Power allocation in cell-free massive MIMO

Power allocation in cell-free massive MIMO: Using deep learning methods

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op dinsdag 31 januari 2022 om 11:00 uur

door

Yu Zhao

geboren te Shanxi, China

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. P.G.M. Baltus
1e promotor:	prof.dr.ir. S.M. Heemstra de Groot
co-promotor:	prof.dr.ir. I.G.M.M Niemegeers
leden:	prof.dr. K. Moessner (Technische Unviersitat Chemnitz)
	prof.dr. D. Dahlhaus (University of Kassel)
	prof.dr.ir. F.M.J. Willems
	dr.ir. Ulf Johansen
	dr. A. Lo (Nokia)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

This Ph.D. thesis has been approved by a committe with the following members:

voorzitter:	prof.dr.ir. P.G.M. Baltus
1e promotor:	prof.dr.ir. S.M. Heemstra de Groot
co-promotor:	prof.dr.ir. I.G.M.M Niemegeers
leden:	prof.dr. K. Moessner (Technische Unviersitat Chemnitz)
	prof.dr. D. Dahlhaus (University of Kassel)
	prof.dr.ir. F.M.J. Willems
	dr.ir. Ulf Johansen
	dr. A. Lo (Nokia)

A catalogue record is available from the Eindhoven University of Technology Library.

Title: Power allocation in cell-free massive MIMO: Using deep learning methods.

Author: Yu Zhao.

Eindhoven University of Technology, 2021.

ISBN: 978-90-386-5444-7

NUR: 959

Keywords: Cell-free massive MIMO, Deep learning, Power allocation, 5G and beyond network, Beamforming.

“Some birds are not meant to be caged,
their feathers are just too bright."

Abstract

Since 2016, 5G new radio-access technology known as New Radio has been standardized by the 3rd Generation Partnership Project (3GPP) to satisfy the service demands of a wide range of applications, such as Internet of Things (IoT), autonomous driving, and Virtual Reality (VR); and the first set of 5G New Radio specifications is released in 3GPP Release 15. Despite the advances already made, a number of issues remain, in particular regarding the demand for higher data rates, lower latencies, and improved robustness.

Two key elements that play an essential role in addressing these challenges are the use of mmWave spectrum and massive multiple-input massive multiple-output (MIMO) technology. On the one hand, much more bandwidth is available in the mmWave spectrum than in the sub-6 GHz spectrum which results in higher data rates and, on the other hand, massive MIMO enables a high degree of space division multiplexing, which increases the network capacity for a given spectrum. Recently, cell-free (CF) massive MIMO, has been proposed to further improve the spectral efficiency (SE) and hence the network capacity of the system. In CF massive MIMO, a number of access points (APs) with multiple antennas are spread over the coverage area and are connected to a central controller (CC) via a fronthaul. The APs jointly and coherently provide service to the user equipments (UEs). Compared to massive MIMO, CF massive MIMO is more robust against shadow fading and has a lower average distance between the transmitters and the receivers.

Because of the geographical spreading of antennas, local signal processing is performed at each AP, e.g., by using conjugate beamforming (CB), thereby avoiding the exchange of channel state information (CSI) between the CC and the APs. However, by doing this, the interference between different UEs cannot be suppressed, as would be the case when using zero-forcing beamforming (ZFB). Under these circumstances, the optimization of power allocation becomes non-convex and hence computationally hard.

The use of mmWave brings new challenges for the power allocation in CF massive MIMO. Due to the hardware constraints of present technologies, full-digital beamforming, which requires each antenna to have its own radio frequency (RF) chain and which is common in the microwave domain, is hard to implement. Therefore, hybrid beamforming, where the number of RF chains is less than the number of antennas, is preferred. This degrades the network performance in terms of SE with respect to what could be achieved with full-digital beamforming. In addition, the channel fades faster in the mmWave spectrum than in the sub-6 GHz spectrum, imposing more restrictive time constraints on performing the power allocation.

In this thesis, we address three challenging power allocation problems: Problem 1, max-min power allocation in CF sub-6 GHz massive MIMO; Problem 2, max-sum SE power allocation in CF sub-6 GHz massive MIMO; Problem 3, max-sum SE power allocation in CF mmWave massive MIMO.

For Problem 1, we propose to use deep supervised learning (DSL). The key idea behind the DSL-based method is to use a deep neural network (DNN) to approximate a known but computationally complex algorithm. Specifically, a labeled dataset is generated by the bisection algorithm, and then the DNN is trained by this dataset to determine the mapping from the CSI to the power allocation. To evaluate the performance of the proposed DSL-based method, we use Monte Carlo simulations to compare with the bisection algorithm. Numerical results show that, after training, the DNN approximates the results of the bisection algorithm very closely. In addition, we show that the DSL-based method has a much less execution time than the bisection algorithm.

For Problem 2, we propose to use deep reinforcement learning (DRL). For DRL, the optimal power allocation is obtained by agents, which interact with the environment to maximize a cumulative reward, defined as the sum of all rewards received by the agents so far. Two types of DRL are proposed: a value-based approach deep Q-network (DQN) and a policy-based approach, known as deep deterministic policy gradient (DDPG). We evaluate the performance of the proposed DQN-based method and DDPG-based method, by using Monte Carlo simulations assuming a 3GPP indoor mixed office scenario with mobile UEs. We found that both the DQN- and DDPG-based methods outperform the existing algorithms in terms of sum SE, which is higher than what is achieved by the well-adopted weighted minimum-mean-square-error (WMMSE) algorithm. Moreover, the execution time is substantially less.

For Problem 3, we use both the DQN method and DDPG method. (1) Since hybrid beamforming degrades the performance in terms of SE compared with full-digital beamforming, we study this degradation. By using Monte Carlo simulations, we compare the sum SE achieved by different numbers of RF chains in the APs, assuming a 3GPP indoor mixed office scenario. We found that, when the number

of RF chains is twice the number of receivers, the degradation becomes negligible. In other words, hybrid beamforming achieves almost the same performance as full-digital beamforming. (2) Then we use the DQN- and the DDPG-based methods, for power allocation and evaluate their performance against the WMMSE algorithm. The results of the Monte Carlo simulations show that the sum SE achieved by both methods is higher. Moreover, the execution time is much less.

Finally, based on the insights we gained and on the limitations of our work, we point out future research to be conducted, e.g., different scenarios, traffic and mobility models and implementation issues.

Contents

Abbreviations

3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
6G	Sixth Generation
AWGN	Additive white Gaussian noise
AI	Artificial intelligence
AIoT	Artificial intelligence of things
AoA	Angle of arrival
AoD	Angle of departure
AP	Access point
BS	Base station
CB	Conjugate beamforming
CC	Central controller
CF	Cell-free
CoMP	Cooperative multipoint joint processing
CNN	Convolutional neural network
CSI	Channel state information
DDPG	Deep deterministic policy gradient
DNN	Deep neural network
DL	Deep learning
DQN	Deep Q-network
DRL	Deep reinforcement learning
DSL	Deep supervised learning
FCHB	Fully-connected hybrid beamforming
IMT	International Mobile Telecommunications
IoT	Internet of things

LOS	Line-of-sight
LSTM	Long short-term memory
MIMO	Multiple-input multiple-output
MSE	Mean square error
MMSE	Minimum mean square error
MRB	Maximum-ratio beamforming
MU-MIMO	Multi-user multiple-input multiple-output
NCB	Normalized conjugate beamforming
NLOS	Non-line-of-sight
NOMA	Non-orthogonal multiple access
NP	Nondeterministic polynomial
NPU	Neural processing units
NYU	New York University
PCHB	Partially-connected hybrid beamforming
QoS	Quality-of-service
ReLu	Rectified linear units
ResDense	Residual dense
RF	Radio-frequency
RNN	Recurrent neural network
RSS	Received signal strength
RZFB	Regularization zero-forcing beamforming
SE	Spectral efficiency
SINR	Signal-to-interference-plus-noise-ratio
SNR	Signal-to-noise-ratio
S-V	Saleh-Valenzuela
SVD	Singular value decomposition
TDD	Time-division duplex
UatF	Use-and-then-forget
UE	User equipment
UPA	Uniform planar array
WMMSE	Weighted minimum-mean-square-error
ZFB	Zero-forcing beamforming

Relevant notations

a	Action of agent
B_c	Coherence bandwidth
C	Training interval
d	Distance between AP and UE
D	Batch size
e	Experience of agent
f	Carrier frequency
$\mathbf{g}$	Channel vector
$\mathbf{h}$	Small-scale fading vector
h_{AP}	Height of AP
h_u	Height of UE
$\mathbf{I}$	All-ones vector
I_B	Number of bisection algorithm loops
I_F	Number of feasibility algorithm loops
J	Number of cells
K	Number of UEs
L_p	Number of propagation paths
lr	Learning rate
M	Number of antennas per AP
N	Number of APs
N_{RF}	Number of RF chains per AP
p	Normalized downlink transmission power
p_l	Transmission power limit
p_p	Normalized uplink pilot power
$\mathbf{q}$	Intended signal vector
Q	Number of filters in Chapter 2; Action value of agent in Chapter 3 and 4

r	Reward in DRL
s	State of environment
t	Time slot
u	Parameter of WMMSE algorithm
v	Parameter of WMMSE algorithm
v_l	Velocity of UEs
w	Parameter of WMMSE algorithm
$\mathbf{w}$	Noise matrix
$\mathbf{x}$	Transmitted signal of AP
$\mathbf{y}$	Receive signal
z	Shadow fading coefficient
β	Large-scale fading
γ	Variance of estimated channel
λ	Wavelength
μ	Number of neurons per layer in DNN
ν	Number of layers in DNN
τ_c	Coherence time
τ_l	Time scale on large-scale fading
τ_p	Time duration for channel estimation
$\boldsymbol{\psi}$	Time sequence

Chapter 1

Introduction

With the fast development of the internet-of-things (IoT) over the past decade, the number of connected devices such as mobile phones, laptops, tablets, etc., has increased by 10% per year [1]. Fig.1.1 shows the historical and forecast data of the growth of the number of connected devices from 2019 to 2026. The ever-increasing growth of connected devices demands higher network capacity, and new applications pose more stringent requirements on the quality-of-service (QoS), e.g., on latency and robustness [2]. Triggered by these demands, the first version of the fifth generation (5G) mobile systems has been standardized by the 3rd Generation Partnership Project (3GPP), with initial deployments occurring in 2018 [3]. Even though 5G is still in its infancy, the world has so far seen a substantial expansion of

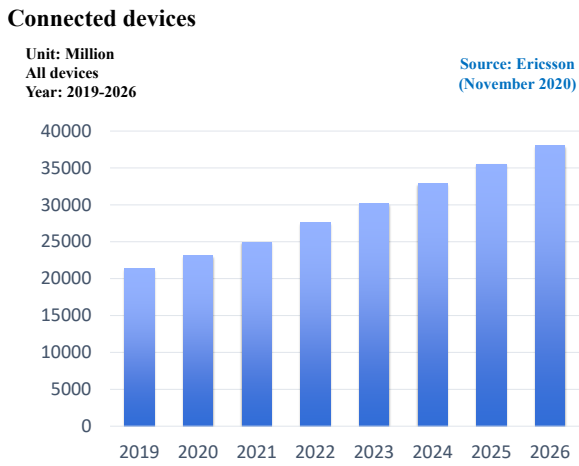


Figure 1.1: Global growth of connected devices [1]

5G with networks providing excellent capacity and services. Meanwhile, research on the sixth generation (6G), which is expected to have its first trials around 2030, is now on its way.

Visions for 6G have been formulated by organizations, companies and individual researchers, and a series of advanced research planning activities have begun [4–7]. In the 6G vision and requirements suggested in [4], special attention was paid to the battery lifetime of mobile device and service classes. [5] provided a human centric vision of 6G network rather than the machine-, application- or data-centric vision, which has been prevailing until now. Therefore, security, user experience and interaction will be key considerations of 6G. In [6], it was pointed out that 6G should be a ubiquitous and integrated network with broad and deep coverage, including airspace, land, sea, etc. Several distinctive scenarios for 6G network were predicted in [7], e.g., unmanned aerial vehicle networks, teleoperated driving and tactile internet. Further, it is foreseen that the same level of reliability as in wired communications will be offered by future wireless networks.

According to the International Mobile Telecommunications (IMT)-2030 promotion group, 6G is expected to cause a shift from IoT to artificial-intelligence-of-things (AIoT) around 2030 [8]. AIoT is where artificial intelligence (AI) and IoT meet, bringing intelligence to the edge of the communication network and into devices such as sensors, cameras, and mobile devices. The enormous amount of data generated by devices, machines, and new time-critical applications using edge intelligence, require significantly higher data rates and much lower latency than can be offered by 5G [8]. Multiple antenna technology, particularly massive multiple-input multiple-output (MIMO) [9], is an important tool to address these challenges. In cellular systems, massive MIMO is a technology that utilizes a very large number of antennas at the base station (BS) [10]. It improves the network capacity for a given amount of spectrum by simultaneously transmitting multiple data streams. Each stream can be beamformed to serve a different user equipment (UE). This achieves a high degree of space division multiplexing. A new way of using massive MIMO is to geographically distribute the antennas over several access points (APs) in the coverage area, each containing a smaller cluster of antennas. In principle, all APs serve, simultaneously and in a cooperative way, each UE in the coverage area. The principle of a cellular network, where each UE is served by a particular BS or AP is abandoned. For this reason, the concept is also known as cell-free (CF) massive MIMO [11]. However, the coherent operation of all antennas, which is essential in massive MIMO, is kept. In this thesis, the term AP is exclusively used in the context of CF massive MIMO rather than BS; while for massive MIMO, we still use the term BS.

Because the antennas are geographically spread, signal processing is done locally in the APs, e.g., by performing conjugate beamforming (CB). This, we will discuss later, makes the downlink power allocation a non-convex problem and hence

computationally hard. Downlink power allocation is an essential function in CF massive MIMO, it controls the inter-UE interference. A larger value of allocated power improves the received signal strength to the intended UE, however leads to a higher interference to the other UEs. The uplink power allocation is another important issue in CF massive MIMO, which we do not address in this thesis. It determines the network capacity in the uplink. Unless otherwise stated, power allocation in this thesis refers to the downlink.

Although many effective methods have been developed for power allocation in wireless communication, their implementation in real systems faces serious obstacles. In particular, the execution time is problematic. Since the state of the channel evolves in time, the power allocation should stay in tune with this state. The present methods, e.g., the well-adopted weighted minimum mean square error (WMMSE) algorithm [12], typically find a near-optimal solution in an iterative manner at the cost of considerable time. This potentially compromises the timeliness of the power allocation. In this thesis, we address this problem by exploring the use of data-driven machine learning methods since they can achieve near-optimal performance with substantially lower execution time. Deep learning (DL) [13] is one such method. It only requires a multilayer deep neural network (DNN) to solve a complicated optimization problem. The DL-based power allocation method for CF massive MIMO is the main subject of this thesis.

In the rest of this chapter, we first introduce massive MIMO and CF massive MIMO. Then we introduce the problems that we are addressing in this thesis, followed by the proposed solutions and methods, i.e., the contributions of this thesis. Lastly, we present the outline of the thesis.

1.1 Background

1.1.1 Massive MIMO

MIMO is a multiple antenna technology where two devices with multiple antennas communicate with each other. The focus of MIMO technology, in recent years, has been shifted to multi-user MIMO (MU-MIMO), where typically a BS with multiple antennas simultaneously serves a set of single-antenna users at the same time-frequency block [14]. MU-MIMO is regarded as a more practical system where expensive equipment is only needed at the BS and the UEs can be relatively cheap single-antenna devices. As a result, MU-MIMO has become an integral part of communications standards such as Wi-Fi (IEEE 802.11n, IEEE 802.11ac) and 802.16 (WiMAX) [10], [15]. Recently, massive MIMO, the new generation of MU-massive MIMO, has been introduced as a technology to further improve the network capacity [16]. The pioneering work [17], defined massive MIMO as a system using large antenna arrays in cellular networks. Later, in [9], the term massive MIMO was

used to denote a system that uses antenna arrays with a few hundred antennas, simultaneously serving tens of UEs using the same time-frequency block. In this thesis, we adopt the canonical definition of massive MIMO proposed in [18]:

Definition 1 (Definition of massive MIMO). A massive MIMO is a multicarrier cellular network with J cells that operate according to a synchronous time-division duplex (TDD) protocol. BS j is equipped with $N_j >> 1$ antennas, to achieve *channel hardening*. BS j communicates with K_j single-antenna UEs simultaneously in each *time-frequency block*, with antenna-UE ratio $N_j/K_j > 1$. Each BS operates individually and processes its signals using linear transmit *beamforming*[1].

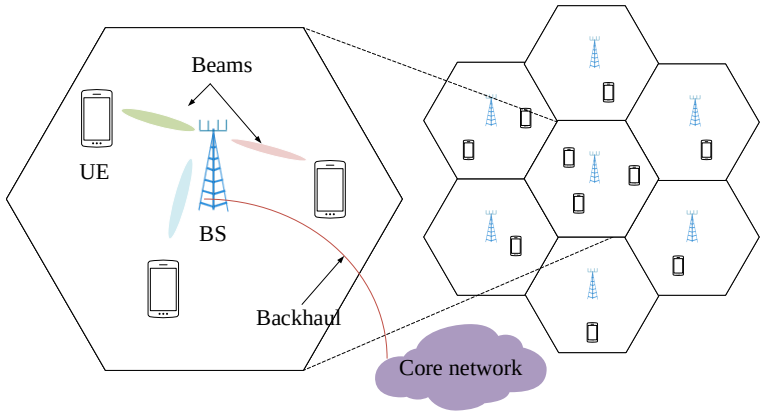


Figure 1.2: An example of massive MIMO, where the coverage area is divided into several cells. Each cell has one dedicated BS. There is no cooperation with different BSs.

Fig.1.2 shows an example of massive MIMO. *Beamforming* is a signal processing technique used in multiple antenna system for directional signal transmission [19]. Using beamforming, the BS forms several independent beams to serve different UEs, achieving spatial multiplexing. More details about beamforming are discussed in later chapters.

Time-frequency block and *channel hardening* are introduced below.

Time-frequency block

The propagation channels vary over time and frequency. In massive MIMO, the radio resources are divided into time-frequency blocks [18], during which the channels can be regarded as constant and frequency flat. A constant channel means

[1] In [18], the term 'beamforming' is the same as 'precoding'. We use the term 'beamforming' throughout the thesis.

that both large-scale fading and small-scale fading are constant. Frequency flat refers to a type of small-scale fading where all frequency signal components experience the same magnitude of fading. The time duration of the time-frequency block is defined as the coherence time τ_c, the frequency span of the time-frequency block is defined as the coherence bandwidth B_c. In massive MIMO, the time-frequency blocks are operated in TDD mode where the whole bandwidth is used for both downlink and uplink transmission but separated in time. Fig.1.3 shows an illustration of the time-frequency blocks.

In Fig.1.3, each time-frequency block is divided into two parts, namely the part for uplink channel estimation and the part for downlink data transmission. To perform the beamforming, the APs require to obtain the channel state information (CSI), which is estimated via the uplink pilots transmitted by the UEs. A common assumption in massive MIMO is that the uplink and downlink channels are reciprocal, which means that uplink and downlink channels are identical within a time-frequency block [20]. Therefore, the estimated CSI by the uplink can then be directly used for downlink beamforming. Then the APs transmit the payload data to the UEs in the downlink data transmission part.

Channel hardening

The term channel hardening is used to describe a fading channel that behaves almost deterministically due to spatial diversity [21]. Channel hardening is a direct consequence of the law of large numbers, i.e., the instantaneous channel gain tends to its mean value as the number of antennas increases. Let $\mathbf{g} \in C^{1\times M}$ represent the channel vector between an arbitrary BS with M antennas and an arbitrary UE, then

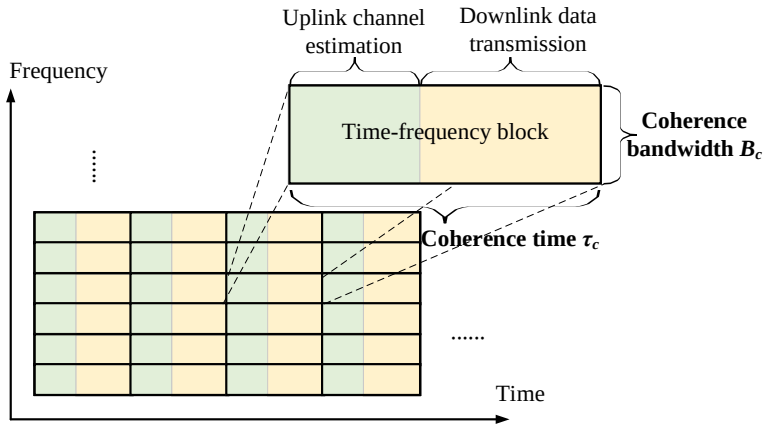


Figure 1.3: The illustration of the time-frequency blocks in TDD mode.

the channel hardening property is formalized as follows:

$$\lim_{M\to\infty} \frac{||\mathbf{g}||^2}{\mathbb{E}\{||\mathbf{g}||^2\}} = 1 \tag{1.1}$$

(1.1) is interpreted as $||\mathbf{g}||^2$ being close to the expected value $\mathbb{E}\{||\mathbf{g}||^2\}$ if the BS is equipped with a sufficiently large number of antennas. This important property demonstrates the disappearance of the small-scale fading effect and allows massive MIMO to use the average channel gains, i.e., deterministic numbers, rather than the corresponding instantaneous values when computing the spectral efficiency (SE) of the UEs and making power allocation decisions [18] [22].

Different time scales

If one has the benefit of channel hardening, power allocation can be done on a long-time scale, commensurate with τ_l, where τ_l is defined as the time during which the effect of small-scale fading can be negligible. However, the beamforming used during the data transmission phase, which is based on the CSI values collected by uplink channel estimation, will be valid for approximately the coherence time τ_c. This means that the power allocation should be adapted every τ_l and the beamforming should be adapted every τ_c. To clarify the different time scales, we use τ_l as the period for power allocation and the coherence time τ_c as the period for beamforming. τ_p is the time used for uplink channel estimation. Fig.1.4 shows the different time scales.

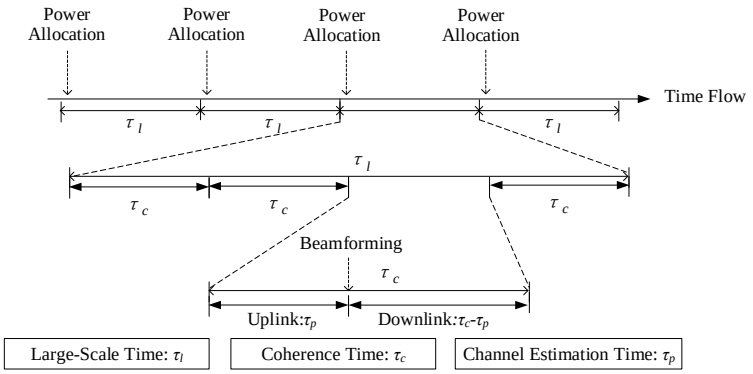


Figure 1.4: Time scales for power allocation, beamforming and channel estimation.

1.1.2 CF massive MIMO

The network capacity of massive MIMO is limited by the uncoordinated inter-cell interference [22]. The co-processing for inter-cell interference suppression in massive MIMO was proposed in [11]. This distributed architecture of massive MIMO is known as CF massive MIMO. In Fig.1.5 we show an example of CF massive MIMO, where the APs are spread over the coverage area. Each AP contains a small number of antennas. The APs cooperatively serve the UEs.

There are three main differences between massive MIMO and CF massive MIMO:

(1) In CF massive MIMO, the APs are connected to a central controller (CC) via a fronthaul network, enabling the coordinated data transmission, while in massive MIMO the BSs do not coordinate their transmission.

(2) In CF massive MIMO, the UEs are served by all APs in the whole system, using cooperative multipoint joint processing (CoMP). In massive MIMO, a BS only serves the UEs within its own cell.

(3) In CF massive MIMO, the channel estimation and the beamforming are performed locally at the APs to avoid exchanging CSI within each coherence time and the power allocation is performed centrally at the CC. In massive MIMO, the channel estimation, beamforming, and power allocation are all performed at the BSs.

CF massive MIMO has the benefits of massive MIMO [24], in particular, channel hardening and favorable propagation, provided the APs are equipped with a

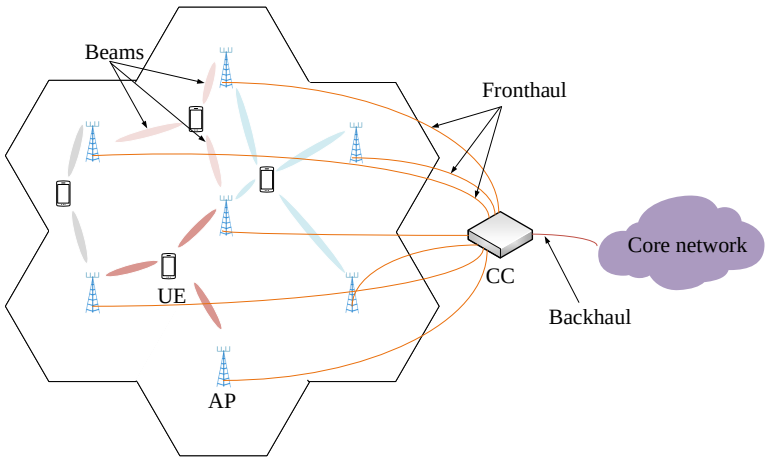


Figure 1.5: An example of CF massive MIMO.

large enough number of antennas [22]. Favorable propagation means that the UEs' channel vectors are almost orthogonal. These benefits simplify the closed-form expression of the SE. The details are discussed later in Chapter 2. In addition, CF massive MIMO can effectively mitigate the inter-cell interference that UEs at the edge of a cell might experience [23].

In recent literature, e.g., [24–27], CF massive MIMO has also been designated as distributed massive MIMO to distinguish it from massive MIMO in a single cell, where all functions are centralized in the BS (Fig.1.6).

There is no standard definition of CF massive MIMO in the literature. In this thesis, we use **Definition 2**, which is based on [11] and [28].

Definition 2 (Definition of CF massive MIMO). In a CF massive MIMO, N APs are geographically distributed over the coverage area, each AP is equipped with M antennas. The APs are jointly serving K single-antenna UEs in TDD mode and on each time-frequency block. The APs are connected via fronthaul links to a CC, which performs AP coordination.

Compared to massive MIMO, the merits of CF massive MIMO are:

(1) A smaller physical size of the APs, which facilitates their deployment.

(2) A higher robustness against shadow fading due to the geographic distribution of the APs; this is particularly important for mmWave.

(3) A higher achievable data rate for each UE, provided the distance to the nearest AP is smaller than the distance from the UE to the BS in case of massive MIMO.

There are also some drawbacks of CF massive MIMO compared to massive MIMO:

(1) The higher cost of physical deployment due to the multiple antenna loca-

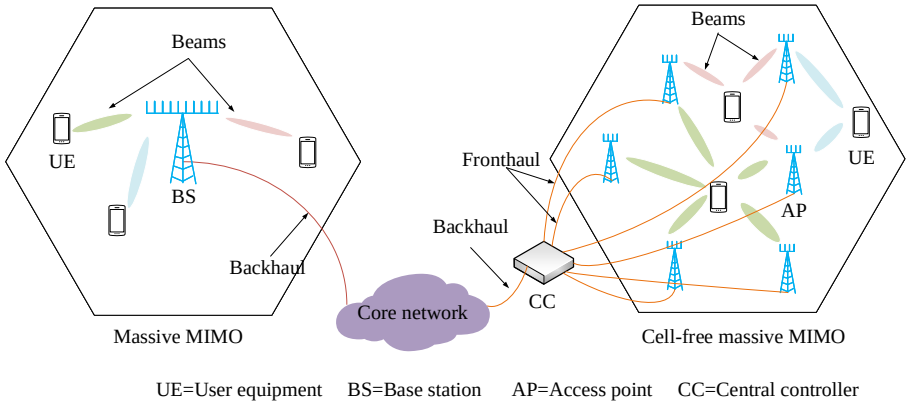


Figure 1.6: Massive MIMO vs CF massive MIMO.

tions and the need for a fronthaul.

(2) The overhead of CSI exchange in the fronthaul. The CC needs to collect CSI from the APs to perform coordinated signal processing, which decreases the time available for payload data transmission.

One should note that CF massive MIMO assumes that all UEs are served by all APs [11]. This is usually impractical and unnecessary in a geographically large network, e.g., in outdoor scenarios, where each UE is physically close to only a subset of the APs. For this case, an alternative topology named CF user-centric massive MIMO [28], has been very recently proposed. In CF user-centric massive MIMO, each UE is only served by a subset of APs. The subset follows a cooperation clustering rule, where there is a threshold of received signal strength (RSS) to determine which APs serve which UEs. In this thesis, we consider indoor scenarios where each UE is served by all APs, i.e., the CF massive MIMO.

CF massive MIMO is regarded as a promising technology for future wireless networks, in which high network capacity, high robustness, and high coverage probability are required [29–33]. However, the research on CF massive MIMO is still in its infancy.

1.2 Scope of the thesis

In this thesis, we study the power allocation problem in CF massive MIMO. Specifically, three problems are addressed:

- **Problem 1** The max-min power allocation in CF sub-6 GHz massive MIMO.
- **Problem 2** The max-sum SE power allocation in CF sub-6 GHz massive MIMO.
- **Problem 3** The max-sum SE power allocation in CF mmWave massive MIMO.

The main contributions of this thesis are the solutions for the three power allocation problems in CF massive MIMO. We study the use of DL methods to optimize the power allocation to meet restricted time constraints in both the sub-6 GHz and mmWave spectrum.

1.2.1 Challenges

Different power allocation schemes

[18] shows several power allocation schemes and their pros and cons. In the CF massive MIMO pioneer paper [11], the authors considered the max-min power allocation scheme in the sub-6 GHz spectrum. This scheme maximizes the minimum SE achieved at all UEs, at the expense of penalizing network capacity. It aims at providing a guaranteed data rate to all UEs. We focus on the network

capacity in this thesis, therefore the max-sum SE, which maximizes the sum SE in the whole system, is the main issue to be addressed. Nevertheless, with respect to the pioneer paper, we also investigate the max-min power allocation scheme in sub-6 GHz.

Because of the geographical spreading of the antennas, local signal processing is performed at each AP, e.g., by using CB, thereby avoiding the exchange of CSI between the CC and the APs. However, by doing this, the interference between different UEs cannot be suppressed, unlike when using zero-forcing beamforming (ZFB). Under these circumstances, max-min or max-sum SE power allocation becomes non-convex and hence computationally hard. Since the state of the channels evolves in time, the power allocation should stay in tune with this state. The present methods, e.g., the bisection algorithm for max-min optimization, and the WMMSE algorithm for max-sum SE optimization, typically find a near-optimal solution in an iterative manner, at the cost of a considerable execution time, potentially compromising the timeliness of the power allocation.

CF massive MIMO in the mmWave domain

mmWave plays an important role in 5G and beyond wireless networks. Indeed, much more bandwidth is available in the mmWave spectrum than in the sub-6 GHz spectrum. However, the operation in the mmWave domain poses different and new challenges for CF massive MIMO:

- **Beamforming.** The full benefit of (CF) massive MIMO can be obtained by providing each antenna with its own radio frequency (RF) chain. This is called full-digital beamforming. However, hardware constraints prevent the realization of full-digital beamforming at mmWave frequencies [34]. Thermal problems, due to the density of the hardware components and the high cost of RF chains, make full-digital beamforming, for the time being, an uneconomical solution. Therefore, one resorts to the more practical hybrid beamforming, which has much less RF chains, each driving an analog beamforming antenna array. This, however, degrades the network capacity. In Section 4.2, we study this degradation, by varying the number of RF chains in CF mmWave massive MIMO, using Monte Carlo simulation. From the numerical results we find that, when the number of RF chains per AP is twice the number of UEs, hybrid ZFB achieves the same performance as full-digital ZFB.

- **Time constraints for signal processing.** The channel varies faster in the mmWave domain than in the sub-6 GHz domain. The coherence time τ_c, during which the channel is regarded as constant, is calculated as follows [35]:

$$\tau_c = \frac{\lambda}{2v_l} \tag{1.2}$$

where λ represents the wavelength, v_l is the relative velocity between the transmitter and receiver. For 2 GHz with $v_l = 1$ m/s, the coherence time τ_c is around 75 ms. For 30 GHz with $v_l = 1$ m/s, the coherence time τ_c is around 5 ms. A relationship between the coherence time τ_c and the large-scale time τ_l is shown in [11]: $\tau_l = 40\tau_c$. Accordingly, the large-scale time τ_l is 3 s for 2 GHz and 0.2 s for 30 GHz. The time constraints for power allocation and beamforming are more restricted in the mmWave domain than in the sub-6 GHz domain.

1.2.2 Solutions for the three problems

Solution for Problem 1

We developed a deep supervised learning (DSL) method: a DNN is used to approximate a known but computationally complex algorithm [36]. Specifically, we proposed a two-stage DNN to approximate the bisection algorithm for optimization. Although the performance of the proposed DNN is slightly worse than that of the algorithm it has the big advantage that its execution time is much less than the bisection algorithm.

Solution for Problem 2

We proposed to use deep reinforcement learning (DRL) methods for this problem. The idea behind DRL is that agents take actions, reacting to their environment, such that a cumulative reward function is maximized [37]. Specifically, we proposed two DRL methods, namely deep Q-network (DQN) and deep deterministic policy gradient (DDPG). Because the DRL methods aim at maximizing the objective function, their performance can be better than the existing algorithms. In addition, the execution time is less.

Solution for Problem 3

Like **Problem 2**, we use both DQN and DDPG for this problem. For mmWave signal processing, hybrid beamforming is used, which makes **Problem 3** different from **Problem 2**. In addition, the channel model used for mmWave is different from the model for sub-6 GHz frequencies.

One should note that we do not study the use of DSL methods for the max-sum SE optimization nor DRL methods for the max-min optimization in this thesis. The reason is that the DSL-based methods for max-sum SE in (CF) massive MIMO are well studied in the literature, e.g., [31] and [38]. For the max-min scheme, which is more complicated than the max-sum SE scheme, the design of the DRL method needs further study. The details are discussed in **Chapter 2** and **Chapter 3**. Table 1.1 gives a short overview of the problems and the solutions we developed in this thesis.

Our research has resulted in several journal and conference papers. The list of publications is provided below.

Journal papers

- [J1] Y. Zhao, I. G. Niemegeers and S. M. Heemstra de Groot, “Power allocation in cell-free massive MIMO: A deep learning method,” *IEEE Access*, vol. 8, pp. 87185-87200, 2020.
- [J2] Y. Zhao, I. G. Niemegeers and S. M. H. De Groot, "Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods," *IEEE Access*, vol. 9, pp. 102953-102965, 2021.

Conference papers

- [C1] Y. Zhao, I. G. Niemegeers and S. M. Heemstra de Groot, “Distributed mmWave massive MIMO: A performance comparison with a centralized architecture for various degrees of hybridization,” *IEEE ICICN 2020*, pp.105-110, 2020. (Awarded by the best oral presentation)
- [C2] Y. Zhao, I. G. Niemegeers and S. M. Heemstra de Groot, “Power allocation in mmWave cell-free massive MIMO with user mobility using deep learning,” *IEEE ICCT 2020*, pp.264-269, 2020. (Awarded by the best oral presentation)
- [C3] Y. Zhao, I. G. Niemegeers and S. M. Heemstra de Groot, “Deep Reinforcement Learning for Dynamic Power Allocation in Cell-free mmWave Massive MIMO,” *Proceedings of the 18th International Conference on Wireless Networks and Mobile Systems*, pp.33-45, 2021. (Chair the session)
- [C4] Y. Zhao, I. G. Niemegeers and S. M. Heemstra de Groot, “Deep Q-network based dynamic power allocation for cell-free massive MIMO,” *IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, pp.1-7, 2021.

Table 1.1: Problem-Solution pattern: P (problem), S (solution).

P / S	**Problem 1**	**Problem 2**	**Problem 3**
DSL	DNN		
DRL		DQN&DDPG	DQN&DDPG

Table 1.2: Problem-Solution pattern: P (problem), S (solution).

	Title	Content
Chapter 1	Introduction	Background, thesis scope and organization
Chapter 2	Max-min power allocation in CF sub-6 GHz massive MIMO	Study of DSL method for Problem 1
Chapter 3	Max-sum SE power allocation in CF sub-6 GHz massive MIMO	Study of DRL method for Problem 2
Chapter 4	Max-sum SE power allocation in CF mmWave massive MIMO	Study of DRL method for Problem 3
Chapter 5	Conclusion and future work	Conclusion and future work

Table 1.3: Publication (Pub) and Chapter (Chap) relation: •(Strong), ∘(Weak).

Chap / Pub	**Chapter 2**	**Chapter 3**	**Chapter 4**
[J1]	•		
[J2]		•	∘
[C1]			•
[C3]			•
[C4]		∘	∘

1.3 Organization of this thesis

The organization of this thesis is shown in Table 1.2. The relations between each chapter and the publications are shown in Table 1.3.

Chapter 2

Max-min power allocation in CF sub-6 GHz massive MIMO

In this chapter, we address **Problem 1** and propose a DSL-based method to solve it. We introduce the system model for CF sub-6 GHz massive MIMO in Section 2.1. In Section 2.2, we formalize the max-min power allocation problem and present the DSL method for optimization. Finally, Section 2.3 summarizes the key points of this chapter.

2.1 System model

Consider a CF sub-6 GHz massive MIMO where N APs serve K UEs. Each AP has M uniform-planar-array (UPA) antennas, whereas each UE has a single antenna. The antenna spacing in the APs is half a wavelength in the APs [39]. All APs are connected to a CC through a fronthaul, see Fig.2.1.

2.1.1 Channel model

The analysis of CF sub-6 GHz massive MIMO typically uses the Rayleigh channel model see, e.g., the pioneer work [11], [40], [41], [42] and recent work [29], [31], [43], [44]. This model assumes enough scatterers and reflectors between the APs and UEs, which holds for most indoor scenarios. Some researchers proposed to use the Rician model, e.g., [45], [46]. The short distance between each UE and a few close APs gives rise to a dominant line-of-sight (LOS) impulse response. However, this is the case for CF user-centric massive MIMO [28], where each UE is served by several close APs. In our case, however, each UE is served by all APs, and therefore, the Rayleigh fading model is generally advocated [18], [46].

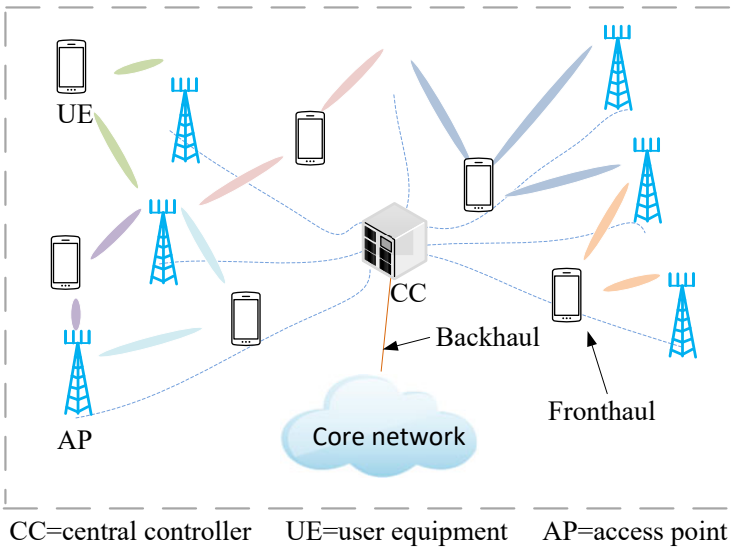


Figure 2.1: CF sub-6 GHz massive MIMO.

We define the Rayleigh channel vector between AP n and UE k as:

$$\mathbf{g}_{k,n} = \beta_{k,n}\mathbf{h}_{k,n} \tag{2.1}$$

where $\beta_{k,n}$ is the large-scale fading between AP n and UE k, and $\mathbf{h}_{k,n} \in C^{1\times M}$ is the small-scale fading vector. The elements of $\mathbf{h}_{k,n}$ are i.i.d. $\mathbb{CN}(0,1)$ random variables. We stress that the channels between UEs and AP antennas are in general not the same. Some may be modeled by the Rayleigh model, others by the Rician model, and some might have several dominant multi-path components. Since we concentrate on the power allocation, we simplify our task by assuming Rayleigh fading for all channels as argued above. However, for assessing the performance gains obtained by the methods more realistically, one would need to consider channel models that are based on measurements.

2.1.2 Uplink channel estimation

To make efficient use of the massive number of antennas, each AP needs to estimate the channel responses from each UE in each time-frequency block, see Fig.1.4. The estimated channel is used for beamforming: several independent beams serve the UEs. The APs estimate the channels via uplink pilots. Recall that τ_p is the channel estimation time, the uplink pilot of UE k is expressed as $\sqrt{\tau_p}\boldsymbol{\psi}_k$, where $\boldsymbol{\psi}_k \in C^{1\times\tau_p}$

is the time sequence with element 0-1 and $|\boldsymbol{\psi}_k|^2 = 1$. The received signal at AP n is the superposition of the pilots from all UEs:

$$\mathbf{y}_n = \sqrt{p_p \tau_p} \sum_{k'=1}^{K} \mathbf{g}_{k',n}^T \boldsymbol{\psi}_{k'} + \mathbf{w}_n \tag{2.2}$$

where $\mathbf{y}_n \in C^{M \times \tau_p}$, p_p is the normalized uplink pilot power, $\mathbf{w}_n \in C^{M \times \tau_p}$ is the normalized additive white Gaussian noise (AWGN) matrix with components that are i.i.d $\mathbb{CN}(0,1)$ random variables. Based on the received signal, the AP performs a conjugate operation to decode the desired signal from UE k:

$$\mathbf{y}_{k,n} = \mathbf{y}_n \boldsymbol{\psi}_k^H = \sqrt{p_p \tau_p} \sum_{k'=1}^{K} \mathbf{g}_{k',n}^T \boldsymbol{\psi}_{k'} \boldsymbol{\psi}_k^H + \mathbf{w}_n \boldsymbol{\psi}_k^H \tag{2.3}$$

where $\mathbf{y}_{k,n} \in C^{M \times 1}$. The minimum mean square error (MMSE) estimator [47] computes the estimated channel (denoted as $\hat{\mathbf{g}}_{k,n}^T$) from the received signal $\mathbf{y}_{k,n}$:

$$\hat{\mathbf{g}}_{k,n}^T = \frac{\sqrt{p_p \tau_p} \beta_{k,n}}{p_p \tau_p \sum_{k'=1}^{K} \beta_{k',n} |\boldsymbol{\psi}_{k'} \boldsymbol{\psi}_k^H|^2 + 1} \mathbf{y}_{k,n} \tag{2.4}$$

From (2.4) one can easily find $\hat{\mathbf{g}}_{k,n} \sim \mathbb{CN}(\mathbf{0}, \gamma_{k,n} \mathbf{I}_{1 \times M})$, where $\mathbf{I}_{1 \times M}$ represents the all-ones vector and:

$$\gamma_{k,n} = \frac{p_p \tau_p \left(\beta_{k,n}\right)^2}{p_p \tau_p \sum_{k'=1}^{K} \beta_{k',n} |\boldsymbol{\psi}_{k'} \boldsymbol{\psi}_k^H|^2 + 1} \tag{2.5}$$

One remark is that if $\tau_p \geq K$, one can choose $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, ... \boldsymbol{\psi}_K$ to be pairwise orthogonal, i.e., $\boldsymbol{\psi}_{k'} \boldsymbol{\psi}_k^H = 0$ for $k' \neq k$, then (2.5) is simplified to:

$$\gamma_{k,n} = \frac{p_p \tau_p \left(\beta_{k,n}\right)^2}{p_p \tau_p \beta_{k,n} + 1} \tag{2.6}$$

However, because of the limited length of the coherence time τ_c, in general $\tau_p < K$, some pilot sequences are reused, leading to pilot contamination. This degrades the performance of the CF massive MIMO in terms of achievable SE.

2.1.3 Downlink data transmission

Based on the estimated channels, the APs transmit their signals to the UEs using beamforming. Note that because of time constraints for beamforming (see Fig.1.4), linear beamforming techniques such as ZFB and CB, are typically preferred to non-linear beamforming techniques, due to their lower computational complexity [49].

Unless otherwise stated, in the rest of the thesis, beamforming refers to linear beamforming. ZFB aims to suppress the inter-UE interference, while CB aims to maximize the signal-to-noise-ratio (SNR) [50]. A more advanced linear beamforming, named Regularized ZFB (RZFB), takes both the inter-UE interference and the SNR into consideration to improve the network capacity [18]. However, one should note that ZFB and RZFB rely on the full CSI, i.e., the CSI of all links between the APs and UEs, which requires a prohibitive cost of CSI exchanging in the fronthaul. Therefore, CB is used in CF massive MIMO since it can be done at each AP, which implies that there is no CSI exchanging from CC to the APs via the fronthaul [11].

Let the vector $\mathbf{q} = [q_1, q_2, ..., q_K]^T$, with $|q_k|^2 = 1 (k = 1, 2, ...K)$ be the intended signal for the UEs , then the transmitted signal from AP n is:

$$\mathbf{x}_n = \sum_{k'=1}^{K} \sqrt{p_{k',n}} \frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{\mathbb{E}\{||\hat{\mathbf{g}}_{k',n}||^2\}}} q_{k'} = \sum_{k'=1}^{K} \sqrt{p_{k',n}} \frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}} q_{k'} \tag{2.7}$$

where $p_{k',n}$ is the normalized downlink transmission power from AP n to UE k'. UE k will receive the signal y_k, which is the superposition of the signals from all APs:

$$y_k = \sum_{n=1}^{N} \mathbf{g}_{k,n} \mathbf{x}_n = \sum_{n=1}^{N} \sum_{k'=1}^{K} \sqrt{p_{k',n}} \mathbf{g}_{k,n} \frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}} q_{k'} + w_k \tag{2.8}$$

where $w_k \sim \mathbb{CN}(0, 1)$ is the AWGN at UE k.

2.1.4 SE of the UEs

The ergodic SE of UE k is expressed as follows [22] [51]:

$$\begin{aligned} SE_k^{Rigorous} &= \left(1 - \frac{\tau_p}{\tau_c}\right) \mathbb{E}\{\log_2(1 + SINR_k)\} \\ &= \left(1 - \frac{\tau_p}{\tau_c}\right) \mathbb{E}\left\{\log_2\left(1 + \frac{|\sum_{n=1}^{N} \sqrt{p_{k,n}} \mathbf{g}_{k,n} \frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}}|^2}{\sum_{k' \neq k} |\sum_{n=1}^{N} \sqrt{p_{k',n}} \mathbf{g}_{k,n} \frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}}|^2 + 1}\right)\right\} \end{aligned} \tag{2.9}$$

where the factor $\left(1 - \frac{\tau_p}{\tau_c}\right)$ refers to the time for downlink data transmission as τ_p is used for channel estimation (see Fig.1.4). Equation in (2.9) is called the rigorous expression of SE [22]. The closed form of the rigorous expression is difficult to formalize and calculate because there is an expectation operation outside the logarithm [50]. There is an alternative expression, named use-and-then-forget

(UatF) [22] [50], giving a simpler lower bound which, in high-level channel hardening scenarios, is fairly tight.

The UatF expression is derived as follows: According to (2.8), the intended signal for UE k can be obtained from y_k:

$$\begin{aligned} y_k = & \underbrace{\mathbb{E}\left\{\sum_{n=1}^{N}\sqrt{p_{k,n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}}\right\}q_k}_{\text{Intended}} + \\ & \underbrace{\left(\sum_{n=1}^{N}\sqrt{p_{k,n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}} - \mathbb{E}\left\{\sum_{n=1}^{N}\sqrt{p_{k,n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}}\right\}\right)q_k}_{\text{Fluctuation}} + \\ & \underbrace{\sum_{k'\neq k}\sum_{n=1}^{N}\sqrt{p_{k',n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}}q_{k'}}_{\text{Interference}} + \underbrace{w_k}_{\text{Noise}} \end{aligned} \tag{2.10}$$

There are four parts in (2.10), namely, the intended signal, the fluctuation caused by the uncertain channel gains, the interference from other UEs and the AWGN. Then the downlink SE for UE k, is given by:

$$SE_k^{UatF} = \left(1 - \frac{\tau_p}{\tau_c}\right)\log_2(1 + SINR_k) \tag{2.11}$$

with:

$$SINR_k = \frac{M(\sum_{n=1}^{N}\sqrt{p_{k,n}\gamma_{k,n}})^2}{M\sum_{k'\neq k}\left(\sum_{n=1}^{N}\sqrt{p_{k',n}\gamma_{k',n}}\frac{\beta_{k,n}}{\beta_{k',n}}\right)^2|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2 + \sum_{k'=1}^{K}\sum_{n=1}^{N}p_{k',n}\beta_{k,n} + 1} \tag{2.12}$$

The details are shown in Appendix A.

The UatF expression is named after the fact that the received signals are used for signal detection but then are 'forgotten' for blind estimation of the instantaneous channel realizations [22]. The benefit of using the UatF expression is that the small-scale fading is neglected in the expression, which simplifies the power allocation [18] [22]. However, one should note that neglecting the small-scale fading for power allocation degrades the SE, especially in scenarios with a low level of channel hardening. In other words, to achieve the best performance, the power allocation should not neglect small-scale fading [31]. Nevertheless, [22] pointed out that, when using a sufficient number of antennas at the APs, e.g., 5 to 10, CF massive MIMO benefits from a high level of channel hardening. This implies that the degradation caused by using the UatF expression is negligible. That is why most publications, e.g., [52] [53] , used the UatF expression for the SE. In

this thesis, we use the UatF expression for analysis. Power allocation according to the rigorous expression is an open issue for further study. For convenience, we will use the notation SE_k rather than SE_k^{UatF} in the rest of the thesis.

2.1.5 Power allocation

The SE analysis in the previous subsection applies to arbitrary transmitted powers of the APs. However, a set of randomly or uniformly selected power in (2.12) is generally not the optimal strategy if one wants to achieve a particular optimization scheme [18]. There are typically two types of schemes, which have different optimization objectives for power allocation. The first scheme, called the max-min power allocation, focuses on fairness among the UEs, at the expense of not achieving the maximum network capacity. The second scheme, called the max-sum SE power allocation, aims at maximizing the sum SE to maximize the network capacity. To demonstrate these two types of power allocation, Fig.2.2 shows an example of the SEs achieved by the two schemes for a scenario with 1 AP serving 2 UEs. For the max-min power allocation, two UEs have the same SE values, namely 2.5 bit/s/Hz, the sum SE is 5 bit/s/Hz. For the max-sum SE power allocation, UE 1 has a SE of 4.8 bit/s/Hz, while UE 2 has a SE of 1.5 bit/s/Hz; the sum SE is 6.3 bit/s/Hz.

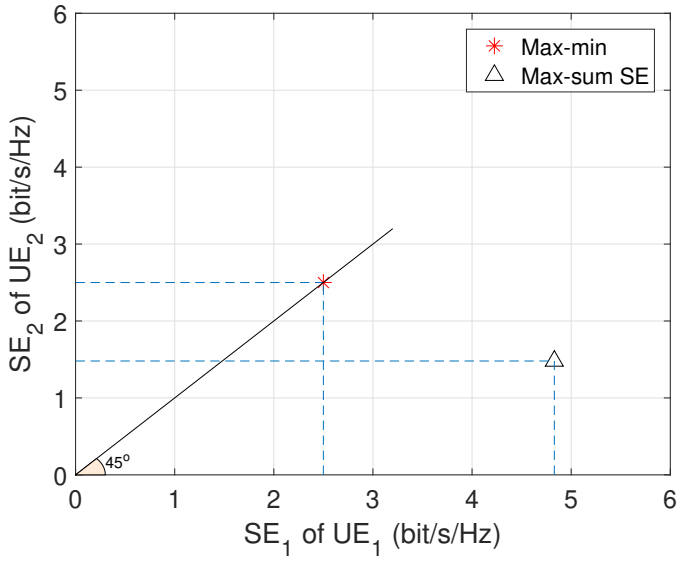


Figure 2.2: Example of SEs that can be achieved by different power allocation schemes [18].

There are also other power allocation schemes such as max-product SINR, max-energy efficiency, etc. The fact depends on the optimization objective that the utility function intends to maximize [18]. The optimization objectives are typically classified into two subsets, the first one focuses on the peak rate, the representative scheme is max-sum SE; the second one focuses on the rate that can be guaranteed in the coverage area, the max-min is a representative scheme thereof [28]. Other schemes are in between these two schemes, namely with the objective considering different weights for the peak rate and the rate that can be guaranteed in the coverage area.

There is no answer to which power allocation scheme is the best. Fairness among the UEs and network capacity are two essential elements that need to be considered by the network designer. In this thesis, we propose methods to achieve these two power allocation schemes.

2.2 Max-min power allocation scheme

The max-min power allocation scheme is formulated as follows:

$$\begin{aligned} &\max_{p_{k,n}} \min_{k'} SE_k \\ &s.t. \sum_{k=1}^{K} p_{k,n} \leq p_l, \forall n \end{aligned} \tag{2.13}$$

where p_l is the transmission power limit of each AP. An exact solution for (2.13) is not feasible, since the computational complexity increases exponentially as N and K increase linearly, i.e., it is nondeterministic polynomial (NP)-hard. Therefore, one must resort to a heuristic to solve it.

2.2.1 Bisection algorithm for max-min power allocation

A widely adopted heuristic (see, e.g., [11]) for (2.13) is the bisection algorithm [57], shown in Algorithm 1, where η is a parameter to control the termination of the iteration and, $SINR_{up}$, and $SINR_{down}$ denote the upper and lower bound for $SINR_{candidate}$. $SINR_{candidate}$ denotes the achievable SINR for all UEs. The key idea behind the bisection algorithm for solving(2.13) is to divide the problem into two sub-problems, namely, the candidate value problem and the feasibility problem. In each iteration of the bisection algorithm, a candidate value is chosen to determine the constraints of the subsequent feasibility problem.

The challenge of this algorithm is that one needs to solve a non-convex feasibility problem in each iteration (Step 3~5 in Algorithm 1), which is a nonlinear

Algorithm 1 Pseudo code of the bisection algorithm.

1: Initialize $SINR_{up}$, $SINR_{down}$ and the tolerance $\eta > 0$.
2: Set: $SINR_{candidate} = (SINR_{up} + SINR_{down})/2$.
3: Check feasibility:
4: $\max\limits_{k=1,2,...K} SINR_k \geq SINR_{candidate}$
5: s.t. $\sum_{k=1}^{K} p_{k,n} \leq p_l, \forall n$
6: If it is feasible: set $SINR_{up} = SINR_{candidate}$.
7: Otherwise, set:$SINR_{down} = SINR_{candidate}$.
8: Repeat 2~7 until $SINR_{up} - SINR_{down} < \eta$.

inequalities problem with $K + N$ constraints and KN variables. There is no analytical solution, therefore we propose the numerical solution given by Algorithm 2. This, of course, will give a sub-optimal solution.

Algorithm 2 Pseudo code of the feasibility algorithm.

1: Initialize power allocation and the tolerance $\eta' > 0$.
2: **While** $SINR_{max} - SINR_{min} > \eta'$:
3: If $SINR_{max} \leq SINR_{candidate}$, break **While**, output: **unfeasible**.
4: If $SINR_{min} \geq SINR_{candidate}$, break **While**, output: **feasible**.
5: Label the UE with the highest SINR as UE_{max}.
6: Label the UE with the lowest SINR as UE_{min}.
7: Go through all APs, decrease the allocated power of UE_{max} to $SINR_{max} = SINR_{candidate}$. Increase the allocated power of UE_{min} under constraint of power limit. If $SINR_{min} \geq SINR_{candidate}$, continue **While**.
8: End **While**

The key idea of Algorithm 2 is that, after an initial power allocation, the UE with the highest SINR will always give its allocated power to the UE with the lowest SINR, provided that its SINR does not go below $SINR_{candidate}$. At the stopping point of the algorithm, the difference in SINR between the UE with the highest and the one with the lowest SINR, will be no more than η'. If we set η' small enough, the SINR, accordingly the SE, of all UEs will be regarded as sufficiently equal, achieving fairness.

Even if the bisection algorithm fairly achieves the max-min power allocation scheme, the computational complexity may be too high to meet the tight time constraint in Fig.1.4.

There are two loops in the bisection algorithm, namely the bisection search loop in Algorithm 1 and the feasibility loop in Algorithm 2. Here we use I_B and I_F to represent the bisection search loop and the feasibility loop, respectively. The com-

putational complexity mainly lies in Algorithm 2. Specifically, the computational complexity to calculate $\gamma_{k,n}$ is $\mathrm{O}(K)$; then derive from (2.12), the computational complexity to calculate the SINR of each UE is $\mathrm{O}(KN^2)$. Therefore, for K UEs, the total computational complexity is $\mathrm{O}(KN^3)$. The computational complexity to find the maximum and minimum value of SINR is $\mathrm{O}(K)$.

The computational complexity of Step 7 in Algorithm 2 is $\mathrm{O}(N)$, which is an inner loop of Algorithm 2. Accordingly, the computational complexity of Algorithm 2 is $\mathrm{O}(I_F(KN^3+K)N)=\mathrm{O}(I_F N^2K^3)$. Finally, the computational complexity of the bisection algorithm for (2.13) is $\mathrm{O}(I_B I_F N^2K^3)$.

2.2.2 DSL for max-min power allocation

The problem of the bisection algorithm for power allocation is that the computational cost may be excessive for the given time budget. This led us to propose a DL method, namely a DNN, to perform the task. The computational complexity of a DNN is low: it only requires a few layers of simple operations such as matrix or vector multiplications [59]. In addition, a DNN is expected to run on neural processing units (NPU) [60], specifically designed to support DL, making it possible to parallelize parts of the computation. Hence it should be easy to meet the time constraints.

It is worth pointing out that the max-sum SE scheme is simpler to solve than the max-min scheme. The problem in (2.13) has two sub-problems, namely the candidate value problem and the feasibility problem, wherein the latter one is non-convex and NP-hard. In other words, (2.13) poses a sequence of non-convex and NP-hard problems. While the max-sum SE power allocation, we will show in the next section, comprises one non-convex and NP-hard problem. However, one may find that most existing literature studies the max-sum SE scheme rather than the max-min scheme, especially when using the low-complexity DL methods for power allocation. For example, in [59], a fully connected DNN mimicked the processing of the WMMSE algorithm to maximize the sum SE in multi-cell networks. The numerical results show that with a DNN one can closely approximate the performance of WMMSE but, in substantially less time. Similarly, in [61], the authors proposed a DL method to control the power to maximize a utility function in a non-orthogonal multiple access (NOMA) downlink system. The DL-based power allocation method in massive MIMO is studied in [31] [38], and [62]. Specifically, [31] exploited a DNN to maximize the uplink sum SE in CF sub-6 GHz massive MIMO. The state-of-the-art residual dense block (ResDense) method was applied in [38] to allocate power in multi-cell massive MIMO. In [62] a two-layer DNN was used for power allocation to combat inter-cell interference in massive MIMO. [63] proposed a recurrent neural network (RNN) to implement the max-min power allocation in cellular massive MIMO. However, this work did not address the max-min power

allocation in CF massive MIMO.

All the above works considered the DSL method, i.e., using a DNN to approximate a known but complicated algorithm, to allocate the power. Inspired by these works, we propose a DSL method to perform the max-min power allocation in CF massive MIMO. The main idea of the proposed DSL method is to use a DNN to approximate the bisection algorithm. Specifically, a DNN is used to determine the mapping between the large-scale fading and the optimized power obtained by the bisection algorithm. In other words, the DNN takes as input the large-scale fading and outputs the allocated power for each AP to each UE. We show in Appendix B an explanation of general concepts used in DNN.

Design of the DNN

As discussed in Section 2.2.1, (2.13) is optimized with the bisection algorithm by solving two sub-problems. Therefore, we propose a two-stage DNN to approximate the bisection algorithm.

We note that the candidate value $SINR_{candidate}$ is determined by solving a sequence of feasibility problems, while the power allocation is based on this $SINR_{candidate}$. If this $SINR_{candidate}$ is known or easily obtained, the loop in Algorithm 1 can be substantially decreased.

Based on the above analysis, and according to the universal approximation theory [64], we expect to get $SINR_{candidate}$ (to simplify the statement, we use S to represent $SINR_{candidate}$) by using a DNN with several layers. We use a convolutional neural network (CNN) for the regression of S, because it achieves higher performance than fully connected DNN for many applications [64] [65]. In Appendix C, the concept of CNN is explained. After we get S, we need an iterative algorithm to solve the feasibility problem to determine the allocated power. Referring to [59], we can use several fully connected layers to approximate this iterative algorithm. So, we design the structure of our DNN as consisting of two stages: regression processing and allocation processing, see Fig.2.3.

Regression:

The objective of the regression part is to output S from the input, i.e., from the large-scale fading matrix β, where:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{1,1}\mathbf{I}_M & \beta_{1,2}\mathbf{I}_M & ... & \beta_{1,N}\mathbf{I}_M \\ \beta_{2,1}\mathbf{I}_M & \beta_{2,2}\mathbf{I}_M & ... & \beta_{2,N}\mathbf{I}_M \\ ... & ... & ... & ... \\ \beta_{K,1}\mathbf{I}_M & \beta_{K,2}\mathbf{I}_M & ... & \beta_{K,N}\mathbf{I}_M \end{bmatrix} \tag{2.14}$$

We use two convolutional layers and two fully connected layers for this process. Specifically, for the first convolutional layer, we use $5 \times M \times Q$ filters with stride $[1, M]$ to operate on the input matrix. The result of this convolutional operation yields Q feature matrices with $K \times N$ elements by zero padding 2. Note that we

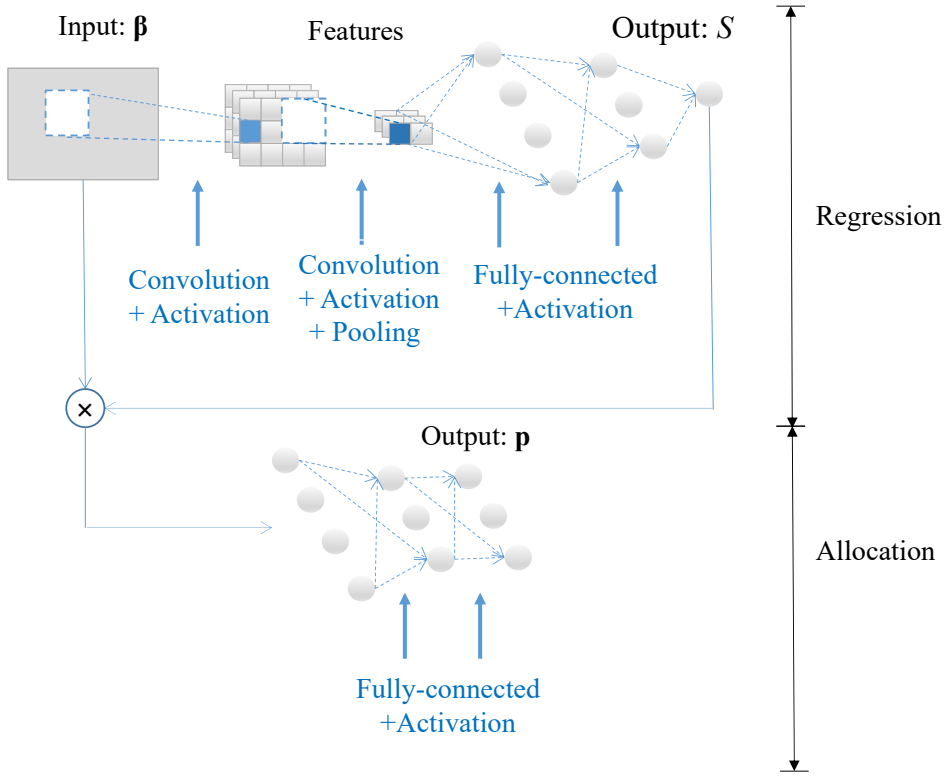


Figure 2.3: The two-stage DNN for power allocation.

do not use a pooling operation after feature extraction in this layer. In the second layer, we use $5 \times 5 \times Q$ filters with stride $[1,1]$ and zero padding 2 to guarantee the same number of inputs and outputs in this layer. Then a max-pooling operation is used to decrease the number of parameters. Here we use a 2×2 kernel size with stride $[2,2]$. After that we adopt a two-layer fully connected network to get S. The numbers of neurons in these two fully connected layers are $\lceil K/2 \rceil \times \lceil N/2 \rceil \times Q$ and $\lceil K/2 \rceil \times \lceil N/2 \rceil \times \lceil Q/2 \rceil$, respectively, where $\lceil . \rceil$ represents the ceiling operation.

Allocation:

When S is obtained, the next step is to output the allocated power $\mathbf{p} = [p_1, p_2, ...p_{KN}]$. We derive from (2.12) that there is a multiplication operation of $\boldsymbol{\beta}$ and SINR to calculate the transmission power, so $S\boldsymbol{\beta}$ is the input in this phase. Finally, two fully connected layers with $2KN$ and KN neurons are employed to map the nonlinear

relationship between $S\boldsymbol{\beta}$ and $\mathbf{p}$.

Rectified linear units (ReLu) are used as our activation function in all layers. While we propose a specific DNN structure, it is worth to point out that finding the best DNN structure and the values of the hyperparameters can also be seen as optimization problems, requiring further research. Referring to the literature ([59], [67]), we tried several structures and hyperparameters, including fully connected networks (from one layer to six layers) and traditional CNN (two convolutional layers with a number of fully connected layers varying from one to four), to choose the best configuration, i.e., the one that gives us the lowest mean square error (MSE) on the training dataset.

Training of the DNN

The DNN is used to approximate the bisection algorithm, therefore a training dataset generated by the bisection algorithm is required. The training dataset is generated with sufficiently large snapshots: in each snapshot, K UEs are uniformly and randomly distributed in the coverage area. The max-min power allocation is computed by the bisection algorithm. Three elements are recorded in each snapshot, namely the large-scale fading matrix $\boldsymbol{\beta}$, the target outputs $\mathbf{p}^{\text{target}}$ and S^{target} by the bisection algorithm, to generate a training data sample.

The DNN is trained in two phases, i.e., the input-output pair ($\boldsymbol{\beta}$, S) is used to adjust the filters and the first two fully connected layers (regression processing), while ($S\boldsymbol{\beta}$,$\mathbf{p}$) is used to train the following two fully connected layers (allocation processing). The MSE is used as the loss function:

$$MSE = \frac{1}{i}\sum_{i=1}^{I_T}\left(\underbrace{\frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(\frac{p_{k,n}^{pre,i}-p_{k,n}^{target,i}}{p_l}\right)^2}_{\text{average power error}}+\underbrace{\left(\frac{S^{\text{pre},i}-S^{\text{target},i}}{S^{\text{target},i}}\right)^2}_{\text{regression error}}\right) \tag{2.15}$$

where I_T is the number of training data, and $p_{k,n}^{pre,i}$, $p_{k,n}^{target,i}$, $S^{\text{pre,i}}$, $S^{\text{targe,i}}$ represent the output of the power allocation by the DNN, the target output of the power allocation by the training dataset, the final $SINR_{candidate}$ by DNN, and the final $SINR_{candidate}$ by the training dataset for sample i, respectively. The first term in (2.15) represents the average power error while the second one is the regression error of S.

The DNN is trained using the adapted gradient descent method described in [68], which means that the learning rate is updated in each epoch: in initial epochs the learning rates are large for sufficient searching in the feasible region, while in the later epochs the learning rates are getting smaller to guarantee that the loss function will converge. Ideally the training of DNN should use all data in each processing, however, this is very expensive when the training set is huge. An

efficient way is to use a random subset of the training set, called minibatch, to evaluate the gradients [69]. The MSE of the DNN for a random minibatch D is:

$$MSE = \frac{1}{i}\sum_{i=1}^{D}\left(\underbrace{\frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(\frac{p_{k,n}^{pre,i}-p_{k,n}^{target,i}}{p_l}\right)^2}_{\text{average power error}}+\underbrace{\left(\frac{S^{\text{pre},i}-S^{\text{target},i}}{S^{\text{target},i}}\right)^2}_{\text{regression error}}\right) \tag{2.16}$$

Computational complexity of the DSL method

The computational complexity considers the operational phase, i.e., the time that a well-trained DNN performs the power allocation. Therefore, the computational complexity of the DSL-based methods is only determined by the DNN. In the first convolutional layer, we use $5 \times M \times Q$ filters with stride $[1, M]$. The input is one $K \times MN$ matrix so the output of the first convolutional layer is a $K \times N \times Q$ matrix, which implies $K \times N \times Q$ convolutional computations. Therefore, the computational complexity of the first convolutional layer is $\mathrm{O}(5KNQ)$. Similarly, the computational complexity of the second convolutional layer is $\mathrm{O}(25KNQ^2)$. The computational complexity of the fully connected layers is $\mathrm{O}(\nu\mu^2)$, where ν is the number of layers and μ is the number of neurons in the widest layer, i.e., the layer with the most neurons. This number depends on the output of the convolutional layer, i.e., $\mathrm{O}(\mu)=\mathrm{O}(KN)$ in our case. The number of layers for a DNN is independent of the scale of the problem. Therefore, the computational complexity of the fully connected layers is $\mathrm{O}(K^2N^2)$. Finally, the total computational complexity of the proposed DNN is:

$$\mathrm{O}(5KNQ + 25KNQ^2 + K^2N^2) = \mathrm{O}(KNQ^2 + K^2N^2) \tag{2.17}$$

Even if it is not obvious, we will show by examining the execution time, that the computational complexity of (2.17) is lower than the one of the bisection algorithm $\mathrm{O}(I_B I_F N^2 K^3)$. This implies that the DNN can meet the tight time constraints more easily than the bisection algorithm.

2.2.3 DSL vs bisection algorithm

In this section, we show by simulations that the DSL method can closely approximate the performance of the bisection algorithm and that it has a lower execution time.

Scenario and configuration

Consider a 200m×200m square coverage area with a total of 100 antennas to serve $K = 5$ UEs. There are 9 APs ($N = 9$) placed in a regular grid as shown in

Fig.2.4. Each AP has 11 antennas ($M = 11$), except for the central one which has 12. This has no particular significance, except that the simulations were originally done for studying the effects of the degree of distribution of the antennas on the network capacity. Note that determining the optimal degree of distribution and geographical deployment of APs in CF massive MIMO is an open issue. Finding the optimal configuration, given a particular number of antennas requires further research. We have used a sample scenario of a CF massive MIMO to demonstrate the potential of the DSL method. However, for a new configuration (e.g., different deployment of APs), the DNN might need to be retrained.

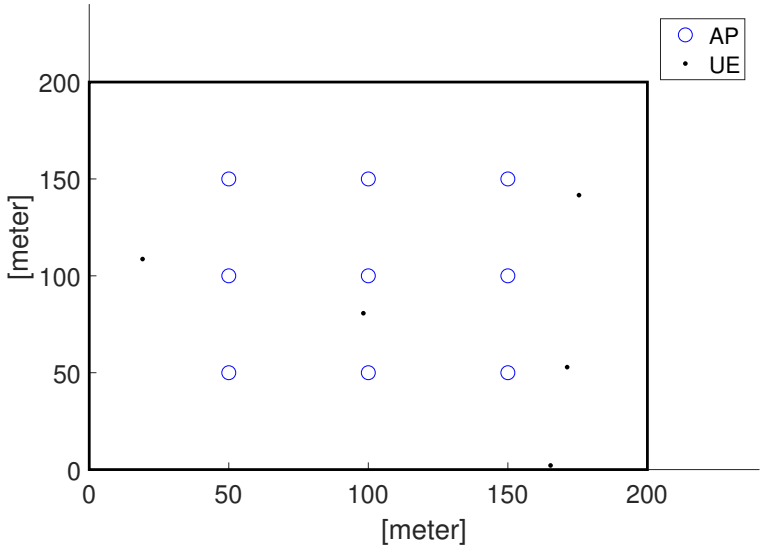


Figure 2.4: The scenario used in simulations.

As in [11], the maximum power levels for each AP and each UE are 23 dBm and 20 dBm, respectively. The carrier frequency is 1.9 GHz and the available bandwidth is 20 MHz. We set the height of the APs to 5 m and for the UEs to 1.65 m. We assume that the noise power is -94 dBm, and the standard deviation of shadow fading is 8 dB. The number of modulation samples in each coherence interval is assumed to be 200. The minibatch size, i.e., the number of samples to process before the parameters are updated in the DNN, is 500. The initial learning rate is 0.002 and the maximum number of iterations is 800. The initial weights and bias are Gaussian random variables that have an $\mathbb{N}(0, 0.01)$ distribution. The number of filters in each convolutional layer is 60. For the first two fully connected layers, we

Table 2.1: Parameters used in simulations.

Parameters for MIMO	
Parameter	value
Carrier frequency	1.9GHz
Bandwidth	20MHz
Noise/contamination power	-94dBm
Height of APs	5m
Height of UEs	1.65m
Power of pilots, downlink transmission	23dBm, 20dBm
Standard deviation of shadowing fading	8dB
Parameters for DNN	
Parameter	value
Batch size	500
Initial learning rate	0.002
Maximum number of iterations	800
Number of filters	60
Initial weights and bias	$\mathbb{N}(0.01)$
Neurons in fully connected layers	[900, 450; 1,000, 500]

set the number of neurons to 900 and 450, respectively; while for the latter fully connected layers, we select 1,000 and 500, respectively. The simulation parameters are listed in Table 2.1.

The large-scale fading from AP n to UE k is given by:

$$\beta_{k,n} = PL_{k,n} \times 10^{\frac{\sigma z_{k,n}}{10}} \tag{2.18}$$

where $PL_{k,n}$ is the path loss in dB, the second factor represents the shadow fading with standard deviation σ and $z_{k,n}$ is the shadow fading coefficient defined as in [11] [70]:

$$z_{k,n} = \sqrt{\kappa} a_n + \sqrt{1-\kappa} b_k \tag{2.19}$$

where $a_n \sim \mathbb{N}(0,1)$ and $b_k \sim \mathbb{N}(0,1)$ are independent random variables, and κ, where $0 \leq \kappa \leq 1$, is a parameter. When $\kappa = 0$, the shadowing from a given UE is the same for all APs, which means that the obstacle is close to the UE; while for $\kappa = 1$, the shadowing from a given AP is the same to all UEs, means that the obstacle is close to the AP. In our simulation, we set $\kappa = 0.5$ and adopt the covariance functions of a_n and b_k in [70]. The three-slope path loss model [71] is formulated as follows:

$$PL_{k,n} = \begin{cases} -La - 35\log_{10}(d_{k,n}), d_{k,n} > d_1 \\ -La - 15\log_{10}(d_1) - 20\log_{10}(d_{k,n}), d_0 < d_{k,n} \leq d_1 \\ -La - 15\log_{10}(d_1) - 20\log_{10}(d_0), d_{k,n} \leq d_0 \end{cases} \tag{2.20}$$

where $d_{k,n}$ is the distance between UE k and AP n, La is defined in [72] as:

$$\begin{aligned} La =& 46.3 + 33.9\log_{10}(f) - 13.82\log_{10}(h_{AP}) - \\ & (1.1\log_{10}(f) - 0.7)h_u + 1.56\log_{10}(f) - 0.8 \end{aligned} \tag{2.21}$$

where f is the carrier frequency, h_{AP} is the height of the APs and h_u is the UE antenna height.

Performance of the DSL method

We generate a dataset containing 432,000 data samples, wherein 430,000 data samples are used to train the DNN, the remaining 2,000 data samples are used for testing. Fig.2.5 shows the training process, i.e., how the average power error, the regression error, and the loss function, behave with the number of iterations.

Two hyperparameters influence the behavior of the loss function. The first one is the learning rate, i.e., the evolution rate of the DNN. A higher rate results in a faster convergence but runs the risk of ending up in a local optimum, which means that insufficient searching has been done. A low rate, on the other hand, may lead to a loss function that does not converge. The second influential hyperparameter is the number of filters (Q) in each convolutional layer. More filters improve the prediction accuracy, i.e., the loss becomes smaller, however, it also implies that the training time becomes longer. It is worth to point out that the number of filters dominates the loss function of the regression while the learning rate has more influence on the loss function of the allocation. We tried different hyperparameters, considering the tradeoff between the number of filters and the implementation complexity and, between the learning rate and the training time. We see from Fig.2.5 that after 800 iterations, the loss function is close to 0; its value is actually 10^{-5}. This value is expected to further decrease as the number of iterations grows.

The remaining 2,000 data samples are used to test the DNN. Fig.2.6 compares the power allocation obtained by the DNN and the target, determined by the bisection algorithm. It does this by comparing the CDF of the power allocation from 2,000 testing data samples. Each data sample consists of 45 transmit-receive pairs (KN), so the data in Fig.2.6 is the result of 90,000 pairs. A specific transmit-receive pair corresponds to a particular AP and the receiver of a UE. We found that the power allocated to around 30% of the channels is zero. This means that some APs are not providing power to some UEs. We also observe that the power distribution generated by the DNN and the target almost overlap, which means that the two methods achieve similar results.

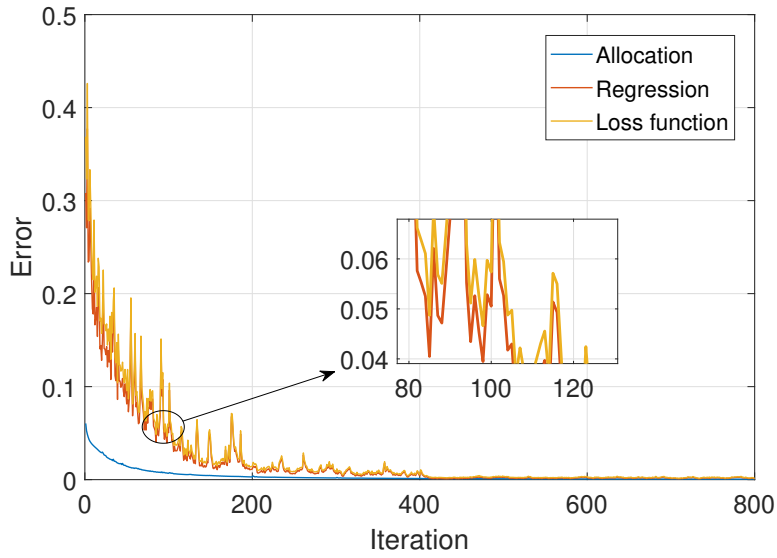


Figure 2.5: Average power error, regression error and loss function as a function of the number of training iterations.

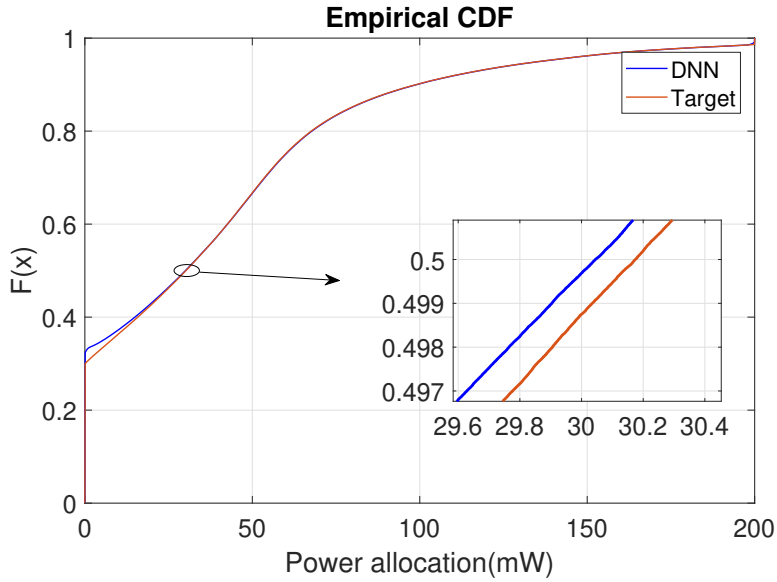


Figure 2.6: CDF of power allocation with $N = 9$, $M = 11, 12$.

However, the CDF only describes the results statistically, different transmit receive pairs may get a very close transmission power, as a result, a similar CDF is obtained. To make it clear, we choose the results of one data sample, randomly taken from the test dataset, which contains 2,000 data samples, see Fig.2.7. Fig.2.7 shows the histogram of the power allocation for one randomly chosen data from 2,000 test data samples. For this specific data, the target (bisection) and the DNN for each of the 45 (NK) transmit-receive pairs are compared. The errors between DNN and target (bisection) power allocation are mostly less than 1 mW. Fig.2.7 demonstrates that the DNN performs as well as Fig.2.6 shows. Following the power allocation, Fig.2.8 compares the CDF of the per-UE SE for the DNN and the target bisection algorithm, based on the total test dataset containing 2,000 data points. From Fig.2.8 one can find that the gap for the DNN and the target bisection algorithm is no more than 0.2 bit/s/Hz, which maps to at least 94% accuracy of approximation, measured as (1-|Target-DNN|/target).

Then we test the DNN with mobile UEs. Each UE has an initial position that is uniformly distributed over the coverage area. Each UE moves in a random direction (up, down, left and right) with a randomly chosen velocity distributed uniformly between 0 and 3 m/s. It maintains its speed and direction for 1s, before selecting a new speed and direction. When the UE reaches the boundary of the coverage area it reverses its direction of movement to stay within the coverage area. A trace of a realization of such a scenario, over the simulation period, for five UEs, is shown in Fig.2.9.

We simulate a duration of 200 seconds starting from a random initial position of the UEs. The DNN is performed every second, corresponding to 40 times the coherence time. We compare again the DNN and the bisection algorithm. Fig.2.10 shows the CDF of the power allocation for both methods, over a 200 seconds period. These results are thus based on 9,000 instances of power allocation.

From Fig.2.10 we can see that the DNN performs worse than in the fixed-position scenario. Much more errors occurred in 1~70 mW. We also show the results of one randomly chosen one second time period out of the 200 seconds in Fig.2.11.

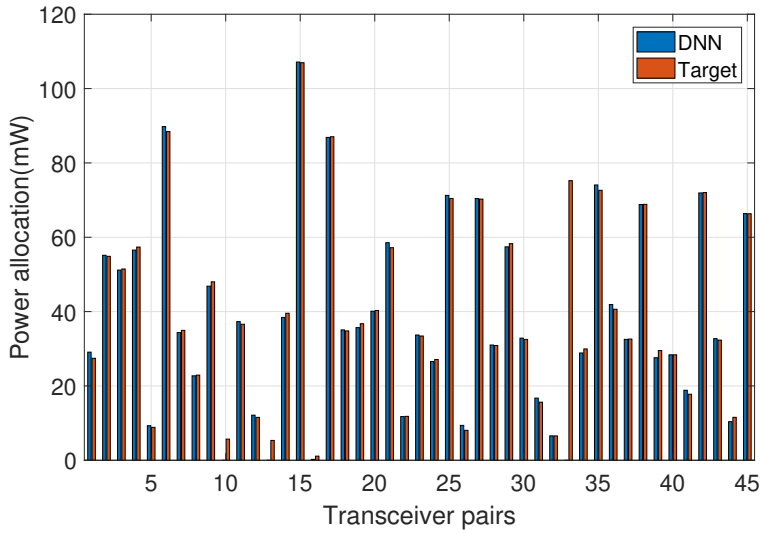


Figure 2.7: One random data from 2,000 data samples with $N = 9$, $M = 11, 12$.

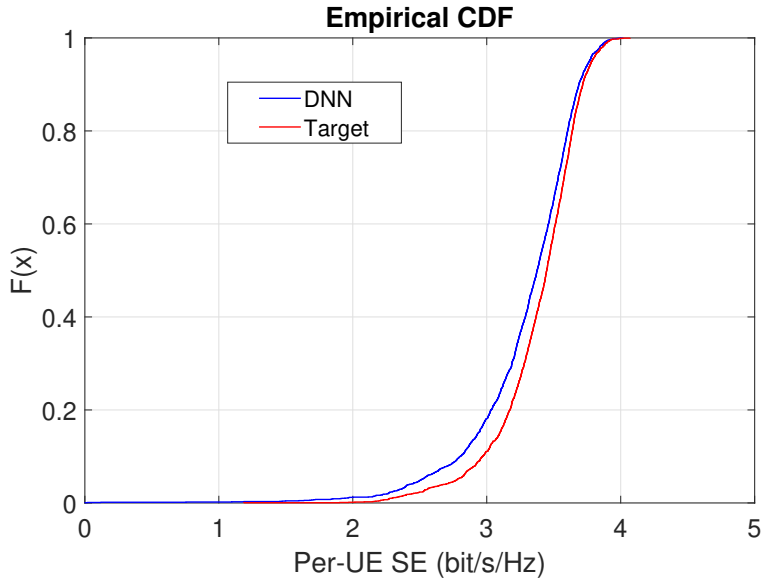


Figure 2.8: CDF of per-UE SE with $N = 9$, $M = 11, 12$.

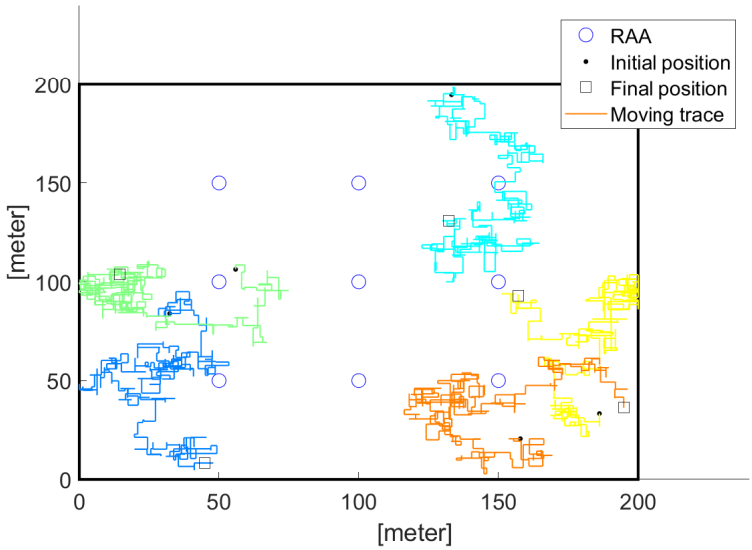


Figure 2.9: Mobile UE scenario over 200 seconds with $N = 9$, $M = 11,12$.

Compared to Fig.2.7, Fig.2.11, shows that the DNN approximates the trend of the target bisection algorithm but has larger errors with respect to the target than in the fixed-position UE scenario. This is because in the mobile-UE scenario, the random directions and velocity of the UEs induce more noise for the input of the DNN, leading to a less accurate approximation of the bisection algorithm. As in the fixed-position UE scenario, we also compare the per-UE SE of the DNN and the bisection algorithm. Fig.2.12 shows the simulation results. The largest difference between the two methods occurs around 3.2 bit/s/Hz, corresponding to around 85% accuracy of approximation. We also observe that the difference is the largest in the left side of the CDF figure, which implies that for the cases where UEs are in an unfavorable location, e.g., due to shadowing, the DNN does not perform as well.

Finally, to see what impact a different deployment of the APs has on the effectiveness of the DNN method compared to the bisection algorithm, we consider the AP configuration shown in Fig.2.13. 4 APs are placed in the coverage area and each AP is equipped with 25 antennas, 5 single-antenna UEs are served.

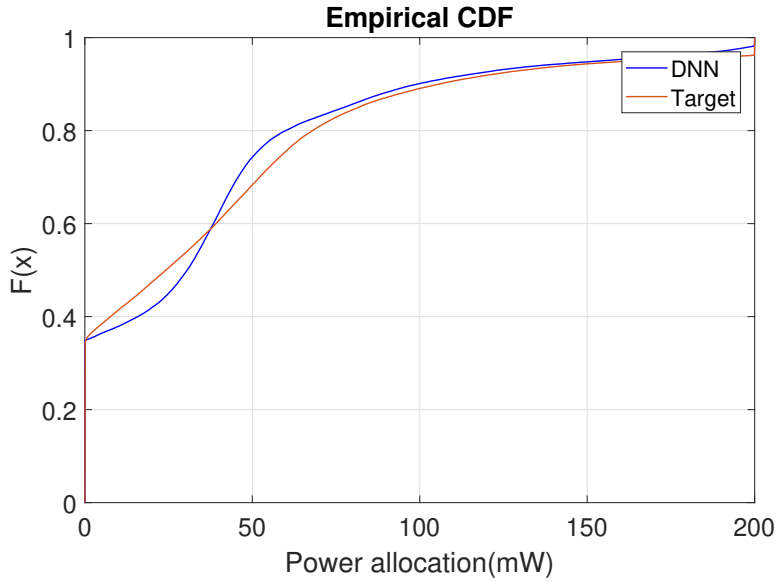


Figure 2.10: CDF of power allocation with $N = 9$, $M = 11, 12$ and mobile-UEs.

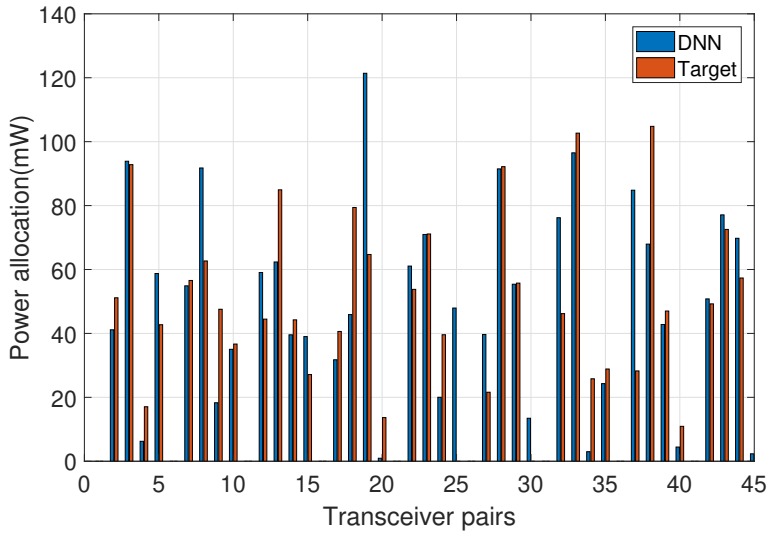


Figure 2.11: One random data from 200 second with $N = 9$, $M = 11, 12$ and mobile-UEs.

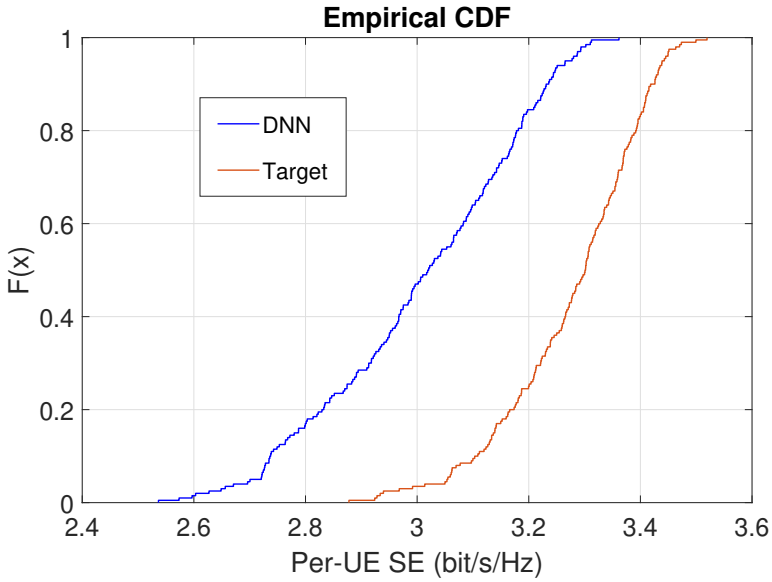


Figure 2.12: CDF of per-UE SE with $N = 9$, $M = 11,12$ and mobile-UEs.

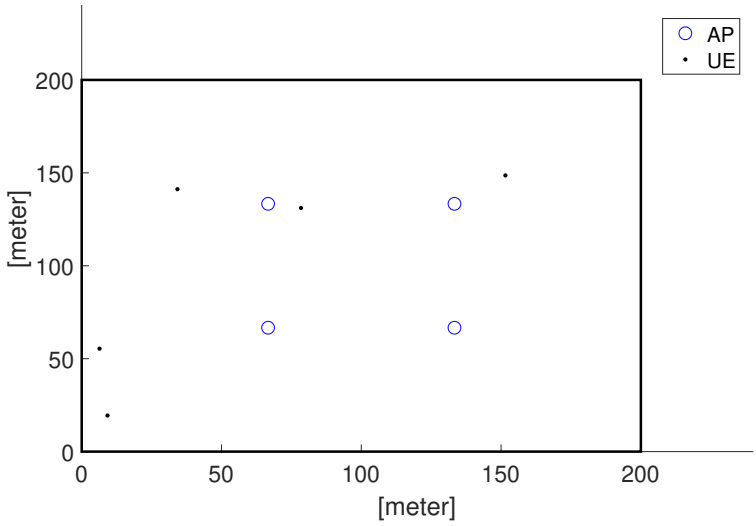


Figure 2.13: Different deployment of APs.

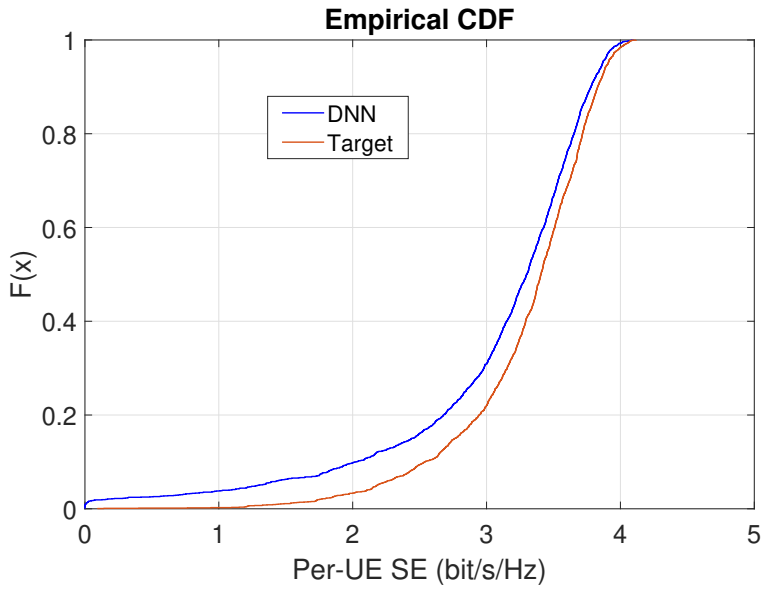


Figure 2.14: CDF of per-UE SE with $N = 4$, $M = 25$ scenario.

We generate 173,800 data samples in scenario Fig.2.13 using the bisection algorithm. Then 171,800 data samples are used for training the DNN that has been trained in Fig.2.5. That is, a total of 601,800 (171,800+430,000) data samples have been used to train the DNN. The rest 2,000 data samples are used for testing. Fig.2.14 shows the CDF of per-UE SE. The largest errors occur in the low SE part (from 0 to 3bit/s/Hz). While for the other part, the difference between the DNN and the bisection algorithm is no more than 0.3 bit/s/Hz, corresponding to around 88% accuracy.

Comparison of the execution time

It is difficult to do a fair comparison in terms of processing time in a real implementation. One important reason is that the implementations will be based on different hardware architectures. The bisection algorithm is well suited to be executed on a classical (multicore) CPU architecture, the DNN will likely run on a hardware architecture optimized for machine learning, e.g., an NPU. Nevertheless, it is revealing to see the huge difference, when both are executed on the same hardware. Table 2.2 shows the statistical characteristics of the execution time for the simulation over 2,000 testing data samples. We use the same platform, a 4 core Intel Core i5-7300 CPU with 2.6 GHz frequency. The programs are both written in Python 3.7.2. From Table 2.2 it is obvious that the DNN requires much less pro-

Table 2.2: Execution time of the DNN and bisection algorithm (in seconds).

Method	Mean	Max	Min	Standard deviation
DNN	0.0683	0.0832	0.0562	0.0068
Bisection	29.6202	46.6688	12.1361	8.0527

cessing time than the bisection algorithm and has less variation. For the DNN, the number of calculations is constant. The fluctuation of processing time comes from the calculation of different floating-point numbers and the inaccuracy of reading the system time. The CPU load also plays a role in this fluctuation. For the bisection algorithm, the time fluctuation mainly comes from different initializations, i.e., a different starting point of the search can make a large difference in the time needed to find the optimum.

2.3 Summary of key points in Chapter 2

In this chapter, we have proposed a DNN to perform the power allocation in a CF sub-6 GHz massive MIMO. The max-min power allocation scheme, which provides a fair quality of service for all UEs, was considered. We showed that this non-convex and NP hard problem can be better solved by a well-trained DNN. The DNN method has a low computational complexity while exhibiting a performance very close to the commonly used bisection search heuristic. The cost of using a DNN is the lengthy training required. But this should not be a problem in practice since it is done offline, before the network becomes operational. We demonstrated the performance of the DNN solution using a particular network configuration and scenario. Similar performance, i.e., the close approximation of the behavior of a known algorithm, should in principle hold for any network and scenario, since DNNs have been proven, given enough training, to be capable of approximating any function arbitrarily close. The numerical results, obtained by Monte Carlo simulations, showed that the proposed DSL method has a competitive performance compared to the existing algorithms and the execution time is much lower. The key points are listed as follows:

- The Rayleigh channel model is used for the analysis in CF sub-6 GHz massive MIMO.

- The APs estimate the channel via the uplink pilots in each coherence time. Accordingly, the estimated channel is used to perform the beamforming in each coherence time.

- Because of the geographical spreading of antennas, perform local signal processing using CB to avoid CSI exchange between the CC and the APs.
- Benefiting from the property of channel hardening, the UatF bound, which neglects the effect of small-scale fading, is derived as a closed-form expression to calculate the SE. Therefore, the power allocation can be performed on the large-scale fading time.
- The DSL method is used to approximate the bisection algorithm to implement the max-min power allocation scheme. Because the target of the DSL method is to approximate a known algorithm, therefore the per-UE SE achieved by the DSL method is typically worse than the algorithm that is being approximated.

Chapter 3

Max-sum SE power allocation in CF sub-6 GHz massive MIMO

In this chapter, we address **Problem 2** and investigate solutions that use DRL-based methods. In Section 3.1, we formalize the max-sum SE power allocation problem. In Section 3.2, we present the DRL methods for the optimization. Then in Section 3.3, we evaluate the performance of the proposed DRL methods by comparing with the WMMSE algorithm. Finally, Section 3.4 summarizes the key points of this chapter. The system model for analysis is based on Section 2.1.

3.1 Max-sum SE power allocation scheme

The max-sum SE power allocation scheme aims at maximizing the network capacity, which is formulated as follows:

$$\begin{aligned} &\max_{p_{k,n}} \sum_{k=1}^{K} SE_k \\ &s.t. \sum_{k=1}^{K} p_{k,n} \leq p_l, \forall n \end{aligned} \tag{3.1}$$

where p_l is the transmission power limit of each AP. Like (2.13), the problem in (3.1) is non-convex and NP-hard, since the computational complexity increases exponentially as N and K increase linearly.

A well-adopted method to solve (3.1) is the WMMSE algorithm, which converts the sum-SE maximization problem to an equivalent minimization problem of the MSE [12]. Specifically, the algorithm works as follows:

From an initial point $v_{k,n}^0$ satisfying the constraints in (3.1), the optimal power allocation is obtained by updating $\{v_{k,n}, u_{k,n}, w_{k,n}\}$ in an iterative manner, where

$v_{k,n}, u_{k,n}, w_{k,n}$ are optimization variables. Specifically, $v_{k,n}$ denotes the root of the normalized power, $u_{k,n}$ represents the MSE of $v_{k,n}$, $w_{k,n}$ represents the weight of the MSE. The variables $v_{k,n}, u_{k,n}, w_{k,n}$, for $k = 1,2,...,K$ and $n = 1,2,...N$ in iteration i, are updated using (3.2) (3.3) and (3.4), where (3.4) implies that the variable $v_{k,n}$ should be in range from 0 to $\sqrt{p_l/K}$. The WMMSE algorithm for solving (3.1) is given in Algorithm 3. The algorithm stops when the condition $w_{k,n} < \epsilon$ is fulfilled. The value of ϵ depends on the convergence behavior of the WMMSE algorithm. Like in [12] we set $\epsilon = 0.01$.

$$u_{k,n}^{i} = \frac{\sqrt{M} v_{k,n}^{i-1} \sqrt{\gamma_{k,n}}}{M \sum_{k'=1}^{K} (\sum_{n'=1}^{N} v_{k',n'}^{i-1} \sqrt{\gamma_{k',n'}} \frac{\beta_{k,n'}}{\beta_{k',n'}})^2 |\boldsymbol{\psi}_k \boldsymbol{\psi}_{k'}^H|^2 + \sum_{k'=1}^{K} \sum_{n'=1}^{N} (v_{k',n'}^{i-1})^2 \beta_{k,n'} + 1} \tag{3.2}$$

$$w_{k,n}^{i} = \frac{1}{1 - u_{k,n}^{i-1} \sqrt{M \gamma_{k,n}} v_{k,n}^{i-1}} \tag{3.3}$$

$$v_{k,n}^{i} = \left[\frac{w_{k,n}^{i} u_{k,n}^{i} \sqrt{M \gamma_{k,n}}}{\sum_{k'=1}^{K} \sum_{n'=1}^{N} w_{k',n'}^{i} (u_{k',n'}^{i})^2 M \gamma_{k',n'}} \right]_{0}^{\sqrt{p_l/K}} \tag{3.4}$$

Algorithm 3 Pseudo code of the WMMSE algorithm.

1: Initialize $v_{k,n}^{0}$ such that $(v_{k,n}^{0})^2 \leq p_l/K, \forall k, n$.
2: Set $i = 1$, repeat:
3: Update the variables $u_{k,n}^{i}$ for all k, n, by using (3.2).
4: Update the variables $w_{k,n}^{i}$ for all k, n, by using (3.3).
5: Update the variables $v_{k,n}^{i}$ for all k, n, by using (3.4).
6: Set $i = i + 1$.
7: Stop until $w_{k,n} < \epsilon$.
8: Output: $p_{k,n} = (v_{k,n})^2$.

The computational complexity mainly lies in Step 3 and 5. The calculation of $\gamma_{k,n}$ has a complexity of $O(K)$. For the denominator of (3.2) and (3.4), the complexity is $O(KN^2)$. The computational complexity for updating $p_{k,n}$ is $O(I_W KN^2)$, where I_W represents the number of iterations. Finally, the total computational complexity to update KN links is $O(I_W N^2 K^3)$.

3.2 DRL for max-sum SE power allocation

Like the bisection algorithm for max-min power allocation, the computational complexity of the WMMSE algorithm is problematic for max-sum SE power allocation.

Here, we use the DRL methods to perform the task. On the on hand, one could find that the performance of the DSL method is usually slightly worse than the algorithm that is being approximated, e.g., in [31] and [38]. This is because the training target of the DSL method is to approximate, which implies that when the algorithm that is being approximated falls into a local optimum, the DSL accordingly obtains a sub-optimal solution. Unlike the DSL method, the training of a DRL is done by means of rewards obtained by trial-and-error interactions with the environment. The training target of the DRL is to maximize these rewards, which enables the DRL to potentially get better performance. On the other hand, the regime of the DRL method fairly matches the max-sum SE scheme. The sum SE can be directly set as the rewards to maximize; therefore, it is easy to implement the DRL method to achieve the max-sum SE scheme.

As discussed in the previous section, the max-min scheme is more complicated than the max-sum SE scheme in CF massive MIMO. The DRL-based max-min power allocation in CF massive MIMO is an open issue that needs to be further studied. One could expect that the DRL method works better than the bisection algorithm in terms of per-UE SE and execution time.

Recently, several DRL-based methods have been applied to solve power allocation problems in wireless communication networks, e.g., [73–77]. Specifically, [73] proposed a DQN-based method power allocation method to maximize the overall capacity in a cellular network. Similarly, in [74], the authors used a multi-agent DRL method to maximize the total capacity in cellular networks. In [75], the authors proposed a DDPG-based method to allocate the power in a cellular network. Compared to the DQN-based method, the DDPG method achieves better performance in terms of sum SE. [76] proposed a multi-agent proximal policy optimization method for power allocation in a two-tier heterogeneous network. The numerical results show that the proposed method achieves better performance than existing algorithms in terms of the overall throughput. [77] proposed a Deep-Q-Full-Connected-Network to maximize the overall capacity in a multi-cell network. It has been shown that the proposed method achieves a higher overall capacity than the traditional water-filling algorithm.

All the above works considered single-antenna BSs in multi-cell networks, in this section we propose DRL-based power allocation for CF sub-6 GHz massive MIMO. Two DRL methods are investigated, namely the DQN and the DDPG.

3.2.1 Background of DRL

DRL is a category of machine learning where an agent learns by interacting with its dynamic environment through a repeated sequence of observations, actions and rewards [37]. At time slot t, where t is an integer, by observing the state s^t, the agent takes action $a^t \in \boldsymbol{A}$ according to a certain policy π, then gets the reward r^t

from the environment and enters the next state s^{t+1}. The four-tuple (s^t, a^t, r^t, s^{t+1}), describes a single interaction with the environment. We define e^t, where $e^t = (s^t, a^t, r^t, s^{t+1})$ as an experience sequence. The agent aims to find the optimal policy to maximize the cumulative reward:

$$R^t = r^t + \omega r^{t+1} + \omega^2 r^{t+2} + ... \tag{3.5}$$

where $\omega \in [0, 1)$ is a discount factor that trades off the importance of immediate and future rewards. To evaluate a policy, the action value is defined:

$$Q^\pi(s, a) = \mathbb{E}_\pi\{R^t | s^t = s, a^t = a\} \tag{3.6}$$

where the right side of (3.6) is the expected reward once action a^t is taken under state s^t and policy π. Reinforcement learning makes use of the Bellman equation [78] for the cumulative calculation:

$$Q^\pi(s, a) = \mathbb{E}\{r^t + \omega Q^\pi(s', \pi(s')) | s^t = s, a^t = a, s^{t+1} = s'\} \tag{3.7}$$

The optimal policy op, is the one that maximizes (3.7), i.e.,

$$op = \arg\max_\pi Q^\pi(s, a) \tag{3.8}$$

For convenience, we use $Q^{op}(s, a)$ to represent the action value of the optimal policy. The main two methods to find the optimal policy are: the value-based method DQN and the policy-based method DDPG.

3.2.2 DQN method

The key to find the optimal policy for the DQN method is to obtain the action-value function of $Q^\pi(s, a)$. It is common to use a function approximator to estimate the action-value function, typically a lookup table or a linear function. If this approximator is a DNN, it is called DQN. The DQN is defined as $Q(s, a, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the parameters (weights between neurons). The DQN is trained to estimate the optimal action-value function, i.e., $\boldsymbol{\theta}$ is updated to estimate the action-value function of $Q^\pi(s, a)$. The agent stores the experiences in the dataset $D = \{e^1, e^2, \dots e^t\}$, which is used to train the DQN by the gradient descent algorithm [13]. As with the DSL, the training of the DQN uses random minibatches. The loss function of the DQN for a random minibatch D^t (random sample over D) at time slot t is:

$$L(\boldsymbol{\theta}) = \sum_{e \in D^t} \underbrace{(r + \omega \max_{a'} Q(s', a', \hat{\boldsymbol{\theta}})}_{\text{target}} - Q(s, a, \boldsymbol{\theta}))^2 \tag{3.9}$$

where $e = (s, a, r, s')$, $\hat{\boldsymbol{\theta}}$ represents the network parameters to compute the target at time slot t, which is only updated every C time slots. Finally the optimal parameters $\boldsymbol{\theta}^{op}$ is:

$$\boldsymbol{\theta}^{op} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{3.10}$$

To use the DQN for solving the power allocation problem in CF massive MIMO, we define the duration of each time slot t as a time unit during which the large-scale fading stays constant, i.e., the large-scale time τ_l in Fig.1.4. Referring to [73] and [74], we define for each AP-UE link an agent, thus the power allocation is performed by a multi-agent system. The agents interact with the environment to collect training data and each agent contains a DQN, which outputs the Q-value of the action (see Fig.3.1). The optimal policy follows the maximization of the Q-value. There is a total of KN agents in the whole system. At time slot t, each agent (k, n) allocates the transmit power from AP n to UE k. One should note that the proposed multi-agent system follows a centralized training with a decentralized execution framework:

The agents are deployed in the APs, accordingly each AP consists of K agents. There is an extra agent, named **Agent 0**, in the CC. During the training, **Agent 0** is randomly initialized. The agents in the APs copy the parameters of **Agent 0** to allocate the power; the interaction information $e^t = (s^t, a^t, r^t, s^{t+1})$ is recorded and then sent to the CC. In each time slot, NK data samples are collected by the CC. **Agent 0** is trained by the collected interaction information of all agents.

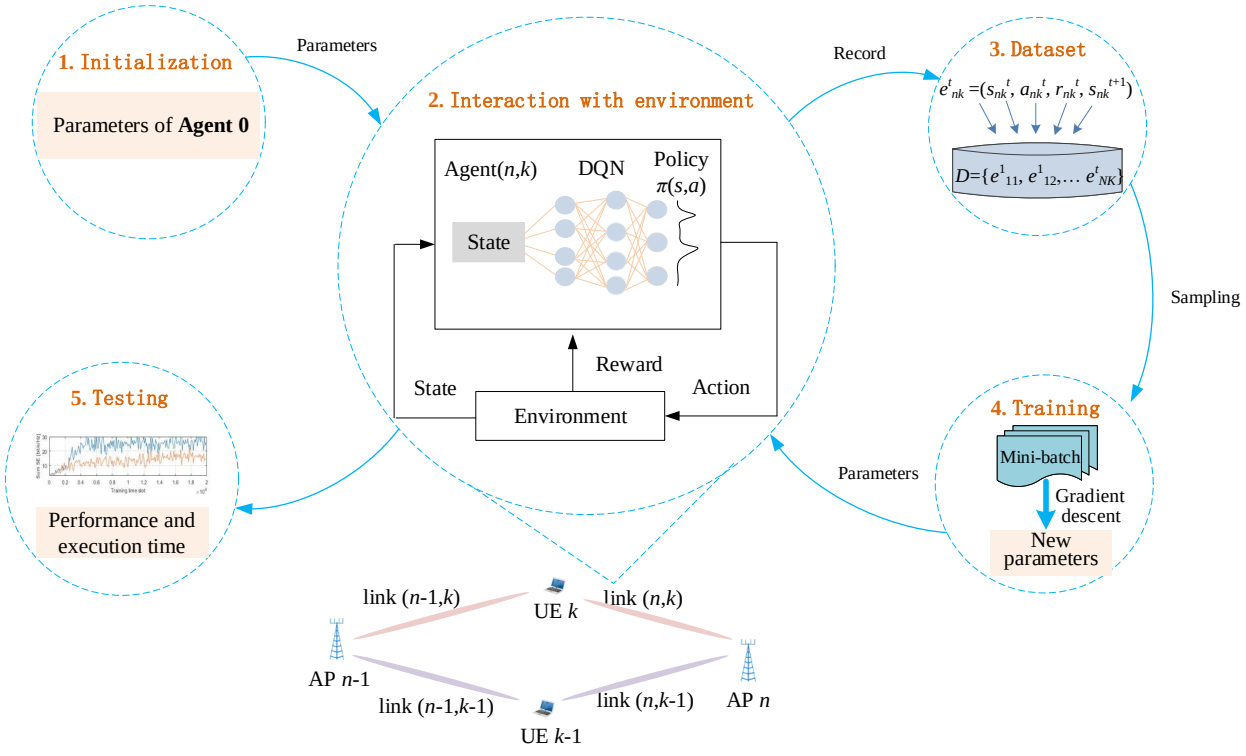


Figure 3.1: DQN-based power allocation method.

In every C (defined as the training interval) time slots, the agents in the APs update their DQN parameters by copying the new parameters from **Agent 0**, as shown in Fig.3.1.

We define $e_{k,n}^t = (s_{k,n}^t, a_{k,n}^t, r_{k,n}^t, s_{k,n}^{t+1},)$ as the experience sequence of agent (k,n) at time slot t. The DQN is trained by the dataset $D = \{e_{1,1}^1, e_{1,2}^1, ..., e_{k,n}^t \ldots\}$, which describes the agents' relation with their environment. The key to use the DQN for solving (3.1) is to model the decision variables as the action of agents. Obviously the normalized downlink transmission power $p_{k,n}$ is the decision variable for SE, therefore the action of agent (k,n) is $p_{k,n}$. As a time series we define $p_{k,n}^t$ as the action of agent (k,n) at time slot t. The agent (k,n) takes action according to the current state $s_{k,n}^t$, which features the independent variables. From (2.12) we find that the large-scale fading is the independent variable for SE, therefore the large-scale fading information is a key element for $s_{k,n}^t$. The objective function, which describes the target of the agents, can be defined as the reward in each time slot. Based on the above analysis, the elements of the experience $e_{k,n}^t$. for CF massive MIMO power allocation are defined as following:

1) State $s_{k,n}^t$:

The SINR is the key element of the SE. The *signal* in SINR of UE k comes from the agent set $\{(k, 1), (k, 2), ..., (k, N)\}$, while the *interference* in SINR of UE k comes from the agent set $\{(k', 1), (k', 2), ..., (k', N)\}$, where $k' = 1, 2, ..., K$ and $k' \neq k$. The agent (k, n) allocates the power considering the *signal* of its own link and the *interference* caused by other links. Therefore, the information $\beta_{k,n}^t$ and $\boldsymbol{\zeta_1} = \{\beta_{k',1}^t, \beta_{k',2}^t, ..., \beta_{k',N}^t\}$ are the components of the state $s_{k,n}^t$. Referring to [74], the information of power allocation and SE at time slot $t-1$ can improve the sum-SE performance of the DQN, therefore we also consider $p_{k,n}^{t-1}$, $\boldsymbol{\zeta_2} = \{p_{k',1}^{t-1}, p_{k',2}^{t-1}, ..., p_{k',N}^{t-1}\}$ and $\boldsymbol{\zeta_3} = \{SE_1^{t-1}, SE_2^{t-1}, ..., SE_K^{t-1}\}$ as the components of the state $s_{k,n}^t$. Finally, $s_{k,n}^t$ is formalized as follows:

$$s_{k,n}^t = \{\beta_{k,n}^t, \boldsymbol{\zeta_1}, p_{k,n}^{t-1}, \boldsymbol{\zeta_2}, \boldsymbol{\zeta_3}\} \tag{3.11}$$

The size of $s_{k,n}^t$, i.e., the input dimension of the DQN, is $2N(K-1)+K+2$. To reduce the input dimension and the complexity, the elements in $\boldsymbol{\zeta_1}$ are sorted in decreasing order and only the first L components remain. The corresponding links of components are also reduced in $\boldsymbol{\zeta_2}$. Therefore, there are $2(1+L)+K$ components remaining in $s_{k,n}^t$.

2) Action $a_{k,n}^t$

The allocated power is a continuous value, while the agent's action is a discrete value. Therefore, we should discretize the transmission power as follows:

$$\mathbf{A} = \left\{0, \frac{p_l}{K|\mathbf{A}|-1}, \frac{2p_l}{K|\mathbf{A}|-1}, ..., \frac{p_l}{K}\right\} \tag{3.12}$$

where $|\mathbf{A}|$ represents the number of power levels.

3) Reward $r_{k,n}^t$

The target is to maximize the sum SE. Therefore, the reward is the sum SE at time slot t:

$$r_{k,n}^t = \sum_{k=1}^{K} SE_k^t \tag{3.13}$$

3.2.3 DDPG method

For the DDPG-based method, a DNN is used as a policy network, that is, the output of the DNN is the action. DDPG is an actor-critic method [79]: an actor $Ac(\boldsymbol{\delta}_a)$ takes action a by observing the state s, where $Ac(\boldsymbol{\delta}_a)$ is the policy network and $\boldsymbol{\delta}_a$ represents the parameters of this network. A critic $Cr(\boldsymbol{\delta}_c)$ evaluates the action a with the critic state, where $Cr(\boldsymbol{\delta}_c)$ is the critic network (a different DNN) and $\boldsymbol{\delta}_c$ represents the parameters of this network, see Fig. 3.2. The optimal policy by DDPG is:

$$op_{DDPG} = \arg \max_{Ac(\boldsymbol{\delta}_a^{op})} Cr(\boldsymbol{\delta}_c^{op}) \tag{3.14}$$

The actor $Ac(\boldsymbol{\delta}_a)$ and the critic $Cr(\boldsymbol{\delta}_c)$ cooperate to get the optimal parameters of $\boldsymbol{\delta}_a^{op}$ and $\boldsymbol{\delta}_c^{op}$. Similarly to the DQN-based method, a random minibatch D^t is used to train $Ac(\boldsymbol{\delta}_a)$ and $Cr(\boldsymbol{\delta}_c)$. The loss functions are defined as follows:

$$L(\boldsymbol{\delta}_a) = Cr(\boldsymbol{\delta}_c)|_{a=Ac(\boldsymbol{\delta}_a)} \tag{3.15}$$

$$L(\boldsymbol{\delta}_c) = Cr(\boldsymbol{\delta}_c)|_{a=Ac(\boldsymbol{\delta}_a)} - r \tag{3.16}$$

Finally the optimal parameters $\boldsymbol{\delta}_a^{op}$ and $\boldsymbol{\delta}_c^{op}$ are:

$$\boldsymbol{\delta}_a^{op} = \arg\max_{\boldsymbol{\delta}_a} L(\boldsymbol{\delta}_a) \tag{3.17}$$

$$\boldsymbol{\delta}_c^{op} = \arg\min_{\boldsymbol{\delta}_c} L(\boldsymbol{\delta}_c) \tag{3.18}$$

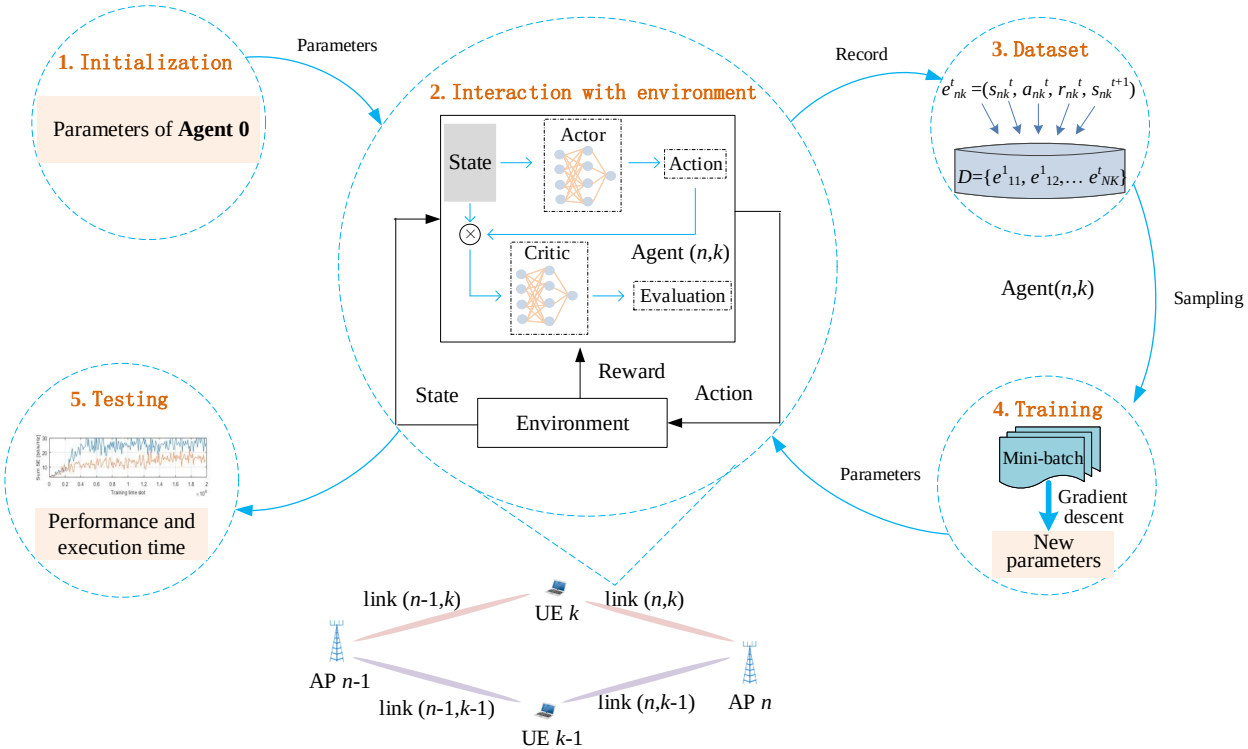


Figure 3.2: DDPG-based power allocation method.

(3.17) implies that the actor $Ac(\boldsymbol{\delta}_a)$ strives to get the maximum value of the evaluation from the critic; (3.18) aims to get the precise assessment from the critic $Cr(\boldsymbol{\delta}_c)$.

Similar to the DQN method, the DDPG method uses the multi-agent system to train $Ac(\boldsymbol{\delta}_a)$ and $Cr(\boldsymbol{\delta}_c)$. The agents interact with the environment to collect training data and each agent contains an actor network and a critic network. The actor network outputs the action; the critic network evaluates the action. The optimal policy follows the action that has the maximum evaluation. It is a centralized training with a decentralized execution framework, see Fig. 3.2. The state s and reward r are the same as in the DQN method. The only difference between the DQN method and the DDPG method is the action a. The output of the DQN is the action value of different actions, and is determined by (3.8). The output of actor $Ac(\boldsymbol{\delta}_a)$ is

$$a_{k,n}^t = \left[Ac(\boldsymbol{\delta}_a)|_{s_{k,n}^t} \right]_0^{p_l/K} \tag{3.19}$$

resulting in a continuous value, which is different from the action generating discrete values obtained by the DQN-based method in (3.12).

3.2.4 Computational complexity of the DRL method

The computational complexity of the DRL methods considers the operational phase, i.e., the time that a trained DRL performs the power allocation. For a fully connected neural network, the complexity is $\mathrm{O}(\nu\mu^2)$, where ν is the number of layers and μ is the number of neurons in the layers. The number of neurons in each layer depends on the dimension of the input layer, i.e., $\mathrm{O}(\mu)=\mathrm{O}(N+K+1)$ in our case. The number of layers for a neural network is independent of the scale of the problem. Therefore, the computational complexity of one agent is $\mathrm{O}(N^2+K^2)$. As we adopted a multi-agent system, the total computational complexity of the proposed DRL methods is $\mathrm{O}(N^3K+NK^3)$. Compared to the computational complexity of the WMMSE algorithm $\mathrm{O}(I_W N^2 K^3)$, the DRL methods show an even higher computational complexity. However, one should note that the agents work in parallel, which implies that the execution time of the DRL methods only counts for one single agent operation. We will show later in Section 3.3.4, that the execution time of the DRL method is much less than the WMMSE algorithm.

3.3 DRL vs WMMSE algorithm

In this section we show by simulations that the DRL-based power allocation methods in CF massive MIMO are competitive in terms of performance and complexity, compared to the non-DL WMMSE algorithm.

3.3.1 Scenario and Configuration

We consider the 3GPP TR 38.901 indoor mixed office scenario (120m×50m×3m) [80] with 12 APs, positioned as shown in Fig.3.3. Each AP contains 10 antennas in a horizontally mounted and downward radiating 2×5 UPA at a height of 3m. We assume $K = 10$ single-antenna UEs moving within the coverage area. Each UE moves in a random direction (up, down, left, and right) with a randomly chosen velocity distributed uniformly between 0 and 1 m/s. We consider a discrete time system where the duration of each time slot is 3 s, corresponding to 40 coherence times. For a UE velocity of 1 m/s, the channel coherence time τ_c is about 75 ms, calculated as (1.2). Each UE maintains its speed and direction in each second before selecting a new speed and direction. The initial positions of the UEs at time $t = 0$, are uniformly distributed over the coverage area (Fig.3.3). We model the large-scale fading as the combination of pathloss and shadowing, as in [80].

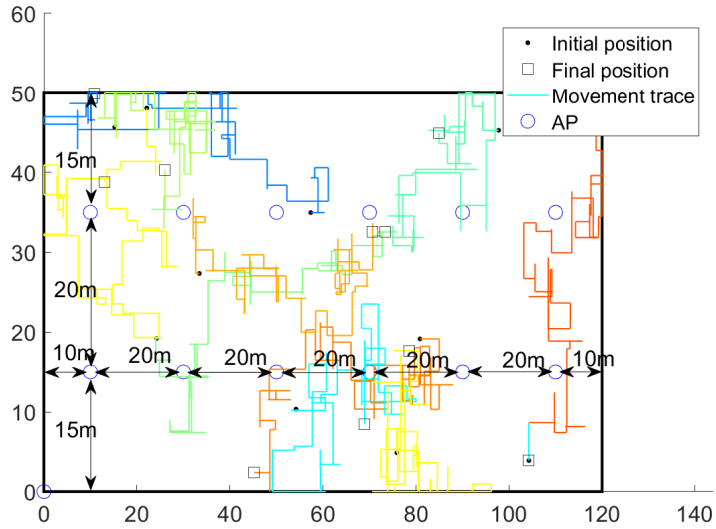


Figure 3.3: Example of UE movement traces in a 3GPP TR 38.901 scenario for 100 time slots.

The carrier frequency is 2 GHz, the bandwidth is 20 MHz. The maximum power constraint, i.e., p_l, is 23 dBm and the noise power is assumed to be -94 dBm. The uplink pilot power is 20 dBm. We set the coherence time to 200 modulation symbols as in [11]. The length of the uplink pilot is 5 symbols for channel estimation in our simulations. As discussed in Section 2.1.2, when $\tau_p < K$ some pilot sequences

Table 3.1: Parameters used in simulations

Parameter	value
Coverage volume	120m×50m×3m
K, number of UEs	10
M, number of antennas per AP	10
N, number of APs	12
p_l, maximum power constraint	23 dBm
p_p, pilot power	20 dBm
τ_c, length of coherence time in symbols	200
τ_p, length of pilot in symbols	5
Carrier frequency	2 GHz
Bandwidth	20 MHz
Noise power	-94dBm
Distribution of UE velocity	U(0, 1) m/s
Time slot duration	3 s

are reused, hence the simulations take the pilot contamination into consideration. The parameters used in the simulations are summarized in Table 3.1.

3.3.2 Architecture of the DNN

We next describe the hyperparameters used for the DQN method and the DDPG method. We first tried three DNN candidate architectures to investigate the impact of the number of hidden layers and the number of neurons per hidden layer on the training process of both the DQN and the DDPG.

For the DRL-based power allocation in cellular networks, [73] proposed a DNN using two fully connected hidden layers with 128 and 64 neurons, respectively; [74] proposed a DNN using three fully connected hidden layers with 200, 100 and 40 neurons, respectively. Based on this, we chose the candidate architectures of the DNNs as follows:

(1) DNN_1 with two fully connected hidden layers, with 128 and 64 neurons, respectively;

(2) DNN_2 with two fully connected hidden layers, with 256 and 128 neurons, respectively;

(3) DNN_3 with three fully connected hidden layers, with 256, 128 and 64 neurons, respectively.

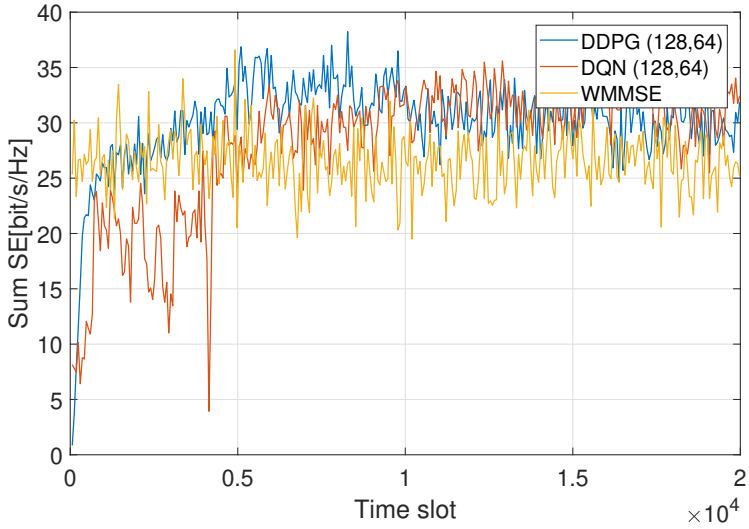
The number of neurons in the input layers for the DNNs are $2(1+L)+K$ as we explained in the description of state $s^t_{k,n}$, see (3.11). We set $L = 20$ therefore 52 neurons in the input layer. The selection of L affects the number of interfering

links for a given agent. The value of L is in the range [0, KN-1]. A larger value of L makes the power allocation of the agent closer to the optimal solution but results in a higher complexity. Referring to [73] using $L = 16$ and [74] using $L = 5$, we choose $L = 20$ to gain a near-optimal power allocation.

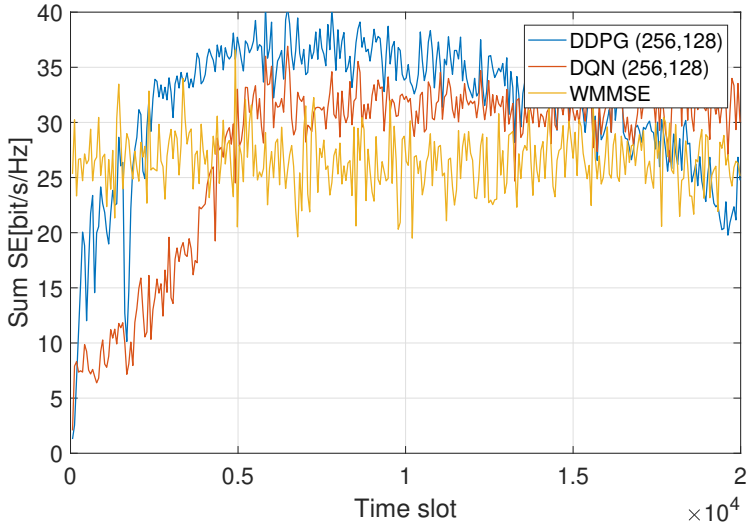
For the DQN method, the number of power levels is set to 10, hence the number of neurons in the output layer for $Q(s, a, \boldsymbol{\theta})$ is also 10. For the DDPG method, since the output of the actor network is an action, i.e., the allocated power, there is only one neuron in the output layer of $Ac(\boldsymbol{\delta}_a)$; the critic network produces an evaluation value of the action, therefore there is one neuron in the output layer of $Cr(\boldsymbol{\delta}_c)$. All the active functions of the DNNs use the ReLU activation function, except for the output layer of $Ac(\boldsymbol{\delta}_a)$, which uses the Sigmoid function to obtain the normalized allocated power.

There are other hyperparameters affecting the training process of the DQN-based method and the DDPG-based method, namely, the discount factor ω, the training interval C, the initial adaptive learning rate lr, the adaptive ϵ-greedy algorithm and the minibatch size $|D^t|$. Adaptive learning means that the learning rate decays with the number of training time slots. Generally, a large learning rate allows the model to learn faster but may end up with a sub-optimal final set of weights. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer. Adaptive learning balances the training time and performance. The ϵ-greedy algorithm is a learning method that makes use of the exploration-exploitation tradeoff, in which the agent takes a random action with probability ϵ or takes an action using the policy of DQN or DDPG with probability $1-\epsilon$. A random action may cause the training to "jump out" of a local optimum and explore new convergence regions. In the adaptive ϵ-greedy algorithm the value of ϵ decays each training time slot. A large ϵ avoids the training ending up in local optima during the initial training time slots, a small value of ϵ makes sure that the training will converge in the later training time slots.

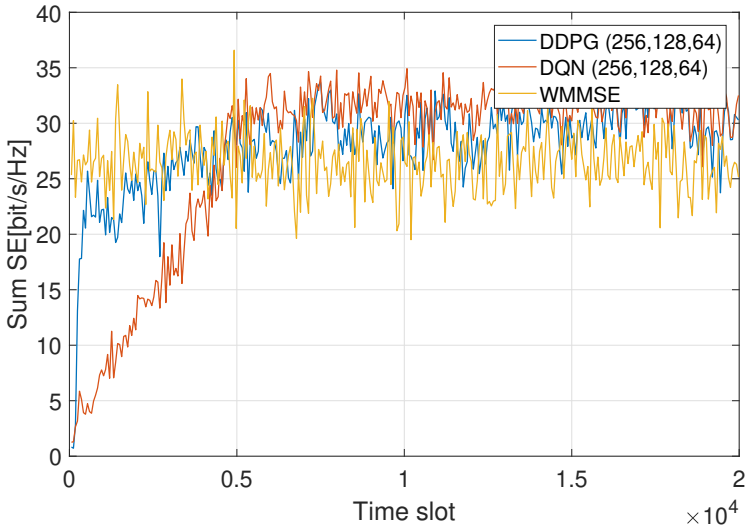
Referring to the literature, we choose: $\omega \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ [37], [73]; $C \in \{10, 50, 100, 200, 500\}$ [37], [79]; $lr \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ [13], [59]; $\epsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ [73], [74]; $|D^t| \in \{500, 1000, 2000, 5000, 10000\}$ [59] to find the optimal hyperparameters. We found that for different values of ω, C and $|D^t|$, the behavior of the sum SE as a function of the training time slot is very similar, implying our proposed DRL-based methods are not very sensitive to these hyperparameters. However, for different values of lr and ϵ, we get very different behaviors of the sum SE as a function of the training time slot. This is because the learning rate and ϵ-greedy algorithm affect the exploration-exploitation tradeoff, which is important for the updating of the DNN parameters, i.e., weights and bias. We chose the parameters $\omega = 0.1$, $C = 10$, $lr = 0.005$, $\epsilon = 0.1$ and $|D^t| = 500$, which gave us the highest sum-SE values, in our simulations.



(a) Training process of DQN and DDPG with DNN_1.



(b) Training process of DQN and DDPG with DNN_2.



(c) Training process of DQN and DDPG with DNN_3.

Figure 3.4: Sum SE as a function of time during the training of the DQN-based method and the DDPG-based method for three candidate architectures of DNN.

Fig.3.4 shows the result of the training process, i.e., the sum SE as a function of the training time slot, where the DQN and DDPG use the three DNN candidate architectures. The length of the training period we choose is 20,000 time slots, corresponding to roughly 16.6 hours. The time it takes for the sum SE to converge was estimated by the naked eye, i.e., a longer training period does not result in fluctuations around a significantly different value of sum SE. The fluctuations of the sum SE in Fig.3.4 are caused by the random mobility of the UEs.

Fig.3.4 (a) shows the training process of DQN and DDPG for the DNN_1 architecture: the DQN method and the DDPG method converge to fluctuations around 30 bit/s/Hz during 5,000 training time slots. It is obvious that the DDPG method converges faster than the DQN method, by observing that the blue line is always higher than the red line during most of the training time slots between 0 and 5,000. Afterwards, the DQN method and the DDPG method achieve similar sum-SE values, that are higher than the values obtained by the WMMSE algorithm which is shown as a reference.

Fig.3.4 (b) shows the training process of DQN and DDPG for the DNN_2 architecture: the DQN method converges to fluctuations around 32 bit/s/Hz during 5,000 training time slots, achieving higher values than the WMMSE algorithm. For the DDPG method, it converges to fluctuations around almost 40 bit/s/Hz through

7,000 training time slots. Afterwards however, the sum SE decreases at around the 15,000th time slot. Moreover, the sum-SE values of the DDPG-based method become even lower than the WMMSE algorithm.

Fig.3.4 (c) shows the training process of DQN and DDPG for the DNN_3 architecture: similar observations can be made as for Fig.3.4 (a), the DQN method and the DDPG method converge to fluctuations around 32 bit/s/Hz through 5,000 training time slots. Afterwards, the DQN method and the DDPG method achieve similar sum-SE values, higher than the WMMSE algorithm. In addition, the DDPG method converges faster than the DQN method.

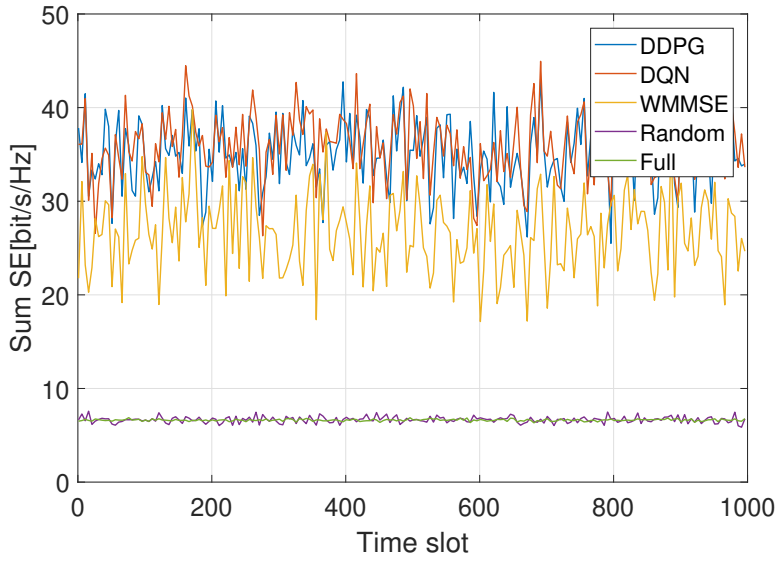
Based on the above observations, we choose DNN_1 as the architecture for the DQN method and the DDPG method for the following reasons: (1) the DDPG method has higher sum-SE values for DNN_1 than for DNN_2, (2) DNN_1 achieves a similar performance as DNN_3, which has one more layer implying longer execution time.

Fig.3.5 (a) shows the sum SE over a period of 1,000 time slots, Fig.3.5 (b) shows the empirical CDF of the sum SE. As expected, the DQN-based method and the DDPG-based method have better performance than the three benchmark algorithms. The random mobility of UEs causes the fluctuations of the sum SE for the above methods in Fig.3.5 (a). Although the differences are not pronounced in Fig.3.5 (b), it is clear that the DQN-based method slightly outperforms the DDPG-based method, by observing that the red line is to the right of the blue line in Fig.3.5 (b). Nevertheless, we can conclude that the DQN-based method and the DDPG-based method achieve a similar performance, which results in fluctuations around a value about 10bit/s/Hz higher than the WMMSE algorithm.

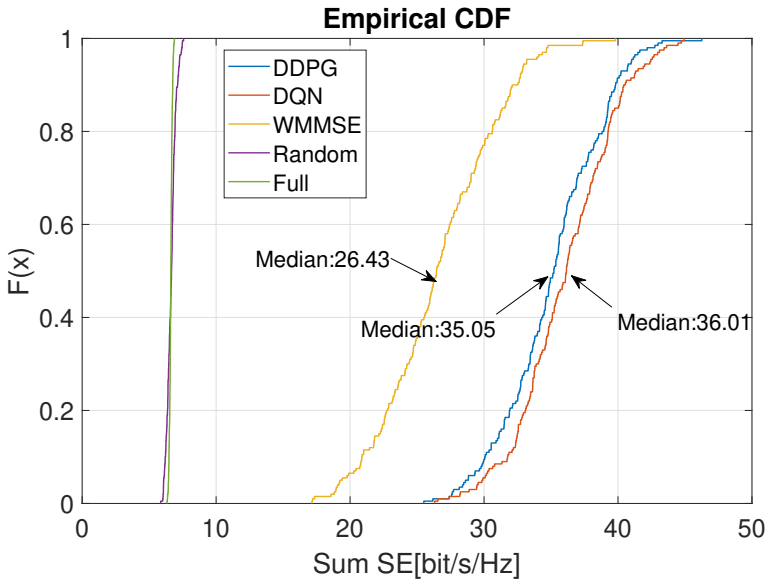
3.3.3 Sum-SE performance

To evaluate the performance of the DQN-based method and the DDPG-based method, we have used three benchmark algorithms. The first benchmark is the WMMSE algorithm. The second is random power allocation where $p_{k,n} \sim \mathbb{U}(0, p_l/K)$ for $k = 1,2,...,K$ and $n = 1,2,...N$. The third one is full-power allocation, i.e., $p_{k,n} = p_l/K$ for $k = 1,2,...,K$ and $n = 1,2,...N$. For the DQN-based method and the DDPG-based method we used the DNN_1 architecture. The training period is 20,000 time slots. After training we run the systems for 1,000 time slots, during which we record the sum-SE values obtained by the five power allocation methods, as shown in Fig.3.5. We chose 1,000 time slots for testing and comparing our proposed DRL-based methods, based on the fact that in [73], for the same purpose, 500 time slots were estimated to be sufficient to have a reliable comparison.

We then ran some experiments with different values of N, K and M. We first set $N = 8$ while keeping $K = 10$ and $M = 10$. The APs are positioned as shown in Fig.3.6. The UE mobility model is kept the same as in Fig.3.3.



(a) Sum SE as a function of time slot.



(b) CDF of the sum SE.

Figure 3.5: Comparison of the sum SE over 1,000 time slots with $N = 12$, $K = 10$ and $M = 10$.

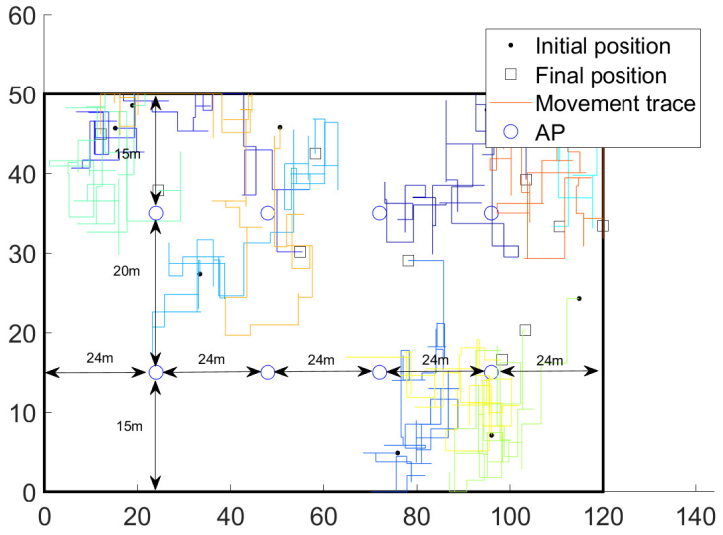
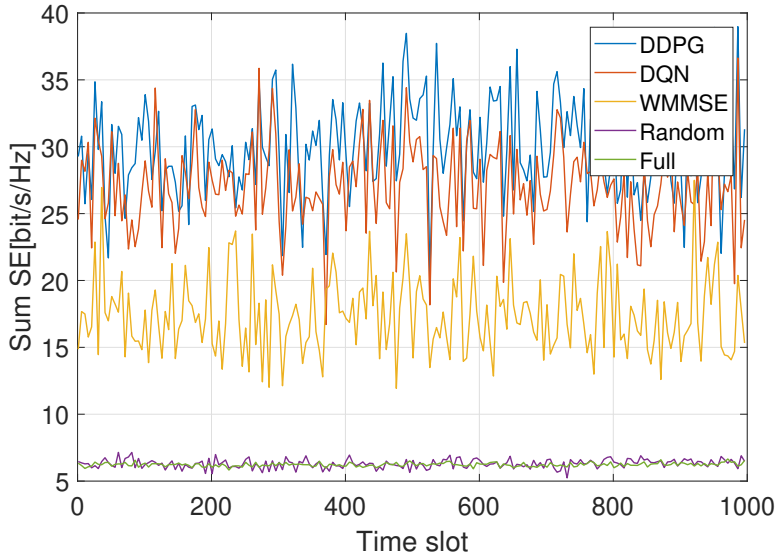


Figure 3.6: Example of UE movement traces in a 3GPP TR 38.901 scenario for 100 time slots with $N = 8$ APs.

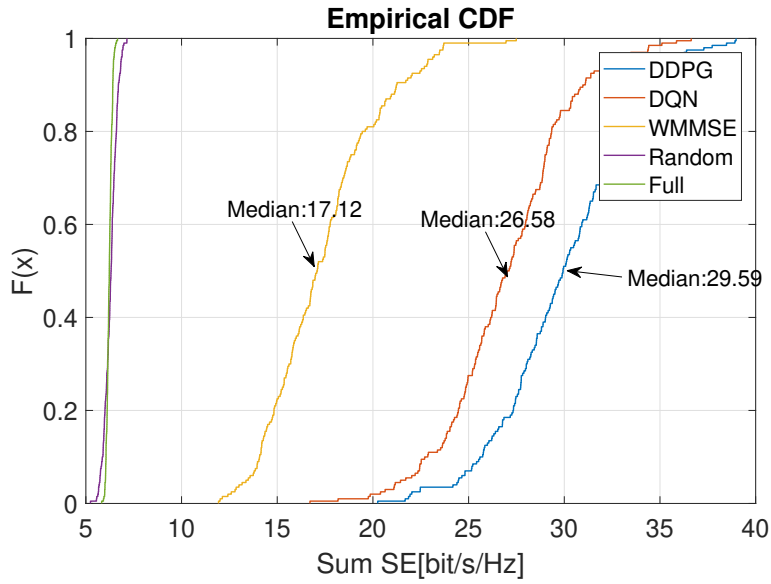
Fig.3.7 shows the simulation result over a period of 1,000 time slots. Observe from Fig.3.7 (a) that DQN and DDPG still have much better performance than the WMMSE algorithm. The DQN method has a sum-SE average that is 9 bit/s/Hz higher than the value for the WMMSE algorithm. For DDPG the difference is 13 bit/s/Hz. It is also clear that DDPG works better than DQN, by observing that the blue line is to the right of the red line in Fig.3.7 (b).

Next, we set $K = 5$ while keeping $N = 12$ and $M = 10$. Fig.3.8 shows the simulation result over a period of 1,000 time slots. Observe from Fig.3.8 (a), we find that the DQN method, the DDPG method and the WMMSE algorithm achieve similar sum-SE performance. However, the empirical CDF in Fig.3.8 (b) shows that the DQN method has lower sum-SE values than the WMMSE algorithm and DDPG method. From Fig.3.8 we conclude that the DRL-based power allocation methods are sensitive to the number of UEs. This implies that, when the number of UEs changes, the DQN and DDPG methods are expected to be continuously trained to get better performance. The length of the training process depends on the convergence of the sum SE.

Finally, we set $M = 15$ while keeping $N = 12$ and $K = 10$. Fig.3.9 shows the simulation result over a period of 1,000 time slots. The result in Fig.3.9 is very similar to Fig.3.5. Because the power allocation in CF massive MIMO is optimized

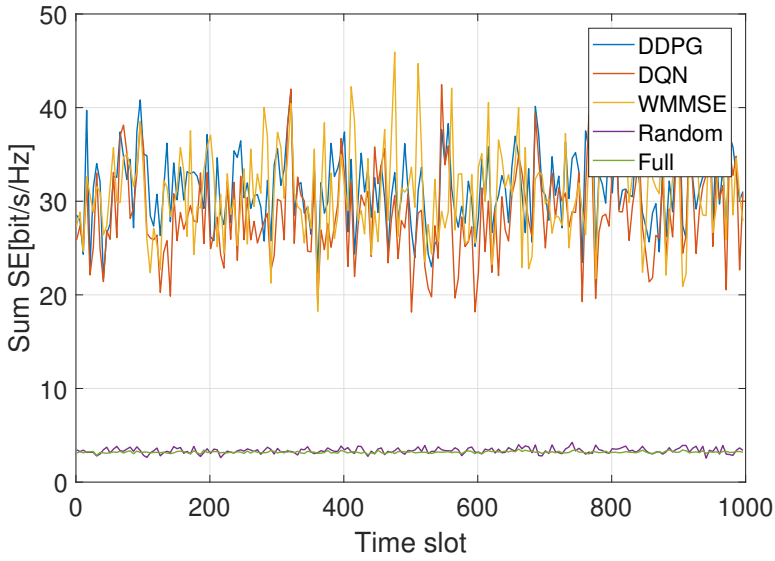


(a) Sum SE as a function of time slot.

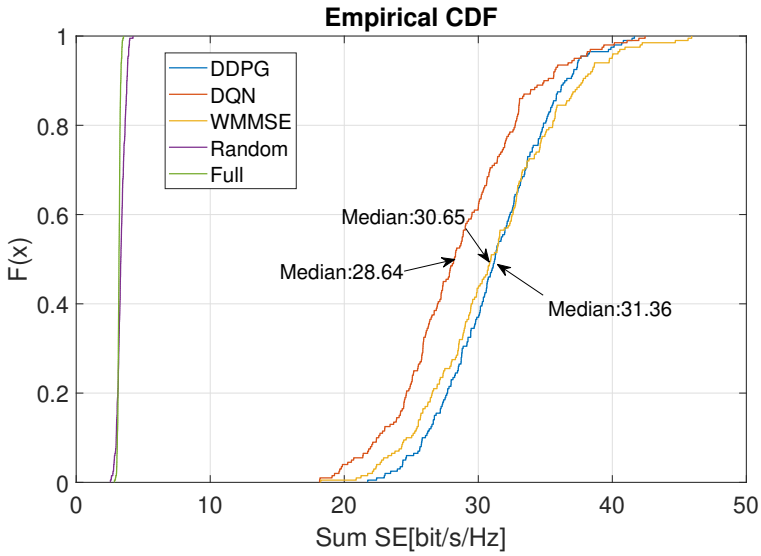


(b) CDF of the sum SE.

Figure 3.7: Comparison of the sum SE over 1,000 time slots with $N = 8$, $K = 10$ and $M = 10$.

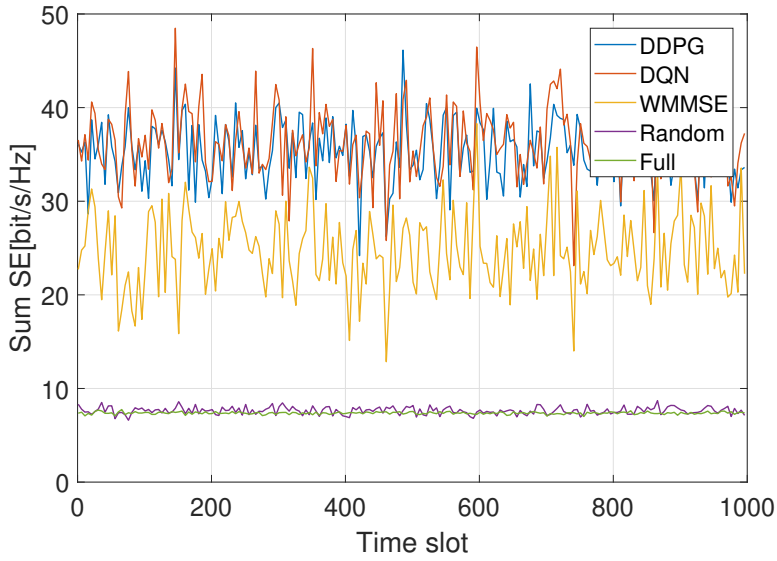


(a) Sum SE as a function of time slot.

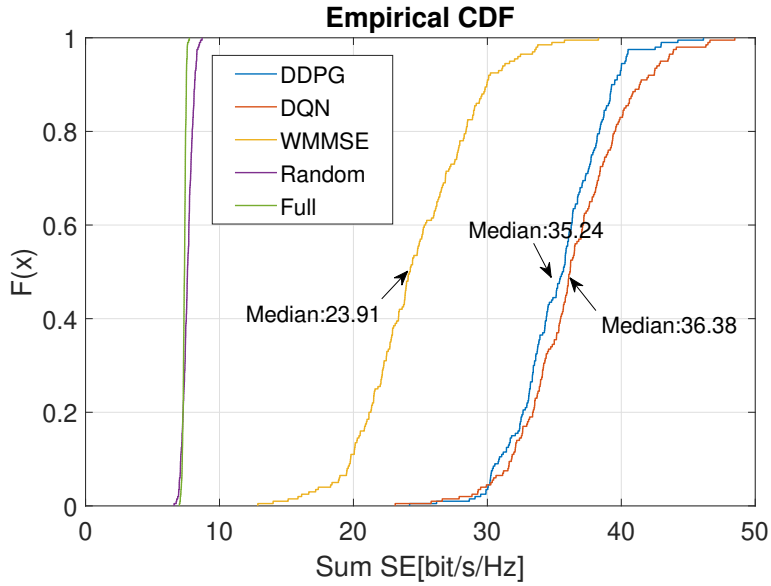


(b) CDF of the sum SE.

Figure 3.8: Comparison of the sum SE over 1,000 time slots with $N = 12$, $K = 5$ and $M = 10$.



(a) Sum SE as a function of time slot.



(b) CDF of the sum SE.

Figure 3.9: Comparison of the sum SE over 1,000 time slots with $N = 12$, $K = 10$ and $M = 15$.

based on the AP-UE links, which does not change when the number of antennas per AP is varied, so the number of antennas per AP does not affect the power allocation. Therefore, when M changes, the power allocation solution still holds.

3.3.4 Comparison of execution time

To get an indication of the difference in computational complexity of the DRL methods and the WMMSE algorithm, we measured the execution time in each of the 1,000 time slots. We ran the algorithms on a personal laptop with CPU Intel i5-7300. The programs are coded in Python 3.7.2. Table 3.2 shows the statistical characteristics of the execution time for a simulation over 1,000 time slots.

Table 3.2: Execution time of the DRL methods and the WMMSE algorithm (in ms).

Method	Mean	Max	Min	Standard deviation
DQN	0.66	0.97	0.52	0.03
DDPG	0.63	0.99	0.51	0.04
WMMSE	621.23	759.63	592.16	16.35

From Table 3.2, it is obvious that the DQN method and the DDPG method require much less processing time than the WMMSE algorithm and have less variation. For the DQN method and the DDPG method, the number of calculations is constant, as the number of neurons and layers does not change over 1,000 time slots. Observe that there are still some slight fluctuations of execution time for DQN and DDPG, which come from the calculation of different floating-point numbers and the inaccuracy of reading the system time. For the WMMSE algorithm, the time fluctuation mainly comes from different initializations, i.e., a different initial point of the algorithm can make a large difference in the time needed to find the optimum.

3.4 Summary of key points in Chapter 3

In this chapter, we studied two DRL power allocation methods, namely the DQN and the DDPG, in CF sub-6 GHz massive MIMO with mobile UEs, with the objective of maximizing the sum SE in the downlink. Unlike supervised learning that needs a huge training data set that is generated by an algorithm with high computational complexity, the DRL methods are trained by interacting with the environment. The objective function is directly defined as the reward to train the DNN. We found that for the scenario we considered, the performance (in terms of sum SE) of the DRL methods is competitive with the popular and well-adopted WMMSE algorithm. For

the same configuration and scenario, the well-trained DRL methods achieved about 38% higher sum SE than the WMMSE algorithm. For a different configuration of scenarios, the well-trained DRL-based methods still achieved an average of 33% higher sum SE than the WMMSE algorithm. In addition, the execution time of the DRL power allocation methods is significantly less than the WMMSE algorithm. On our simulation platform, only 0.1% of the execution time of the WMMSE algorithm is needed. The key points are listed as follows:

- Two DRL methods, namely the DQN method and the DDPG method, are used to perform the max-sum SE power allocation scheme.
- The power allocation is implemented by a multi-agent system, which follows a centralized training with a decentralized execution framework.
- The target of the DRL methods is to maximize the objective function, therefore the sum SE achieved by the DRL methods can be potentially higher than the benchmark algorithms.
- A particular 3GPP indoor mixed office scenario with mobile UEs is considered in the analysis.
- The max-min scheme is more complicated than the max-sum SE scheme. The use of DRL methods to achieve the max-min scheme is an open issue for further studies.

Chapter 4

Max-sum SE power allocation in CF mmWave massive MIMO

In this chapter, we investigate DRL-based methods for solving **Problem 3**. We first introduce the channel model in Section 4.1. Then in Section 4.2, we study hybrid beamforming in the context of CF mmWave massive MIMO. In Section 4.3, we use DRL methods for power allocation to achieve the max-sum SE scheme in CF mmWave massive MIMO. Finally, Section 4.4 summarizes the key points of this chapter.

4.1 Channel model

Consider a CF mmWave massive MIMO with N APs that are spread over the coverage area; each AP is equipped with M antennas. All APs are connected to a CC through a fronthaul. K single-antenna UEs are served within the coverage area, like Fig.2.1.

An accurate channel model is necessary for the analysis of the performance of a CF mmWave massive MIMO. Over the past decades, numerous channel measurements and modeling efforts at mmWave frequencies motivated a series of channel measurement campaigns [81–88]. For example, 28 GHz and 140 GHz channel measurement campaigns were conducted by New York University (NYU) WIRELESS research center in 2014 and 2019 [81] [86]. They conducted propagation measurements at mmWave in indoor scenarios. Several 3-D statistical channel models, e.g., [82], [87] and [88], were proposed to describe the signal propagation. Saleh and Valenzuela proposed a clustered channel model [89]. The channel is assumed to be a sum of contributions of multiple scattering clusters. Its modified version, known as the extended Saleh-Valenzuela (S-V) channel model [90] [91], is well-adopted in the analysis of mmWave massive MIMO [92–97]. Other efforts in

mmWave channel modeling such as [84] and [85], were proposed in different scenarios. Specifically, [84] proposed channel models for outdoor environments; [85] conducted channel measurements in a specific office scenario at 60 GHz frequency.

Standard documents such as IEEE 802.11 ad/ay and 3GPP TR 38.901 presented statistical channel models for up to 100 GHz for indoor scenarios such as residences, offices, shopping malls, and factories [80], [98], [99]. IEEE 802.11 ad/ay channel models adopted a double-directional CIR model for 60 GHz with dual polarization based on field measurements and complimentary ray-tracing simulations, which provided detailed temporal and angular channel statistics for conference rooms, cubical environments, and living rooms [98] [99]. 3GPP TR 38.901 proposed a unified geometry-based statistical channel model for indoor and outdoor scenarios for frequencies from 0.5 to 100 GHz, where different scenarios have different values of large-scale parameters which are required in the channel generation procedure [80].

It is difficult to find a generally advocated channel model for mmWave frequencies. Nevertheless, it is known that mmWave channels in massive MIMO are expected to be sparse and have a limited number of propagation paths, typically described as a 3-D geometric model [91] [97]. In this thesis, we adopt the 3-D geometric model, which is more known as the extended S-V channel model, to describe the signal propagation at mmWave frequencies. The extended S-V channel model makes use of so-called scattering clusters, i.e., the impulse response is seen as the superposition of multiple clustering scatterers, see Fig.4.1.

Mathematically, the channel vector from AP n to UE k can be formalized as in [92] [97]:

$$\mathbf{g}_{k,n} = \sqrt{\frac{M}{L_p}} \sum_{l=1}^{L_p} \alpha_l^{k,n} \mathbf{f}_r(\phi_{l,r}^{k,n}, \theta_{l,r}^{k,n}) \mathbf{f}_t^H(\phi_{l,t}^{k,n}, \theta_{l,t}^{k,n}) \tag{4.1}$$

where L_p is the number of propagation paths from AP n to UE k, $\alpha_l^{k,n}$is the complex gain of the l-th path, $\phi_{l,r}^{k,n}$ and $\theta_{l,r}^{k,n}$ are the azimuth and elevation angle of arrival (AoA) of path l, where the subscript r stands for reception, $\phi_{l,t}^{k,n}$ and $\theta_{l,t}^{k,n}$ are the azimuth and elevation angle of departure (AoD) of path l, where the subscript t stands for transmission, and $\mathbf{f}_r(\phi_{l,r}^{k,n}, \theta_{l,r}^{k,n})$ and $\mathbf{f}_t(\phi_{l,t}^{k,n}, \theta_{l,t}^{k,n})$ are the receive and transmit antenna array response vectors with dimensions equal to the number of receive and transmit antennas, respectively. The antenna response vectors $\mathbf{f}$ depend on the antenna design and array configuration. In our downlink analysis we assume the transmission antenna is an UPA with $U_1 \times U_2$ elements. For this case, the response vector $\mathbf{f}_t(\phi_{l,t}^{k,n}, \theta_{l,t}^{k,n})$ has $U_1 \times U_2$ elements and takes the following form [92] [97]:

$$\mathbf{f}_{\text{UPA}}(\phi, \theta) = \frac{1}{\sqrt{U_1 U_2}} \left[1, ..., e^{j2\pi \frac{d}{\lambda}[u_1 \sin(\phi)\sin(\theta) + u_2 \cos(\theta)]}, ..., e^{j2\pi \frac{d}{\lambda}[(U_1-1)\sin(\phi)\sin(\theta) + (U_2-1)\cos(\theta)]}\right]^H \tag{4.2}$$

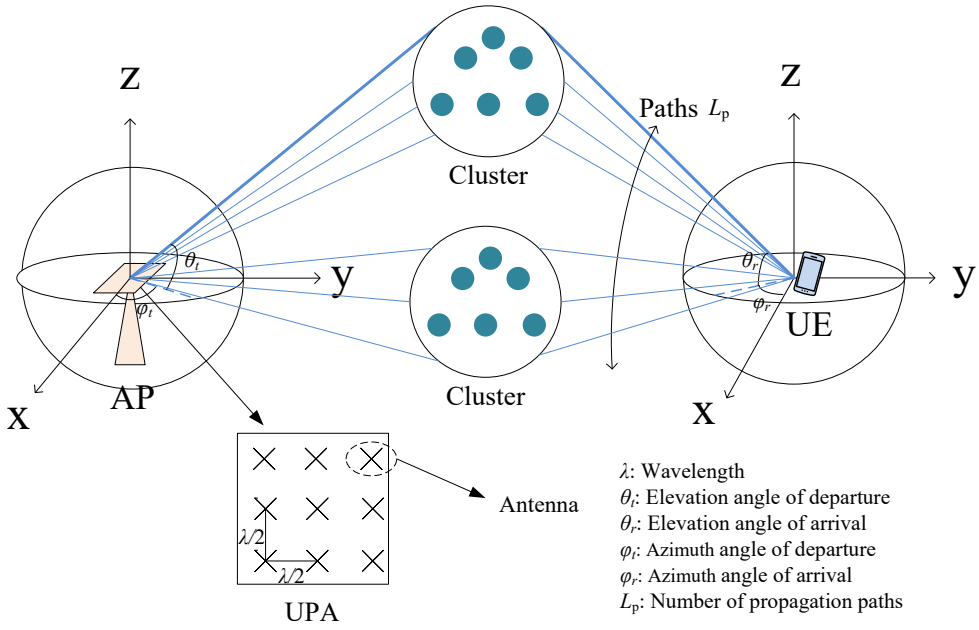


Figure 4.1: Cluster scattering concept in the 3D model.

where $u_1 = [1,2,...,U_1]$, $u_2 = [1,2,...,U_2]$, d is the antenna spacing and λ is the wavelength. In our case we choose $U_1 = U_2 = \sqrt{M}$. Since we assume that all UEs are equipped with a single omnidirectional antenna, the reception response vector $\mathbf{f}_r(\phi_{l,r}^{k,n}, \theta_{l,r}^{k,n})$ in (4.1), reduces to a scalar with value 1.

4.2 Hybrid beamforming

In Section 2.1, we mentioned that CB is preferred in CF sub-6 GHZ massive MIMO because it only uses the local CSI (e.g., $\mathbf{g}_{1,n}, \mathbf{g}_{2,n}, ..., \mathbf{g}_{K,n}$ in (2.7)). Therefore beamforming can be performed locally at each AP. On the contrary, ZFB and RZFB rely on the full CSI to perform the beamforming, i.e.,

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_{1,1} & \mathbf{g}_{1,2} & \cdots & \mathbf{g}_{1,N} \\ \mathbf{g}_{2,1} & \mathbf{g}_{2,2} & \cdots & \mathbf{g}_{2,N} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{g}_{K,1} & \mathbf{g}_{K,2} & \cdots & \mathbf{g}_{K,N} \end{bmatrix} \tag{4.3}$$

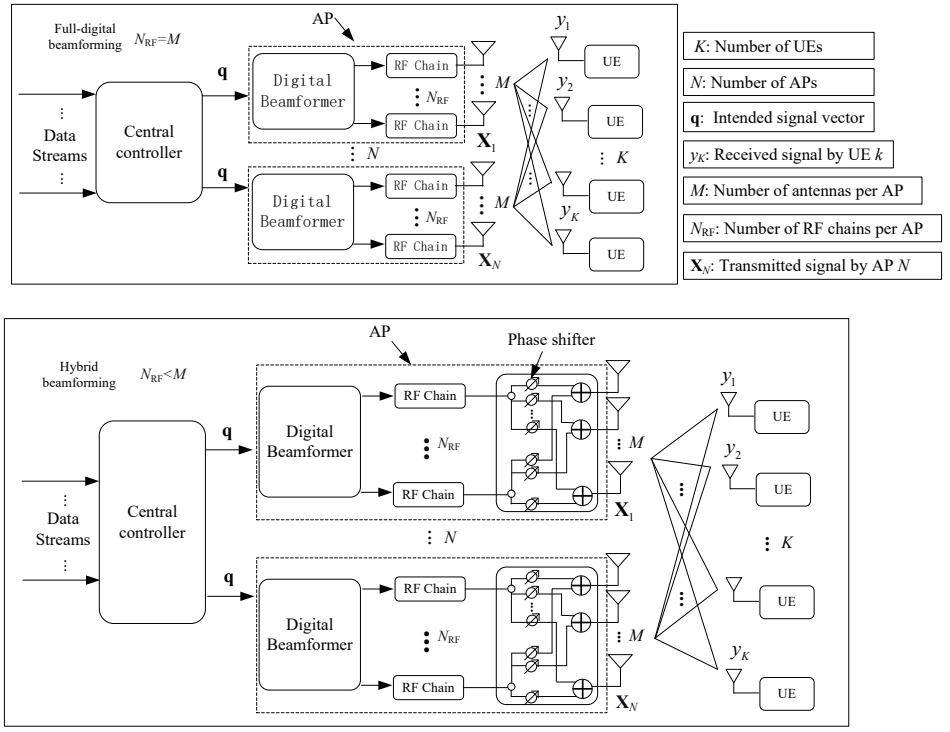


Figure 4.2: Full-digital beamforming and hybrid beamforming in CF mmWave massive MIMO.

This implies that the CC needs to collect the full CSI in each coherence time. Recently local ZFB technologies were proposed in [100] and [101], wherein the APs use the local CSI to perform the ZFB. Local ZFB technologies are used under the constraint $M >> K$, otherwise the interference for a given UE cannot be suppressed. It has been shown in [100] that local ZFB significantly outperforms CB in terms of the SE. We do not address centralized ZFB that relies on the full CSI in this thesis. Centralized ZFB may improve the SINR of the UEs, but it involves overhead for CSI exchanging which degrades the factor $\left(1-\frac{\tau_p}{\tau_c}\right)$ of the SE. The tradeoff between the factor $\left(1-\frac{\tau_p}{\tau_c}\right)$ and the improvement of SINR is an open issue that still needs to be studied. Unless otherwise stated, ZFB refers to local ZFB in the rest of this thesis.

Because the condition $M >> K$ is not necessarily met in CF massive MIMO, we did not address local ZFB in Chapter 2 and Chapter 3. Here, we consider local ZFB under the assumption that $M >> K$.

Due to the hardware constraints of present technologies, full-digital beamforming, which requires each antenna to have its own RF chain (including bandpass filter, amplifier, mixer, etc.) and which is common in the sub-6 GHz domain, is hard to implement. Therefore, hybrid beamforming, where the number of RF chains is less than the number of antennas, is required [102].

4.2.1 SE in mmWave massive MIMO

Fig.4.2 shows the architectures of full-digital beamforming and hybrid beamforming. In the full-digital beamforming architecture, each antenna has one dedicated RF chain therefore, it can achieve multiplexing gains and allows a high design freedom [103]. In the hybrid beamforming architecture, the antennas are driven by a lower number of RF chains; $N_{RF} \times M$ phase shifters are used to connect the RF chains and the antennas within each AP, where N_{RF} is the number of RF chains in each AP. Compared to the full-digital architecture, the hybrid architecture degrades the performance of the network in terms of SE.

Let the vector $\mathbf{q} = [q_1, q_2, ..., q_K]^T$, with $|q_k|^2 = 1$ $(k = 1, 2, ..., K)$ be the intended signal for the UEs , then the transmitted signal from AP n is:

$$\mathbf{x}_n = \boldsymbol{\iota}_n \sqrt{\mathbf{p}_n} \mathbf{q} \tag{4.4}$$

where $\mathbf{p}_n$ is a $K \times K$ diagonal matrix with element $p_{k,n}$ denoting the normalized power allocation for AP n to UE k, and $\boldsymbol{\iota}_n$ is the hybrid beamforming matrix of AP n with:

$$\boldsymbol{\iota}_n = \boldsymbol{\iota}_n^a \boldsymbol{\iota}_n^d \tag{4.5}$$

where $\boldsymbol{\iota}_n^a$ is the analog part with size $M \times N_{RF}$. The elements are under constant modulus constraints $|\iota_{u,v}|^2 = 1$, where $\iota_{u,v}$ is the (u, v)-th element of $\boldsymbol{\iota}_n^a$. $\boldsymbol{\iota}_n^d$ is the digital part with $N_{RF} \times K$. One can observe that after the digital part of the beamforming, the intended signal vector $\mathbf{q}$ with dimension $K \times 1$ becomes a vector of dimension $N_{RF} \times 1$. If $N_{RF} < K$, some parts of the intended signal are missing, which means that the number of transmitted data streams is less than K. So, we assume that $N_{RF} \geq K$, to guarantee that K UEs can be served simultaneously. One remark is that in massive MIMO, each RF chain is connected to all antennas, which is known as fully connected hybrid beamforming (FCHB) architecture [104], while for CF massive MIMO, each RF chain is only connected to the antennas of its own AP. Here we refer to it as the partially connected hybrid beamforming (PCHB).

Similar to (2.8) , UE k will receive y_k, the superposition of the signals from all APs in the whole system, which is given by:

$$y_k = \sum_{n=1}^{N} \mathbf{g}_{k,n} \boldsymbol{\iota}_n \sqrt{\mathbf{p}_n} \mathbf{q} + w_k = \sum_{n=1}^{N} \mathbf{g}_{k,n} \boldsymbol{\iota}_n^a \sum_{k'=1}^{K} \boldsymbol{\iota}_{n,k'}^d \sqrt{p_{k',n}} q_{k'} + w_k \tag{4.6}$$

where $\boldsymbol{\iota}_{n,k'}^{d}$ is the k'-th column of $\boldsymbol{\iota}_{n}^{d}$. The SE of UE k under mmWave channel, is given by [105]:

$$SE_k = \left(1 - \frac{\tau_p}{\tau_c}\right) \log_2 (1 + SINR_k) \tag{4.7}$$

with the SINR:

$$SINR_k = \frac{\left|\sum_{n=1}^{N} \mathbf{g}_{k,n} \boldsymbol{\iota}_{n}^{a} \boldsymbol{\iota}_{n,k}^{d} \sqrt{p_{k,n}}\right|^2}{\sum_{k' \neq k} \left|\sum_{n=1}^{N} \sqrt{p_{k',n}} \mathbf{g}_{k,n} \boldsymbol{\iota}_{n}^{a} \boldsymbol{\iota}_{n,k'}^{d}\right|^2 + 1} \tag{4.8}$$

Please note the difference between (2.9) and (4.7). In (2.9), the SE is derived by the ergodic capacity [18] [54], which is evaluated by averaging the capacity obtained at a particular time instance on a fading channel over an infinite time interval. The ergodic capacity is usually used for statistic channel models [54], e.g., Rayleigh or Rician. The UatF expression is a simplified version of ergodic capacity in massive MIMO. While for a non-statistic channel model, e.g., the extended S-V channel model[1], the instantaneous capacity is used to calculate the SE, i.e., (4.7) is used.

4.2.2 Design of hybrid beamforming

The design objective of hybrid beamforming is to find a pair of ($\boldsymbol{\iota}_{n}^{a}$, $\boldsymbol{\iota}_{n}^{d}$) to approximate full-digital beamforming. This is typically achieved by performing zero-forcing in the digital part [92]. Assuming that the perfect CSI is obtained in the uplink channel estimation, with a given analog part $\boldsymbol{\iota}_{n}^{a}$, an equivalent channel $\mathbf{g}_{n}^{eq} = \mathbf{g}_{n} \boldsymbol{\iota}_{n}^{a}$ is used to perform ZF for the digital part in AP n:

$$\boldsymbol{\iota}_{n}^{d} = \left(\mathbf{g}_{n}^{eq}\right)^{H} \left[\mathbf{g}_{n}^{eq} \left(\mathbf{g}_{n}^{eq}\right)^{H}\right]^{-1} = \left(\mathbf{g}_{n} \boldsymbol{\iota}_{n}^{a}\right)^{H} \left[\mathbf{g}_{n} \boldsymbol{\iota}_{n}^{a} \left(\mathbf{g}_{n} \boldsymbol{\iota}_{n}^{a}\right)^{H}\right]^{-1} \tag{4.9}$$

where $\mathbf{g}_{n} \in C^{K \times M}$ is the channel matrix between AP n to all UEs, i.e., $\mathbf{g}_{n} = [(\mathbf{g}_{1,n})^{T}, (\mathbf{g}_{2,n})^{T}, ..., (\mathbf{g}_{K,n})^{T}]^{T}$. Combining (4.5) and (4.9), one has

$$\boldsymbol{\iota}_{n} = \boldsymbol{\iota}_{n}^{a} (\boldsymbol{\iota}_{n}^{a})^{H} \left(\mathbf{g}_{n}\right)^{H} \left[\mathbf{g}_{n} \boldsymbol{\iota}_{n}^{a} (\boldsymbol{\iota}_{n}^{a})^{H} \left(\mathbf{g}_{n}\right)^{H}\right]^{-1} \tag{4.10}$$

Then the problem turns out to be the optimization of $\boldsymbol{\iota}_{n}^{a}$ to approximate the full-digital ZFB. Numerous methods have been developed for designing the optimal analog beamforming, see, e.g., [92–97]. However, most of the proposed methods have a high computational complexity. For example, [92] proposed an orthogonal matching pursuit algorithm to design the analog beamforming. This method needs

[1] There are only $3 \sim 5$ multi-path components at mmWave domain measured from realistic environments [106] [107]. The extended S-V channel model assumes a limited number of propagation paths in mmWave, it is a non-statistical channel model.

a sequence of matrix inversion operations; recall the time constraint for beamforming in Fig.1.4. It is difficult to obtain the optimal analog beamforming within the coherence time duration. In addition, all above works considered massive MIMO where the FCHB architecture can be used. For CF mmWave massive MIMO, PCHB should be used. [94] proposed a low-complexity design, which directly uses the normalized conjugate channel as the analog part. Although it suffers a slight performance loss, this design of analog beamforming does not need an iterative calculation of matrices. Moreover, it can be easily implemented in CF mmWave massive MIMO. In case that $N_{RF} = K$, the (u, v)-th element of $\boldsymbol{\iota}_n^a$ is set as [94]:

$$\iota_{u,v} = e^{j\phi_{u,v}} \tag{4.11}$$

where $\phi_{u,v}$ $(u = 1,2,\dots M, v = 1,2,\dots K)$ is the phase of the (u, v)-th element of the conjugate transpose of the channel $\mathbf{g}_{k,n}$. In other words,

$$\boldsymbol{\iota}_n^a = \begin{bmatrix} e^{j\phi_{1,1}} & e^{j\phi_{1,2}} & \dots & e^{j\phi_{1,K}} \\ e^{j\phi_{2,1}} & e^{j\phi_{2,2}} & \dots & e^{j\phi_{2,K}} \\ \dots & \dots & \dots & \dots \\ e^{j\phi_{M,1}} & e^{j\phi_{M,2}} & \dots & e^{j\phi_{M,K}} \end{bmatrix} \tag{4.12}$$

Now let us consider the case $N_{RF} > K$:

$$\boldsymbol{\iota}_n^a = \begin{bmatrix} e^{j\phi_{1,1}} & \dots & e^{j\phi_{1,K}} & \dots & \iota_{1,N_{RF}} \\ e^{j\phi_{2,1}} & \dots & e^{j\phi_{2,K}} & \dots & \iota_{2,N_{RF}} \\ \dots & \dots & \dots & \dots & \\ e^{j\phi_{M,1}} & \dots & e^{j\phi_{M,K}} & \dots & \iota_{M,N_{RF}} \end{bmatrix} \tag{4.13}$$

When comparing (4.10) to full-digital ZFB

$$\boldsymbol{\iota}_n^{\text{full}} = \left(\mathbf{g}_n\right)^H \left[\mathbf{g}_n \left(\mathbf{g}_n\right)^H\right]^{-1} \tag{4.14}$$

one can find that the optimal design of analog beamforming is to approximate $\boldsymbol{\iota}_n^a (\boldsymbol{\iota}_n^a)^H$ to a diagonal matrix, i.e.,

$$\sum_{v=1}^{K} e^{j(\phi_{u,v} + \phi_{u',v})} + \sum_{v=K+1}^{N_{RF}} \iota_{u,v} \iota_{u',v} \to 0, u \neq u'; u, u' = 1,2,\dots M \tag{4.15}$$

one can set

$$\iota_{u,v+K} = -j e^{j\phi_{u,v}}, v = 1,2,\dots, N_{RF} - K \tag{4.16}$$

One can observe that when $N_{RF} = 2K$, hybrid beamforming asymptotically approximates full-digital beamforming. When $K < N_{RF} < 2K$, we can choose the "best" $(N_{RF} - K)$ UEs[2] to achieve 'full-digital beamforming'. When $N_{RF} > 2K$, $\iota_{u,v}$, where

[2]The "best" $(N_{RF} - K)$ UEs refer to the UEs corresponding to the largest $(N_{RF} - K)$ singular values of the channel $\mathbf{g}_n$.

$u = 1,2,...M, v = 2K+1,... N_{RF}$, can be set to "0"[3] to achieve the same performance as $N_{RF} = 2K$.

4.2.3 Performance evaluation

To verify the above analysis, we compare, using Monte Carlo simulations, the performance of hybrid beamforming and full-digital beamforming in mmWave massive MIMO. Like in Section 3.3, we adopt the 3GPP indoor mixed office scenario, where 12 APs serve 10 single-antennas UEs; see Fig.4.3.

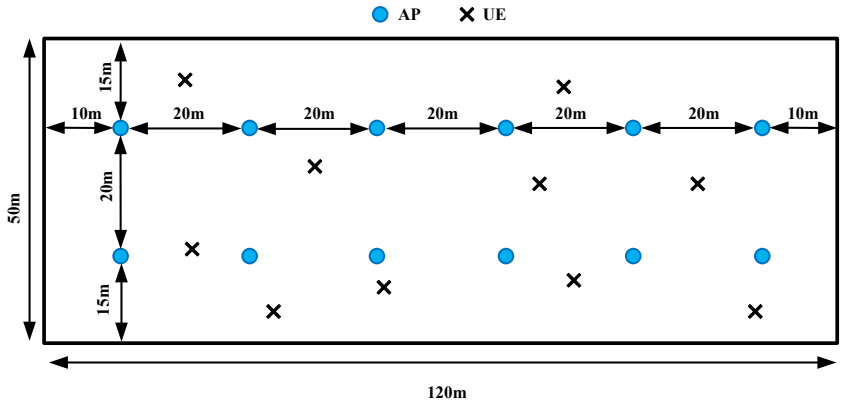


Figure 4.3: CF mmWave massive MIMO configuration used in simulations.

To enable ZFB, the number of antennas per AP is set as 25. The antennas are placed in UPA of 5×5 in each AP. The antenna spacing is half a wavelength. The carrier frequency is 28 GHz, the bandwidth is 200 MHz. The number of propagation paths L_p is set to be 5. The maximum power constraint is 23 dBm and the noise power is -74 dBm. See Table 4.1.

We do the comparisons, by varying the number of RF chains per AP for the hybrid beamforming. Fig.4.4 shows the empirical CDF of the per-UE SE under max-min power allocation and the performance of the sum SE under max-SE power allocation. The results come from 1,000 simulations. In each simulation 10 UEs are randomly placed in the coverage area, according to a uniform distribution. The bisection algorithm[4], proposed in Section 2.2.1, is used for the max-min power

[3] Here "0" means that the phase shifter is turned off.

[4] Here we focus on the performance of different ZFB techniques, the execution time of the algorithm is neglected. On the other hand, because ZFB is used, the power allocation problems are convex. Hence, the computational complexity to optimize the power allocation is much lower than in the case of CB.

Table 4.1: Parameters used in simulations

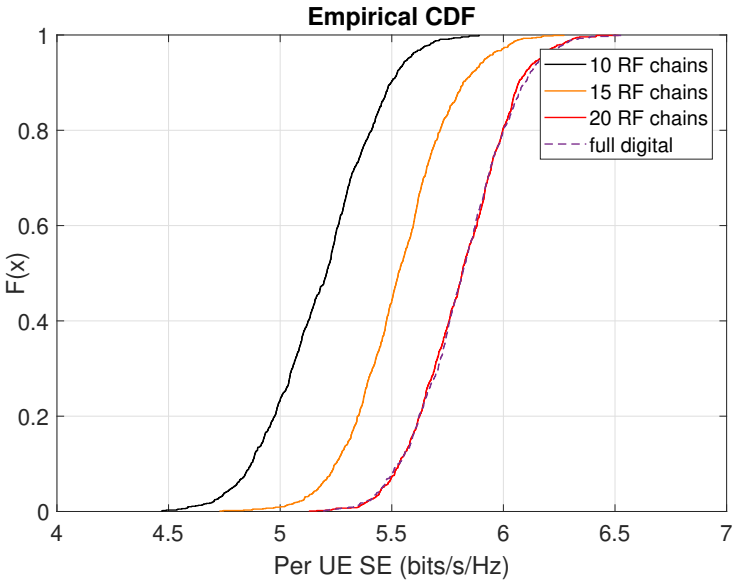
Parameter	value
Coverage volume	120m×50m×3m
K, number of UEs	10
M, number of antennas per AP	10
N, number of APs	25
L_p, number of propagation paths	5
Antennas spacing in each AP	Half wavelength
p_l, maximum power constraint	23 dBm
Carrier frequency	28 GHz
Bandwidth	200 MHz
Noise power	-74dBm

allocation and the water-filling algorithm is used for the max-sum SE power allocation.

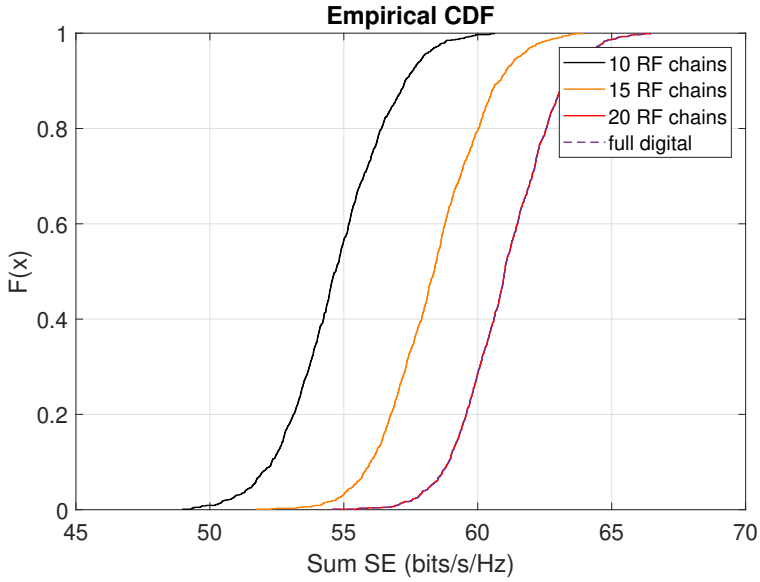
From Fig.4.4 we observe that both the per-UE SE and sum SE increase with the number of RF chains. For example, this can be seen by comparing the yellow curve and the black curve in Fig.4.4 (a). When the number of RF chains in each AP is twice the number of UEs, hybrid beamforming achieves the same performance as the full-digital beamforming. This can be observed in both Fig.4.4 (a) and (b) where the red solid curves and the purple dashed curves overlap.

4.2.4 Discussion

In this section, we studied hybrid beamforming techniques in CF mmWave massive MIMO. We first reviewed the channel model used in mmWave frequencies. Referring to the literature, we adopted the 3-D geometric model, i.e., the extended S-V model, to describe the signal propagation in mmWave. Then we formalized the SE of the UEs in the hybrid beamforming architecture. Following this, we proposed a low-complexity design of hybrid ZFB, which can be easily implemented in the PCHB architecture. Finally, we studied the effects of the number of RF chains on the performance of the hybrid ZFB. By analysis and numerical results, we found that, when the number of RF chains per AP is twice the number of UEs, hybrid ZFB achieves the same performance as full-digital beamforming in terms of the achievable SE. This conclusion holds on in arbitrary scenario where the number of antennas per AP is much larger than the number of UEs.



(a) Per-UE SE under max-min power allocation.



(b) Sum SE under max-sum SE power allocation.

Figure 4.4: Full-digital ZFB VS Hybrid ZFB with different number of RF chains.

4.3 Sum-SE power allocation in CF mmWave massive MIMO

The use of ZFB simplifies the power allocation by suppressing the inter-UE interference, enabling convex optimization methods (e.g., using the water-filling algorithm). However, the price to pay is the cost of a large number of RF chains in the network. Consider the 3GPP indoor mixed office scenario in Fig.4.3, where 12 APs serve 10 UEs in a coverage area of 120m×50m. The network needs at least 120 RF chains to serve the UEs, which is high. In addition, the condition $M >> K$ is typically not fulfilled. For example, the pioneer works [11] [41] considered single antenna APs. Later, [22] proposed that 5 to 10 antennas per AP is a good tradeoff between macro diversity and channel hardening in CF massive MIMO. Therefore, ZFB is typically not expected to be used in CF massive MIMO. The alternative option is NCB. This however, like the use of CB in CF sub-6 GHz massive MIMO, leads to the power allocation being a non-convex problem in CF mmWave massive MIMO.

The power allocation is expected to be performed based on the SE obtained by the instantaneous capacity, i.e., by (4.7) (4.8) [31], which implies the power allocation should be updated according to the small-scale fading variation in every coherence time. However, this is difficult to achieve because the coherence time is quite short in the mmWave domain. For example, the coherence time τ_c is around 5 ms at 28 GHz with a UE velocity of 1 m/s. Nevertheless, it has been shown in [31] that power allocation based on the small-scale fading variation can significantly improve the SE in CF sub-6 GHz massive MIMO. It is still an open issue in CF mmWave massive MIMO. In this thesis, we resort to the UatF expression to optimize the power in CF mmWave massive MIMO.

We consider the max-sum SE power allocation scheme. The proposed DQN method and DDPG method are used to optimize the power. We evaluate the performance of the DQN method and DDPG method in CF mmWave massive MIMO. Unlike Section 3.3, in this section we use a 28 GHz frequency carrier and the extended S-V channel model. Please note that the coherence time τ_c is 5 ms at 28 GHz domain, therefore the large-scale time τ_l, i.e., the duration of time slot for DRL methods, is 200 ms. For the DQN and DDPG, we directly adopt the architecture of DNN_1. Other parameters are the same as the configuration in Section 3.3.

4.3.1 Hyperparameter selection

We first studied the impact of the parameters, namely discount factor ω, the training interval C, the initial adaptive learning rate lr, the adaptive ϵ-greedy algorithm and the minibatch size $|D^t|$ on the training of the DQN and DDPG. Fig.4.5 ~4.9

show the effect of different parameters on the training process of DQN. Fig.4.10 ~4.14 show the effect of different parameters on the training process of DDPG. The graphs in the upper show the sum SE as a function of time over a period of 20,000 time slots. The corresponding graphs in the lower show the empirical CDF of the sum SE. In each of the figures we vary one parameter, while keeping the others constant.

Fig.4.5 shows the effect of the discount factor ω on the training of the DQN. Although the differences are not pronounced, we see that the sum SE for $\omega = 0.1$ is always larger than for $\omega = 0.9$, by observing that the red line is to the right of the blue line in the empirical CDF graph. The fluctuations of the sum SE as a function of the training time, is due to the random mobility of the UEs, which leads to a variation of the large-scale fading.

Fig.4.6 shows the effect of the training interval C on the training of the DQN. Similar observations can be found in Fig.4.5; the differences between the graphs are not pronounced. Nevertheless, we find that $C = 100$ achieves the highest sum SE, by observing the light blue line in the empirical CDF graph.

Fig.4.7 shows the effect of the initial learning rate lr on the training of the DQN. The differences between the lines are not obvious, but we still see that the sum SE for $lr = 0.005$ achieves the highest value, by observing the light red line is to the right of the other lines in the empirical CDF graph.

Fig.4.8 shows the effect of the ϵ-algorithm on the training of DQN. We find that for different values of ϵ, the values of the sum SE are very different. It is obvious that $\epsilon = 0.1$ achieves the highest sum SE, by observing the red line in both the training process and the empirical CDF graphs.

Fig.4.9 shows the effect of the batch size $|D^t|$ on the training of the DQN. The differences of the lines are not pronounced. Similarly, we find that $|D^t|$=2,000 achieves the highest sum SE, by observing the light green line in the empirical CDF graph.

Fig.4.10 shows the effect of the discount factor ω on the training of the DDPG. Like Fig.4.5, the differences are not pronounced. Nevertheless, the sum SE for $\omega = 0.1$ is always larger than for $\omega = 0.9$, as can be seen by observing that the red line is to the right of the blue line in the empirical CDF graph.

Fig.4.11 shows the effect of the training interval C on the training of the DDPG. The observation is that the sum-SE values for $C = 10$ and $C = 20$ are clearly higher than the others, as shown in the CDF graph.

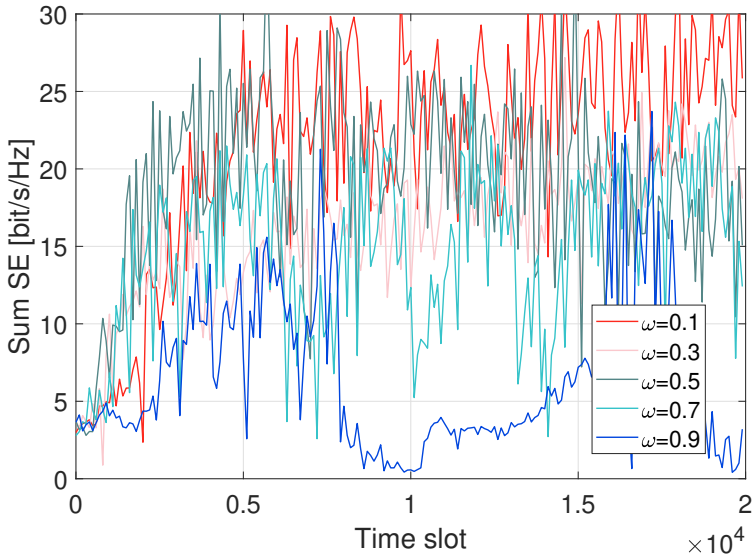
Fig.4.12 shows the effect of the initial learning rate lr on the training of the DDPG. The differences between the lines are obvious. We see that the sum SE for $lr = 0.001$ achieves the highest value, since the red line is to the right of other lines in the empirical CDF graph.

Fig.4.13 shows the effect of the ϵ-algorithm on the training of the DDPG. Similar to Fig.4.8, we find that for different values of ϵ, the values of sum SE are very

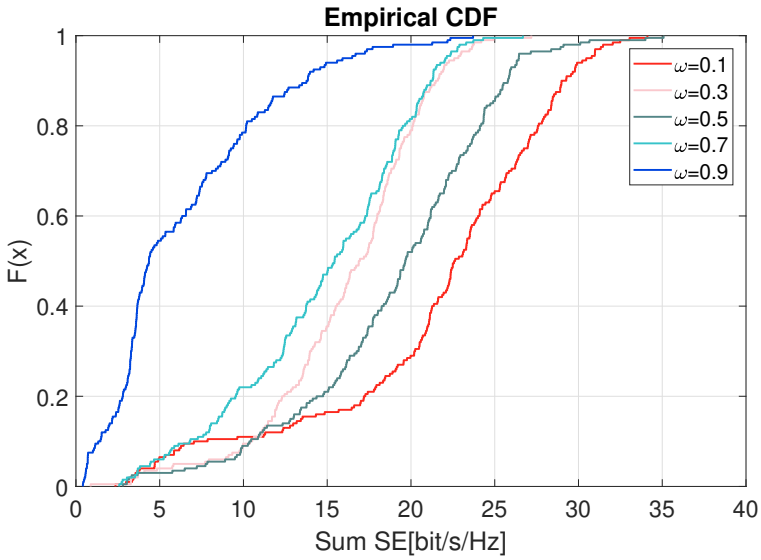
different. It is obvious that $\epsilon = 0.1$ achieves the highest sum SE, by observing the red line in both training process and empirical CDF graphs.

Fig.4.14 shows the effect of the batch size $|D^t|$ on the training of the DDPG. The differences of the lines are not pronounced. We find $|D^t|$=500 achieves the highest sum SE, by observing the red curve in the empirical CDF graph.

Based on the above observations, we choose the parameters $\omega = 0.1$, $C = 100$, $lr = 0.005$, $\epsilon = 0.1$ and $|D^t|$=2,000 to train the DQN, and $\omega = 0.1$, $C = 10$, $lr = 0.001$, $\epsilon = 0.1$ and $|D^t|$=500 to train the DDPG. The length of the training period we choose is determined by the time it takes for the time average of the sum SE to converge to a stable value, i.e., a longer training period does not result in a significantly different time average. In our case 20,000 time slots, appears to be sufficiently long, as can be observed from Fig.4.15. Observe that DQN achieves sum-SE values fluctuating around 23 bit/s/Hz for about 5,000 training time slots. Afterwards, the average rises slowly and finally converges to around 28 bit/s/Hz after 10,000 training time slots. The DDPG achieves sum-SE values fluctuating around 30 bit/s/Hz for about 5,000 training time slots. Afterward the average rises slowly and finally converges to around 32 bit/s/Hz after 10,000 training time slots. These are obviously better than the value obtained by the WMMSE algorithm, which is also shown in the figure as a reference. The random mobility of UEs causes the fluctuations of the sum SE for both methods. It is clear that the DQN method and the DDPG method, after sufficient training, achieve significantly better average sum-SE values than WMMSE.

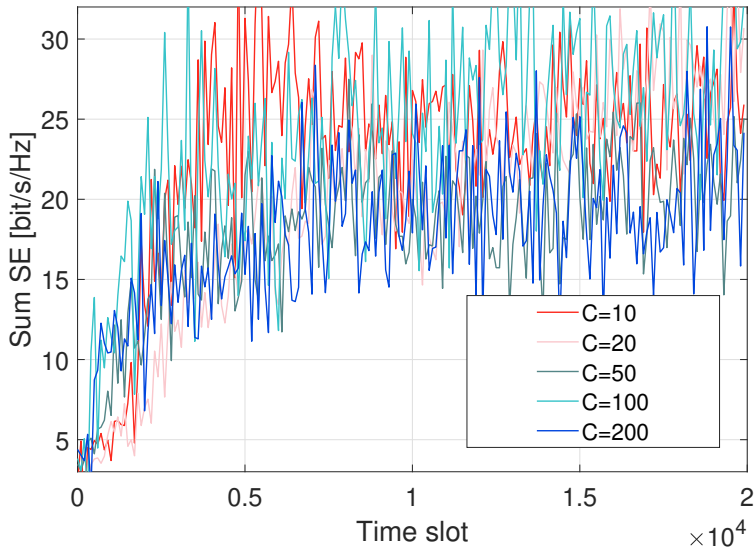


(a) Sum SE as a function of time slot.

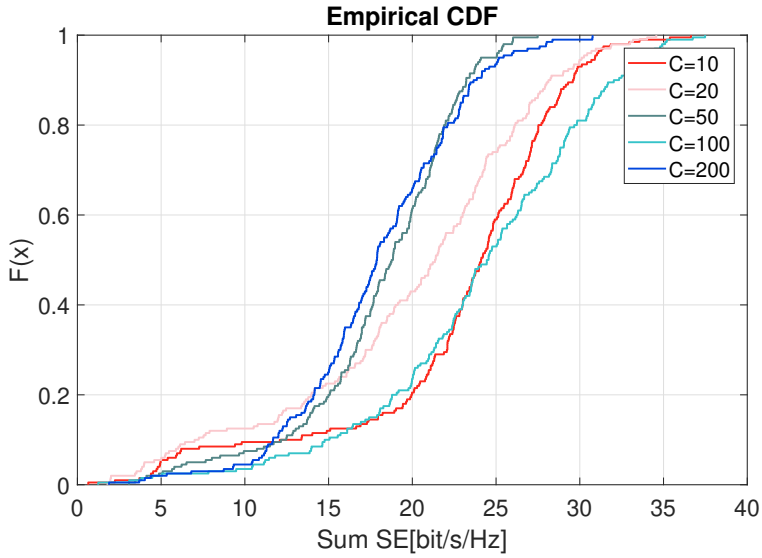


(b) CDF of the sum SE.

Figure 4.5: Effect of discount factor ω on DQN, with training interval $C = 10$, initial learning rate $lr = 0.005$, $\epsilon = 0.1$ and batch size $|D^t| = 500$.

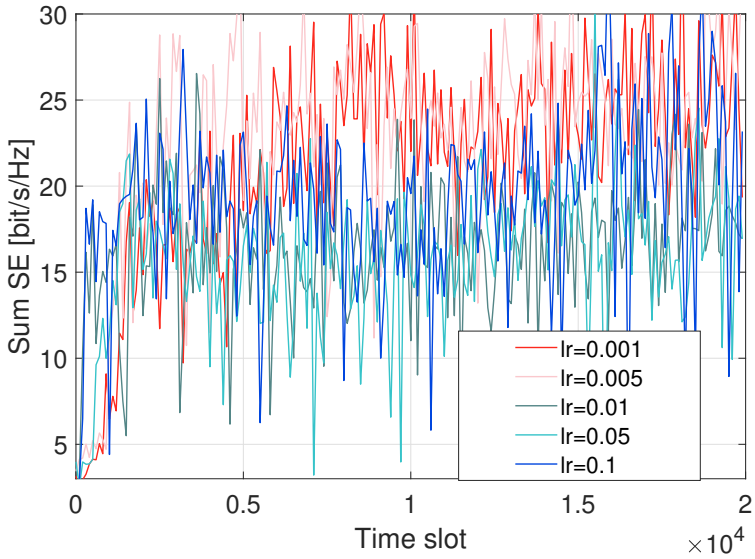


(a) Sum SE as a function of time slot.

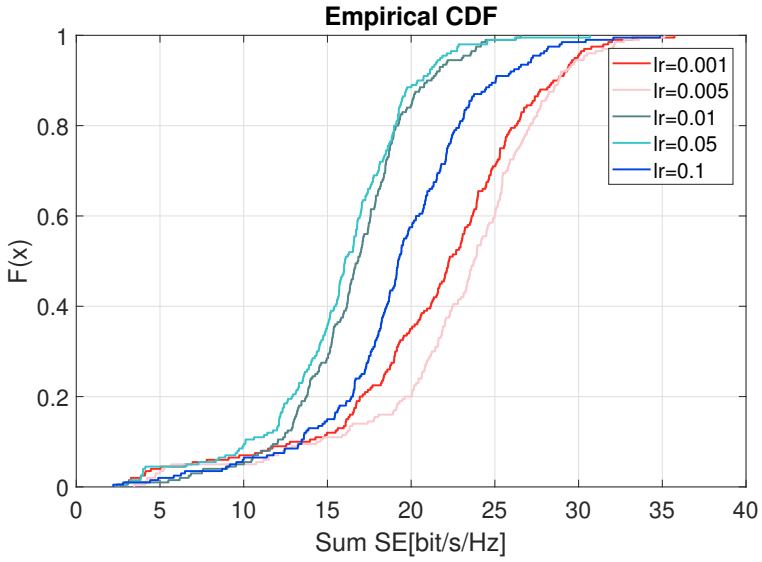


(b) CDF of the sum SE.

Figure 4.6: Effect of training interval C on DQN, with discount factor $\omega = 0.1$, initial learning rate $lr = 0.005$, $\epsilon = 0.1$ and batch size $|D^t| = 500$.

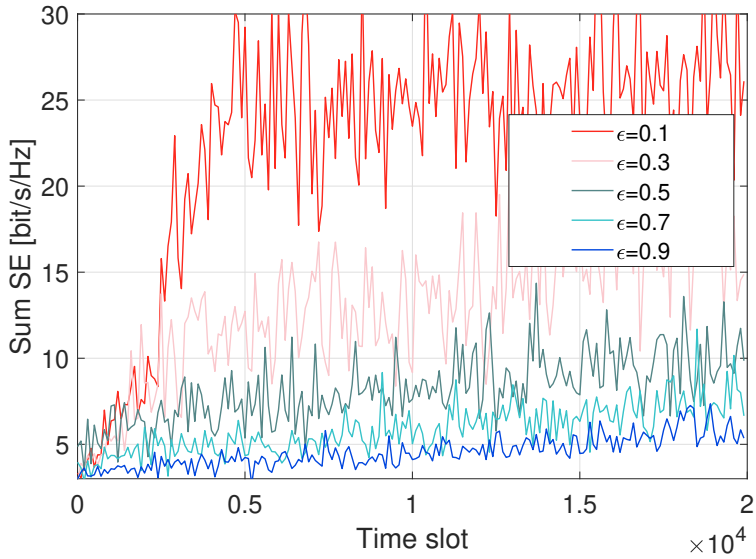


(a) Sum SE as a function of time slot.

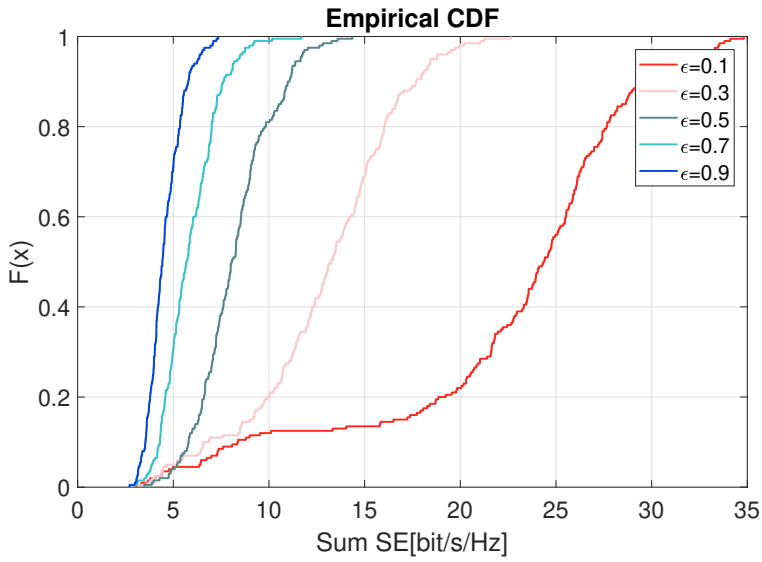


(b) CDF of the sum SE.

Figure 4.7: Effect of training interval lr on DQN, with discount factor $\omega = 0.1$, training interval $C = 10$, $\epsilon = 0.1$ and batch size $|D^t| = 500$.

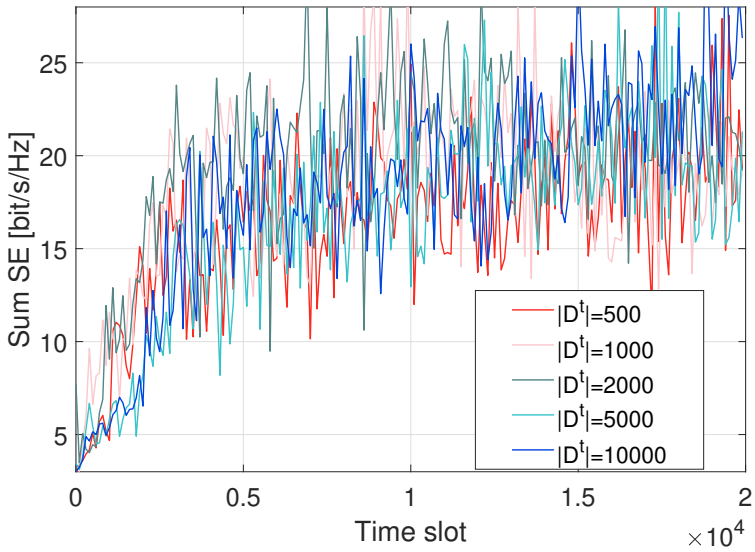


(a) Sum SE as a function of time slot.

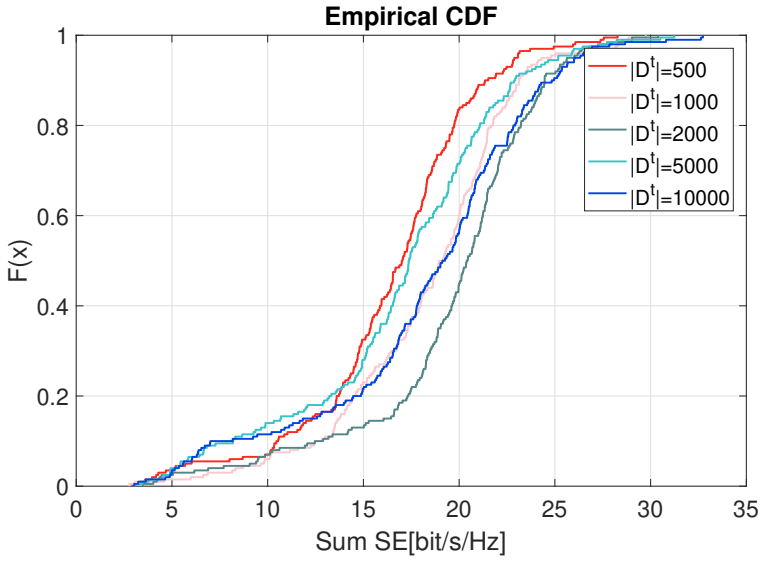


(b) CDF of the sum SE.

Figure 4.8: Effect of ϵ on DQN, with discount factor $\omega = 0.1$, training interval $C = 10$, initial learning rate $lr = 0.005$, and batch size $|D^t| = 500$.

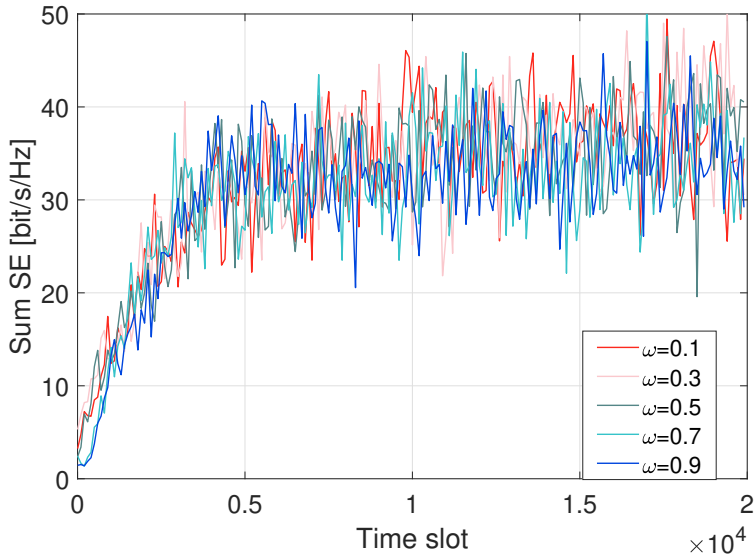


(a) Sum SE as a function of time slot.

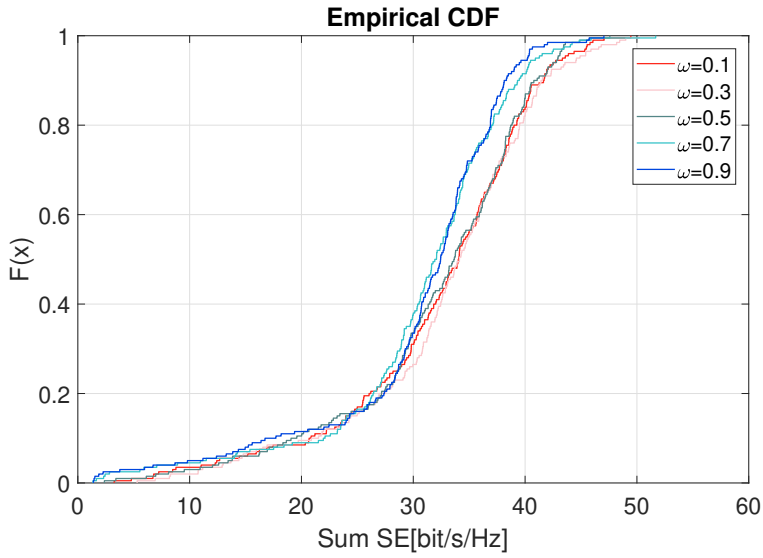


(b) CDF of the sum SE.

Figure 4.9: Effect of batch size $|D^t|$ on DQN, with discount factor $\omega = 0.1$, training interval $C = 10$, initial learning rate $lr = 0.005$, and $\epsilon = 0.1$.

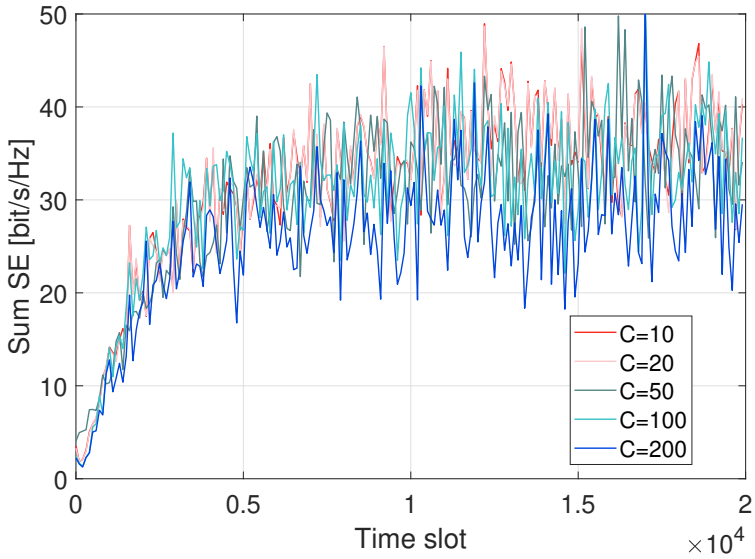


(a) Sum SE as a function of time slot.

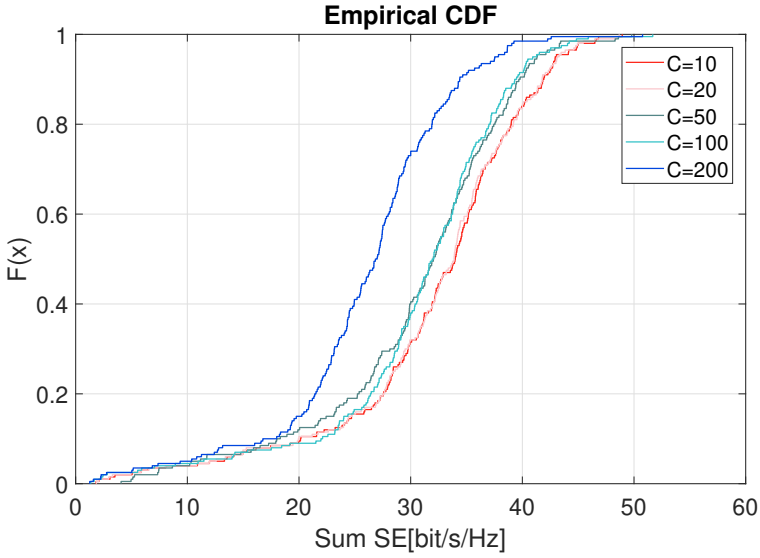


(b) CDF of the sum SE.

Figure 4.10: Effect of discount factor ω on DDPG, with training interval $C = 10$, initial learning rate $lr = 0.005$, $\epsilon = 0.1$ and batch size $|D^t| = 500$.

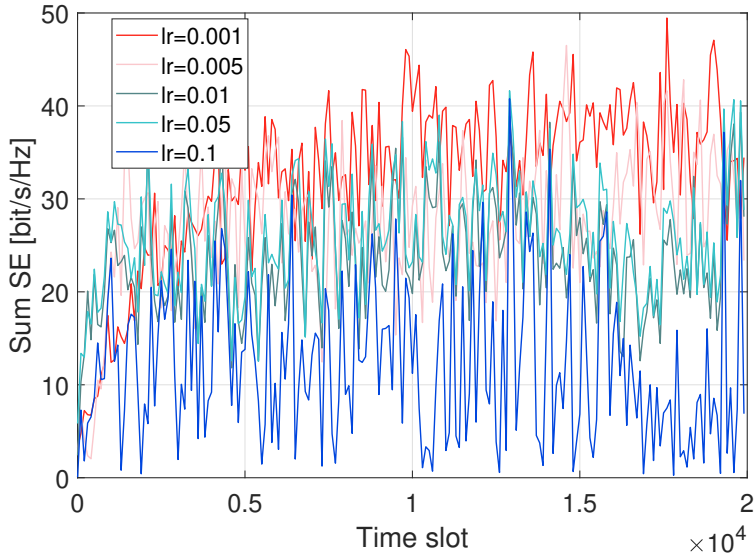


(a) Sum SE as a function of time slot.

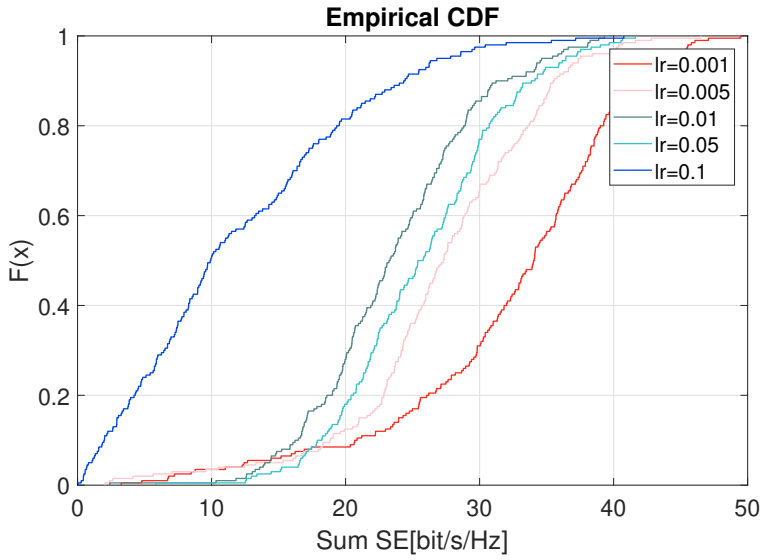


(b) CDF of the sum SE.

Figure 4.11: Effect of training interval C on DDPG, with discount factor $\omega = 0.1$, initial learning rate $lr = 0.005$, $\epsilon = 0.1$ and batch size $|D^t| = 500$.

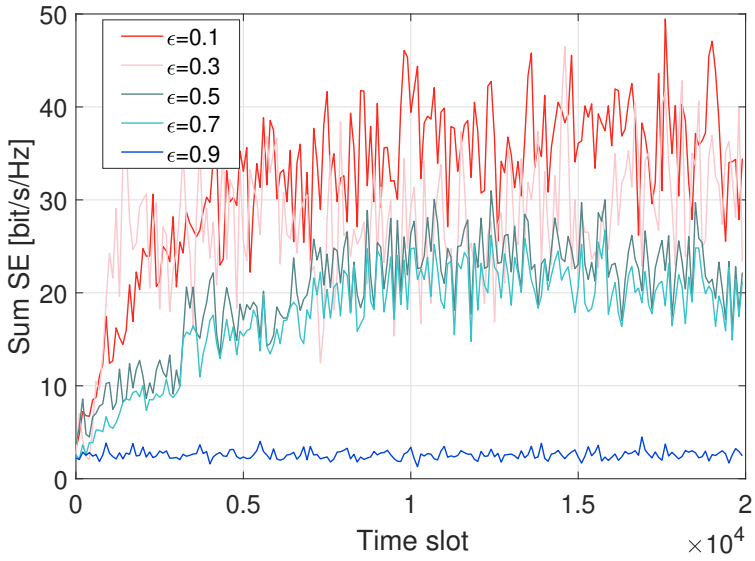


(a) Sum SE as a function of time slot.

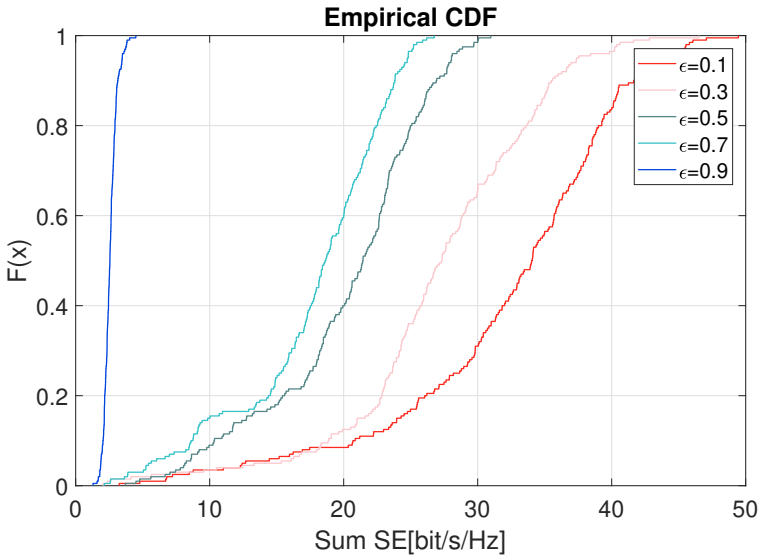


(b) CDF of the sum SE.

Figure 4.12: Effect of training interval lr on DDPG, with discount factor $\omega = 0.1$, training interval $C = 10$, $\epsilon = 0.1$ and batch size $|D^t| = 500$.

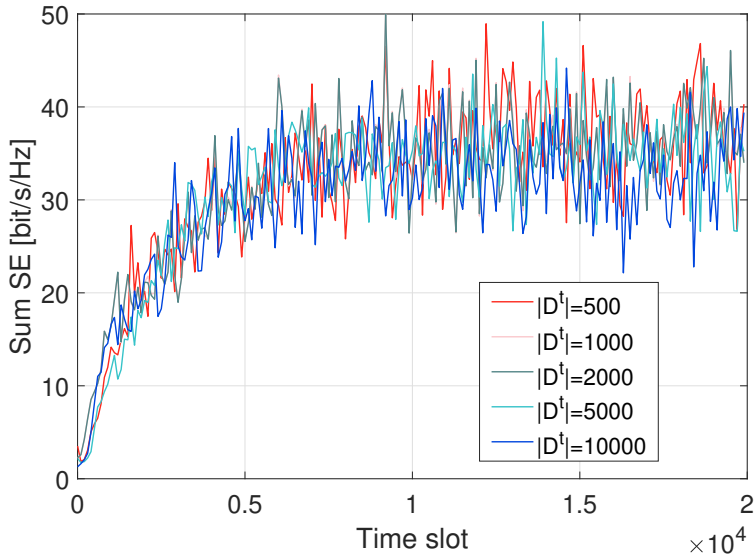


(a) Sum SE as a function of time slot.

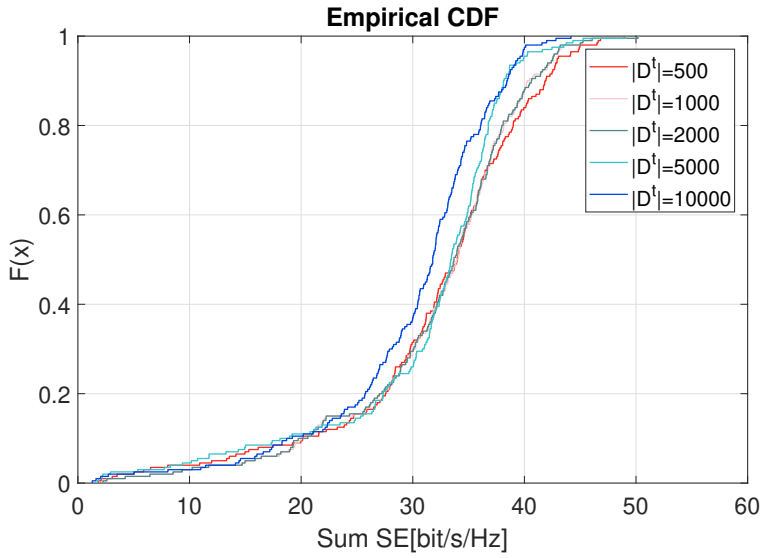


(b) CDF of the sum SE.

Figure 4.13: Effect of ϵ on DDPG, with discount factor $\omega = 0.1$, training interval $C = 10$, initial learning rate $lr = 0.005$, and batch size $|D^t| = 500$.



(a) Sum SE as a function of time slot.



(b) CDF of the sum SE.

Figure 4.14: Effect of batch size $|D^t|$ on DDPG, with discount factor $\omega = 0.1$, training interval $C = 10$, initial learning rate $lr = 0.005$, and $\epsilon = 0.1$.

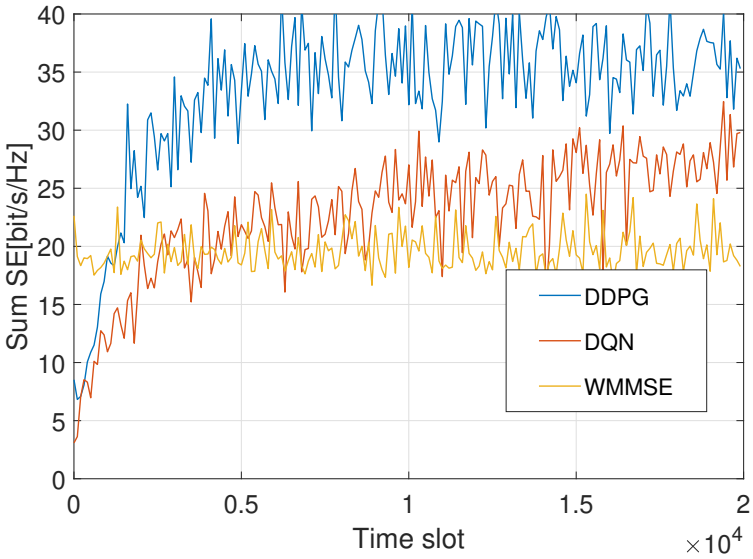


Figure 4.15: Sum SE of WMMSE, DQN and DDPG during training.

4.3.2 Sum-SE performance and execution time

We have used three benchmark algorithms to evaluate the performance of the DQN and DDPG methods for power allocation. The first benchmark is the WMMSE algorithm. The second is random power allocation where $p_{k,n} \sim \mathbb{U}(0, p_l/K)$ for $k = 1,2,...,K$ and $n = 1,2,...N$. The third one is full-power allocation, i.e., $p_{k,n} = p_l/K$ for $k = 1,2,...,K$ and $n = 1,2,...N$. We use the DQN and DDPG that have been trained for 20,000 time slots, as shown in Fig.4.15 and run it for 1,000 slots. Fig.4.16 shows the sum SE of the five methods over a period of 1,000 time slots. As expected, DDPG and DQN have much better performance than other methods. In addition, the DDPG method performs better than DQN. The DDPG method achieves an around 6 bit/s/Hz higher sum SE than the DQN method.

Finally, to get an indication of the difference in computational complexity of DQN and DDPG, we measured the execution time in each of the 1,000 time slots. We ran the algorithms on a 4-core Intel Core i5-7300 CPU with 2.6 GHz frequency. The programs are coded in Python 3.7.2 (DQN with Tensorflow 1.13.1). Fig.4.17 shows the empirical CDF of the execution times that we recorded for the two methods. From Fig.4.17 it is obvious that DQN and DDPG require much less processing time than WMMSE and have less variation. It is around 0.632 ms for DQN and

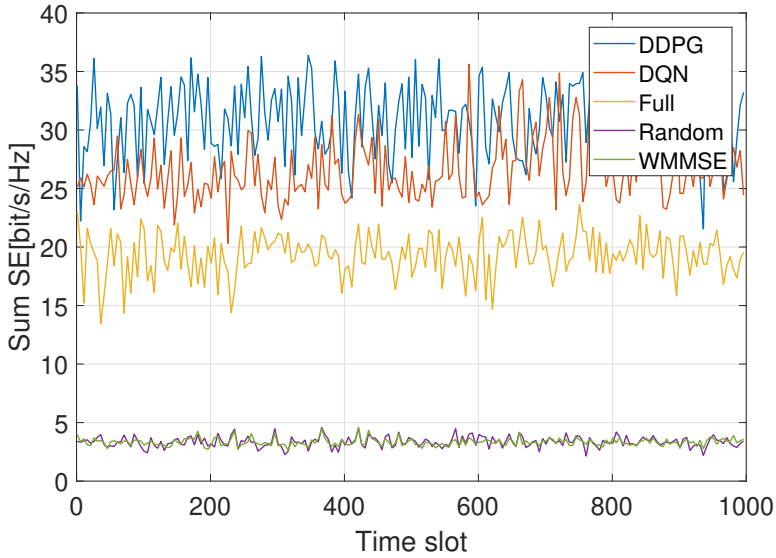


Figure 4.16: Comparison of the sum SE over 1,000 time slots.

0.628 ms for DDPG, while for WMMSE the execution time ranges from 600 ms to 750 ms. Recall that the power allocation is performed within each large-scale time, namely 200 ms. It is obvious the DQN method and DDPG method meet this time constraint, while the WMMSE does not.

In addition, for DQN and DDPG, the number of calculations is constant, as the number of neurons and layers does not change. From the magnifier in Fig.4.17, there are still some slight fluctuations of execution time, which come from the calculation of different floating-point numbers and the inaccuracy of reading the system time. For WMMSE, the time fluctuation mainly comes from different initializations, i.e., a different initial point of the algorithm can make a large difference in the time needed to find the optimum.

4.3.3 Discussion

In this section, we studied the use of DQN and DDPG to achieve the max-sum SE power allocation scheme in CF mmWave massive MIMO. The high cost of ZFB in CF mmWave massive MIMO makes the NCB an alternative option for signal processing. This however makes power allocation a non-convex problem. Like in CF sub-6 GHz massive MIMO, we use the DQN and the DDPG methods, for the

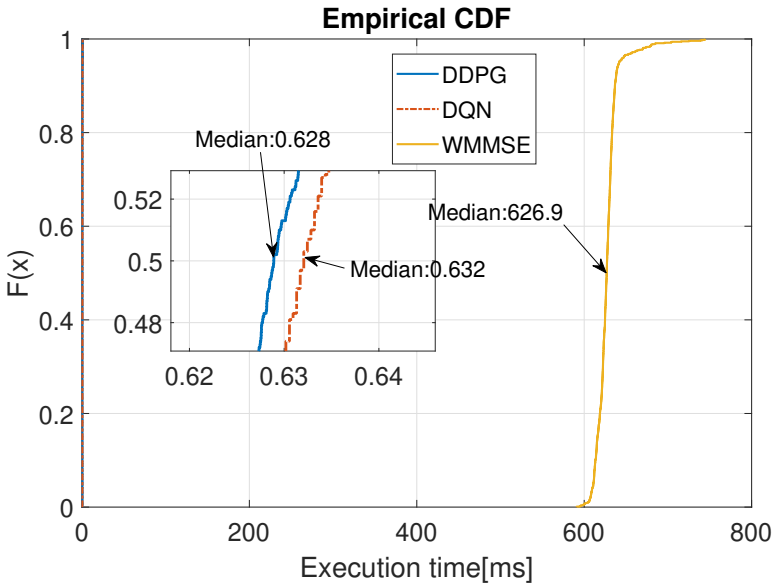


Figure 4.17: Execution time for DQN, DDPG and WMMSE.

max-sum SE power allocation and evaluate their performance against the WMMSE algorithm. The numerical result showed, the sum SE achieved by both methods is higher. Moreover, the execution times of the DRL methods, are much less than the WMMSE algorithm in the same simulation platform.

4.4 Summary of key points in Chapter 4

In this chapter, we investigated the hybrid beamforming and power allocation problem in CF mmWave massive MIMO. Specifically, we first introduced the system model, including the channel model, in CF mmWave massive MIMO. Due to the hardware constraints of present technologies, full-digital beamforming, which requires each antenna to have its own RF chain, is hard to implement. Therefore, hybrid beamforming, where the number of RF chains is less than the number of antennas, is preferred. We proposed a low-complexity design of hybrid ZFB. We studied the effect of the number of RF chains on the performance of hybrid ZFB. From the analysis and the numerical results, we found that when the number of RF chains per AP is twice the number of UEs, hybrid ZFB achieves the same performance as full-digital beamforming. However, the deployment of ZFB in CF mmWave massive MIMO is expensive. Therefore, we proposed NCB as an alter-

native option for the signal processing. Under this condition, we used the DRL methods to allocate power to achieve the max-sum SE scheme. The numerical result showed that, the sum SE achieved by the DRL methods is higher than for the well-adopted WMMSE algorithm. In addition, the execution times are much lower. The key points are:

- The extended S-V channel model is used for the analysis in CF mmWave massive MIMO.
- The condition for ZFB to be used in CF massive MIMO is $M >> K$, where M is the number of antennas per AP, and K is the number of UEs in the coverage area.
- The SE of UEs in CF mmWave massive MIMO is derived from the instantaneous capacity, which is different from the ergodic capacity in sub-6 GHz frequencies. Nevertheless, the power allocation is performed by the expression of the UatF bound (derived by the ergodic capacity technology) to avoid exchanging the CSI in each coherence time.
- In CF mmWave massive MIMO, when the number of RF chains is twice the number of UEs, hybrid ZFB achieves the same performance as full-digital ZFB.
- To enable simultaneous service to all UEs, the number of RF chains per AP should be no less than the number of UEs. This, however, requires a huge number of RF chains in the network. To avoid expensive deployment costs, NCB was proposed for the signal processing.
- Two DRL methods, namely the DQN the DDPG methods, were used to achieve the max-sum SE power allocation scheme.

Chapter 5

Conclusions and Future Work

In this chapter, we summarize the contributions of this thesis and discuss some future research topics in the light of our studies.

5.1 Conclusion

In this thesis, we investigated three power allocation problems in CF massive MIMO. For these non-convex and NP-hard problems, we proposed different DL methods. These methods have a relatively less execution time, which enables the optimization of power allocation to happen in a timely manner.

Problem 1: Max-min power allocation in CF sub-6 GHz massive MIMO.

We proposed a DSL method, namely a two-stage DNN, for implementing the max-min power allocation scheme.

We considered a CF sub-6 GHz massive MIMO with conjugate beamforming in the downlink. The Rayleigh channel model was used for analysis. We derived the closed-form UatF expression to evaluate the SE of the UEs. The analysis considered the effects of channel estimation error and pilot contamination. We compared two types of power allocation schemes: max-min scheme and max-sum SE scheme. Then we devised an algorithm based on the bisection search to perform the max-min power allocation. The optimal solution can be computed by solving a sequence of non-convex feasibility problems. To reduce the computational time of the power allocation, we proposed to use DSL. We used a two-stage DNN to approximate the bisection algorithm. By exploiting the universal approximation theorem of DNN, one can determine the mapping from the large-scale fading information to the optimal power allocation. The execution time of the DNN is much less than the bisection algorithm. We quantitatively compared the performance of the proposed

DNN method to the bisection algorithm. The numerical results showed that the DNN can approximate the bisection algorithm very closely.

Problem 2: Max-sum SE power allocation in CF sub-6 GHz massive MIMO.

We proposed two DRL methods, namely the DQN method and DDPG method, to achieve the max-sum SE power allocation.

We first introduced as a benchmark the WMMSE algorithm, which converts the sum-SE maximization problem to an equivalent minimization problem of the MSE in the data detection. Then we proposed DRL methods, which are based on a multi-agent system. The agents take actions by the optimal policy which is determined by a DNN. The agents interact with the environment to obtain the optimal policy that maximizes a cumulative reward, defined as the sum SE. By exploiting the different learning methods of the agents, two DRL methods were proposed, namely the value-based method DQN and the policy-based method DDPG.

The key to find the optimal policy for the DQN method is to obtain the action-value function, which is approximated by a DNN. The learning of the agents is to adjust the parameters of the DNN, i.e., the weights and bias between the neurons, to approximate the action-value function. Accordingly, the optimal policy of the agents is to perform the action that maximizes the action value. The power allocation of the DQN method is a discrete value.

The idea behind the DDPG method is the actor-critic pattern, which has two DNNs for optimization. The first DNN is called the actor, and interacts with the environment. The second DNN is called the critic, and evaluates the action that has been taken by the actor. The learning process, for the actor, is to get a maximum evaluation from the critic; for the critic, it is to get a precise evaluation of the actor. The actor and the critic work cooperatively to adjust the parameters of the two DNNs. The optimal policy, for the DDPG method, is that the actor performs the action that has the maximum evaluation from the critic. The power allocation of the DDPG method is a continuous value.

We evaluated the performance of the two proposed DRL methods, by comparing with the benchmark WMMSE algorithm, in a 3GPP indoor mixed office scenario. The numerical results showed that, the proposed DRL methods outperform the WMMSE algorithm, in terms of the sum SE. In addition, the execution times of the DRL methods are much less than the WMMSE algorithm running on the same hardware.

Problem 3: Max-sum SE power allocation in CF mmWave massive MIMO.

We used both the DQN method and DDPG method to achieve the max-sum SE power allocation scheme.

We adopted the extended S-V channel model to describe the signal propagation in CF mmWave massive MIMO. Due to the hardware constraints in mmWave signal processing, hybrid beamforming is assumed in the network. We derived the SE

of the UEs by instantaneous capacity. The performance of hybrid beamforming in terms of SE is inferior to the performance of full-digital beamforming. We have studied this degradation. By analyzing the design of hybrid ZFB, we found that when the number of RF chains per AP is twice the number of UEs, hybrid ZFB achieves the same performance as full-digital beamforming. The numerical results, obtained by the Monte Carlo simulations, in the 3GPP indoor mixed office scenario, demonstrated this.

The condition of using hybrid ZFB is not always fulfilled in CF mmWave massive MIMO. The number of antennas per AP should be sufficiently larger than the number of UEs. In addition, to enable the UEs being served simultaneously in the network, the number of RF chains per AP should be no less than the number of UEs. These constraints make hybrid ZFB expensive. An alternative option is the NCB, which only performs a conjugate operation in the APs and can be easily deployed. This however, unlike ZFB that suppresses the inter-UE interference, makes the optimization of power allocation non-convex and hence computationally hard. Therefore, we proposed the DQN method and the DDPG method to optimize the power allocation, under the UatF expression. The results of the Monte Carlo simulations showed that the sum SE achieved by DRL methods is higher than for the WMMSE algorithm. Moreover, the execution time is much less on the same hardware.

5.2 Future work

Although it became clear from our results that DL is a viable method to perform power allocation in CF massive MIMO, i.e., it achieves results that are as good as conventional non-DL methods but requires a significantly less execution time, the current work only represents a first step towards understanding the capability of DNNs for this type of problems. Based on the insights we have gained in this thesis, several points for future work regarding DL and CF massive MIMO can be identified.

Channel models based on measurements

Accurate channel models are required for an accurate performance analysis of CF massive MIMO. In this thesis, we considered the Rayleigh model for sub-6 GHz channels and the extended S-V model for mmWave channels. Although the results based on these channel models convincingly show the promising characteristics of the proposed DL-based power allocation, in a realistic environment these models may not reflect the channel characteristics well enough to get accurate performance predictions. Therefore channel models obtained from measurements in a variety of realistic scenarios should be developed and used in further studies.

DRL for max-min power allocation scheme

In Chapter 2 and Chapter 3, we found that the max-min power allocation scheme is more complicated than the max-sum SE scheme. Therefore, we proposed a DSL method to approximate the existing algorithm, rather than optimize the objective function. However, by doing so, the solution obtained by the DSL method heavily depends on the algorithm that is being approximated. This implies that when the algorithm that is being approximated ends up in a local optimum, the DSL accordingly obtains a sub-optimal solution. A promising method for the max-min power allocation is the DRL method. The DRL method usually optimizes the objective function by its own regime, therefore it potentially outperforms the existing algorithms after sufficient training. However, the design of a DRL to perform the max-min power allocation is difficult. The idea of the DRL method is to maximize an objective function that is represented by a cumulative reward; the fairness target of the max-min scheme is not fit for the DRL regime. Nevertheless, we conducted some first studies that define the minimum SE as the cumulative reward. From our simulation results, we have not found any evidence that the DRL method performs better than the other methods. Nevertheless, one could expect a novel design of the DRL method that works better than the bisection algorithm in terms of per-UE SE. This should be further explored.

Different scenarios for CF massive MIMO

The numerical results in this thesis were based on some particular scenarios, e.g., the 3GPP indoor mixed office scenario in Chapter 4. This however potentially might limit the applicability of our results. The assumptions in these scenarios consider that the APs are evenly spread in a grid over the coverage area. Determining the AP placement to optimize particular performance measures is a non-trivial issue. It depends, e.g., on the channel response, environment (e.g., the materials of the walls, obstructions, objects, or people in the area, etc.) and the behavior of the UEs. In addition, different UE mobility models should be considered in further studies. Therefore, to generalize our results and insights, different and potentially extreme scenarios should be explored.

Power allocation for CF massive MIMO based on small-scale fading

The power allocation we considered in this thesis is performed on the time scale of the large-scale fading variation. As we argued in Section 2.1 and Section 3.2, the power is expected to be controlled on the time scale of the small-scale fading variation. It has been shown in [31] that small-scale fading based power control significantly improves the sum SE in the uplink of CF massive MIMO compared to the large-scale fading based power control. This is because the CF massive MIMO experiences a low channel hardening level when the number of antennas per AP is limited. Recall that the power allocation is performed following the large-scale fading based UatF expression, which is tight only under the condition of a high level

of channel hardening. The big difference between the UatF expression and the instantaneous SE makes the large-scale fading based power allocation no longer optimal. Therefore, also methods should be explored to allocate the power based on the small-scale fading.

Tradeoff between overhead cost and the improvement of SINR

In Section 4.1.2, we discussed centralized ZFB in CF massive MIMO. Centralized ZFB relies on the full CSI, i.e., the CSI of all links between the APs and UEs. To perform centralized ZFB, the CC needs to collect the full CSI from all the links and forward the beamforming coefficients to the APs via the fronthaul. This requires more overhead processing than CB does. The benefit is a higher SINR for the UEs. Therefore, the tradeoff between the overhead cost and the improvement of SINR is of interest and needs to be studied further.

Multi-antenna UEs

In the definitions of massive MIMO (**Definition 1**) and CF massive MIMO (**Definition 2**), the UEs were assumed to be single-antenna devices. However more complex antenna configurations, involving multiple antennas or even multiple cooperating antenna arrays are being introduced in UEs, in particular for mmWave communication (e.g., the iPhone Pro 12, https://www.patentlyapple.com/patently-apple/2021/06/apple-describes-the-use-of-multiple-millimeter-wave-antennas-in-future-iphones-that-will-provide-superior-5g-communications.html). A further study should address the power allocation problem in CF massive MIMO networks when multi-antenna UEs are involved.

Appendix A

Use-and-then-forget expression

In this appendix we derive the closed form of the UatF expression, which is used for calculating the SE of the UEs in high-level channel hardening scenarios. [54] shows that achievable data rate of reliable data transmission [55] will not exceed the Shannon limit [56]. Therefore, the key to derive the closed form of UatF expression is to formalize the SINR, where:

$$SINR_k = \frac{|\text{Intended}|^2}{\mathbb{E}\{|\text{Fluctuation}|^2\} + \sum_{k' \neq k} \mathbb{E}\{|\text{Interference}_{k'}|^2\} + \mathbb{E}\{|\text{Noise}|^2\}} \tag{A.1}$$

1. $\left|\text{Intended}\right|^2$

   For UE k, the *signal* in SINR is calculated as follows:

$$\begin{aligned} |\text{Intended}|^2 &= \left| \mathbb{E}\left\{ \sum_{n=1}^{N} \sqrt{p_{k,n}} \mathbf{g}_{k,n} \frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}} \right\} \right|^2 \\ &= \left| \sum_{n=1}^{N} \sqrt{\frac{p_{k,n}}{M\gamma_{k,n}}} \mathbb{E}\left\{ \mathbf{g}_{k,n} \hat{\mathbf{g}}_{k,n}^H \right\} \right|^2 \\ &\overset{(a)}{=} M \left(\sum_{n=1}^{N} \sqrt{p_{k,n}\gamma_{k,n}} \right)^2 \end{aligned} \tag{A.2}$$

where (a) follows the fact that:

$$\begin{aligned}\mathbb{E}\left\{\mathbf{g}_{k,n}\hat{\mathbf{g}}_{k,n}^{H}\right\} &= \mathbb{E}\left\{\mathbf{g}_{k,n}\frac{\sqrt{p_p\tau_p}\beta_{k,n}}{p_p\tau_p\sum_{k'=1}^{K}\beta_{k',n}|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2+1}\left(\sqrt{p_p\tau_p}\sum_{k'=1}^{K}\mathbf{g}_{k',n}^{T}\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H+\mathbf{w}_n\boldsymbol{\psi}_k^H\right)^H\right\}\\ &= \frac{\sqrt{p_p\tau_p}\beta_{k,n}}{p_p\tau_p\sum_{k'=1}^{K}\beta_{k',n}|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2+1}\sqrt{p_p\tau_p}\mathbb{E}\left\{|\mathbf{g}_{k,n}|^2\right\}\\ &= M\frac{p_p\tau_p\beta_{k,n}^2}{p_p\tau_p\sum_{k'=1}^{K}\beta_{k',n}|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2+1}\\ &= M\gamma_{n,k}\end{aligned} \tag{A.3}$$

2. $\mathbb{E}\left\{\left|\text{Fluctuation}\right|^2\right\}$

For UE k, the fluctuation is calculated as follows:

$$\begin{aligned}\mathbb{E}\left\{\left|\text{Fluctuation}\right|^2\right\} &= \mathbb{E}\left\{\left|\sum_{n=1}^{N}\sqrt{p_{k,n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}}-\mathbb{E}\left\{\sum_{n=1}^{N}\sqrt{p_{k,n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}}\right\}\right|^2\right\}\\ &= \mathbb{D}\left\{\sum_{n=1}^{N}\sqrt{p_{k,n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k,n}^H}{\sqrt{M\gamma_{k,n}}}\right\}\\ &= \sum_{n=1}^{N}\frac{p_{k,n}}{M\gamma_{k,n}}\mathbb{D}\left\{\mathbf{g}_{k,n}\hat{\mathbf{g}}_{k,n}^H\right\}\\ &= \sum_{n=1}^{N}\frac{p_{k,n}}{M\gamma_{k,n}}\left(\frac{\sqrt{p_p\tau_p}\beta_{k,n}}{p_p\tau_p\sum_{k'=1}^{K}\beta_{k',n}|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2+1}\right)^2 b\end{aligned} \tag{A.4}$$

where

$$\begin{aligned}b &= \mathbb{D}\left\{\mathbf{g}_{k,n}\left(\sqrt{p_p\tau_p}\sum_{k'=1}^{K}\mathbf{g}_{k',n}^{T}\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H+\mathbf{w}_n\boldsymbol{\psi}_k^H\right)^*\right\}\\ &= p_p\tau_p\sum_{k'=1}^{K}\mathbb{D}\left\{\mathbf{g}_{k,n}\mathbf{g}_{k',n}^H\right\}\left|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H\right|^2+M\beta_{k,n}\\ &= p_p\tau_p\mathbb{D}\left\{\mathbf{g}_{k,n}\mathbf{g}_{k,n}^H\right\}+p_p\tau_p\sum_{k'\neq k}\mathbb{D}\left\{\mathbf{g}_{k,n}\mathbf{g}_{k',n}^H\right\}\left|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H\right|^2+M\beta_{k,n}\\ &= p_p\tau_p\mathbb{D}\left\{\left|\mathbf{g}_{k,n}\right|^2\right\}+p_p\tau_pM\beta_{k,n}\beta_{k',n}\left|\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H\right|^2+M\beta_{k,n}\end{aligned} \tag{A.5}$$

Since $\mathbf{g}_{k,n}\sim\mathbb{CN}(\mathbf{0},\beta_{k,n}\mathbf{I}_M)$, one has

$$\mathbb{D}\left\{\left|\mathbf{g}_{k,n}\right|^2\right\} = \mathbb{D}\left\{\left[\text{Re}\left(\mathbf{g}_{k,n}\right)\right]^2\right\}+\mathbb{D}\left\{\left[\text{Im}\left(\mathbf{g}_{k,n}\right)\right]^2\right\} \tag{A.6}$$

where $\mathrm{Re}(\mathbf{g}_{k,n}), \mathrm{Im}(\mathbf{g}_{k,n}) \sim \mathbb{N}(\mathbf{0}, \beta_{k,n}\mathbf{I}_M/2)$, then:

$$\left[\frac{\mathrm{Re}(\mathbf{g}_{k,n})}{\sqrt{\beta_{k,n}/2}}\right]^2, \left[\frac{\mathrm{Im}(\mathbf{g}_{k,n})}{\sqrt{\beta_{k,n}/2}}\right]^2 \sim \chi(\mathbf{1}) \tag{A.7}$$

So one can obtain:

$$\begin{aligned} \mathbb{D}\left\{\left[\frac{\mathrm{Re}(\mathbf{g}_{k,n})}{\sqrt{\beta_{k,n}/2}}\right]^2\right\} &= \mathbb{D}\left\{\left(\frac{\mathrm{Im}(\mathbf{g}_{k,n})}{\sqrt{\beta_{k,n}/2}}\right)^2\right\} = 2M \\ \mathbb{D}\left\{\left[\mathrm{Re}(\mathbf{g}_{k,n})\right]^2\right\} &= \mathbb{D}\left\{\left[\mathrm{Im}(\mathbf{g}_{k,n})\right]^2\right\} = \frac{M\beta_{k,n}^2}{2} \end{aligned} \tag{A.8}$$

i.e.,

$$\mathbb{D}\left\{\left|\mathbf{g}_{k,n}\right|^2\right\} = M\beta_{k,n}^2 \tag{A.9}$$

Plugging (A.4), (A.5) and (A.9), one obtains:

$$\mathbb{E}\left\{\left|\mathrm{Fluctuation}\right|^2\right\} = \sum_{n=1}^{N} p_{k,n}\beta_{k,n} \tag{A.10}$$

3. $\mathbb{E}\left\{\left|\mathrm{Interference}_{k'}\right|^2\right\}$

$$\begin{aligned} &\sum_{k'\neq k} \mathbb{E}\left\{\left|\mathrm{Interference}_{k'}\right|^2\right\} = \sum_{k'\neq k} \mathbb{E}\left\{\left|\sum_{n=1}^{N} \sqrt{p_{k',n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}}\right|^2\right\} \\ &= \sum_{k'\neq k}\left(\mathbb{E}\left\{\sum_{n=1}^{N} \sqrt{p_{k',n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}}\right\}\right)^2 + \sum_{k'\neq k}\mathbb{D}\left\{\sum_{n=1}^{N} \sqrt{p_{k',n}}\mathbf{g}_{k,n}\frac{\hat{\mathbf{g}}_{k',n}^H}{\sqrt{M\gamma_{k',n}}}\right\} \\ &= \sum_{k'\neq k}\left(\sum_{n=1}^{N}\sqrt{\frac{p_{k',n}}{M\gamma_{k',n}}}\mathbb{E}\left\{\mathbf{g}_{k,n}\hat{\mathbf{g}}_{k',n}^H\right\}\right)^2 + \sum_{k'\neq k}\sum_{n=1}^{N}\frac{p_{k',n}}{M\gamma_{k',n}}\mathbb{D}\left\{\mathbf{g}_{k,n}\hat{\mathbf{g}}_{k',n}^H\right\} \\ &\overset{(c)}{=} M\sum_{k'\neq k}\left(\sum_{n=1}^{N}\sqrt{p_{k',n}\gamma_{k',n}}\frac{\beta_{k,n}}{\beta_{k',n}}\right)^2 |\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H|^2 + \sum_{k'\neq k}\sum_{n=1}^{N} p_{k',n}\beta_{k,n} \end{aligned} \tag{A.11}$$

where (c) follow:

$$\mathbb{E}\left\{\mathbf{g}_{k,n}\hat{\mathbf{g}}_{k',n}^H\right\} = M\gamma_{k',n}\frac{\beta_{k,n}}{\beta_{k',n}}\boldsymbol{\psi}_{k'}\boldsymbol{\psi}_k^H \tag{A.12}$$

and

$$\mathbb{D}\left\{\mathbf{g}_{k,n}\hat{\mathbf{g}}_{k',n}^H\right\} = M\gamma_{k',n}\beta_{k,n} \tag{A.13}$$

Plugging (A.1), (A.2), (A.10) and (A.11), one obtains (2.12).

Appendix B

Deep learning and deep neural networks

DL is an emerging technology to be utilized in future wireless communications to solve the computational complexity issues [10]. This appendix provides a brief introduction of this technology to help understand the proposed DL methods in this thesis.

B.1 Overview

DL is a class of machine learning that uses multilayer neural networks for computing systems that can 'learn' the features of functions from the input data [108] [109]. The 'deep' derives from the fact that the neural network has multiple layers. The neural network used in DL is called DNN. DNN is inspired by information processing and distributed communication nodes in biological systems. Different from biological brains, the neurons in DNN work in a mathematical way, see Fig.B.1. Fig.B.1 shows an example of a fully-connected DNN, which means that each neuron in a given layer is connected to all the neurons of the next layer. The output of the neurons is determined by the input via a deterministic function:

$$\mathbf{x}_{i+1} = \hat{f}_{i+1}(\mathbf{x}_i, \boldsymbol{\theta}_{i+1}) \tag{B.1}$$

where $\mathbf{x}_i$ is input of i-th layer, $\mathbf{x}_{i+1}$ is the output of the i-th layer (also the input for the $(i+1)$-th layer), $\boldsymbol{\theta}_{i+1}$ represents the parameters between the i-th layer and the $(i+1)$-th layer. Typically the function $\hat{f}_{i+1}$ is determined by the parameters $\boldsymbol{\theta}_{i+1} = \{\mathbf{W}_{i+1}, \mathbf{b}_{i+1}\}$ and modeled as:

$$\hat{f}_{i+1}(\mathbf{x}_i, \boldsymbol{\theta}_{i+1}) = \Gamma\left(\mathbf{W}_{i+1}\mathbf{x}_i + \mathbf{b}_{i+1}\right) \tag{B.2}$$

where $\mathbf{W}_{i+1}$ is called the weight matrix, $\mathbf{b}_{i+1}$ is called the bias vector and Γ is an element-wise nonlinear function that is called the activation function. Following the forward propagation, the complete input-output relation for the case in Fig.B.1 is:

$$\mathbf{y} = \hat{f}_3\left(\hat{f}_2\left(\hat{f}_1\left(\mathbf{x}_0, \boldsymbol{\theta_1}\right), \boldsymbol{\theta_2}\right), \boldsymbol{\theta_3}\right) \tag{B.3}$$

The initial parameters $\boldsymbol{\theta}_i$ $(i = 1, 2, ...)$ are chosen randomly and then optimized by training. The training process of the DNN is the update of the parameters $\boldsymbol{\theta}_i$ in each layer. Depending on the training method, the DL is classified into three categories, namely supervised learning, unsupervised learning, and reinforcement learning.

For supervised learning, the training data consist of the input-output pair, i.e., $\mathbf{x}_0 - \mathbf{y}$ pair in Fig.B.1. 'Supervised' means that the output element is known and labeled by other methods. In other words, the DNN is 'supervised' by other methods.

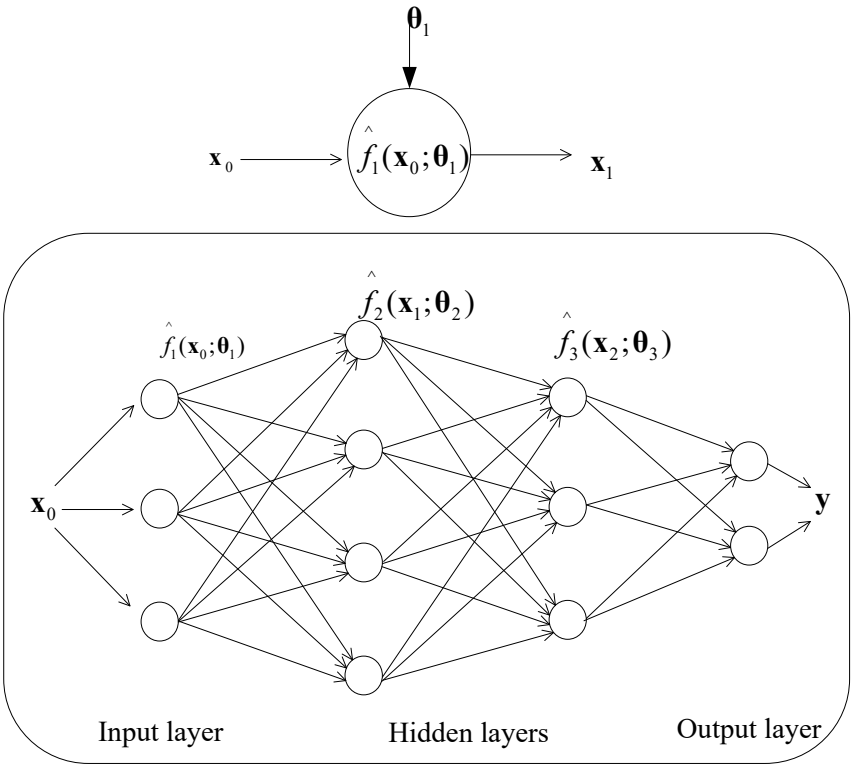


Figure B.1: An example of a fully-connected DNN.

Supervised learning is used for classification and for approximating other methods.

For unsupervised learning, the training data only consist of the input element, i.e., $\mathbf{x}_0$ in Fig.B.1. The output element is unknown for the DNN. The DNN labels the output elements according to the similarities and differences of the input elements. Unsupervised learning is usually used for clustering tasks.

For reinforcement learning, the training data consist of the input element, i.e., $\mathbf{x}_0$ in Fig.B.1. Like unsupervised learning, there is no labeled output element in the training data. The DNN labels the output elements by maximizing a cumulative reward obtained from the environment. Reinforcement learning is used for the optimization problems.

B.2 Training

The labeled output elements, generated by other methods or cumulative reward, are used to adjust the parameters of the DNN, i.e., $\boldsymbol{\theta}_i$ in Fig.B.1. The training process is done by simple gradient descent. A loss function (see (B.4)), typically defined as the error between the labeled output element and the output of the forward propagation by the DNN, is used to compute the gradient.

$$loss = ||\mathbf{y}_{\text{labeled}} - \mathbf{y}_{\text{DNN}}||^2 \tag{B.4}$$

By the back propagation procedure [13], the parameters in layer i are updated as follows:

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i - lr\partial\frac{loss}{\partial\boldsymbol{\theta}_i} \tag{B.5}$$

where lr is the learning rate, $\partial\frac{loss}{\partial\boldsymbol{\theta}_i}$ is the gradient in layer i. The learning rate controls how fast the DNN learns the input-output pair. In general, a large learning rate may lead to unstable training whereas a small value leads to a failure to train the DNN. The update of the parameters in (B.5) repeats until the desired loss function value is obtained. Labeled with the iteration, (B.5) is rewritten as:

$$\boldsymbol{\theta}_i^{(j)} = \boldsymbol{\theta}_i^{(j)} - lr\partial\frac{loss}{\partial\boldsymbol{\theta}_i^{(j)}} \tag{B.6}$$

Typically, the DNN needs a huge number of training data to adjust the parameters $\boldsymbol{\theta}_i$; (B.4) shows the loss function of an example of one training data. Here we rewrite (B.4) in case of multiple training data:

$$loss = \frac{1}{D}\sum_{d=1}^{D}||\mathbf{y}_{\text{labeled}}^{(d)} - \mathbf{y}_{\text{DNN}}^{(d)}||^2 \tag{B.7}$$

where D is the number of training data for the DNN. The method using (B.7) to update the parameters in each iteration is called full batch learning. However, this is very expensive when the training set is huge. An efficient way is to use a random subset of the training set, called minibatch, to evaluate the gradients for the update of the parameters. By using the minibatch training, the loss function is formalized as follows:

$$loss = \frac{1}{\bar{D}} \sum_{d=1}^{\bar{D}} ||\mathbf{y}_{\text{labeled}}^{(d)} - \mathbf{y}_{\text{DNN}}^{(d)}||^2 \tag{B.8}$$

where $\bar{D}$ is the size of the minibatch.

B.3 Framework

After the training, the DNN can be used for solving complicated problems. Fig.B.2 shows the way of exploiting a DNN for complicated problems.

Typically, the DNN requires an extensive, offline, training process before it becomes operational. However, once the DNN is well-trained, the operation phase only takes into account the described forward propagation (B.3). The low complexity of the DNN method benefits from "making efforts in advance".

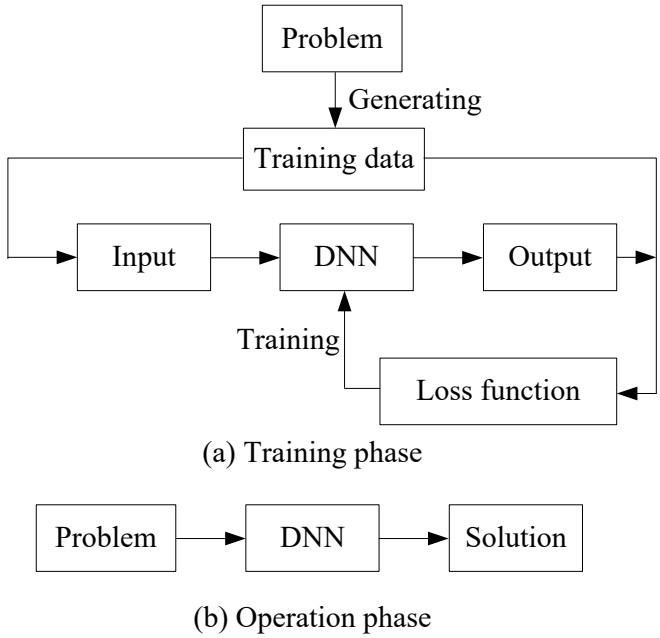


Figure B.2: The framework of DNN method for complicated problems.

Appendix C

Convolutional neural networks

CNN is a class of DNN, applied to process data that come in the form of multiple arrays [13] [110]. Different from the structure of fully connected DNN, the architecture of a typical CNN is structured as a series of stages. The first few stages are composed of two types of layers: convolutional layer and pooling layers, see Fig.C.1.

Fig.C.1 shows an example of a convolutional layer and a pooling layer in CNN,

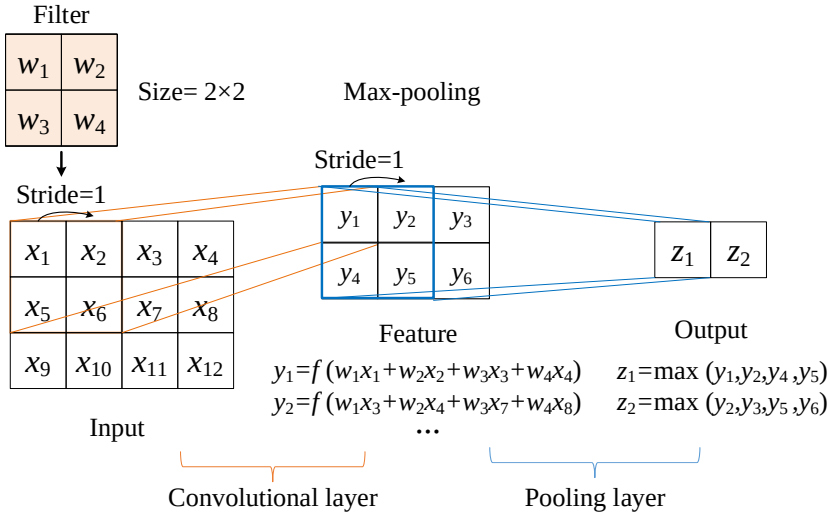


Figure C.1: Convolutional layer and pooling layer in CNN.

where one filter operates on the input data and then a max-pooling operation is processed after the convolutional layer. The filter, with size of 2×2, slides on the input data to extract the feature. The calculation is shown in the bottom of the figure, where f represents the activation function. There is a stride to control the filter shifts over the input data. If there are Q filters in a CNN, Q features are obtained after the convolutional layer. Like the convolution operation, the pooling operation has a size and stride. In Fig.C.1, we show the max-pooling operation, which selects the maximum element within its window. There are other pooling operations such as mean-pooling, which calculate the mean of the elements within its window.

A CNN comprises two or three stages of convolution and pooling, then several fully connected layers are followed to compute the final output, see Fig. C.2.

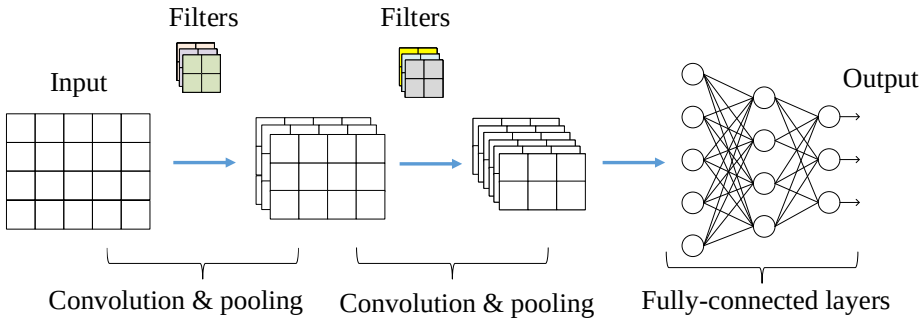


Figure C.2: Convolutional neural network.

Bibliography

[1] Ericsson, *Ericsson Mobility Visualizer*. Available: https://www.ericsson.com/en.

[2] Z. Zhang *et al.*, "6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28-41, Sept. 2019.

[3] S. Ahmadi, *5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards*. New York, NY, USA: Academic, 2019.

[4] K. David and H. Berndt, "6G Vision and Requirements: Is There Any Need for Beyond 5G?," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72-80, Sept. 2018.

[5] Shuping Dang *et al.*, "What should 6G be?." *Nature Electronics* 3.1 (2020): 20-29.

[6] S. Chen, Y. C. Liang, S. Sun, S. Kang, W. Cheng and M. Peng, "Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 218-228, April 2020.

[7] A. Yastrebova, R. Kirichek, Y. Koucheryavy, A. Borodin and A. Koucheryavy, "Future Networks 2030: Architecture Requirements," *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2018, pp. 1-8.

[8] International Mobile Telecommunications, "6G vision and candidate technologies," China, White Paper, 2021.

[9] E. G. Larsson, O. Edfors, F. Tufvesson and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186-195, February 2014.

[10] Van Chien T. Spatial Resource Allocation in Massive MIMO Communications: From Cellular to Cell-Free. Linköping University Electronic Press, 2019.

[11] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834-1850, March 2017.

[12] Q. Shi, M. Razaviyayn, Z. Luo and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[13] LeCun Y, Bengio Y and Hinton G, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin and R. Zhang, "An Overview of Massive MIMO: Benefits and Challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742-758, Oct. 2014.

[15] Q. Li *et al.*, "MIMO techniques in WiMAX and LTE: a feature overview," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 86-92, May 2010.

[16] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, "Introduction," *Fundamentals of Massive MIMO*, Cambridge: Cambridge University Press, 2016, pp. 5–18.

[17] T. L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590-3600, November 2010.

[18] Emil Björnson, Jakob Hoydis and Luca Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *Foundations and Trends in Signal Processing*, vol. 11, no. 3, pp. 154–655, 2017.

[19] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4-24, April 1988.

[20] Kaltenberger, H. Jiang, M. Guillaud and R. Knopp, "Relative channel reciprocity calibration in MIMO/TDD systems," *2010 Future Network Mobile Summit*, 2010, pp. 1-10.

[21] B. M. Hochwald, T. L. Marzetta and V. Tarokh, "Multiple-antenna channel hardening and its implications for rate feedback and scheduling," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 1893-1909, Sept. 2004.

[22] Z. Chen and E. Björnson, "Channel Hardening and Favorable Propagation in Cell-Free Massive MIMO With Stochastic Geometry," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5205-5219, Nov. 2018.

[23] J. Qiu, K. Xu, X. Xia, Z. Shen and W. Xie, "Downlink Power Optimization for Cell-Free Massive MIMO Over Spatially Correlated Rayleigh Fading Channels," *IEEE Access*, vol. 8, pp. 56214-56227, 2020.

[24] T. C. Mai, H. Q. Ngo and T. Q. Duong, "Downlink Spectral Efficiency of Cell-Free Massive MIMO Systems With Multi-Antenna Users," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4803-4815, Aug. 2020.

[25] J. Zhang, Y. Wei, E. Björnson, Y. Han and S. Jin, "Performance Analysis and Power Control of Cell-Free Massive MIMO Systems with Hardware Impairments," *IEEE Access*, vol. 6, pp. 55302-55314, 2018.

[26] A. Papazafeiropoulos, P. Kourtessis, M. D. Renzo, S. Chatzinotas and J. M. Senior, "Performance Analysis of Cell-Free Massive MIMO Systems: A Stochastic Geometry Approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3523-3537, April 2020.

[27] H. V. Nguyen *et al.*, "On the Spectral and Energy Efficiencies of Full-Duplex Cell-Free Massive MIMO," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1698-1718, Aug. 2020.

[28] Özlem Tugfe Demir, Emil Björnson and Luca Sanguinetti, "Foundations of User-Centric Cell-Free Massive MIMO", *Foundations and Trends in Signal Processing*, vol. 14: no. 3-4, pp 162-472.

[29] E. Björnson and L. Sanguinetti, "Scalable Cell-Free Massive MIMO Systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247-4261, July 2020.

[30] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao and R. H. Middleton, "Cell-Free Massive MIMO for Wireless Federated Learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6377-6392, Oct. 2020.

[31] M. Bashar *et al.*, "Exploiting Deep Learning in Limited-Fronthaul Cell-Free Massive MIMO Uplink," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1678-1697, Aug. 2020.

[32] M. Ke, Z. Gao, Y. Wu, X. Gao and K. -K. Wong, "Massive Access in Cell-Free Massive MIMO-Based Internet of Things: Cloud Computing and Edge Computing Paradigms," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 756-772, March 2021.

[33] S. Chen, J. Zhang, E. Björnson, J. Zhang and B. Ai, "Structured Massive Access for Scalable Cell-Free Massive MIMO Systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1086-1100, April 2021.

[34] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic and R. W. Heath, "MIMO Precoding and Combining Solutions for Millimeter-Wave Systems," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 122-131, December 2014.

[35] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, “Models and Preliminaries,” *Fundamentals of Massive MIMO*, Cambridge: Cambridge University Press, 2016, pp. 19–44.

[36] E. Bjornson and P. Giselsson, "Two Applications of Deep Learning in the Physical Layer of Communication Systems [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 134-140, Sept. 2020.

[37] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 56, no. 518, pp. 529–533, 2015.

[38] T. Van Chien, T. Nguyen Canh, E. Björnson and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5732-5748, Sept. 2020.

[39] M. Wang, F. Gao, S. Jin and H. Lin, "An Overview of Enhanced Massive MIMO With Array Signal Processing Techniques," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 886-901, Sept. 2019.

[40] H. Q. Ngo, L. Tran, T. Q. Duong, M. Matthaiou and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 25-39, March 2018.

[41] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, "Cell-free massive MIMO: Uniformly great service for everyone," *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2015, pp. 201-205.

[42] E. Nayebi, A. Ashikhmin, T. L. Marzetta and H. Yang, "Cell-free massive MIMO systems," *2015 49th Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 695-699.

[43] G. Interdonato, H. Q. Ngo and E. G. Larsson, "Enhanced normalized conjugate beamforming for cell-free massive MIMO," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 2863-2877, May 2021.

[44] A. Papazafeiropoulos, H. Q. Ngo, P. Kourtessis, S. Chatzinotas and J. M. Senior, "Towards Optimal Energy Efficiency in Cell-Free Massive MIMO Systems," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 816-831, June 2021.

[45] S. -N. Jin, D. -W. Yue and H. H. Nguyen, "Spectral and energy efficiency in cell-free massive MIMO systems over correlated Rician fading," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2822-2833, June 2021.

[46] Y. Zhang, L. Yang and H. Zhu, "Cell-Free Massive MIMO Systems With Low-Resolution ADCs: The Rician Fading Case," *IEEE Systems Journal*, doi: 10.1109/JSYST.2020.

[47] J. -. van de Beek, O. Edfors, M. Sandell, S. K. Wilson and P. O. Borjesson, "On channel estimation in OFDM systems," *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, 1995, pp. 815-819.

[48] Efron B, “Bayes’ theorem in the 21st century,” *Science*, vol. 340, no. 6137, pp. 436–444, 2013.

[49] N. Fatema, G. Hua, Y. Xiang, D. Peng and I. Natgunanathan, "Massive MIMO Linear Precoding: A Survey," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3920-3931, Dec. 2018.

[50] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, “Single-Cell Systems,” *Fundamentals of Massive MIMO*, Cambridge: Cambridge University Press, 2016, pp. 45–76.

[51] G. Caire, "On the Ergodic Rate Lower Bounds With Applications to Massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3258-3268, May 2018.

[52] W. Zeng, Y. He, B. Li and S. Wang, "Pilot Assignment for Cell Free Massive MIMO Systems Using a Weighted Graphic Framework," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6190-6194, June 2021.

[53] T. Van Chien, E. Björnson and E. G. Larsson, "Joint Power Allocation and Load Balancing Optimization for Energy-Efficient Cell-Free Massive MIMO Networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6798-6812, Oct. 2020.

[54] D. Tse and P. Viswanath, Fundamentals of wireless communication. Cambridge university press, 2005.

[55] D. Hammerschmidt, T. Dittfeld and S. Brewerton, "Reliable data transmission with reduced bit error rate," Nov. 11, 2014, US Patent 8887022.

[56] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," *Proceedings of ICC '93 - IEEE International Conference on Communications*, 1993, pp. 1064-1070 vol.2.

[57] Sikorski, K. "Bisection is optimal, " *Numer. Math.* 40, 111–117 (1982).

[58] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein, Clifford. "3.1: Asymptotic notation". *Introduction to Algorithms 2nd*. MIT Press and McGraw–Hill. 2001.

[59] 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 Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438-5453, 15 Oct.15, 2018.

[60] D. A. Palmer and M. Florea, "Neural processing unit," Feb. 18, 2014, US Patent 8,655,815.

[61] Y. Sun, Y. Wang, J. Jiao, S. Wu and Q. Zhang, "Deep Learning-Based Long-Term Power Allocation Scheme for NOMA Downlink System in S-IoT," *IEEE Access*, vol. 7, pp. 86288-86296, 2019.

[62] M. A. Wijaya, K. Fukawa and H. Suzuki, "Intercell-Interference Cancellation and Neural Network Transmit Power Optimization for MIMO Channels," *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, 2015, pp. 1-5.

[63] L. Sanguinetti, A. Zappone and M. Debbah, "Deep Learning Power Allocation in Massive MIMO," *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 1257-1261.

[64] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." *Neural networks* 2.5 (1989): 359-366.

[65] T. N. Sainath *et al.*, "Improvements to Deep Convolutional Neural Networks for LVCSR," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 315-320.

[66] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142-3155, July 2017.

[67] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press, 2016.

[68] Zeiler, Matthew D. "Adadelta: an adaptive learning rate method." arXiv preprint :1212.5701 (2012).

[69] Li M, *et al.*, "Efficient mini-batch training for stochastic optimization," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 661–670.

[70] Z. Wang, E. K. Tameh and A. R. Nix, "Joint Shadowing Process in Urban Peer-to-Peer Radio Channels," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 52-64, Jan. 2008.

[71] Ao Tang, JiXian Sun and Ke Gong, "Mobile propagation loss with a low base station antenna for NLOS street microcells in urban area," *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)*, 2001, pp. 333-336 vol.1.

[72] Singh, Yuvraj. "Comparison of okumura, hata and cost-231 models on the basis of path loss and signal strength." *International journal of computer applications* 59.11 (2012).

[73] F. Meng, P. Chen and L. Wu, "Power allocation in multi-user cellular networks with deep Q learning approach," *ICC 2019* , Shanghai, China.

[74] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[75] F. Meng and P. Chen, "Power Allocation in Multi-User Cellular Networks: Deep Reinforcement Learning Approaches," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6255–6267, 2020.

[76] D. Guo, L. Tang, X. Zhang and Y. -C. Liang, "Joint Optimization of Handover Control and Power Allocation Based on Multi-Agent Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13124-13138, Nov. 2020.

[77] Y. Zhang, C. Kang, T. Ma, Y. Teng and D. Guo, "Power Allocation in Multi-Cell Networks Using Deep Reinforcement Learning," *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1-6.

[78] Barron E N and Ishii H, “The Bellman equation for minimizing the maximum cost,” *Nonlinear Analysis: Theory, Methods Applications*, vol. 13, no. 9, pp. 1067–1090, 1989.

[79] Lillicrap, Timothy P. *et al.*, "Continuous control with deep reinforcement learning." arXiv preprint:1509.02971 (2015).

[80] 3GPP Radio Access Network Working Group,“Study on channel model for frequencies from 0.5 to 100 GHz (Release 15),” 3GPP TR 38.901, 2018.

[81] T. A. Thomas, H. C. Nguyen, G. R. MacCartney and T. S. Rappaport, "3D mmWave Channel Model Proposal,"*2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1-6.

[82] S. Deng, M. K. Samimi and T. S. Rappaport, "28 GHz and 73 GHz millimeter-wave indoor propagation measurements and path loss models," *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1244-1250.

[83] M. K. Samimi and T. S. Rappaport, "3-D Millimeter-Wave Statistical Channel Model for 5G Wireless System Design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 7, pp. 2207-2225, July 2016.

[84] K. Haneda *et al.*, "5G 3GPP-Like Channel Models for Outdoor Urban Microcellular and Macro cellular Environments," *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1-7.

[85] X. Wu *et al.*, "60-GHz Millimeter-Wave Channel Measurements and Modeling for Indoor Office Environments," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 4, pp. 1912-1924, April 2017.

[86] Y. Xing, O. Kanhere, S. Ju and T. S. Rappaport, "Indoor Wireless Channel Properties at Millimeter Wave and Sub-Terahertz Frequencies," *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1-6.

[87] S. Ju, Y. Xing, O. Kanhere and T. S. Rappaport, "3-D Statistical Indoor Channel Model for Millimeter-Wave and Sub-Terahertz Bands," *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1-7.

[88] S. Ju, Y. Xing, O. Kanhere and T. S. Rappaport, "Millimeter Wave and Sub-Terahertz Spatial Statistical Channel Model for an Indoor Office Building," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 6, pp. 1561-1575, June 2021.

[89] A. A. M. Saleh and R. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128-137, February 1987.

[90] I. Ferreira, S. Mota and A. Rocha, "Results from MIMO channel measurements and comparison to the extended Saleh-Valenzuela model," *Proceedings of the Fourth European Conference on Antennas and Propagation*, Barcelona, 2010, pp. 1-5.

[91] C. Gustafson, K. Haneda, S. Wyne and F. Tufvesson, "On mm-Wave Multipath Clustering and Channel Modeling," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 3, pp. 1445-1455, March 2014.

[92] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi and R. W. Heath, "Spatially Sparse Precoding in Millimeter Wave MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1499-1513, March 2014.

[93] X. Yu, J. Shen, J. Zhang and K. B. Letaief, "Alternating Minimization Algorithms for Hybrid Precoding in Millimeter Wave MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 485-500, April 2016.

[94] L. Liang, W. Xu and X. Dong, "Low-Complexity Hybrid Precoding in Massive Multiuser MIMO Systems," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 653-656, Dec. 2014.

[95] F. Sohrabi and W. Yu, "Hybrid Digital and Analog Beamforming Design for Large-Scale Antenna Arrays," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 501-513, April 2016.

[96] J. P. González-Coma, J. Rodríguez-Fernández, N. González-Prelcic, L. Castedo and R. W. Heath, "Channel Estimation and Hybrid Precoding for Frequency Selective Multiuser mmWave MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 2, pp. 353-367, May 2018.

[97] X. Gao, L. Dai, S. Han, C. I and R. W. Heath, "Energy-Efficient Hybrid Analog and Digital Precoding for MmWave MIMO Systems With Large Antenna Arrays," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 998-1009, April 2016.

[98] A. Maltsev *et al.*, Channel Models for 60 GHz WLAN Systems, IEEE Standard 802.11-09/0334r8, May 2010.

[99] A. Maltsev *et al.*, Channel Models for IEEE 802.11ay, IEEE Standard 802.11-15/1150r9, May 2016.

[100] G. Interdonato, M. Karlsson, E. Björnson and E. G. Larsson, "Local Partial Zero-Forcing Precoding for Cell-Free Massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4758-4774, July 2020.

[101] G. Interdonato, M. Karlsson, E. Bjornson and E. G. Larsson, "Downlink Spectral Efficiency of Cell-Free Massive MIMO with Full-Pilot Zero-Forcing," *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 1003-1007.

[102] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436-453, April 2016.

[103] X. Gao, L. Dai and A. M. Sayeed, "Low RF-Complexity Technologies to Enable Millimeter-Wave MIMO with Large Antenna Array for 5G Wireless Communications," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 211-217, April 2018.

[104] X. Song, T. Kühne and G. Caire, "Fully-/Partially-Connected Hybrid Beamforming Architectures for mmWave MU-MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1754-1769, March 2020.

[105] M. Alonzo, S. Buzzi, A. Zappone and C. D'Elia, "Energy-Efficient Power Control in Cell-Free and User-Centric Massive MIMO at Millimeter Wave," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 651-663, Sept. 2019.

[106] Z. Gao, L. Dai, D. Mi, Z. Wang, M. A. Imran and M. Z. Shakir, "MmWave massive-MIMO-based wireless backhaul for the 5G ultra-dense network," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 13-21, October 2015.

[107] A. Alkhateeb, O. El Ayach, G. Leus and R. W. Heath, "Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831-846, Oct. 2014.

[108] Deng, Li, and Dong Yu. "Deep learning: methods and applications." *Foundations and trends in signal processing* 7.3–4 (2014): 197-387.

[109] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85-117.

[110] Valueva, Maria V. *et al.*, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation." *Mathematics and Computers in Simulation* 177 (2020): 232-243.

A letter from aspiring Dr. Zhao to PhD student Yu

Dear Yu:

Congratulations! On behalf of your future, I am glad to see that you are reaching the end of your PhD journey!

Shortly, you will have your thesis defense as well as the ceremony. At that moment, you come to be Dr. Zhao! It will be a fantastic moment, like the past four years which I would like to bring to your attention...

I am sure you remember the scene when you first met your supervisor Prof. Sonia Heemstra de Groot. Even though you were shy and not good at English at the time, she was endlessly kind and patient with you, reassuring you to "take your time". In the following period, her guidance and support significantly helped you with the quality of your research, and importantly, she gave you sufficient freedom to pursue your ideas. I think you are having a vivid memory of your second supervisor Prof. Ignas Niemegeers, who is a kind, patient and funny man. He seems to have infinite tolerance and patience for improving your manuscript. Usually, he reviewed your manuscripts word for word and gave you comments, like "People do not know what 'hop-on-hop-off' means, if he or she has never been on a city sight-seeing bus!", "What is a small number, is 3 small enough?", etc. These, later you will know, make your manuscripts very readable and explicit. That's why your submitted papers have never been rejected for English writing. You are so lucky having had Sonia and Ignas as your supervisors during your PhD period. Now, let me on behalf of you, express sincere thanks to both.

You must remember your first time arriving in the Netherlands, the trees were tall, the moon crossed the clouds illuminating the leaves. It was the moment that you were in your dream – go abroad, which you had since childhood. You once often thought you would never achieve this dream, but it is real now. Please do appreciate your Chinese supervisor Prof. Fengming Zhang, who gave you a great deal of help with the going abroad process. A sincere thanks should also be given

to your cousin Dan Zhao, you may not get your PhD offer without her help. A special appreciation should go to Prof. Emil Björnson from Linköping University, thanks for his help with the questions you had on massive MIMO.

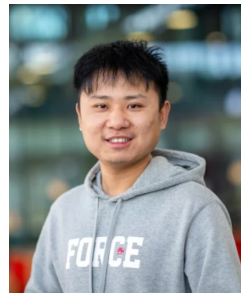
I would like to remind you to be grateful to your committee members: Prof. Klaus Moessner, Prof. Dirk Dahlhaus, Prof. Frans Willems, Dr. Ulf Johansen and Dr. Anthony Lo, for their time to review and approve your thesis.

I think you will never forget your colleagues from the ECO group. A special thank goes to Jose Hakkens, who is the group secretary and gave you help countless times. Here again, Yu, on behalf of you, I would like to thank Prof. Ton Koonen, Dr. Oded Raz, Dr. Georgios Exarchakos, Dr. Zizheng Cao, Dr. Nicola Calabretta, Dr. Chetan Belagal Math, Dr. Fulong Yan, Dr. Chenhui Li, Dr. Xuwei Xue, Dr. Xiaotao Guo, Dr. Bitao Pan, Dr. Xuebing Zhang, Dr. Haotian Zheng, Dr. Teng li, Dr. Mingyang Zhao, Yu Lei, Yu Wang, Shaojuan Zhang, Bin Shi, Jianou Huang, Liuyan Chen. A warm thanks will be given to your friends: Dr. Wang Miao and his wife Dr. Shuli Wang, Zezhou Wang, Dr. Yangjun Gao, Dr. Qiang Liu, Lu Huang, Dr. Guangming Li, Wenjing Tian, Dr. Weiming Yao, Coco (Weitian Kou), Dr. Fu Wang and Pan Gao.

Importantly, I would like to express special thanks to your family (a.k.a. my family) members, for their love and encouragements. Finally, dear Yu, I would like to genuinely appreciate you! Thanks for your braveness and persistence, and thanks for trusting there is always a light in the dark. You definitely know what you want and you are now approaching it. I wouldn't be here without you.

Best regards,
Aspiring Dr. Yu Zhao

Curriculum Vitae



Yu Zhao was born in Yuncheng, China, on February 25, 1993. From December of 2017, Yu started his PhD study in the Electrical-Optical Communication (ECO) group of Eindhoven University of Technology (TU/e), the Netherlands.

During his PhD study, Yu's research topics were cell-free massive MIMO, deep learning and artificial intelligence algorithms. He has published two journal papers and four conference papers. He was awarded with the "Best Oral Presentation" in the 8th IEEE ICICN conference and the "Best Oral Presentation" in the 20th IEEE ICCT conference. He chaired the "Internet of Things" session in the 18th WINSYS conference. He is now a Graduate Student Member of IEEE.