. Available: http://dx.doi.org/10.1016/0166-5316(90)90018-E

[124] S. Schiessl, J. Gross, and H. Al-Zubaidy, "Delay analysis for wireless fading channels with finite blocklength channel coding," in *Proc. ACM MSWiM*, 2015.

[125] C. Sun, C. She, C. Yang, T. Q. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 402–415, Jan. 2019.

[126] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[127] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1412.htmlKingmaB14

[128] 3GPP, LTE ETSI TR 36.931 v9.0.0, "Evolved universal terrestrial radio access." May 2011.

[129] Z. Hou, C. She, Y. Li, T. Q. S. Quek, and B. Vucetic, "Burstiness aware bandwidth reservation for ultra-reliable and low-latency communications (URLLC) in tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 1–10, Nov. 2018.

[130] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, May 1993.

[131] 3GPP, "Study on energy efficiency aspects of 3GPP standards; services and system aspects (release 15)." TR 21.866 V15.0.0, Jun. 2017.

[132] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H. V. Poor, "Cluster content caching: An energy-efficient approach to improve quality of service in cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1207–1221, May 2016.

[133] 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.

[134] E. C. Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Kténas, N. Cassiau, and C. Dehos, "6G: The next frontier," *arXiv preprint arXiv:1901.03239*, 2019.

[135] S. Xu, T.-H. Chang, S.-C. Lin, C. Shen, and G. Zhu, "Energy-efficient packet scheduling with finite blocklength codes: Convexity analysis and efficient algorithms," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5527–5540, Aug. 2016.

[136] C. Sun, C. She, C. Yang, T. Q. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and

low-latency communications," *IEEE Trans. on Wireless Commun.*, vol. 18, no. 1, pp. 402–415, Jan. 2019.

[137] C. Ye, M. C. Gursoy, and S. Velipasalar, "Power control for wireless VBR video streaming: From optimization to reinforcement learning," *IEEE Trans. Commun., early access*, 2019.

[138] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[139] 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.

[140] A. Zappone, M. Debbah, and Z. Altman, "Online energy-efficient power control in wireless networks by deep neural networks," in *IEEE SPAWC*, 2018.

[141] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks towards wireless systems optimization," *arXiv preprint arXiv:1808.01672*, 2018.

[142] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *arXiv preprint arXiv:1807.10025*, 2018.

[143] L. Lei, Y. Yuan, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Learning-based resource allocation: Efficient content delivery enabled by convolutional neural network," in *IEEE SPAWC*, 2019.

[144] Z. Xu, C. Yang, G. Y. Li *et al.*, "Energy-efficient configuration of spatial and frequency resources in MIMO-OFDMA systems," *IEEE Trans. Commun.*, vol. 28, no. 2, pp. 564 – 575, Feb. 2013.

[145] C. Xiong, G. Y. Li, Y. Liu *et al.*, "Energy-efficient design for downlink OFDMA with delay-sensitive traffic," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 3085–3095, Jun. 2013.

[146] W. Yu, L. Musavian, and Q. Ni, "Statistical delay QoS driven energy efficiency and effective capacity tradeoff for uplink multi-user multi-carrier systems," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3494–3508, Aug. 2017.

[147] Y. Zhu, Y. Hu, A. Schmeink, and J. Gross, "Energy minimization of mobile edge computing networks with finite retransmissions in the finite blocklength regime," in *IEEE SPAWC*, 2019.

[148] W. Lee, M. Kim, and D. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276–1279, Jun. 2018.

[149] A. Zappone, E. Björnson, L. Sanguinetti, and E. Jorswieck, "Globally optimal energy-efficient power control and receiver design in wireless networks," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2844–2859, Jun. 2017.

[150] M. Eisen, C. Zhang, L. F. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775–2790, May 2019.

[151] H. Lee, S. H. Lee, and T. Q. S. Quek, "Deep learning for distributed optimization: Applications to wireless resource management," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2251–2266, Oct. 2019.

[152] M. Chen, W. Saad, C. Yin, and M. Debbah, "Data Correlation-Aware Resource Management in Wireless Virtual Reality (VR): An Echo State Transfer Learning Approach," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4267–4280, Jun. 2019.

[153] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[154] Q. Yao, H. Yang, A. Yu, and J. Zhang, "Transductive transfer learning-based spectrum optimization for resource reservation in seven-core elastic optical networks," *J. Lightw. Technol.*, pp. 1–1, 2019.

[155] I. Chaturvedi, Y. Ong, and R. Arumugam, "Deep transfer learning for classification of time-delayed gaussian networks," *Signal Processing*, vol. 110, pp. 250 – 262, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168414004198

[156] L. Liu, "Energy-efficient power allocation for delay-sensitive traffic over wireless systems," in *IEEE ICC Workshop, Ottawa, Canada*, 2012.

[157] C. Chang and J. A. Thomas, "Effective bandwidth in high-speed digital networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1091–1100, Aug. 1995.

[158] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, Jul. 2003.

[159] F. P. Kelly, "Notes on effective bandwidths," *Stochastic Networks: Theory and Applications*, London, U.K.: Oxford Univ. Press. 1996.

[160] M. Ozmen and M. C. Gursoy, "Wireless throughput and energy efficiency with random arrivals and statistical queuing constraints," *IEEE Trans. Inf. Theory*, vol. 62, no. 3, pp. 1375–1395, Mar. 2016.

[161] M. Condoluci, T. Mahmoodi, E. Steinbach, and M. Dohler, "Soft resource reservation for low-delayed teleoperation over mobile networks," *IEEE Access*, vol. 5, pp. 10 445–10 455, May 2017.

[162] B. Debaillie, C. Desset, and F. Louagie, "A flexible and future-proof power model for cellular base stations," in *2015 IEEE 81st VTC Spring, Glasgow, UK*, May 2015, pp. 1–7.

[163] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[164] M. Conforti, G. Cornuejols, and G. Zambelli, *Integer Programming (Graduate texts in mathematics)*. Springer Heidelberg, 2014.

[165] H. He, H. Daume III, and J. M. Eisner, "Learning to search in branch and bound algorithms," in *Proc. Adv. Neural Inform. Process. Syst.*, Dec. 2014, pp. 3293–3301.

[166] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*. Cambridge, UK: Cambridge Univ. Press, 1998, revised, oct 2012. [Online]. Available: http://leon.bottou.org/papers/bottou-98x

[167] F. Rusek, D. Persson, B. K. Lau *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40 – 60, Jan. 2013.

[168] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Transfer learning for mixed-integer resource allocation problems in wireless networks," in *Proc. IEEE ICC, Shanghai, China*, 2019.