DEVELOPMENT OF DEEP REINFORCEMENT LEARNING BASED RESOURCE ALLOCATION TECHNIQUES IN CLOUD RADIO ACCESS NETWORK

AMJAD IQBAL

DOCTOR OF PHILOSOPHY (ENGINEERING)

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

JUNE 2022

DEVELOPMENT OF DEEP REINFORCEMENT LEARNING BASED RESOURCE ALLOCATION TECHNIQUES IN CLOUD RADIO ACCESS NETWORK

By

AMJAD IQBAL

A thesis submitted to the Department of Electrical and Electronic Engineering,

Lee Kong Chian Faculty of Engineering and Science,

Universiti Tunku Abdul Rahman,

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Engineering)

June 2022

ABSTRACT

DEVELOPMENT OF DEEP REINFORCEMENT LEARNING BASED RESOURCE ALLOCATION TECHNIQUES IN CLOUD RADIO ACCESS NETWORK

Amjad Iqbal

Next-generation networks are envisioned to support dynamic and agile network management to maximize the users' quality of service (QoS). Cloud radio access network (CRAN) emerges as a promising candidate since the limited network resources can be virtualized and shared among distributed remote radio heads (RRHs). Conventional approaches formulate resource allocation as an optimization problem and solve it with instantaneous environment knowledge without considering the consequences of actions. A step towards long-term network performance optimization is the use of deep reinforcement learning (DRL), which can learn the best policy via interaction with the environment. This thesis proposes three DRL-based resource allocation algorithms that optimize the CRAN performance in terms of energy efficiency (EE), spectral efficiency (SE), and total power consumption. The first proposed algorithm aims to optimize the EE by controlling the on/off status of RRH via a deep Q network (DQN) and subsequently solving a power optimization problem. To capture the spatio-temporal channel state information (CSI), the second proposed algorithm adopts machine learning with anchor graph hashing techniques to extract generalized features before feeding them into the DQN. The goal here is to optimize the long-term tradeoff between EE and SE. In the last proposed scheme, additional EE savings are facilitated by designing and

integrating a convolutional neural network (CNN), which can better learn the feature of environment states. Simulation results show that all proposed DRL algorithms outperform 20-25% compared to existing techniques while achieving faster convergence. All performance benchmarking was carried out based on 100 testing episodes after properly training the DRL agent with 1000 episodes.

ACKNOWLEDGEMENT

The grant of this thesis work has been made possible by Allah Almighty, the Most Beneficent, the Most Merciful, who has provided me with strength, patience, health, thoughts, and cooperative people through which I can complete it.

It is impossible for a single mind to produce research of such calibre. First and foremost, I am highly thankful to my Supervisor, **Ir Ts Dr Tham Mau Luen,** for his continuous support of my PhD study and related research. He is also grateful for his patience, motivation, and immense knowledge. His excellent guidance helped me in all the time of analysis. I could not have imagined having a better advisor and mentor for my PhD study.

Secondly, I would like to express my deep and sincere gratitude to my Co-Supervisor, **Ir Dr Chang Yoong Choon,** for allowing me to do research and providing priceless guidance throughout this research. His sincerity, diligence, vision, and motivation have deeply inspired me. I am incredibly grateful for your assistance, directions and suggestions throughout my PhD study.

Thirdly, I am highly thankful to the Institute of Postgraduate Studies and Research (IPSR) Universiti Tunku Abdul Rahman (UTAR) Malaysia under UTARRF (IPSR/RMC/UTARRF/2021-C1/T05) for providing me with the fund throughout my PhD study.

For their endless support and guidance throughout my life, my parents, **Mr Mehboob Karim** and **Mrs Rubina Bibi,** deserve special thanks. Lastly, I would like to thank my uncle, **Mr Muhammad Zahid**, brothers, **Mr Sajid**

Iqbal, **Mr Imtiaz Iqbal**, **Mr Rashid Iqbal**, sisters, **Miss Shabina Bibi**, **Miss Shabnam Bibi**, and friends, **Mr Hashmat Aziz**, **Mr Muhammad Siraj**, and **Miss Salva Asghar** for supporting me throughout the PhD journey.

APPROVAL SHEET

This thesis entitled **"DEVELOPMENT OF DEEP REINFORCEMENT LEARNING BASED RESOURCE ALLOCATION TECHNIQUES IN CLOUD RADIO ACCESS NETWORK"** was prepared by AMJAD IQBAL and submitted as partial fulfilment of the requirements for the degree of Doctor of Philosophy (**Engineering)** at Universiti Tunku Abdul Rahman.

Approved by:

(Assistant Prof. Ir. Ts. Dr. THAM MAU LUEN)

Date: 8 June 2022

Supervisor

Department of Electrical and Electronic Engineering

Lee Kong Chian Faculty of Engineering and Science

Universiti Tunku Abdul Rahman

(Associate Prof. Ir. Dr. CHANG YOONG CHOON)

Date: 10 June 2022

Co-supervisor

Department of Electrical and Electronic Engineering

Lee Kong Chian Faculty of Engineering and Science

Universiti Tunku Abdul Rahman

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name AMJAD IQBAL

Date 08/06/2022

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 08/06/2022

SUBMISSION OF THESIS

It is hereby certified that ***AMJAD IQBAL*** (ID No: ***19UED00837***) has completed this thesis entitled *"DEVELOPMENT OF DEEP REINFORCEMENT LEARNING BASED RESOURCE ALLOCATION TECHNIQUES IN CLOUD RADIO ACCESS NETWORK"* under the supervision of ***Ir Ts Dr THAM MAU LUEN*** (Supervisor) from the Department of Electrical and Electronic Engineering, Lee Kong Chian Faculty of Engineering and Science, and ***Ir Dr CHANG YOONG CHOON*** (Co-Supervisor)* from the Department of Electrical and Electronic Engineering, Lee Kong Chian Faculty of Engineering and Science.

I understand that the university will upload a softcopy of my thesis in pdf format into UTAR Institutional Repository, which may be accessible to the UTAR community and public.

Yours truly,

_____

(*AMJAD IQBAL*)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

1G	first generation
4G	fourth generation
5G	fifth generation
AGH	anchor graph hashing
AI	artificial Intelligence
AL	augmented Lagrangian
AP	access points
BSs	base stations
BBU	baseband unit
CAPEX	capital expenditures
CNN	convolutional neural network
CRAN	cloud radio access network
CSI	channel state information
D2D	device-to-device
D2QN	dueling DQN
DAS	distributed antenna system
DDPG	deep deterministic policy gradient
DL	deep learning
DNN	deep neural network
DQN	deep Q network
DRL	deep reinforcement learning
DSP	digital signal processing
E2E	end-to-end

EB	exabytes
EE	energy efficiency
FA	full coordinate association
GBDT	gradient boosting decision tree
Gbps	Gigabits per second
IoT	internet of things
Mbps	megabits per second
MDP	Markov decision process
MEC	mobile edge computing
MINO	mixed-integer nonlinear optimization
ML	machine learning
MNOs	mobile network operators
MOOP	multiple objective optimization problems
NFV	network function virtualization
NN	neural networks
OFDMA	orthogonal frequency division multiple access
OPEX	operating expenses
QoS	quality of service
RA	resource allocation
RAN	radio access network
RAT	radio access technology
RE	resource efficiency
RL	reinforcement learning
RRHs	remote radio heads
SARSA	state-action-reward-state-action

SE	spectral efficiency
SINR	signal-to-interference-plus-noise ratio
SL	supervised learning
SOCP	second-order cone programming
SOOP	single objective optimization problem
UEs	user equipment’s
UL	unsupervised learning
V2V	vehicle-to-vehicle

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The past few decades have seen rapid progress in mobile communication technologies, that is, from being able to make analog voice calls for a limited number of users in the first generation (1G) to provides high data rates to millions of devices in the fourth generation (4G). The emerging fifth-generation (5G) of wireless communications originates with even more promising features, including a high data rate (10 Gigabits per second (Gbps)), lower latency (less than one millisecond (ms)), and 10-100 times higher number of connected devices for the purpose of Internet of Things (IoT) application (Tullberg *et al.*, 2016). According to the Cisco annual report 2020, mobile subscriptions are expected to grow to 5.7 billion (71 % of the total population) by 2023 from 5.1 billion (66% of the total population) in 2018. As shown in Figure 1.1, the total number of internet users is expected to reach 5.3 billion by 2023 as opposed to 3.9 billion in 2018, which indicates a 6% annual growth (Cisco, 2020).

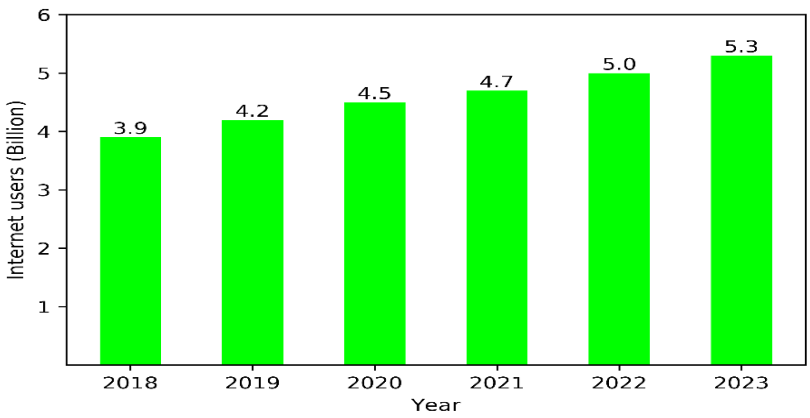


Figure 1.1: Global internet user growth (Cisco, 2020)

Obviously, as the number of users and service requirements grow, the number of access points (APs) and base stations (BSs) will increase, leading to several challenges regarding interference complexity management, cost inflation (capital expenditures (CAPEX) and operating expenses (OPEX)), and deployment strategy (Rony *et al.*, 2017). Furthermore, designing and upgrading the 5G network in a multi-environment is more complex and challenging (Hassani, Haidine and Jebbar, 2020). As a result, Mobile Network Operators (MNOs) are under high pressure to design and adopt a new and cost-effective Radio Access Network (RAN).

A BS is physically connected to a fixed number of antennas in a typical RAN, limiting the potential performance gain due to spatial correlation (Chih-Lin *et al.*, 2018). Cloud-RAN, commonly known as CRAN, is a new approach to addressing the challenges faced by MNOs and reducing their CAPEX and OPEX costs (Checko *et al.*, 2016). In the CRAN philosophy, baseband processing is shifted away from the physical location of BS into a "virtual BS pool." A CRAN is basically comprised of two main components, a baseband unit (BBU) and remote radio heads (RRHs). The BBU handles the signal processing function, whereas the RRHs handle the radio signal transmission to the user equipment (UEs). Fronthaul links perform the interconnection between RRHs and BBU. Furthermore, the traffic processing is accomplished via the backhaul connection between the BBU and the core network. As the CRAN is adopted from the cloud computing concept where resources are shared in a centralized manner and allocated on demand (Checko *et al.*, 2016). The difference between typical RAN and CRAN is shown in Figure 1.2. The

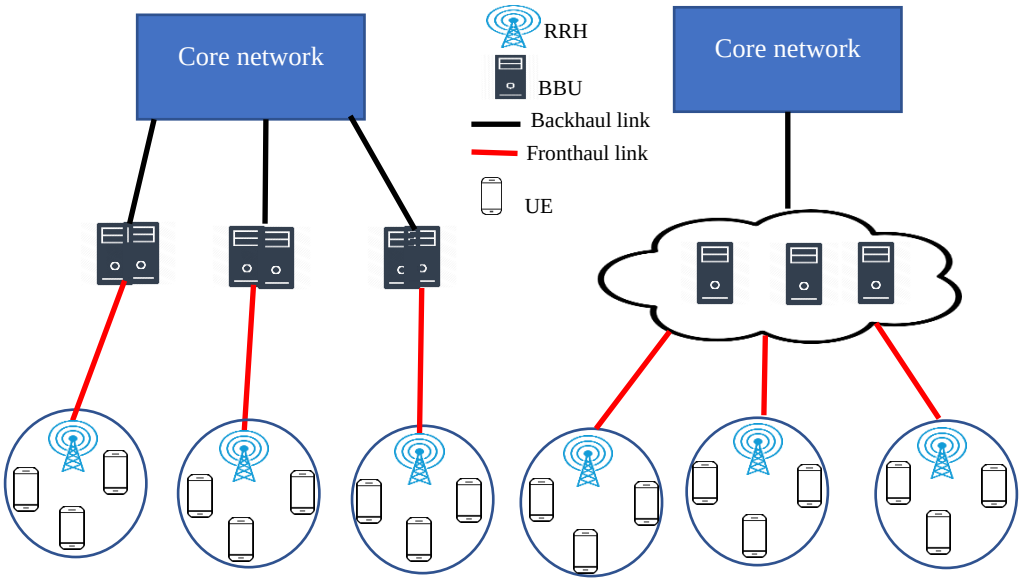


Figure 1.2: Architecture of a Radio Access Network

baseband resources can be employed more efficiently in the CRAN application based on the whole network. In addition, this concept allows the processing power in the BBU pool to be adapted to the network's instantaneous load.

1.2 MOTIVATION

The main motivation of this thesis is to manage the complex resource allocation (RA) optimization for next-generation wireless networks efficiently and intelligently. Generally, it is assumed that information about the environment, such as power consumption and wireless channels, is completely known. In practice, however, the wireless channel gains change in a fading environment, and when the BS configuration is dynamically updated, the wireless channel gain can be uncertain. Accordingly, the BS cannot always know the exact channel gain. Machine learning (ML) is an emerging tool that has the potential to manage the network's resources in such a way that network efficiency,

reliability, and robustness goals are achieved while meeting the quality of service (QoS) demand (Gressling, 2020). The main advantage of ML is to manage the network resources in an agile and flexible way, which provides the network with more autonomy and reduces the amount of computational and time expenses needed to perform manual configuration and maintenance. ML can also provide real-time analysis and dynamic control, which reduces human intervention. In fact, ML-enabled 5G and beyond networks offer several advantages over previous generations of wireless networks due to the opportunities that arise from learning the environment's parameters under varying channel behavior. Despite these advancements, some challenges still need to be addressed, such as RA, computational complexity, adaptability to network dynamics, etc.

A family of ML algorithms called reinforcement learning (RL) (Sutton and Barto, 2018) can learn from experience and solve problems in which finding analytical solutions would be difficult or impractical. RL follows the trial-and-error process to learn the environmental behavior. The training process of the RL algorithm is different from supervisor learning (Michael W. Berry, 2020), as previously acquired data is not required. Also, the RL algorithm is different from unsupervised learning (Michael W. Berry, 2020), as some knowledge from the engineer is required. The engineer's knowledge is conveyed to the algorithm through a reward function, which measures the RL algorithm's action quality. Therefore, the RL algorithm needs to learn the process and its behavior to achieve high rewards during the training process. Furthermore, the channel gains change and power control techniques for the upcoming wireless network generation are expected to address a plethora of dynamic situations. Analytical

models, which generally require full knowledge about their environment, can be inconvenient for displaying complex situations. ML algorithms, such as RL, can excel in such situations and aid in reaching satisfactory solutions.

The fields of Artificial Intelligence (AI) and ML have been greatly influenced by advances in Deep Learning (DL) during the past few years (Ian Goodfellow, Yoshua Bengio, 2016). DL uses Neural Networks (NN) with multiple hidden layers. These hidden layers allow the NN to represent more complex functions. RL methods leveraging DL have been invented recently, giving birth to Deep Reinforcement Learning (DRL) (Sutton and Barto, 2018). Given the present opportunities and the benefits of DRL, this thesis proposes and investigates DRL-based techniques to balance the power consumption, Energy Efficiency (EE), and the joint tradeoff between EE and Spectral Efficiency (SE) while maintaining the user's QoS requirements in a downlink CRAN framework.

This thesis focuses on investigating three DRL algorithms that serve as the core for the proposed RA frameworks. The aim is to compare different RL paradigms and application strategies and see how they work when applied to the RA problem. Furthermore, this thesis evaluates the behavior of RL algorithms and their actions in specific controlled scenarios.

1.3 PROBLEM STATEMENTS

This thesis aims to address the near-optimal RA problem in terms of power minimization, maximizing energy efficiency, and satisfying users' QoS requirements in downlink CRAN by using different DRL algorithms. This is motivated by the existence of several constraints for designing a centralized RA strategy, such as:

- User data rate demand and QoS requirements
- Beamforming weights
- RRH transmission power limitation

Wireless communication network relies heavily on a concept called channel state information (CSI) (C. Luo *et al.*, 2020) (Li *et al.*, 2016). As its name implies, CSI represents the characteristics of a radio channel. Specifically, CSI describes how path loss, scattering, diffraction, fading, shadowing, etc., are combined when a signal propagates from a transmitter to its receiver. Therefore, it is important to obtain accurate CSI to guarantee the performance of radio links in wireless communication systems. In addition, CSI can be used to identify whether a radio link is in good or bad condition.

A majority of wireless network problems define the states of users with hand-crafted characteristics and do not take into consideration the relationship between the RRHs and the users (H. Li *et al.*, 2018) and (Xiao *et al.*, 2020). The features may be extracted artificially, and the learning agent may be forced to make sub-optimal decisions. The main drawback of such works is that the users report their information to the respective RRHs, increasing the burden on signaling overhead as feedback. If such information is present between the users and RRHs, then RRHs are responsible for recording all the valid information. The users do not need to provide any such information for signaling. Such a process reduces the signaling burden in the network.

Secondly, most of the works focus only on maximizing SE or EE. They do not consider the joint summation between these two metrics (Vu *et al.*, 2017) and (Tan *et al.*, 2018) because these two metrics are usually used to contradict each

other. Furthermore, these two metrics cannot be added directly due to different units. To optimize the long-term tradeoff between SE and EE is targeted in this approach. This is achieved by adopting the weighted sum of the system EE and SE, where a tunable parameter is used to adjust the priority among EE and SE.

Besides that, most existing works use a Deep Neural Network (DNN) to train the neural network, which significantly increases the training parameters (H. Li *et al.*, 2018) and (Luong *et al.*, 2021). This motivates using a Convolutional Neural Network (CNN) to extract the input features. The extracted feature of CNN is then fed to the input of the DRL agent. This means that the CNN phase is responsible for extracting the input information. In contrast, the DRL phase finds the optimal policy to turn on/off RRHs based on the user demand. This will speed up the algorithm learning process and achieve better network performance.

In this thesis, the relationship between users and RRHs is explicitly considered at the input of the network state, and generalized features are extracted by adopting ML techniques. Therefore, this thesis considers the DRL approach to optimize the RA problem over a long operational period. Furthermore, the advanced version of the DRL algorithms is used to find the optimal control policy to dynamically turn on/off the RRHs based on the user demands to save more power. The two main advantages of using DRL are:

1) The agents can be trained through each learning stage to determine the on/off status of RRHs at each time slot t. This ensures the RRH switching decision to not rely on physical RA customization for the delay, rate, and jitter optimization.

2) The agent can survey the entire network by considering every possible state to ensure that the users' QoS requirements are met.

1.4 THESIS OBJECTIVES

The utmost objective of this thesis is to maximize the long-term RA performance by adjusting the per RRH transmit power and user data rate. RA performance has been extensively investigated in terms of power control and EE since the invention of radio communication. In summary, the RA performance can be achieved using mathematical and heuristic approaches for short-term goals without considering any future consequences. Furthermore, the latest introduction of ML surpasses human-level performance, especially using DRL approaches. Therefore, the main objectives of this thesis can be summarized as:

1) To optimize the EE subject to the constraints on per-RRH transmission power and user data rates.

2) To optimize the long-term tradeoff between EE and SE while considering the spatio-temporal CSI.

3) To further optimize EE by learning the feature of environment states via a CNN.

1.5 CONTRIBUTIONS

The key contributions of this thesis are summarized as follows:

1) **An energy-efficient resource allocation scheme based on a Double deep Q-network in CRANs.**

A Double deep Q-network based algorithm is proposed first to minimize total power consumption and satisfy the user QoS demand. The proposed solution is then compared with the DQN algorithm (Xu *et al.*, 2017) and the traditional approach (Dai and Yu, 2016). Furthermore, the proposed solution is extended for EE by adding CSI at the input of the network state and solving with the function approximation method. In the end, three different scenarios are considered to verify the infeasibility issue that may arise due to insufficient active RRHs.

2) **A deep reinforcement learning-based resource allocation for joint energy efficiency and spectral efficiency in CRANs.**

   An approach based on DRL has been proposed to achieve a long-term tradeoff between EE and SE. The multiple objective optimization problems (MOOP) that EE and SE have jointly optimized are transformed into a single objective optimization problem (SOOP) by dynamically weighing EE and SE. The same metric unit for EE and SE in a weighted summation is ensured first. Therefore, a tunable parameter is used to adjust the EE and SE priority. Furthermore, the CSI is explicitly considered at the input of the network state. However, the CSI is updated continuously at each time step t, making the network exploration difficult in practice. Therefore, the anchor graph hashing (AGH) method is applied to limit the CSI and then map AGH to a hash code where the hash code can easily match to the DRL input.

3) **Resource management in CRAN using convolutional neural networks-based deep Q-networks (CNN-DQN).**

A CNN-based deep Q-network (CNN-DQN) is proposed to balance energy consumption and guarantee the user quality of service (QoS) demand in a downlink CRAN. The CSI characteristic has been assumed at the input of the network state in the previous two approaches, where function approximation and AGH approach are used to solve its dynamic nature. However, CSI is updated continuously at each time step t and takes the continuous values. This creates the convergence problem; therefore, the CSI is discretized by using the CNN framework in this approach. Furthermore, the RA performance is optimized specifically by the CNN-based DQN method, where a CNN is responsible for carrying out the CSI feature extraction process. In contrast, the DQN phase is responsible for turning on/off the RRHs. Finally, the proposed solution is compared with DQN and a traditional approach.

1.6 THESIS ORGANIZATION

The rest of the thesis is structured into six chapters as follows:

• Chapter 2 provides a critical overview of the RA schemes found in the literature. The majority of the proposed solutions in the literature can be categorized into either: i) addressing the C-RAN power minimization problem with a limit on the total transmission power; ii) focusing on the EE with users' QoS requirements subject to per RRHs transmit power and users' target rate constraint, or iii) proposing solutions to jointly optimize the EE-SE. This kind of strategies can be based on the following criteria: i) static or ii) dynamic approaches. It is worth noting that most related solutions are based on model-

based approaches, which require accurate information in advance before solving the RA problem. In other words, these approaches usually find restrictions for large-scale networks. They are only applicable for low user data rates and QoS requirements, which are impractical for the future 5G and beyond networks. In order to address such limitations, different DRL algorithms are studied in this thesis. A concise overview of the RL, including the basic probabilistic formalism of the Markov decision process (MDP), is explained. Furthermore, the value-based and policy-based methods are also discussed to solve the required RA problem.

• Chapter 3 presents a Double DQN-based RA framework that optimizes the long-term RA performance in terms of power minimization and EE maximization while taking into account the transmission power selection at each RRH and user rates. The CSI is added at the input of the network state and then uses the function approximation approach to solve the optimization problem. The starting point of this approach is the traditional approach, where the reward function is achieved from the immediate action while ignoring its effect to the future. In order to consider the future action consequence, a DQN approach is proposed using the past learning experiences and considering the future effects based on the current action decision. However, the action overestimation problem gives a lower probability limit to estimating the maximum Q-value. Therefore, in this approach, a Double DQN is presented that separates the selected action from the target Q-value generation leading to a higher value of energy savings at the CRAN.

• Chapter 4 presents a Dueling DQN-based RA scheme intended to maximize the long-term tradeoff between EE-SE and satisfy the users' QoS requirements.

The MOOP is first solved for EE and SE and then converted into a SOOP. However, EE and SE have different units and are thus inappropriate to directly add EE and SE. Therefore, the same metric unit is ensured in the weighted summation of EE and SE. A tunable parameter is used to adjust the priority of EE and SE. The AGH method is set up to limit the CSI generalized features before feeding them into the input of DRL. The Dueling DQN method is then configured to learn the near-optimal control strategy to turn on/off the RRHs to maximize the joint EE-SE performance and satisfy the users' QoS requirements. The improved EE-SE performance is examined with the Dueling DQN based-AGH method. Finally, the proposed Dueling DQN based-AGH method is evaluated by comparing it with the Dueling DQN without CSI generalization, Q-learning, and myopic approach.

• Chapter 5 presents the CNN-based DQN (CNN-DQN) approach in the downlink CRAN to simultaneously balance the EE performance and satisfy the users' QoS demand. In this method, the CNN approach is combined with DQN, where the CNN phase is responsible for extracting the input state information containing the CSI feature. The extracted feature of CNN is then fed to DQN, which is responsible for finding the optimal policy for turning on/off the RRHs based on the user demand.

• Chapter 6 concludes this thesis and presents some of the possible directions for future research work.

CHAPTER 2

LITERATURE REVIEW

Massive connectivity between heterogeneous users, including vehicles, humans, and machines, are expected in the next generation of wireless communication systems, resulting in diverse QoS requirements. Furthermore, the network dynamics, traffic variation, and user mobility mandate efficient utilization of network resources. A network resource is a process of allocating resources, like power and spectrum, to the wireless network, commonly known as resource allocation (RA), in order to provide high-quality QoS to wireless communication networks. This thesis considers RA functionalities concerned with power allocation, user QoS satisfaction, and EE applied to the 5G and beyond wireless networks. This chapter presents the state of the artwork related to solving wireless networks' RA problems. Specifically, chapter 2 focuses on two methods, i.e., the traditional and machine learning-based methods. The starting point is to provide an overview of the traditional methods that have been widely used to solve the RA problem in a typical wireless network. Afterwards, the machine learning-based methods, including RL and DRL, are presented. The various algorithms adopted by the wireless network to solve the RA problem are summarized in Table 2.1.

Table 2.1: Research work summary for RA in wireless network

Method	Ref.	Constraints	Objective	Approach used

Traditional Methods	(Chai *et al.*, 2021)	Average transmission rate, subcarriers, total transmit power	Maximize EE performance	Mathematical Approach
	(Luo, Chen and Tang, 2018)	Maximum transmit power, fronthaul capacity limit, and total transmission rate	Jointly optimize the system power consumption and delay performance while guaranteeing user QoS and fronthaul capacity.	Mathematical Approach
	(Wang, Zhou and Mao, 2016)	Maximum Transmit power, QoS constraint, queue stability, fronthaul capacity limit	Maximize the EE optimization problem.	Mathematical Approach
	(Peng *et al.*, 2016)	Minimum data rate, maximum transmit power	Improve the EE performance in a downlink heterogeneous	Mathematical Approach

			CRAN (H-CRAN).	
	(Huang *et al.*, 2020)	Transmission rate, maximum transmission power	Maximize the EE RA problem in fog computing under the transmission power constraints.	Mathematical Approach
	(Chughtai *et al.*, 2018)	Transmit power, energy causality, QoS	Maximize EE performance.	Programming Approach
	(AlQerm and Shihada, 2018)	User association, transmit power, QoS requirements,	To maximize EE and mitigate interference while maintaining users' QoS requirements.	Programming Approach
	(Tham *et al.*, 2017)	Transmit power, data rates	Maximize EE in a downlink multiuser distributed antenna system (DAS).	Programming Approach
	(Tang *et al.*, 2014)	QoS requirements,	Joint tradeoff performance of	Programming

		maximum transmit power	EE-SE as a resource efficiency (RE).	Approach
	(Farhadi Zavleh and Bakhshi, 2021)	User's association, fronthaul capacity, QoS requirements, transmit power	To maximize the total sum rate.	Programming Approach
	(Ari *et al.*, 2019)	Transmit power, fronthaul capacity,	To reduce overall network cost while maintaining user QoS and QoE.	Heuristic Approach
	(Aqeeli, Moubayed and Shami, 2018)	QoS requirements, Transmit power,	To minimize the power consumption.	Heuristic Approach
	(Lin and Liu, 2019)	Maximum power, fronthaul capacity	To maximize system throughput.	Heuristic Approach
	(Zeng *et al.*, 2018)	Maximum power, data rate, bandwidth	To minimize the network power consumption.	Heuristic Approach

	(Dinh *et al.*, 2021)	Maximum transmit power, fronthaul capacity	To maximize the EE performance.	Heuristic Approach
Machine Learning-based Methods	(Sun, Boateng, Huang, *et al.*, 2019)	Maximum transmit power, interference threshold	To balance energy consumption and satisfy user QoS demand.	Q-learning
	(Sun, Boateng, Ayepah-Mensah, *et al.*, 2019)	Maximum transmit power, throughput	To maximize EE and maintain QoS requirements.	Q-learning
	(Khan *et al.*, 2020)	Maximum transmit power, minimum level of SE	To improve joint energy and spectral efficiency.	Q-learning
	(Peesapati *et al.*, 2021)	Average UE rate, a sum of encoding and decoding power consumption of the BS	To reduce the energy consumption of a BS under variable input traffic demand	Q-learning
	(Xu *et al.*, 2017)	Transmit power, user demand	To achieve a significant amount of	DQN

			power savings while meeting user demands simultaneously	
	(Y. Luo *et al.*, 2020)	User demand, transmit power	To save the dynamic power consumption.	DQN
	(Hsieh, Chan and Chien, 2021)	Transmit power, backhaul capacity, user rates	To enhance EE while satisfying the user QoS.	DQN
	(Tasnim Rodoshi, Kim and Choi, 2020)	Maximum BBU capacity, user demands	To minimize resource waste and unsatisfied user demands by allocating resources optimally.	DQN
	(Zhang *et al.*, 2020)	Transmit power, data rate	To improve the system performance in terms of energy-saving and QoS guarantee.	Double DQN
	(Zhao *et al.*, 2020)	Transmit power, delay	To maximize the total system	Double DQN

			capacity while guaranteeing strict transmission delay and reliability.	
	(Li, Xu and Li, 2021)	Transmit power, data rate, delay, maximum computational resources	To minimize the energy consumption and latency	Double DQN
	(Yuan *et al.*, 2021)	Maximum total power, user's interference temperature	Jointly optimizes cognitive users' spectrum efficiency and quality of experience through the cognitive user's channel selection and power control.	Double DQN
	(Sun, Ayepah-Mensah,	Transmit power, user data rate	To minimize power consumption and guarantee QoS	Dueling DQN

	Xu, *et al.*, 2020)		satisfaction by using CNN-based relational dueling DQN.	
	(Liu *et al.*, 2019)	Throughput, RB association, finite amount of UE.	Joint optimization of EE and SE of the network.	Dueling DQN
	(Sun, Ayepah-Mensah, Budkevich, *et al.*, 2020)	The user data rate, maximum power	To minimize total energy consumption.	Dueling DQN
	(Gholipoor *et al.*, 2021)	Transmit power, data rate, server time, CPU cycle, storage size, delay,	To maximize EE and guarantee E2E QoS.	Actor-critic
	(Wei *et al.*, 2018)	Maximum power, data rate	Maximizing EE of the overall network.	Actor-critic
	(Li *et al.*, 2021)	Beamforming vector, transmit power	Maximize the long-term EE.	DDPG
	(Zhang, Zhu and	QoS requirement,	To maximize the EE in the	DDPG

	Wang, 2021)	association relationships between UEs and BSs	long term for D2D.	
	(Meng *et al.*, 2020)	Maximum power, QoS requirements	To maximize the sum-rate	DDPG

2.1 TRADITIONAL OPTIMIZATION METHODS

One of the significant challenges of the future wireless networks (5G and beyond) is successfully managing power consumption, maximizing EE, and satisfying the user's QoS requirements due to the increasing popularity of smartphone applications. Thus, many scholars have expressed their interest in proposing a lasting solution to the aforementioned problems. The traditional methods (Chughtai *et al.*, 2018), (Labana and Hamouda, 2020) and (Zhang *et al.*, 2019) effectively optimize the RA problem from the short-term perspective. The traditional methods are primarily used to achieve the objective function at each time slot. For example, excessive switching between RRHs in adjacent time slots may increase network deployment costs. The traditional methods generally include the mathematical, programming, and heuristic approaches to solve the required objective function. Some of the related work based on the traditional methods are summarized as:

An energy-efficient based RA algorithm for multi-radio access technology (RAT) networks is presented in (Chai *et al.*, 2021), allowing the UEs to transmit data over multiple radio interfaces in order to leverage the complementary

advantages of the different RATs. The RA is modelled as a stochastic EE maximization problem. Furthermore, the virtual queue is established for each UE to offer more flexibility for RA over time-varying channel fading. The Lyapunov optimization approach is adopted to convert the non-concave EE maximization problem into a Mixed-Integer Nonlinear Optimization (MINO) problem. The MINO problem is then solved by using Lagrange dual methods to develop an energy efficient-based dynamic joint subcarrier and power allocation algorithm that does not rely on prior knowledge of CSI. Finally, the simulation performance is derived to maximize the EE and satisfy the time average QoS constraints.

In (Luo, Chen and Tang, 2018), the authors provide a modified BS power consumption model based on the well-known EARTH model to make the power consumption model more compatible with CRAN. The BS sleeping strategy is proposed based on the Lyapunov method to reduce power consumption significantly. The problem formulation is based on two factors, i.e., the number of handovers and the delay. However, these factors affect QoS and power consumption. Therefore, these factors need to be considered and resolved together for CRAN.

In (Wang, Zhou and Mao, 2016), an energy-efficient joint resource scheduling scheme based on BBU computation and RRH resources is proposed in CRAN. A weighted minimum mean square error (WMMSE) approach is used to obtain the energy-efficient beamforming vectors under per-UE QoS requirements and fronthaul capacity constraints. The derived theoretical results show that simulation results prove the tradeoff between EE and delay.

An energy-efficient based RA in queue-aware multimedia heterogeneous CRAN (H-CRAN) is studied in (Peng *et al.*, 2016), where CRAN maintains a queue for each RRH and user. However, the RA problem has not examined the delay requirement, which is an important QoS parameter for delay-sensitive applications.

In (Huang *et al.*, 2020), the RA problem in fog computing networks is examined based on the fog nodes mechanism to balance network load under transmission rate performance constraints. A fog node reporting a nonzero computation capability becomes the candidate of the fog node. Furthermore, Lyapunov optimization for each time slot is used to maximize the network EE performance.

The problem explained in (Chai *et al.*, 2021), (Luo, Chen and Tang, 2018), (Wang, Zhou and Mao, 2016), (Peng *et al.*, 2016), and (Huang *et al.*, 2020) belongs to the classical mathematical method, where a Lyapunov optimization algorithm is used to solve the RA problem. The primary advantage of this method is that it can obtain a closed-form expression for its objective function in each time slot. However, this method explicitly relies on exact objective functional expressions that are difficult to abstract from many real-world optimization scenarios. Moreover, such methods cannot be guaranteed in a highly dimensional scenario.

The second approach to the traditional method is the programming method, which has been widely used to solve various RA problems in wireless networks.

In (Chughtai *et al.*, 2018), the EE problem is explored for H-CRAN. The optimization problem for EE is solved by mixed-integer nonlinear

programming. Based on the simulation results, higher EE is achieved with low complexity and lower grid power consumption.

The work in (AlQerm and Shihada, 2018) proposes the online RA model for H-CRAN to maximize the EE while maintaining the user QoS requirements. The proposed method reduces the convergence time and overcomes the curse of dimensionality since resources are allocated based on the number of UEs with high QoS constraints. Thus, it is unfair to those with low QoS constraints. Therefore, finding a mechanism to solve this tradeoff problem is essential.

In (Tham *et al.*, 2017), the energy-efficient power allocation scheme is proposed for a downlink distributed antenna system with the objective of maximizing the EE on per antenna transmit power and data rate constraints. The authors convert the nonlinear fractional EE problem to a single variable nonlinear equation by Charnes cooper transformation, which is then solved via Karush-Kuhn-Tucker to achieve optimal power. The authors further proposed full power mode operation to deliver higher SE at the rate of losing EE.

In (Tang *et al.*, 2014), the tradeoff between EE-SE is proposed for Orthogonal Frequency Division Multiple Access (OFDMA) cellular networks via different transmission bandwidth requirements. The proposed algorithm simultaneously optimizes the EE and SE performance and balances the power consumption and occupied bandwidth. Furthermore, the authors propose a suboptimal algorithm based on a uniform power allocation scheme to reduce the complexity.

In (Farhadi Zavleh and Bakhshi, 2021), the joint user association and power allocation for a sparse code multiple access are investigated in CRAN. The objective is to accomplish the maximum sum rate under the constraints of total

RRHs available power, user association, fronthaul capacity, user power, and QoS requirements for each user. The RA problem is solved by considering the successive convex approximation method.

The literature discussed in (Chughtai *et al.*, 2018), (AlQerm and Shihada, 2018), (Tham *et al.*, 2017), (Tang *et al.*, 2014), and (Farhadi Zavleh and Bakhshi, 2021) solves the RA problem based on the programming method, which can be helpful to solve a sequence of the optimization problem. However, this approach relies on the iteration function, where the objective function is recalculated at the beginning of each iteration. Thus, this method requires a high calculation cost to realize real-time decision-making problems. Furthermore, this method also relies on the accurate predictions of wireless networks, which are difficult to achieve in real scenarios.

The third category of the traditional approach is generally called as a heuristic method, which is mainly used to solve the non-convex optimization problem and achieve a local optimum solution with a certain probability.

The work in (Ari *et al.*, 2019) presents an efficient RA scheme for 5G in CRAN called Bee-Ant-CRAN. This work aims to minimize the overall network cost and maintain the user QoS and QoE requirements. The RA optimization problem is then decomposed into two stages, i.e., UE-RRH association and BBU-RRH mapping. The UE-RRH association is performed using a swarm intelligent-based approach, while an ameliorated ant colony optimization algorithm is used to accomplish the BBU-RRH mapping.

An optimal computational RA between RRHs and UE is studied in (Aqeeli, Moubayed and Shami, 2018). The formulated problem relies on the physical

RA to determine the necessary computational resources for the users. The RA problem is formulated by utilizing the decomposition model. Furthermore, the decomposition model is solved by using the heuristic solution to achieve optimal performance with less power consumption for CRAN.

In (Lin and Liu, 2019), maximizing the network throughput via jointly optimizing scheduling, power allocation, subcarrier assignment, and user association in a user-centric OFDM-based CRAN is considered. In this work, a Lagrange duality approach is proposed to solve the RA problem and a heuristic method that reduces its network complexity. However, it still requires high computational complexity to reach high-quality QoS solutions.

A green CRAN architecture using distributed renewable resources and a traditional power grid is proposed in (Zeng *et al.*, 2018). To reduce non-renewable energy consumption, the author presents a heuristic method to optimize the number of active RRHs at any given time with a given set of QoS constraints.

Similarly, the EE performance in fog RAN (FRAN) is formulated in (Dinh *et al.*, 2021). Augmented Lagrangian (AL) and heuristic methods are formulated to solve the RA problem in FRAN. AL explicitly determines the local edge processing of FRAN, whereas the computational complexity is alleviated by the heuristic method.

The work explained in (Ari *et al.*, 2019), (Aqeeli, Moubayed and Shami, 2018), (Lin and Liu, 2019), (Zeng *et al.*, 2018), and (Dinh *et al.*, 2021) solves the RA problem based on the heuristic method. The heuristic method is beneficial for solving big data problems and complex situations. However, such approaches

are not fastened and cannot be rigorously proven by mathematics.

2.2 CRITICAL ANALYSIS ON TRADITIONAL METHODS

Wireless networks of the future (5G and beyond) must be able to accommodate the rapid growth of mobile data traffic and a growing number of mobile users to utilize various applications and services efficiently. Over time, the networks become more dense, heterogeneous, decentralized, and ad hoc and various network entities are incorporated into them. Due to this, different objectives must be met in terms of service, including high throughput and low latency, and the appropriate allocation of resources must be determined. However, considering the uncertainty, increasing complexity, and data dimension of the future wireless networks (5G and beyond), the traditional methods discussed in the above section require accurate, complete and perfect knowledge of the systems in advance. Such information is always inefficient or even inapplicable when solving the decision-making and control problems.

Furthermore, it is vital to determine the optimized decisions for the future wireless network entities with different objectives, such as minimizing energy consumption, maximizing data rates, and reducing network latency. Moreover, it is challenging for the traditional approaches to achieving optimal resource management and service management in mobile networks, such as time-varying wireless channels, which have a wide range of service requirements. Machine learning-based methods have proven to be a highly useful tool for real-time, dynamic decision-making problems in such time-varying and unpredictable network environments.

2.3 MACHINE LEARNING-BASED OPTIMIZATION METHODS

Data have grown enormously across many fields in the past two decades, resulting in the big data challenge, which has led to a need for intelligent data analysis schemes. Several ML methods have been developed, such as DL, to deal with the big data problem. ML method is generally known as the learn-based method, where historical data is provided as an input to predict the future value. Recently, ML methods have been used for wireless networks.

Supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL) are three of the main categories of ML. These three categories differ in how the algorithm is trained (Kubat, 2017). In SL, a set of labelled data is provided at the input with their corresponding output. SL algorithms are well-suited to the application with already known data, such as feature extraction and classification tasks. A variety of wireless communication problems have been tackled with features extraction and classification, such as weighted throughput maximization (Eisen *et al.*, 2019), EE (Zappone, Technologies and Labs, 2018), and device-to-device (D2D) throughput maximization (Kim *et al.*, 2020). SL methods require large amounts of labelled data, which has limited its applicability for power allocation, as the optimal power values are usually not known in advance. The second category of ML is UL, where the goal of UL is to find the inherent structure from the unlabelled data, and this method is convenient for solving clustering tasks. Similarly, UL has also been applied to solve several wireless communication problems, such as power control (Nikbakht, Jonsson and Lozano, 2021), maximizing the sum rate (Hou *et al.*, 2021) and EE (Chang *et al.*, 2018). However, the agent in the UL algorithm requires a large amount of unclassified data to produce intended target values. Therefore, such methods usually require a huge amount of data to find the

similarities and differences between data points. The third category of ML is generally called RL. In RL, the agent's goal is to predict the optimal action that it should take to achieve the highest reward by using feedback from the environment. As such, the agent uses feedback from the environment to improve its performance in a specific task. RL is unsupervised, but its learning process differs from other UL techniques. RL tries to determine the best actions based on the optimal policy instead of learning the data structure.

This thesis mainly focuses on the RL-based approach to solving the RA problem in wireless networks. Recently RL has shown tremendous improvements in optimizing the RA problems in the wireless network from a long-term perspective. Therefore, the overview of RL is presented first for better readability before discussing different algorithms to solve the RA problems.

2.4 OVERVIEW OF REINFORCEMENT LEARNING

RL refers to the process of learning that occurs when a decision-maker (i.e., an agent) interacts with their environment. Specifically, an RL agent interacts with its environment, executes an action decision, and receives feedback (reward). An agent observes its environment as a state. The state of a domain should efficiently retain relevant information about the environment, including immediate and past observations. A state signal satisfying this condition is said to possess Markov property. Thus, the RL problem is formulated as the Markov Decision Process (MDP). The MDP generally has four-elements tuples, i.e., (state, action, reward, and transition probabilities). Typically, at each iteration $\mathfrak{c}$, the agent obtains some observation of the environmental state $s_{\mathfrak{c}} \in \mathfrak{S}$, where $\mathfrak{S}$ indicates the set of possible states and then executes an action $a_{\mathfrak{c}} \in \mathfrak{A}(s_{\mathfrak{c}})$;

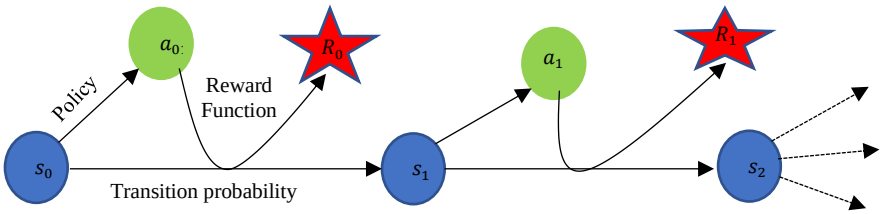


Figure 2.1: MDP illustration

$\mathfrak{A}(s_c)$ specifies the set of possible actions for the state s_c. The agent receives a reward value R_c in response to executing the action. Finally, the agent moves to the next state with a certain probability known as transition probability $p_r(s_{c+1}|s,a) = P\{(s_{c+1}|s_c = s, a_c = a)\}$. The basic illustration of MDP is depicted in Figure 2.1, and the ultimate objective of the RL agent is to find the optimal policy π^* that maximizes the total expected reward, as follows:

$$\pi^* = \underset{\pi(s)}{argmax}\, \mathbb{E}[R_c + \mu R_{c+1} + \mu^2 R_{c+2} + \cdots | s_c = s, a_c = a] \tag{2.1}$$

where $\pi(s)$ represents the policy of a state for optimal action. $\mu \in (0, 1)$ denotes the discount factor and shows the importance of immediate and future rewards. The lower value of μ indicates the immediate reward, while a value close to 1 specifies the future reward. Two main methods are used to solve the RL problem, i.e., the value-based and policy-based methods (Kai Arulkumaran and Miles Brundage, 2017).

2.4.1 Value-Based Method

In the value-based method, the RL agent learns the value of state and action to choose the best action in a particular state. The value-based method is further split into two different functions, i.e., the state-value function $V_\pi(s)$ and the state-action value function $Q_\pi(s,a)$. The state-value function can achieve the expected return when starting from a state s^{th} and acting on our policy as:

$$V_\pi(s_c) = \mathbb{E}_\pi[R_c + \mu R_{c+1} + \mu^2 R_{c+2} + \cdots | s_c = s] \quad (2.2)$$

The optimal policy for the corresponding state-value function can be defined as:

$$V^*(s_c) = \max_\pi V_\pi(s_c)\,, s \in \mathfrak{S} \quad (2.3)$$

Similarly, the state-action value function is the expected return starting with the state s^{th}, then taking action a^{th} followed by a certain policy.

$$Q_\pi(s, a) = \mathbb{E}_\pi[R_c + \mu R_{c+1} + \mu^2 R_{c+2} + \cdots | s_c = s, a_c = a] \quad (2.4)$$

whereas $Q_\pi(s, a)$ represents the state-action-value (also known as the quality-value or Q-value) of policy π. The optimal Q-values can be defined as:

$$Q^*(s, a) = \max_\pi Q_\pi(s, a) \quad (2.5)$$

Temporal difference methods (Sutton and Barto, 2012), such as Q-learning and State-Action-Reward-State-Action (SARSA), are used to estimate the optimal state-action value function. In Q-learning, the agent policy can be approximated using the following update rule:

$$Q_c(s_c, a_c) \leftarrow Q_c(s_c, a_c) + \gamma \left[R_{c+1} + \mu \max_{a_{c+1}} Q_{c+1}(s_{c+1}, a_{c+1}) - Q_c(s_c, a_c) \right] \quad (2.6)$$

where γ and $\max_{a_{c+1}} Q_{c+1}(s_{c+1}, a_{c+1})$ indicates the learning rate and approximate Q-value of the successor state under the best action, respectively. In the same fashion, the SARSA updated the agent policy as:

$$Q_c(s_c, a_c) \leftarrow Q_c(s_c, a_c) + \gamma[R_{c+1} + \mu Q_{c+1}(s_{c+1}, a_{c+1}) - Q_c(s_c, a_c)] \quad (2.7)$$

A fascinating fact about RL agents is that the design of their state and action functions can significantly influence their outcomes. Furthermore, all the state and action values are stored in the form of a lookup table. For each iteration, the lookup table needs to be updated. This process works well when dealing with low-dimensional Q-value function problems. However, when dealing with large

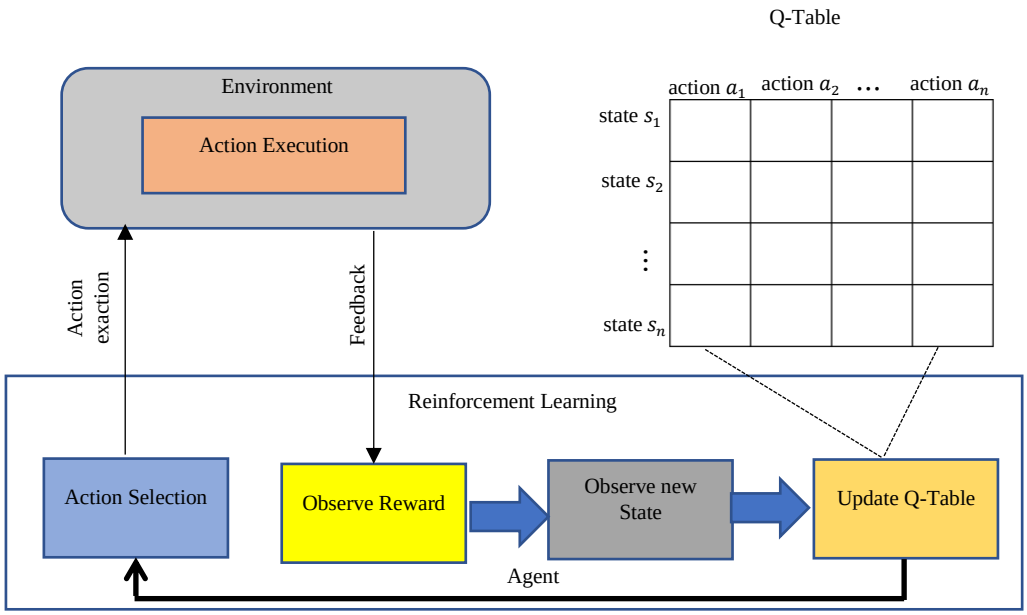


Figure 2.2: Q-Learning intangible operation diagram

Q-value function problems, this method becomes unstable because the RL agent cannot abstract all the valid information from the lookup table in a reasonable time. Figure 2.2 presents an intangible overview of the lookup table methods of RL.

2.4.1.1 RL Algorithm

A Q-learning algorithm is a promising approach for solving many RA optimization problems in wireless communication systems. One widely used model-free RL algorithm is Q-learning for computing optimal policies that maximize long-term rewards. A reward function is introduced to map a state-action pair to the expected cumulative reward (Q-value) in order to estimate and determine the optimal actions in response to different system states. Some of the work based on Q-learning is explained as follows:

In (Sun, Boateng, Ayepah-Mensah, *et al.*, 2019), an autonomous cell activation framework is proposed to balance wireless networks' energy consumption and QoS satisfaction. Furthermore, an Anchor Graph Hashing (AGH) method is

introduced to discretize the state space value. A similar problem is studied in (Sun, Boateng, Huang, *et al.*, 2019), where the fractional power and bandwidth adoption method are formulated to solve the RA problem for energy consumption and QoS requirements. In both cases, the reward function is related to energy consumption with a minimum number of active RRHs. However, the relationship between RRHs and UEs at the network state is not explicitly described.

The work in (Khan *et al.*, 2020) presents joint Energy-Spectral Efficiency (ESE) approach in a multi-hop D2D communication. Improved system performance for ESE is obtained by using Q-learning. However, since EE and SE do not have the same units, it is unclear how the combined utility function should be processed.

2.4.1.2 Deep Learning

Recent advances in wireless networks have attempted to replicate the human brain's neural structure by using Neural Networks (NNs). In particular, NNs consist of three main layers: input layers, hidden layers, and output layers, as shown in Figure 2.3. Each layer contains two different artificial processing:

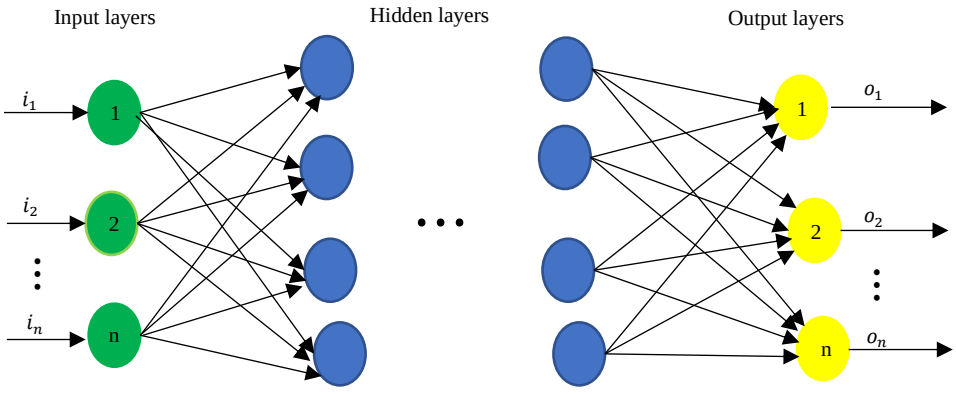


Figure 2.3: Neural network structure

weighted neurons and activation functions. The weighted neurons perform a mathematical function on information received from the input, whereas the activation function introduces non-linearity to the NNs. Softmax, ReLU, Tanh, and Sigmoid are some of the activation functions (Sutton and Barto, 2018). A set of neurons are located in the input layers and perform preprocessing on the input feature vector i. Moreover, a weighted connection of neurons is connected from a specific layer to the preceding layer. Finally, the NN is composed of the output layer o of neurons that create and interpret the outcomes. The weights are adjusted according to inputs and the dataset's expected outputs (inputs and labels).

2.4.1.3 From RL to DRL

The agent must maintain a set of state-action pairs for each Q-value function to maintain and update Q-value functions. However, the future generation of wireless networks will probably be large, decentralized, and heterogeneous, and thus the number of possible system state values will increase exponentially. Moreover, there can be hidden system states or unlimited possible system states due to the vast diversity and uncertainty in system components and environment parameters. As a result, calculating and maintaining all Q-value functions becomes practically impossible, and this is called the curse of dimensionality.

The DRL model addresses this issue by combining RL and deep learning (DL) techniques. The DRL model uses a deep neural network (DNN) to approximate the Q-values functions. DRL parameters (state and reward) can be assigned based on different 5G system objectives, such as power consumption, state of RRHs, user demand, channel gains or throughput maximization, etc. The DQN

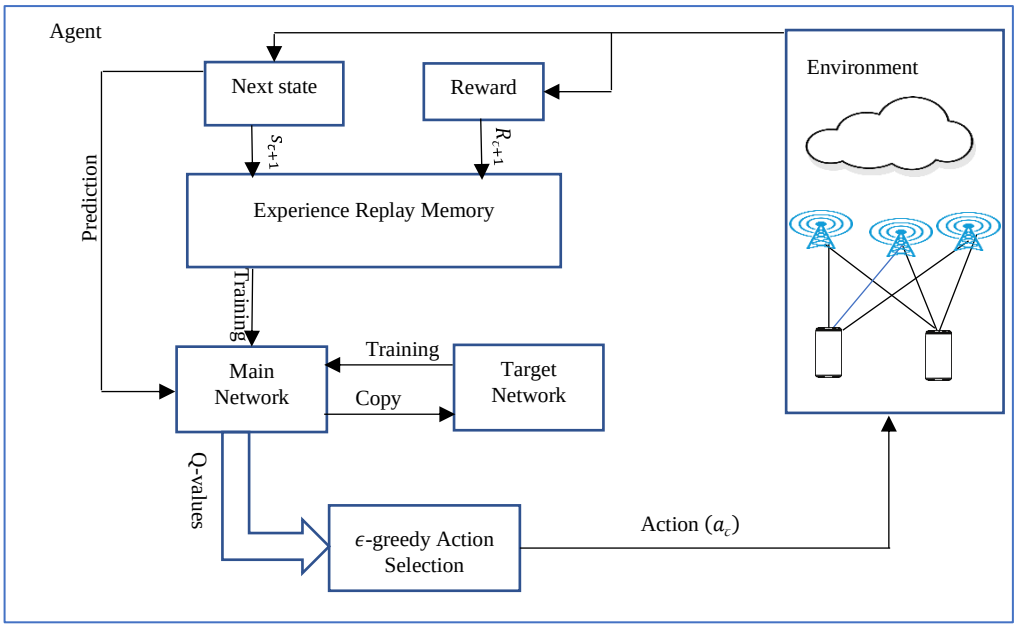


Figure 2.4: Conceptual DQN Architecture

operates similarly to the Q-value function except for the addition of neuron and replay memory. All input states are transferred to different NN layers, each with different weight factors θ and θ'. Finally, DQN generates the Q-value outputs with respect to possible actions. Furthermore, an experience replay memory is also used, where the network training is done by sampling a small batch of tuples from the replay buffer as $e = \{s, a, R, s'\}$, where s, a, R, s' indicates the possible sets of state, action, reward and next state values, respectively. These samples are then used to refine the Q-value estimation at each iteration. A conceptual DRL architecture is shown in Figure 2.4. The goal of DQN is to seek and find the best possible weight factors from historical data, including historical Q-values, actions, and state transitions. The complexity of calculating the Q-values and actions is linear for a DQN based on a multilayer perceptron as the underlying NN.

Moreover, the number of inputs is determined only by all the types of states. With each input, several values can be transferred into different network states

without changing the structure of the DQN, even when the number of all values reaches infinity. Thus, DRL significantly reduces the issues of future communication systems in terms of network complexity.

Furthermore, DRL can also be used for resource management and network optimization when resource capacities, such as those in edge and cloud services, are abundant. In (Xu *et al.*, 2017), the DRL approach is applied in CRAN for power saving while maintaining user QoS demand in highly dynamic cases. The DQN-based algorithm is used to solve the RA problem. Moreover, user demand and state of RRHs are considered at the input of the network state, where the action is restricted to the active set of RRHs. The proposed framework results are compared with two baseline approaches. The DRL-based framework saves 18% more power than the baseline approaches while maintaining the user QoS requirement. In (Y. Luo *et al.*, 2020), a Gradient Boosting Decision Tree (GBDT)-based DQN-framework is proposed to solve the dynamic RA problem in CRAN. The GBDT is first utilized for regression tasks to approximate second-order cone programming (SOCP) problems derived from beamforming design, which generally consumes a high level of computing resources. After that, a DQN-based algorithm is generated to find the robust policy that controls the RRHs switching and saves power over the long-term operation. Like (Xu *et al.*, 2017), the same state features, action values, and rewards function are considered by (Y. Luo *et al.*, 2020).

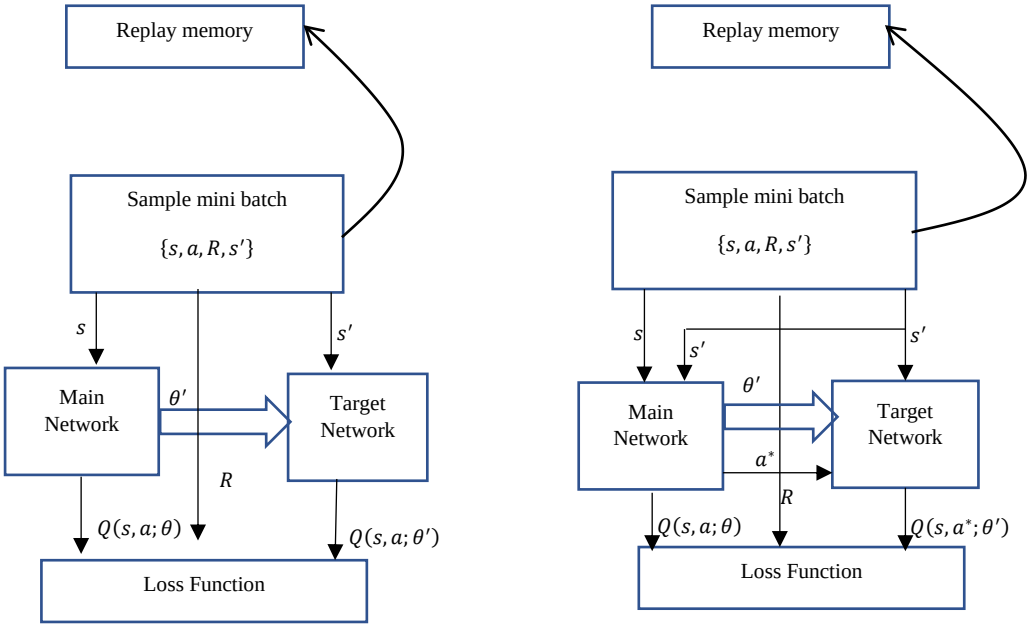
In (Y. Luo *et al.*, 2020), joint power allocation and user association are considered in Heterogeneous CRAN (H-CRAN). This work considers the hybrid action, continuous action for power allocation, and discrete action for device association. The hybrid action is solved by using novel parameterized

DQN (P-DQN) instead of quantizing the continuous power value. Additionally, the only user data rate is considered at the input of the network state. In contrast, the action is based on power allocation and user association with maximizing the overall EE as a reward function. Finally, P-DQN is compared with conventional DQN and the traditional approach.

The work in (Tasnim Rodoshi, Kim and Choi, 2020) presents a dynamically allocated resource solution based on DQN-algorithm that dynamically allocates resources to each virtual machine within a BBU pool in a CRAN. The DQN agent learns the variations in load across RRHs and allocates resources accordingly to the virtual machine. The proposed method can meet the users' demands while minimizing resource waste and unsatisfied requirements based on simulation results. The performance of the proposed algorithm has been evaluated using a real-world cellular dataset.

2.4.1.4 DOUBLE DQN

DQN uses the same max approximator to select and evaluate an action for the Q-values functions, which leads to an overestimation problem and degrades the system performance. In order to avoid the overestimation problem, Double DQN is proposed by (Hado van Hasselt, Arthur Guez, 2016). In Double DQN, two Q-networks are integrated to select and evaluate the action. In Double DQN, the one Q-network is used to determine the greedy policy of the main network for each update, whereas the second network is used to discover the value for the target network. Thus, Double DQN avoids the Q-value from overestimation and improves the objective function compared to DQN. The difference between DQN and Double DQN is illustrated in Figure 2.5.



a) DQN b) Double DQN

Figure 2.5: The flow of DQN and Double DQN

The work in (Zhang *et al.*, 2020) proposes a Double DQN-based BS sleeping algorithm to optimize the system EE and guarantee the users' QoS requirements. Real-world traffic data is collected from commercial RAN for a period of seven days. Furthermore, the state space is defined as BS arrive traffic data, whereas action space is based on BS on/off switching. The simulation results show that Double DQN outperforms energy-saving and QoS requirements as compared to the conventional DQN algorithm.

In (Yuan et al., 2021), a Double DQN algorithm is proposed to optimize the SE and QoE jointly by managing the power control and channel selection in a cognitive radio network. Moreover, the SINR value of the primary users is selected at the input of the network state, whereas the channel selection and power allocation for secondary users are allocated at the action space. Both the state space and action space take the discrete value of the primary and secondary users. Furthermore, simulation experiments are conducted to verify the stability and effectiveness of the Double DQN algorithm in the cognitive radio network,

and the performance is compared with Q-learning and DQN.

A distributed DRL-based framework is introduced to acquire the optimal user association and RA strategy in a heterogeneous downlink network (Zhao *et al.*, 2020). A multi-agent RL approach is proposed, which co-links UEs to BSs and allocates channels to UEs based on the Double DQN strategy. Optimization of the strategy ensures that UE's QoS requirements are guaranteed while maximizing the long-term rewards function.

2.4.1.5 Dueling DQN

Double DQN uses two separate networks to alleviate the overestimation problem. However, achieving fast convergence is still challenging due to large network parameters. In order to further improve the objective function and achieve higher convergence speed (Wang *et al.*, 2016) propose Dueling DQN. The idea behind Dueling DQN is that it is not always necessary to consider the value of each action. For some states, selecting an action has little or no impact. Therefore, selecting the state that strongly influences a particular action is necessary. Thus, the state-action Q-value (s, a) can be disintegrated into two parts, i.e., value function $V(s)$ and advantage function $A(a)$. The value function specifies how good it is to be in a given state, whereas the advantage function indicates the relative importance of a specific action compared with other actions. The $V(s)$ and $A(a)$ are then combined into a single Q-value function at the final layer. This result may lead to a more accurate policy evaluation in the wireless communication networks.

The work in (Sun, Ayepah-Mensah, Xu, *et al.*, 2020) proposes a Dueling DQN algorithm to minimize energy consumption and guarantee users' QoS

requirements. The proposed problem is divided into three parts. Firstly, a two-layer convolutional neural network (CNN) is formulated to capture the raw observation of the environmental input. Secondly, Dueling DQN based framework is developed to turn on/off the RRHs. Finally, the RA problem is formulated based on the users and delay constraints. Moreover, the user transmission rate is considered at the input of the network state, and RRHs on/off switching decision is assumed for the action space. Similar work can also be found in (Sun, Ayepah-Mensah, Budkevich, *et al.*, 2020), where the objective function is only to minimize the energy consumption using the function approximation method. Furthermore, users' data rate and on/off RRHs state are considered at the network input state while the action is performed based on the RRHs on/off switch. The authors do not describe the relationship between RRHs and users at the network input state in both works. A joint tradeoff between EE and SE in a 5G ultra-dense network is proposed in (Liu *et al.*, 2019). A Dueling DQN is developed to deal with the large state space explosion which is caused by the densification of the network. The traditional methods make it difficult to solve the large state space problem for MDP. The joint optimization problem of EE and SE for a multi-objective optimization problem (MOOP) is converted to a single-objective optimization problem (SOOP). Since EE and SE do not have the same units, the process is not clearly explained about the combined utility function.

The literature explained in Q-learning, DQN, Double DQN, and Dueling DQN is based on the value-based method. In the value-based method, the Q-value function is improved at each trajectory iteration sampled from the same environment until the Q-value function converges.

2.4.2 Policy-based Method

RL also uses the policy-based method by redefining the policy at each iteration and computing the Q-value function according to the new policy until the policy converges. The policy-based method optimizes the objective function directly while remaining stable under the function approximation. In (Gholipoor *et al.*, 2021), a joint radio and core RA framework is proposed for Network Function Virtualization (NFV)-enabled networks with the goal of maximizing the EE by guaranteeing the end-to-end (E2E) QoS requirements for different service types. The optimization problem is formulated based on the policy-based optimization problem in which power and spectrum resources are allocated in the radio part. Thus, the joint optimization problem is formulated as an MDP by considering the time-varying characteristics of the resources and wireless channels. Hence, a soft actor-critic DRL algorithm (SAC-DRL) based on a maximum entropy framework is used to solve the proposed MDP problem. According to simulation results, the proposed joint approach using the SAC-DRL algorithm reduced energy consumption significantly compared to the case in which NFV-RA and Radio-RA problems are optimized separately. However, such a process leads to an increase in system complexity. In (Wei *et al.*, 2018), a model-free RL framework is formulated to solve the energy efficient-oriented user scheduling and RA problems. In this work, the authors consider the channel state condition and transmission power are continuous variables. The actor-critic algorithm is used to learn the near-optimal stochastic policy. The actor part generates continuous actions based on parameterized stochastic policy, while the critic part evaluates the policy effectiveness and criticizes the action taken by the actor. Simulation results are presented to illustrate how the

proposed algorithm can improve the network's EE when harvesting more system energy. The work in (Li *et al.*, 2021) examines the DRL for beamforming in cell-free networks based on the closed-form of SINR per user and long-term EE function of MMSE and successive interference cancellation channel estimation. The DDPG-based algorithm is used to perform a centralized beamforming design for the long-term EE maximum problem with continuous state and action space. It shows that the DDPG-based algorithms are concurrent and can reduce the exponential computational complexity to a polynomial. In (Xu *et al.*, 2020), the RA issue is examined in vehicular communications using DDPG, where each vehicle-to-vehicle (V2V) communication acts as an agent and shares the frequency spectrum assigned to vehicle-to-infrastructure (V2I) communications using NOMA technology. A DDPG-based power allocation scheme for V2V is proposed in (Nguyen *et al.*, 2019), which strives to maximize EE without compromising the QoS for the V2V pairs.

As with MDP, policy-based methods have some disadvantages; they generally take longer to become convergent and evaluating policies can be time-consuming. Another disadvantage is that they tend to converge to local maxima rather than global maxima. Therefore, the policy-based method is beyond the scope of this thesis. The policy-based methods are generally used with the continuous state and action values, which requires a considerable amount of memory usage and computation consumption. Therefore, this thesis considers the Q-value function's discrete state and action values.

2.5 SUMMARY OF CHAPTER

This chapter presents the current state of the art work on solving the RA problem in wireless communication networks. Traditional optimization methods,

whether mathematical approach, programming approach, or heuristic approach, usually require lots of iterations to satisfy the required performance and lead to considerable computational power and delay complexity from an in-depth perspective. Secondly, the traditional optimization methods rely on the exact objective function, which means accurate information must be provided before solving the optimization problem, which is challenging to achieve in a highly dynamic scenario. Moreover, the complexity of the upcoming wireless communication networks is increasing exponentially. Therefore, it is very challenging for the traditional methods to find a robust policy with stable convergence results. The machine learning-based methods, especially RL and DRL, use a reward function to evaluate the decision behavior. RL method does not require the exact object function to solve the optimization problem. As RL makes the decision based on the current state and thus makes the online and real-time decision. Furthermore, RL is a decision-making method that provides more robust convergence results. The RL-extension, i.e., Q-learning, DQN, Double DQN, and Dueling DQN, solve the utility function following the value-based method, are briefly discussed. Finally, a policy-based RL algorithm (DDPG and actor-critic) is explained. This thesis considers the discrete state-action pairs (value-based algorithms) to update the Q-value function, where the optimal policy can be implicitly derived directly from the value function. In contrast, the policy-based method (DDPG and actor-critic) solve the continuous DRL problem, where the policy changes with each iteration to update the Q-value function. The policy-based method is used to solve the more complex problem as many hidden layers are required to configure the neural network. It leads to an increment of the computational time and degrades the system

capacity.

CHAPTER 3

DOUBLE DQN BASED RESOURCE ALLOCATION IN CLOUD RADIO ACCESS NETWORK

In this chapter, the proposed model-free reinforcement learning (RL) is presented in order to optimize resource allocation (RA) problems in a cloud radio access network (CRAN). The RA problem is formulated as a Markov decision process (MDP) to reap the accumulative reward function. In particular, the power minimization and energy efficiency (EE) problem is addressed by using a double deep Q-network (Double DQN) algorithm. A simulation was carried out to illustrate the proposed scheme's effectiveness in terms of power minimization and EE, with its results presented at the end of this chapter.

3.1 INTRODUCTION

The unprecedented demand for data traffic has prompted the telecommunications industry to adopt new technology, i.e., fifth-generation (5G). In order to meet the requirements of massively growing data traffic demand, CRAN has become a key enabling technique. However, it is still necessary to improve RA in CRANs over the long operational period. These RA are commonly associated with EE (Mesodiakaki *et al.*, 2014), transmission power (Ali *et al.*, 2017), and throughput (Dhif-Allah *et al.*, 2018). However, these studies do not address the time-correlated scenario in which actions taken at time slot t can alter the future utility distribution. For example, it has been shown that the energy switching costs of RRHs can be quite high (Yu *et al.*, 2016). Furthermore, these works have formulated the RA problem as a

conventional model-based optimization problem.

Markov decision process (MDP) can be formulated to model the sequential decision-making RA problem of the model-free RL framework. As opposed to the model-based approach, the model-free approach maximizes the EE while simultaneously satisfying and meeting the users' QoS demands for the whole operational period in a highly dynamic scenario. The model-free RL framework has two advantages. Firstly, they are capable of generating (sub)-optimal control actions based on feedback received from the environment. Secondly, they can maximize the network utility over long-term operations to make dynamic systems run more efficiently. A model-free RL algorithm called Q-learning has been used to solve the CRAN-RA problem. A Q-learning algorithm based on the base station on/off policy is proposed in (Miozzo *et al.*, 2015) in order to minimize the total power consumption while satisfying the user's data traffic demands. All the state-action pairs are stored in a lookup table in Q-learning, which works fine for a limited state-action pair problem. However, a significant problem with Q-learning is that it is not scalable when multiple actions follow many states (Karunakaran, Worrall and Nebot, 2020).

Q-value function approximation via deep learning (DL) is a step forward from Q-learning known as deep Q-network (DQN), which is introduced to solve the limited state-action pair problem. The work in (H. Li *et al.*, 2018) proposes a dynamic RA problem using the DQN algorithm for self-powered ultra-dense networks to improve the EE performance. A tradeoff problem between spectral efficiency (SE) and EE based on the DQN algorithm is presented in (Liu *et al.*, 2019). Similar approaches can be found in (J. Li *et al.*, 2018) (Ye, Li and Juang, 2019), where the environment scenarios are either mobile edge computing

(MEC) or vehicle-to-vehicle (V2V). However, DQN algorithms work with the same network parameters to update the target Q-value; they may overestimate the Q-value function. Therefore, in this chapter, the Double DQN framework is adopted to solve the overestimation Q-value problem, which incorporates two Q-networks for selecting and evaluating the actions. The proposed Double DQN based algorithm is energy efficient and provides better network performance to the Q-value function as compared to the conventional DQN.

The rest of this chapter is unfolded as follows. Section 3.2 describes the system model, while Section 3.3 presents a Double DQN based RA problem. Finally, Section 3.4 presents simulation details and results. This chapter is summarized in Section 3.5. A list of the key mathematical notations used in this chapter is defined in Table 3.1.

Table 3.1: List of Key Notations

Notation	Description
$\mathcal{T}$	Time-period
D_u	Data rate demands
$\mathcal{R}$	Set of RRHs
$\mathcal{U}$	Set of UEs
$\mathcal{B}$	BBU
PL	Path loss
$d_{r,u}$	Distance between the RRH and UE.
$h_{r,u}$	Channel gain between the RRH and UE.
$\varphi_{r,u}$	Antenna gain between RRH and UE

$\varsigma_{r,u}$	Shadowing coefficient between RRH and UE
$\vartheta_{r,u}$	Small-scale fading between RRH and UE
γ_u	Signal-to-interference-plus-noise ratio
$w_{r,u}$	Beamforming weight between RRH and UE.
σ^2	Background noise
B	Transmission bandwidth
Γ_m	Capacity gap
$p_{r,active}$	Active power
$p_{r,trans}$	Transmit power
$p_{r,switch}$	Transition power
$p_{r,sleep}$	Sleep power
$\mathcal{M}$	Set of active mode RRHs
$\mathcal{N}$	Set of sleep mode RRHs
$\mathcal{S}$	Set of transition mode RRHs
η	Power amplifier efficiency
S	Set of possible states,
A	Set of possible action
G	Reward
P_r	Transition probability
μ	Discount factor
π	Policy

α	Learning rate
θ	The main network weighted factor
θ'	Target network weighted factor
e_t	Experience replay
D_t	Mini-batch samples

3.2 SYSTEM MODEL

This chapter discusses a downlink CRAN, as shown in Figure 3.1, which consists of remote radio heads (RRHs) and a baseband unit (BBU). The BBU handles the digital signal processing (DSP), while RRHs transfer the data from the radio receiver to the end users' equipment (UE), respectively. This work assumes that the BBU acts as an RL agent, which is continuously interacting with the unknown environment, selecting the appropriate action from the inputs of RRHs and UEs. Furthermore, a time-period $\mathcal{T}$ is considered that is uniformly divided into time slot t, denoted as $\mathcal{T}$= {1,2,...,T}. Each UE's position changes randomly and reports its data rate demands $D_u \in [D_{min}, D_{max}]$ and channel state information (CSI) to the BBU during a given time-period $\mathcal{T}$. The BBU pool is aware of the data rate requirements of all UEs. In cases where such information is not available, the service provider will transmit such information to the BBU cloud. The RL agents then send switch decisions to RRHs and monitor the UEs mobility and its effect on switching. Finally, RL agents calculate the accumulative reward by aggregating the user satisfaction and power saving of all RRHs. The proposed model is simplified by assuming that all RRHs and UEs are equipped with a single antenna. However, such a model

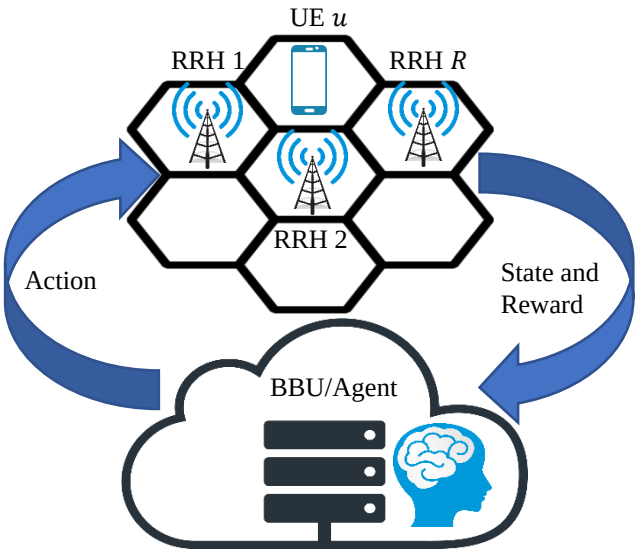


Figure 3.1: DRL Based CRAN scheme

can be generalized in the case of multiple antennas (Dai and Yu, 2016).

3.2.1 Network Model

As shown in Figure 3.1, a set of RRHs, a set of UEs, and a single BBU is considered and can be expressed as $\mathcal{R}$={1,2,…,R}, $\mathcal{U}$={1,2,…,U}, and $\mathcal{B}$, respectively. According to (Y. Luo *et al.*, 2020), the path loss of the system model is defined as:

$$PL(d_{r,u}) = 148.1+37.6 \log_2 d_{r,u} \text{ dB} \tag{3.1}$$

such that $d_{r,u}$ indicates the distance between RRH r and UE u. Furthermore, the channel fading model definition is taken from (Shi, Zhang and Letaief, 2015) as:

$$h_{r,u}=10^{-PL(d_{r,u})/20}\sqrt{\varphi_{r,u}\,\varsigma_{r,u}\,\mathfrak{y}_{r,u}} \tag{3.2}$$

whereas $\varphi_{r,u}$, $\varsigma_{r,u}$ and $\mathfrak{y}_{r,u}$ corresponds to the antenna gain, shadowing coefficient, and small-scale fading between RRH and UE, respectively. The rayleigh channel fading model is considered in this work; where $\mathfrak{y}_{r,u}$ is the

independent and identically distributed (i.i.d) complex Gaussian random variable that captures the small-scale fading effects associated with a radio link between RRH and UE. As stated in (Dai and Yu, 2016), the RRHs r cooperate to serve UEs u jointly by beamforming. Thus, the signal-to-interference-plus-noise ratio (SINR) of the UE u at time slot t, $\gamma_u(t)$ can be expressed mathematically as follows:

$$\gamma_u(t) = \frac{|h_u^H(t)w_u(t)|^2}{\sum_{v \neq u}|h_v^H(t)w_u(t)|^2 + \sigma^2} \tag{3.3}$$

Where $h_u(t)$, $(.)^H$ and $w_u(t)$ specify the channel gain, conjugate transpose of channel gain and beamforming weight between RRH r and UE u at time slot t, respectively, and each element of the channel gain and beamforming weight from RRH r to UE u can be written as $h_u(t) = [h_{1u}, h_{2u}, \dots, h_{Ru}]^T$ and $w_u(t) = [w_{1u}, w_{2u}, \dots, w_{Ru}]^T$. σ^2 denotes the noise. According to Shannon capacity, the user data rate at time slot t, can be given as:

$$C_u(t) = B\log_2\left(1+\frac{\gamma_u(t)}{\Gamma_m}\right) \tag{3.4}$$

B represents the channel bandwidth, and Γ_m denotes the SINR gap, depending on a few practical factors, for example, modulation.

3.2.2 Power Consumption Model

Based on (Auer *et al.*, 2012), the relationship between transmit power and receive power is basically linear. Thus, the linear power model can be applied to each RRH as:

$$p_r = \begin{cases} \frac{1}{\eta}p_{r,trans} + p_{r,active} & ; r \in \mathcal{M} \\ p_{r,sleep} & ; r \in \mathcal{N} \\ p_{r,switch} & ; r \in \mathcal{S} \end{cases} \tag{3.5}$$

such that $p_{r,active}$ is the active RRH power, $p_{r,trans}$ is the transmission power and can be expressed as $p_{r,trans}=\sum_{r\in\mathcal{M}}\sum_{u\in\mathcal{U}}\left|w_{r,u}\right|^2$, η indicates the power amplifier drain efficiency and is assumed as a constant value, $p_{r,switch}$ is the RRH switching power and $p_{r,sleep}$ is the RRH sleep power of RRH r. Whereas $\mathcal{M}$, $\mathcal{N}$ and $\mathcal{S}$ represent the sets of active, sleep, and mode-transition RRHs, respectively. Thus, at time slot t, the total power p_{total} of all RRHs is updated as follows:

$$p_{total}(t)=\sum_{r\in\mathcal{M}_t}\sum_{u\in\mathcal{U}_t}\frac{1}{\eta}\left|w_{r,u}(t)\right|^2+\sum_{r\in\mathcal{M}_t}p_{r,active}+\sum_{r\in\mathcal{N}_t}p_{r,sleep}+\sum_{r\in\mathcal{S}_t}p_{r,switch} \tag{3.6}$$

3.2.3 Definition of Spectral Efficiency and Energy Efficiency in CRAN

SE and EE are the two primary considerations for designing any efficient wireless communication system. The SE is defined as the ratio of throughput C_u to total available bandwidth B and can be represented as bits per Hertz. Mathematically, the SE is expressed as:

$$SE(t)=\frac{\sum_{u=1}^{U}C_u(t)}{B} \tag{3.7}$$

In the same way, EE represents bits of transmitted information per joule and can be defined as the ratio of throughput C_u to total power consumption $p_{total}(t)$ at time slot t. Mathematically, EE is given as:

$$EE(t)=\frac{\sum_{u=1}^{U}C_u(t)}{B\times p_{total}(t)} \tag{3.8}$$

In particular, when the transmission power distribution is uncontrolled, the EE will not be as high even with the high data rate. Therefore, an appropriate power

distribution scheme must be designed to obtain a higher EE.

3.2.4 Problem Formulation

EE is an essential component of the design of future wireless communication networks. Therefore, EE is regarded as a network utility function in this chapter. This chapter aims to maximize the long-term benefits of EE under certain constraints, including per-RRH transmission power and user data rate. The EE optimization can be accomplished by selecting a set of active RRHs during the time slot t and choosing the transmit power levels of the RRHs. Too much on/off switching of RRHs should be avoided in order to avoid a switching penalty $p_{r,\,\text{switch}}$. Let denote the overall beamforming weights by a matrix w with $[w]_{r,u} = w_{r,u}$. Thus, the EE problem can be formulated as follows:

$$\max \quad \sum_{t=1}^{T} EE(t) \tag{3.9}$$

$$\text{subject to } C_u(t) \geq D_u(t), \forall u \in \mathcal{U}, \forall t \in \mathcal{T} \tag{3.9.1}$$

$$\sum_{u \in \mathcal{U}_t} \left| w_{r,u}(t) \right|^2 \leq P_r, \ \forall r \in \mathcal{M}_t, \forall t \in \mathcal{T} \tag{3.9.2}$$

Constraint (3.9.1) stipulates that users' data rates must be greater than or equal to each UEs' target data rate. Constraint (3.9.2) specifies the maximum amount of power transmitted by the RRHs. Due to the interdependence between on/off switching decisions in adjacent time intervals, problem (3.9) cannot be solved directly. Additionally, the network traffic demands fluctuate both in the temporal and spatial domains. Using RL can effectively solve such a problem and motivate the design of subsequent solutions.

3.3 DOUBLE DQN BASED RESOURCE ALLOCATION OPTIMIZATION

In this section, a DRL-based algorithm is applied to solve the Equation (3.9) problem. Since the cumulative reward is to maximize the long-term EE performance based on RRH's actions in the CRAN is inevitably influenced (Liu *et al.*, 2021). It must be taken into account that the network environment state is time-variant. It is also assumed that the current state of the environment and actions influence the total rewards. For the optimization problem, a Markov decision process (MDP) is then formulated, which consists of a tuple of (S, A, G, S'), where S and A denote the sets of possible states and actions, respectively (Zhang, Zhang and Qiu, 2020). The RL agent observes the current state $s_t \in S$ and chooses an action $a_t \in A$ at time slot t. Based on the chosen action, a reward is generated from the environment $G(s_t, a_t)$, and the agent moves to the next state with a certain probability known as transition probability $P_r(S'|s_t, a_t)$. The basic elements of RL used in this chapter are presented first. After that, RL algorithm is proposed to solve the optimization problem.

3.3.1 Basic RL Elements

The essential elements of RL are defined first for the proposed model.

- **State Space**

At the time slot t, the RL agent should have the knowledge of all the UE data rate demand $D_u(t)$, the status of all RRHs $v_r(t)$, and CSI $g_{r,u}(t)$. Mathematically, the state space is defined as:

$$s(t) = \left[D_u(t), v_r(t), g_{R,U}(t)\ \right]^T \tag{3.10}$$

$v_r(t)$ indicates a binary value of RRH r, such that $v_r(t)$=1, means that RRH r is on; otherwise, $v_r(t) = 0$, and $g_{R,U}(t)$ implies the CSI between RRH r and UEs u, which is updated dynamically with the UEs' random movement.

- **Action Space**

The action space is based on the RRHs on/off switching decision at time slot t. The action space can be expressed as $a_r(t) \in \{0,1\}$. However, the RL agent must decide action based on the active set of RRH after successfully exploring the environment state. Note that the action can impact the next state s' in the next time slot t based on the active set of RRH.

- **Reward**

In this chapter, the reward is based on EE improvement, which can be determined as follows:

$$G(t) = EE(t) = \frac{\sum_{u=1}^{U} C_u(t)}{B \times p_{total}(t)} \quad (3.11)$$

Several iterative methods can be used to solve MDP problems, including dynamic programming (DP) and Q-learning (Geramifard *et al.*, 2013). However, DP requires accurate information about the environment as well as the reward function. As the CRAN for the 5G network environment is highly dynamic, this is incredibly very challenging for DP to have this kind of information in advance. Therefore, Q-learning is utilized to maximize the total accumulated expected value without directly modelling the CRAN environment.

3.3.2 Double DQN Based Strategy

Note that the RL agent aims to find the optimal policy $\pi^*: S \rightarrow A$ to maximize

the long-term accumulative reward function. Instead of sending its optimal policy, the RL agent iteratively sends the required user demand to its associated RRHs. Thus, the RL agent tries to learn its optimal policy $\pi^*: S \rightarrow A$ based on the state space elements. In this way, a global channel state is obtained between the RRHs and the UEs. The RL agent then needs to determine the optimal policy to achieve the maximum cumulative discounted reward with the QoS constraints (Fan and Li, 2017). According to (Sutton and Barto, 2018), the cumulative discounted reward value is defined as:

$$V^*(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \mu\, G(s(t), a(t))|_{s'=s,\ a'=a}\right] \tag{3.12}$$

where $\mathbb{E}(i)$ indicates the expectation of i, and μ denotes the discount factor. The s' and a' signifies the next state and action. To solve MDP, Q-learning is one of the popular RL methods. In Q-learning, Bellman's Equation can be used to determine the optimal Q-value function $Q^*(s, a)$, which can be expressed as:

$$Q^*(s, a) = Q(s, a) + \mu \sum_{s'} P_{s's}(a) \max_{a'} Q^*(s', a') \tag{3.13}$$

where $Q(s, a)$ and $P_{s's}(a)$ are the expected value of $G(s, a)$ and transition probability, respectively. Thus, the optimal policy $\pi^*(s)$ for the $Q^*(s, a)$ can be evaluated as:

$$\pi^*(s) = \max_{a'} Q^*(s, a) \tag{3.15}$$

It is always challenging to acquire the exact transition probability. However, Q-learning allows obtaining the optimal strategy based on the information available $(s, a, G,\ s')$ in a recursive manner. The Q-learning update equation is shown as follows:

$$Q(s,a) = (1-\alpha)\, Q(s,a) + \alpha \left[G(t) + \mu \max_{a'} Q(s',a') \right] \tag{3.16}$$

where α denotes the learning rate that influences the $Q(s,a)$-value updating speed.

It is difficult to determine the optimal policy when the state-action spaces become very large. The deep neural network (DNN) algorithm has been introduced recently to solve the sizeable state-action space problem. Deep Q-network (DQN) is a well-known method. In the DQN, the optimal policy and values functions can be approximated by using a DNN which is composed of multiple layers. Each layer has several neurons or nodes. Each neuron receives the weighted linear combination of the previous layers as input, and then a non-linear activation function is applied to generate the target value (output). A neural network from DNN can be considered a deep graph with many processing layers. A neural network function approximator is used as the main network with weights θ. Furthermore, a target network with weights θ' is used to stabilize the overall network performance. At each time step t, the weight θ is updated to minimize the loss function as:

$$L(\theta) = \mathbb{E}[\left(y^t - Q(s,a;\theta)\right)^2] \tag{3.17}$$

where,

$$y^t = G(t) + \mu \max_{a'} Q(s',a';\theta') \tag{3.18}$$

An ϵ-greedy policy is used to select the action from the main network $Q(s,a;\theta)$. The target network is a duplicate copy of the online network and has fixed weights for all the iterations. In contrast, the weights in the online network are continuously modified. In the DQN, an experience replay strategy is used to overcome the instability learning process. During the training stage, mini-

batches of experiences are randomly selected from the replay memory D, instead of using only the current experience (s, a, G, s'). With the experience replay strategy, the correlation between training examples is reduced, which prevents the optimal policy from being driven to a local minimum. Q-learning and DQN methods use the same max operator to select and evaluate actions. The Q-values function may be overestimated by using the same estimator. Thus, the use of Double DQN (Hado van Hasselt, Arthur Guez, 2016) mitigates the overestimation problem by replacing the target DQN y^t with the following target Double DQN:

$$y^t = G(t) + \mu\, Q\left(s', arg\max_{a'} Q(s', a'; \theta); \theta'\right) \tag{3.19}$$

At time step t, the main networks are used to determine the $\epsilon-$greedy policy, whereas target networks determine its value. The $\epsilon-$greedy policy is a random policy that promotes exploration rather than action that is determined by the maximum of the next state's Q-value. The detailed procedure used for the Double DQN is explained in Table 3.2, Algorithm 3.1.

Table 3.2: Algorithm 3.1 Double DQN based Resource Allocation

---

1: Set t=1

Initialize Experience memory N_D and soft update with τ.

Initialize the main network with a random weight and biases as θ

Initialize the target network as a copy of primary network weights and biases as θ'

2: **for** each episode, **do:**

3: Initialize Equation (3.10) for state $s(t)$

4: **for** each time slot, **do:**

5: Select an action a_t based on the $\epsilon-$greedy policy

6: Obtain immediate reward $G(t)$ and observe the next state s'

7: Solve (3.20) and obtain optimal beamforming solution

8: Store experience $(s_t, a_t, G(t), s_t')$ into N_D

9: Randomly sample some mini-batches $(s_t, a_t, G(t), s_t')$ from N_D

10: Calculate the target Q-value in the target deep network

11: **If DRL=DQN** set the target

$y^t = G(t) + \mu \, max \, Q(s', a'; \theta_t')$

12: **If DRL=Double DQN** set the target

$y^t = G(t) + \mu Q\left(s', \underset{a'}{argmax} Q(s', a'; \theta(t)); \theta'(t)\right)$

13: Train the main network to minimize loss function $L(\theta)$ of Equation (3.17)

$$L(\theta) = \mathbb{E}[(y^t - Q(s_t, a_t; \theta))^2]$$

14: Perform gradient descent step on the target network of Equation (3.18)

$(y^t - Q(s_t, a_t; \theta(t)))$

15: Update target deep networks after some steps as

$\theta'(t) = \tau\theta(t) + (1 - \tau)\theta'(t)$

16: $t = t + 1$

17: **end for:**

18: **end for**

3.4 TRANSMIT POWER ALLOCATION

The DQN and Double DQN models determine the RRH selection at each time slot t. Given the set of active RRHs $\mathcal{M}_t$, the Equation (3.6) can be simplified to a slot-by-slot optimization problem as:

$$\min_{\mathbf{w}_t} \sum_{\forall r \in \mathcal{M}_t} \sum_{u \in \mathcal{U}_t} \left|w_{r,u}(t)\right|^2 \tag{3.20}$$

subject to $\gamma_u(t) \geq SINR_u(t), \forall u \in \mathcal{U}_t$ (3.20.1)

$$\sum_{u \in \mathcal{U}_t} \left|w_{r,u}(t)\right|^2 \leq p_{r,transmit}, \ \forall r \in \mathcal{M}_t \tag{3.20.2}$$

According to constraint (3.20.1), the demand of user u is guaranteed and $SINR_u(t) = \Gamma_m\left(2^{D_u(t)/B}\text{-}1\right)$. Problem (3.20) belongs to a convex optimization problem since it can be transformed into a second-order cone optimization problem (SOCP) (Wiesel, Eldar, and Shamai, 2006). Such a problem can be solved efficiently via a conventional algorithm (Soma *et al.*, 1998). A standard interior-point method can be used to solve SOCP for Equation (3.20), e.g., see (Ben-Tal, A. and Nemirovski, 2001, Chapter 6). Thus, the worst-case computational complexity of the proposed Double DQN algorithm is $\mathcal{O}(R^{3.5}U^{3.5}E + \mathcal{D} + |\theta|)$, where E represents the episodes require to converge Algorithm 3.1. $\mathcal{D}$ and $|\theta|$ denotes the number of experience samples in the replay buffer and cardinal of weights, respectively. Even though the proposed Double DQN algorithm achieves the best network performance in comparison to the conventional DQN. However, its computational complexity is higher than other discussed algorithms. In some instances, insufficient active RRHs can lead to an infeasible solution. In such cases, the targeted value is set $G(t) = 0$, which will make the agent more aggressive in turning on the RRH in subsequent time slot t.

3.5 RESULTS AND DISCUSSIONS

This section investigates the performance of the proposed Double DQN-based RA problem in a CRAN and compares its results with conventional DQN and traditional approaches. The traditional approach is assumed as Full Coordinate Association (FA), where all RRHs are turned on, and users can be assigned to associate with multiple RRHs. The RA performance is evaluated based on 100 testing episodes after the RL agent has been trained for 1000 training episodes.

Two scenarios are considered to reach the optimal power-saving versus the user data rate requirements. In order to comply with the fair comparison of the network performance requirement, the same network configuration is maintained as (Dai and Yu, 2016), tabulated in Table 3.3. The selected parameters for Table 3.3 gives the optimal network configuration performance. Furthermore, dynamic channel gain with dynamic UE demand is considered for long-term RA optimization, whereas the UE demand is uniformly distributed in the interval of [10-70] Mbps. After each decision epoch, the UEs' data rate demands and channel gain changes. Note that the total average power consumption is calculated for a 7-minute time slot interval (a 7-minute time slot interval is enough for the RL agent to determine the total average power consumption for the given number of episodes). For each decision epoch, a minimum of five seconds is considered.

Table 3.3: Simulation Setting Parameters

Parameter	Value
Noise power σ^2	-102 dBm
Bandwidth B	10 MHz
Active power $p_{r,\text{active}}$	6.8 W
Transmit power $p_{r,\text{transmit}}$	1 W
Sleep power $p_{r,\text{sleep}}$	4.3 W
Transition power $p_{r,\text{switch}}$	3 W
Antenna Gain $\varphi_{r,u}$	9 dBi
Shadowing coefficient $\varsigma_{r,u}$	8 dB
Capacity gap Γ_m	1

Small scale fading $\eta_{r,u}$	$\mathcal{CN}(0,1)$
Power amplifier efficiency η	25 %
Epsilon-greedy policy ε	0.05

3.5.1 Power Consumption versus User demand

In order to verify the effectiveness of the proposed Double DQN algorithm performance for the average power saving versus user demands, two scenarios are mainly considered, i.e., 1) U=2, R=5, and 2) U=4, R=12. As shown in Figure 3.2, the occupied resources from the given set of RRHs to the UE are dynamically changing with each time slot t, with static user demand. However, one can see from Figure 3.2 that the power consumption increases monotonically as the volume of data traffic increases. This is because more transmission power is required to satisfy the users' QoS requirements. The DQN approach saves 10.39% more power than the FA approach, while the proposed Double DQN approach saves 13.43% more power than the DQN approach. In Figure 3.2, it can also be seen that the DRL approach consistently outperforms

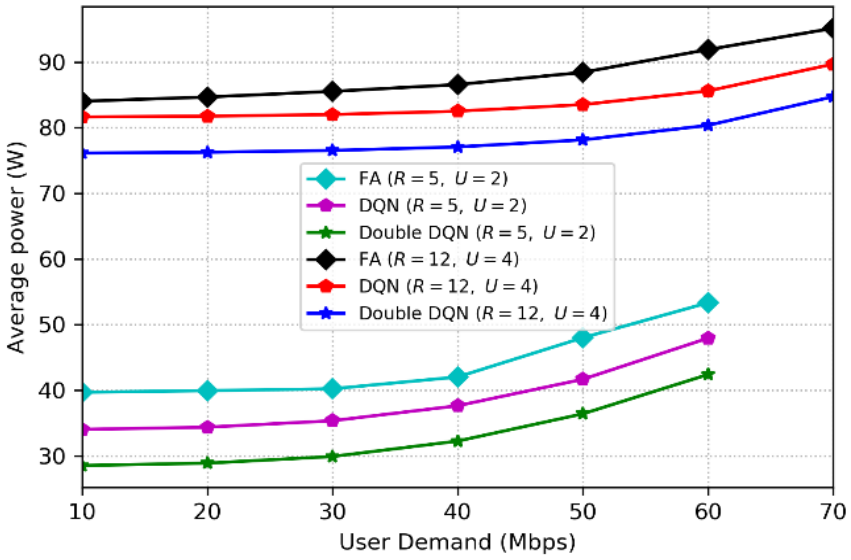


Figure 3.2: Average power consumption versus user demand

the FA method. In addition, the Double DQN has superior performance at all points over the DQN, thereby saving 7%-20% more power. However, due to the resource constraints and the increasing user interface data rates, all three methods cannot satisfy the power demand beyond 60Mbps. The instability problem can be alleviated by increasing the number of RRHs to 12 and the number of UEs to 4, as shown in Figure 3.2. The proposed Double DQN based approach can save more power at all points of user demands. When the DQN approach is used with the 10Mbps user demand, it can save 7% more power than the FA approach. At the same point, the proposed Double DQN can save 8.35% more power than the DQN. A similar performance can be seen for the user demand of 70Mbps. On 70Mbps, the proposed Double DQN approach saves 10.49% and 14.55% more power than DQN and FA approaches, respectively. Furthermore, it can be seen from Figure 3.2 that power increases linearly with increasing user demand for all three methods. However, one can conclude that the proposed Double DQN method is more efficient in power-saving than the DQN and the FA for the upcoming wireless communication networks (5G and beyond).

After this, the long-term performance of the proposed Double DQN algorithm is evaluated in a highly dynamic scenario. The number of RRHs and the number of UEs are set to 12 and 4, respectively. The selected values of RRHs and UEs completely fulfil the users' QoS demands. The user demand is uniformly distributed between 10 Mbps and 70 Mbps. Furthermore, the RL agent is fixed to 5 seconds to calculate the average power consumption for each decision epoch. The UE demand may be unchanged during an epoch but might vary in the next decision epoch. Note that the DRL transition power has been included

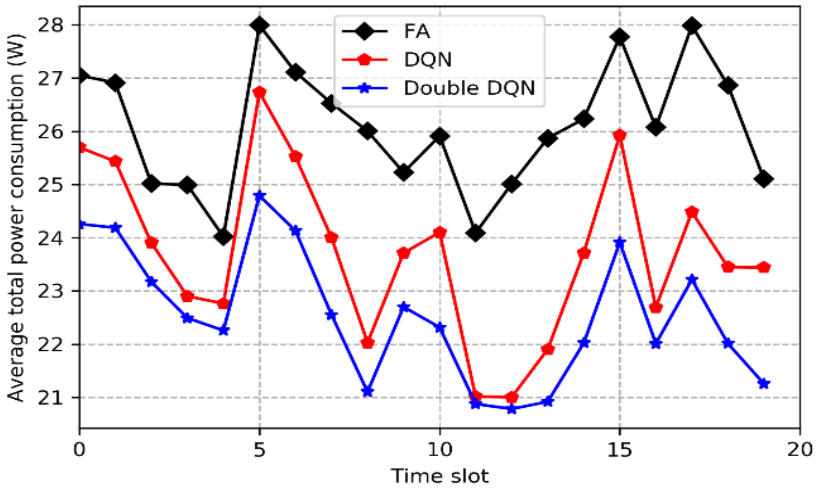


Figure 3.3: Average power consumption versus time slot *t*

in the simulation. Finally, the average power consumption is calculated over a relatively long time interval of 7 minutes. It can be observed from Figure 3.3 that the power consumption fluctuates with the change in the user demand. Thus, it can be examined that the proposed Double DQN consistently outperforms both DQN and FA in power savings and satisfying users' QoS demand. The above performance demonstrates the effectiveness of the proposed Double DQN algorithm in a highly dynamic situation.

3.5.2 Energy Efficiency versus User Demand

The performance of energy efficiency versus growing user demand is shown in Figure 3.4. In this case, the increasing user demand will result in a linear increase in energy efficiency. The algorithm proposed by the Double DQN has achieved a superior performance to the DQN and FA schemes (Dai and Yu, 2016). Figure 3.4 shows that the energy efficiency performance with DRL is superior to the FA approach. This comes from the fact that DRL considers past learning experiences instead of making the decision based on the instantaneous network state, which leads the FA approach to lower network performance. It

can also be observed that the proposed Double DQN algorithm outperforms the DQN (Xu *et al.*, 2017) by about 7-18% at every point of increasing data rate demands. This is because the DQN uses just one Q-value estimator to select and evaluate the action for the Q-value function. In contrast, the proposed Double DQN uses two separate estimators for the Q-value function. The two independent estimators will help the Double DQN agent to select the unbiased Q-value. However, the performance of energy efficiency of all three approaches is monotonically increasing with the user demand. In contrast, when the user data demand approaches 60Mbps, all three methods become unstable with no further increment of energy efficiency. This is because the proposed scheme aims to maximize the energy efficiency performance based on the user data rates which is defined as the logarithmic function in Equation (3.4). The optimal energy efficiency increases with the increasing user demand since a larger number of active RRHs are required to achieve optimal energy efficiency at every point of user demands. The problem of instability occurs, when the user demands approaches to 60Mbps for all three methods. This is because, a large transmission power is required to satisfy the users' QoS demand. In order to

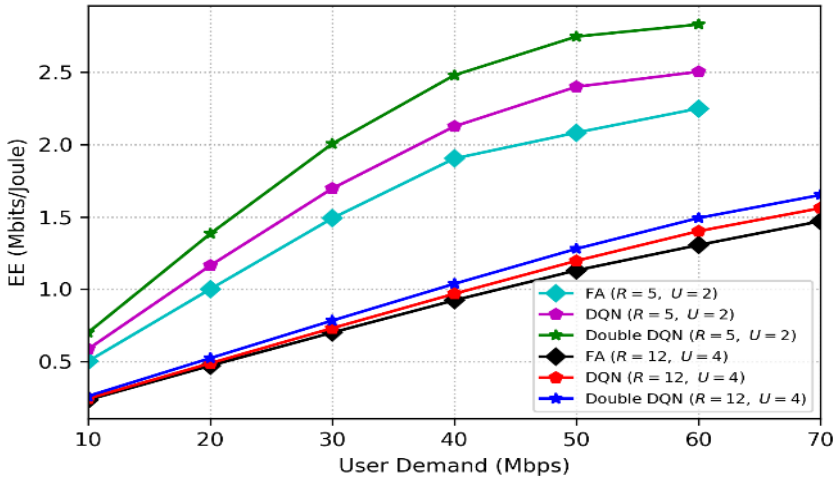


Figure 3.4: EE versus user demand

avoid the instability problem, the number of RRHs and the number of UEs are increased to 12 and 4, respectively, as shown in Figure 3.4. Notably, the energy efficiency performance with the Double DQN algorithm outperforms other algorithms. The Double DQN uses two Q-networks to turn on/off RRH of given users' demands, which results in 6-16% higher energy efficiency than other approaches. From Figure 3.4, it can be concluded that increasing R and U affect EE in a complex way. As a result, adjusting these parameters is necessary to provide higher energy efficiency.

3.5.3 Energy Efficiency versus Power Consumption

The energy efficiency performance is plotted against the obtained average power consumption for $R = 5$ and $U = 2$ in Figure 3.5. The energy efficiency performance is linearly increasing with average power consumption. However, energy efficiency becomes constant when the power consumption reaches the highest value, i.e., the threshold value P_{max}. In other words, the total power consumption $p_{total}(t)$ should be greater than the threshold value P_{max}. It can be seen from Figure 3.5 that the proposed Double DQN algorithm consistently outperforms the DQN and FA approach. The Double DQN consumes the power

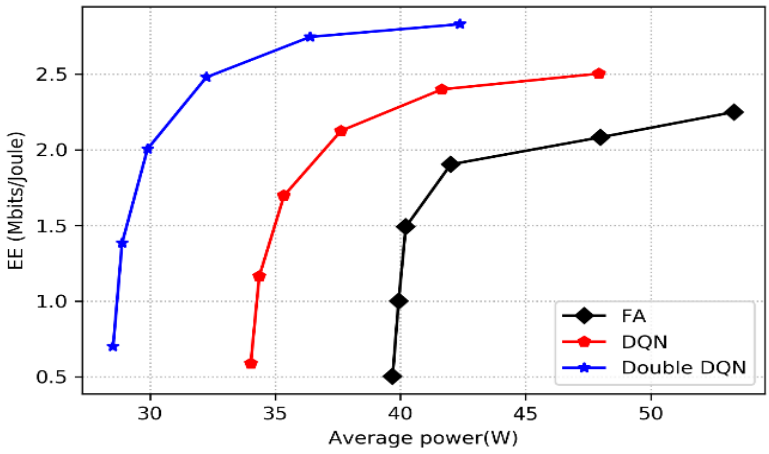


Figure 3.5: EE versus average power consumption for R=5, U=2

of 42W to achieve the energy efficiency of 2.81 Mbits/Joule, whereas the DQN consumes the power of more than 47W to achieve an energy efficiency of 2.521 Mbits/Joule. It means that the proposed Double DQN algorithm is 10%-15% more energy-efficient than the DQN. A similar approach can be found in Figure 3.6 for R=12 and U=4. At the start, the energy efficiency increases slightly with increased power over a short period, as shown in Figure 3.6. After the EE reaches a maximum value, it declines for all three approaches. This is because all three methods attempt to maximize the user data rate, which results in higher transmit power consumption and, ultimately, lower energy efficiency. Thanks to the Double DQN that paves a maximum energy efficiency of 1.62 Mbits/Joule with a value of power consumption of 84.92W, whereas the DQN and FA approaches can obtain the energy efficiency of 1.579 Mbits/Joule and 1.512 Mbits/Joule with 89.21W and 94.982W power consumption, respectively. These performances show that the proposed Double DQN based algorithm saves more power and achieves higher energy efficiency than the DQN and FA schemes.

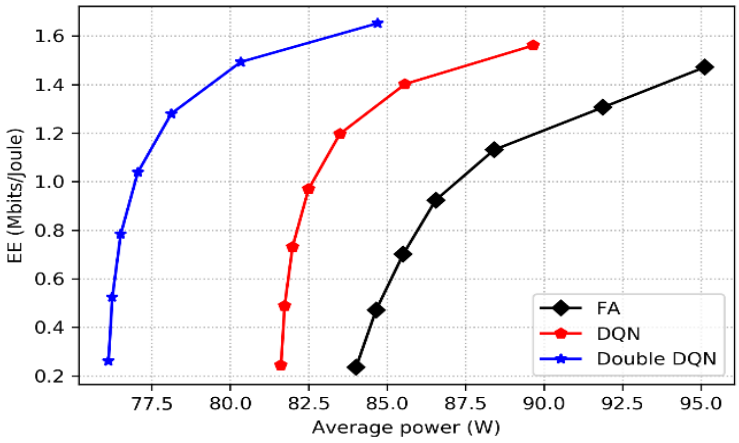


Figure 3.6: EE versus average power consumption for R=12, U=4

3.6 SUMMARY

This chapter examines a Double DQN based RA framework in a CRAN that maximizes the total EE under the constraints associated with transmission power selection by RRHs and user rates. The traditional approach known as the FA approach is modelled first, which relies on immediate actions with no regard for their effects in the future. Next, a DQN scheme is proposed based on past learning experiences and considering future effects. The DQN uses only one estimator to select and evaluate the action for the Q-value function, which generates the over-optimistic Q-value. Thus, it decreases the probability limit to estimating the maximum Q-value function. Finally, a Double DQN algorithm is proposed that separates the selected actions from the corresponding target Q-value generation. The proposed Double DQN algorithm leads to higher power-saving and maximizes the EE while satisfying the user QoS requirements in the CRAN.

CHAPTER 4

DUELING DEEP Q- LEARNING-BASED JOINT RESOURCE ALLOCATION IN CLOUD RADIO ACCESS NETWORK

SE and EE are vital performance evaluation metrics for designing any wireless communications system. However, these metrics always contradict each other and can be linked through their tradeoff. The EE and SE tradeoff in the cloud radio access network (CRAN) has been accurately approximated in the past but only for the static network state. This chapter investigates a deep reinforcement learning (DRL)-based framework to maximize the long-term tradeoff between EE and SE. Specifically, machine learning (ML) techniques are used to extract generalized features of spatio-temporal channel state information (CSI) before feeding them into the input of DRL. A simulation study and its results are presented at the end of the chapter to compare the performance of the proposed scheme in different scenarios.

4.1 INTRODUCTION

As smartphone usage has increased rapidly in recent years, there is an expectation that the total mobile data traffic will grow to 77 exabytes (EB) per month by 2022 (Cisco, 2020). Conventional network architectures may not be able to cope with such an extreme amount of data traffic and satisfy the users' quality of service (QoS) requirements. This is due to the high interference with the reuse of the same radio resources by multiple base stations (BSs).

The CRAN is a promising solution among the available networking solutions since shared network resources can be virtualized and controlled among

distributed remote radio heads (RRHs) (Checko *et al.*, 2015). The main purpose of CRAN is to decouple the remote radio head (RRH) and baseband units (BBU) functionality from the BS. A BBU performs all baseband signal processing, while RRHs handle modulation and amplification. However, due to the non-uniformities in the space and time of the devices being used, it is more challenging to manage CRAN resources (Wang *et al.*, 2015). Thus, allocating resources adaptively in the CRAN should be discussed in more detail.

The resource allocation (RA) problem in heterogeneous network scenarios (HetNet) and CRAN scenarios has been studied from various perspectives to understand their effects on network performance and user experience. In (Ali *et al.*, 2017), a CRAN's overall throughput maximization problem is investigated. In (Ahmad *et al.*, 2020), the depreciation of a weighted transmit power subject to fronthaul capacity and minimum QoS constraints are discussed. The EE optimization problem in HetNet is studied in (Wu, Zeng and Xia, 2017). To balance the joint SE and EE is discussed in (Coskun and Ayanoglu, 2017) and (Xu, Li and Yang, 2018). These studies, however, considered network performance and user requirements only in a fixed state and benefited from short-term rewards. For example, continuously switching off and on RRHs in adjacent time slots can increase deployment costs. Such behaviors which seek immediate benefits are called "myopic."

The model-free reinforcement learning (RL) framework improves the performance over the long term in wireless networks. RL can determine the optimal policy by interacting with the unknown environment (Wei *et al.*, 2018), (Asheralieva, 2017). Q-learning is a well-known and widely used model-free RL algorithm. Using Q-learning in (Shams, Bacci and Luise, 2015), the user's

data traffic demand is satisfied while minimizing the power consumption. However, due to the sizeable state-action space in practical problems, Q-learning convergence time is prolonged, and it is not easy to find the optimal solution. Deep Q networks (DQNs) combine the process of RL with a kind of neural network called a deep neural network to approximate the state-action value functions; this alleviates the limitations of Q-learning.

As a result of CRAN RA schemes that exploit the benefits of DRL, it has been utilized into three basic categories, namely, sum-power minimization (Zhang *et al.*, 2020), the sum of network performance (Gao *et al.*, 2019), and quality-of-service (QoS) requirement (Chen *et al.*, 2021). Several QoS constraints are used to achieve CRAN optimization objectives, such as RRH switching costs, transmission delay, cache management, transmission power, transmission rate, and BBU allocation. These solutions improve network performance beyond the limits of conventional approaches due to the well-formulated state space, action space, and reward functions in every class above. However, the definitions of state space are associated with traffic profiles, not with multiuser diversity.

CRANs can collect continuous CSI from RRHs as a 3-dimensional (3D) matrix and exploit various cooperative diversity gains in the CRANs. However, directly employing CSI as the DRL input will slow down the state-space exploration, especially in large-scale networks. Therefore, limiting the CSI before invoking it to the DRL framework is necessary to improve network efficiency. This chapter first models the RA problem as a Markov Decision Process (MDP) to optimize the long-term tradeoff between EE and SE to improve network efficiency. The Anchor Graph Hashing (AGH) (Liu *et al.*, 2011) is then employed to convert the CSI matrix into binary hash code and

concatenate it with the other elements of DRL, i.e., RRHs features and QoS requirements, leading to one-row feature vectors. Finally, the one-row feature vector is fed into the proposed DRL framework.

The rest of this chapter is divided into four sections. Section 4.2 describes the CRAN model and problem formulation. In Section 4.3, a DRL-based solution is proposed for the long-term RA decision. In Section 4.4, simulation details and results are discussed. Section 4.5 wrap up the chapter. A list of the key mathematical notations used in this chapter is defined in Table 4.1.

Table 4.1: List of Key Notations

Notations	Description
$\mathcal{J}$	Set of RRHs
$\mathcal{U}$	Set of UEs
$\mathbb{T}$	Time-period
D_u	Data rate demand
PL	Path loss
$d_{j,u}$	Distance between RRHs and UEs
$\varphi_{j,u}$	Antenna gain
$\eta_{j,u}$	Shadowing coefficient
$\varsigma_{j,u}$	Small-scale fading
$\delta_u(t)$	Signal-to-interference-plus-noise ratio
h_u	Channel gain
w_u	Beamforming weight

B	Bandwidth
$\mathcal{J}_m$	Capacity gap
$p_{j,trans}$	Transmission power
η	Power amplifier
$p_{j,act}$	Active power
$p_{j,slp}$	Sleep power
$p_{j,tp}$	Transition power
α	Tunable parameter
L	Extracted CSI samples
n	Anchors
b_d	Threshold distance
m	Number of iterations
s_t	State-Space
a_t	Action-space
$K(t)$	Reward
$\mathcal{r}$	Learning rate
μ	Discount factor
$\mathcal{D}_t$	Experience Replay Memory
ϑ	Value function parameters
β	Advantage function parameters

4.2 SYSTEM MODEL

In this chapter, a typical downlink CRAN is considered, which comprises a set of UEs $\mathcal{U}$={1,2,…,U}, a set of RRHs $\mathcal{J}$={1,2,…, J} and a single BBU. The dynamic RA is considered in this work, as shown in Figure 4.1, where at each time slot t, the current states of RRHs and the user data rate demands from the networks is obtained. Since all RRHs are connected to the centralized BBU pool. The DRL agent can share and process all information efficiently. Additionally, a time-period $\mathbb{T}$={1,2, ... T} is uniformly divided into multiple time slots t. Each UE has a specific data rate demand $D_u \in [D_{min}, D_{max}]$. Each UE, regardless of its position, reports its CSI to the BBU during time slot t. To simplify the proposed model, all RRHs and UEs are equipped with a single antenna, which can be extended to multi-antenna cases using the technique (Dai and Yu, 2016). According to (Dai and Yu, 2016), the channel fading model is defined as:

$$h_{j,u}(t) = 10^{-PL(d_{j,u})/20} \sqrt{\varphi_{j,u}\, \varsigma_{j,u}\, \mathfrak{y}_{j,u}} \tag{4.1}$$

such that $PL(d_{j,u}) = 148.1 + 37.6 \log_2 d_{j,u}$ dB denotes the path loss and $d_{j,u}$ represents the distance between RRHs and UEs, whereas $\varphi_{j,u}$, $\mathfrak{y}_{j,u}$ and $\varsigma_{j,u}$ indicates the antenna gain, shadowing coefficient, and small-scale fading between RRHs and UEs. The rayleigh channel fading model is considered in this work; where $\mathfrak{y}_{j,u}$ is the independent and identically distributed (i.i.d) complex Gaussian random variable that captures the small-scale fading effects associated with a radio link between RRH and UE. Then the corresponding signal-to-interference-plus-noise ratio (SINR) at the receiver of UE u at time slot t, $\delta_u(t)$ can be written as:

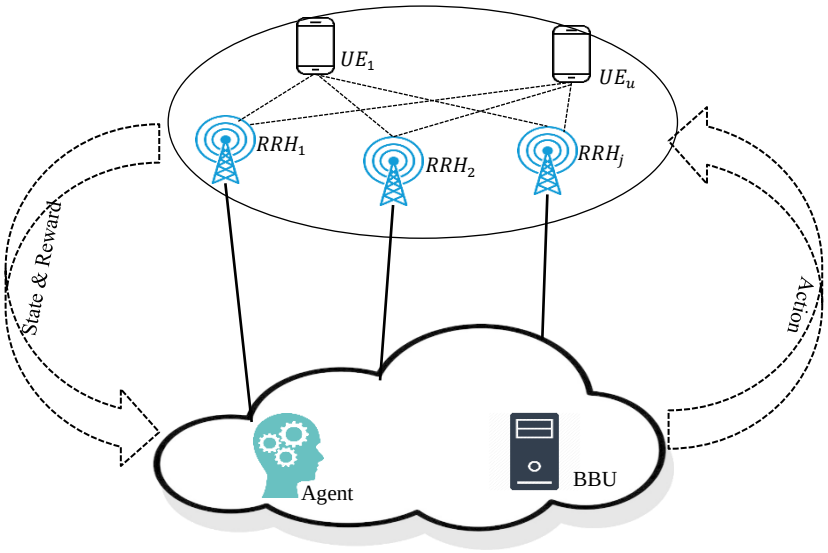


Figure 4.1: Deep reinforcement learning based CRAN model architecture

$$\delta_u(t) = \frac{|h_u^H(t)w_u(t)|^2}{\sum_{v \neq u}|h_v^H(t)w_u(t)|^2 + \sigma^2}, u \in \mathcal{U} \tag{4.2}$$

such that $h_u(t)=\left[h_{1u}, h_{2u}, \ldots, h_{Ju}\right]^T$ means the channel gain vector and each element of h_{ju} represents the channel gain from RRH j to UE u at time slot t. Similarly, $w_u(t)=\left[w_{1u}, w_{2u}, \ldots w_{Ju}\right]^T$ indicates the weighted vector, and each element of w_{ju} showing the beamforming weight from RRH j to UE u at time slot t, whereas σ^2 denotes the background noise. According to Shannon capacity, the data rate of UE u at time slot t can be determined as:

$$C_u(t) = B \log_2\left(1 + \frac{\delta_u(t)}{\mathcal{J}_m}\right), u \in \mathcal{U} \tag{4.3}$$

whereas B is the channel bandwidth and $\mathcal{J}_m$ is the signal-to-noise ratio (SNR) capacity gap.

4.2.1 Power Consumption Model

According to (Auer *et al.*, 2012), the relationship between the BSs transmitting power and consumption power is approximately linear. Thus, in this chapter,

the linear power model is applied to each RRH using the following equations:

$$p_j(t)=\begin{cases}\frac{1}{\eta}p_{j,trans}(t)+p_{j,act}(t) & ;j\in\mathbb{A}\\ p_{j,slp}(t) & ;j\in\mathbb{S}\end{cases} \tag{4.4}$$

where $p_{j,trans}(t)=\sum_{j\in\mathbb{A}}\sum_{u\in\mathcal{U}}\left|w_{j,u}\right|^2$ is the transmission power of RRH j and η is the power amplifier drain efficiency and assumed as a constant. $p_{j,act}(t)$ and $p_{j,slp}(t)$ indicates the active power and sleep power of RRH j, respectively. In contrast, $\mathbb{A}$ and $\mathbb{S}$ denote the active and sleep modes of RRH j, respectively. Thus, one has $\mathbb{A}\cup\mathbb{S}=\mathcal{J}$.

Furthermore, the power consumed by RRHs in changing their states is also considered and known as transition power $\mathcal{S}$. A set of transition powers $\mathcal{S}$ can be defined as $p_{j,tp}\ (t)$ for RRH at the current time slot t, and hence the total power $P_{total}(t)$ of all RRHs can be expressed as:

$$P_{total}(t)=\sum_{j\in\mathbb{A}}\sum_{u\in\mathcal{U}}\left|\frac{1}{\eta}w_{j,u}\right|^2+\sum_{j\in\mathbb{A}}p_{j,act}(t)\ +\sum_{j\in\mathbb{S}}p_{j,slp}(t)+\sum_{j\in\mathcal{S}}p_{j,tp}(t) \tag{4.5}$$

4.2.2 Problem Formulation

According to (Vu *et al.*, 2018), EE (bits/joule) is defined as the ratio between the sum of throughput $C_u(t)$ and total power consumption $P_{total}(t)$, and can be expressed as:

$$EE(t)=\frac{\sum_{u=1}^{U}C_u(t)}{P_{total}(t)} \tag{4.6}$$

In the same manner, the SE (bits/s/Hz) is defined as the ratio between the sum of throughput $C_u(t)$ and available bandwidth B, and can be mathematically expressed as follows:

$$SE(t) = \frac{\sum_{u=1}^{U} C_u(t)}{B} \tag{4.7}$$

This chapter aims to maximize both EE and SE while satisfying the minimum requirements of users. As stated earlier, the problems of maximizing EE and SE of the network usually contradict each other. Multiple objective optimization problems (MOOP) are used to optimize EE and SE simultaneously. The MOOP problems are generally solved by combining the objectives function under a single objective optimization problem (SOOP). In that way, the weighted summation method is utilized to combine the EE and SE metrics. However, EE ($bits/Joule$) and SE($bit/s/Hz$) have different units, so these two metrics are combined in a weighted summation method to ensure that the metric units are the same for the joint optimization problem. Thus, $B/p_{j,trans}(t)$ is multiplied with $SE(t)$ to ensure that the metric units are the same in weighted summation (Coskun and Ayanoglu, 2017). To further tune the objective function, a unitless parameter $\alpha \in (0,1]$ is introduced which helps to decide whether to optimize the network for EE or SE, depending on the network condition. For example, increasing SE is more valuable than EE during peak hours to satisfy the increasing demand of more users. Meanwhile, optimizing network EE is essential to reduce energy consumption during off-peak hours. The joint optimization problem between EE and SE can be expressed mathematically as:

$$\max \sum_{t=1}^{T} \left[(1-\alpha) EE(t) + \alpha \frac{B}{p_{j,trans}(t)} SE(t) \right] \tag{4.8}$$

$$\text{subject to} \quad B \log_2 \left(1 + \frac{\delta_u(t)}{\jmath_m}\right) \geq D_u(t), \forall u \in \mathcal{U}, \forall\, t \in \mathbb{T} \tag{4.8.1}$$

$$\sum_{u \in \mathcal{U}} \left| w_{j,u}(t) \right|^2 \leq P_j, \quad \forall t \in \mathbb{T}, \forall j \in \mathcal{J} \tag{4.8.2}$$

Constraints (4.8.1) specify that the UE's data rate requirements must exceed or be equal to the UE's target data rate, whereas constraint (4.8.2) limits the transmission power per RRH.

4.3 PROPOSED SOLUTION BASED ON DEEP REINFORCEMENT LEARNING

This chapter presents a joint optimization problem of EE and SE in the CRAN with the proposed dueling deep Q-network (D2QN) (Wang *et al.*, 2016) algorithm based on the per RRH activation power. Most of the existing works consider discrete state-space or use function approximation methods to achieve its objective function. However, such approaches do not capture the dynamic environment. This chapter adds a relational matrix to the state-space model, which expresses the spatial-temporal relationship between RRHs and UEs in a mobile environment. Meanwhile, the random user movement at each time slot t increases the state space exponentially, making it impossible for the RL agent to extract all the information in a reasonable amount of time. In order to reduce the state space, the anchor graph hashing (AGH) method is investigated and then map the AGH to hash codes. The obtained hash codes specify the discrete characteristic of state space and can easily fit the input of other elements of DRL.

4.3.1 Anchor Graph Hashing

The concept of hashing is widely used in big data applications as a method of approximate nearest neighbour search due to its low storage cost and speed of retrieval. The objective of hashing is to turn data points into binary-code points from the original space and retain the original space's similarity

(neighbourhood structure) (Jiang and Li, 2015). In this sub-section, the feature of channel gain is extracted by the AGH method. Firstly, the channel gain is defined as $H \in R^{|J|\times|U|}$ and can be expressed mathematically as:

$$H = [h_1(t)\ h_2(t)\ h_U(t)] \tag{4.9}$$

Equation (4.9) belongs to the two-dimensional matrix and cannot directly feed to the input of the other one-dimensional DRL elements. Therefore, the two-dimensional matrix of H is first converted into a one-row vector as a single sample $Z \in R^{|L|}$, where L specifies the extracted channel gain sample elements z. However, the extracted CSI samples are constantly updated at each time slot t, which is challenging for the DRL agent to explore the network space in practice. Therefore, the AGH method limits the extracted channel gain sample and maps AGH to hash code. The hash code specifies discrete characteristics and can easily be matched to other elements in the network state. The AGH uses a small number of anchors n to tie the whole extracted channel gain sample z.

4.3.2 Discretization of State Space

The AGH approximates the data structure of a small set of anchor points n by the neighbours z. In a dynamic environment, it may be challenging to maintain anchor points n, as the sample of z is updated continuously at each time step t. The K-means clustering algorithm is used to avoid this problem with the training sample z. Based on anchor nodes $|N|$, the sample points $|L|$ are partitioned into $|N|$ clusters. The anchor nodes $|N|$ is considered in continuous state-space as $\bar{Z}_n$={1,2, ... $|N|$}, and partition the state-space by the sample points $|L|$ as $\tilde{Z}_l$={1,2, ... $|L|$} . Thus, the state-space for the cluster centroid can

be calculated as follows:

$$\bar{Z} = min\left\{arg\min_{n}|\tilde{z}_l - \bar{z}_n|\,.\right\} \tag{4.10}$$

The proposed structure of CSI discretization based on AGH to hash codes is shown in Figure 4.2. In Equation (4.10), $\tilde{z}_l$ represents the training sample of $\tilde{Z}_l$ in the discrete state-space, while $\bar{z}_n$ denotes the anchor nodes of $\bar{Z}_n$ in the continuous state-space. Equation (4.10) can be transformed into a suitable cluster centroid distribution. Therefore, the predefined elements of K-means clustering are initialized, such as the cluster centroids $|N|$, the distance between the previous centroid and the current centroid as a threshold distance b_d, and the maximum number of iterations $\mathcal{M}$. Once these elements are initialized, the distances of Equation (4.10) are then calculated with each cluster centroid $\bar{z}_n$. If the distance is greater than a threshold distance b_d and the number of current centroids $\bar{z}_n$ elements are less than $|N|$; a new cluster centroid is added to the $\tilde{z}_l$. The process is repeated until $m = \mathcal{M}$ and the distance of $\bar{Z}$ reaches the threshold distance b_d. Consequently, the extracted channel gain $|L|$ training samples are clustered using K-means to attain $|N|(|N| \ll |L|)$ cluster centroids with anchor n as:

$$\tilde{Z} = \left\{\tilde{z}_n \in \mathbb{R}^{(|U|\times|J|)}\right\}_{n=1}^{|N|} \tag{4.11}$$

The K-means approach speeds up clustering and improves the training efficiency. The identified anchor graph in the clusters is then mapped to a hash code by AGH. This hash code is then fed as the input of the DRL agent. Finally, the joint objective function is optimized using a DRL-based algorithm. The process of K-means clustering is described in Table 4.2 Algorithm 4.1.

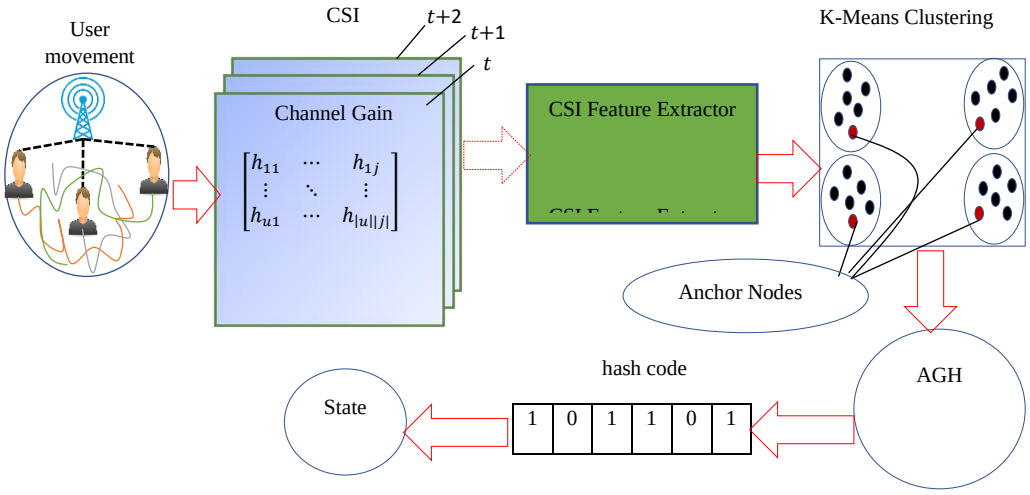


Figure 4.2: CSI discretization based on AGH to hash code

Table 4.2 Algorithm 4.1 K-Means based clustering for the discretization of channel gain

Input: CSI extracted sample

Output: hash code

1: Initialize: $|N|, b_d, \mathcal{M}$

2: Calculate Eq. (4.10) for continuous state-space distance $\bar{Z}$

3: Randomly choose a $\bar{z}_n$ from $\bar{Z}$ as the first centroid $\tilde{z}_0$

4: **while** $l < |L|$

5: Find the continuous state-space $\bar{Z}$

6: Compute the continuous state-space distance between $\bar{z}_n$with each centroid $\tilde{z}_n$

7: **if** $|\tilde{z}_l - \bar{z}_n| > b_d$ and $n > |N|$ **then**

8: Add a new cluster centroid $\tilde{z}_{n+1}$ at the location of $\bar{z}_l$

9: **else**

10: $\tilde{z}_n$ is the cluster centroid of $\bar{z}_n$ from Equation (4.11)

11: Adjust the $\tilde{z}_l$ location till m=$\mathcal{M}$

12: **end if**

13: **end while**

14: Execute the AGH on each centroid $\tilde{z}_n$

4.3.3 Exploration and Exploitation based on Reinforcement Learning

The process of RL is based on the three central elements, i.e., the state space

(s_t), the action space (a_t), and the reward function $\left(K(s_t, a_t)\right)$. These three elements are defined in this chapter as follows:

- **State-space**: At time slot t, the agent needs to know about all data rate demands $D_U(t)$, on/off status of RRHs $v_J(t)$, and constructed hash code. The state-space can be represented mathematically as:

$$s_t=\left[D_1(t), D_2(t), \dots, D_U(t), v_1(t), v_2(t), \dots, v_J(t), hash\ code(t)\right]^T \quad (4.12)$$

  The hash code derived by Equation (4.11) depends on the number of centroids and the length of the hash bits r_b in the channel gain. Thus, a generalized hash function $H_k(\vec{z})$ is used for clustering new channel gain samples with the closest AGH.

- **Action-space**: At each time slot t, the action space specifies the on/off switching decision of RRHs. Whereas the action on any RRHs can be denoted as, $a_J(t) \in \{0,1\}$, $a_J(t)$=0 indicates that RRH is OFF and $a_J(t)$=1 specify that RRH is ON. The action space is obtained based on the exploration and exploitation of the environment, and the action selection function can be represented as follows:

$$a_t = \left[a_1(t), a_2(t), \dots, a_J(t)\right]^T \quad (4.13)$$

- **Reward Function**: The reward function comes from the objective of Equation (4.8) that an agent obtains from the environment after performing an action at time slot t in a particular state. The reward function in this chapter is based on maximizing the joint EE-SE summation at time slot t. Therefore, the reward function can be calculated as:

$$K(t) = [(1\text{-}\alpha)EE(t) + \alpha \frac{B}{p_{j,trans}(t)} SE(t)] \quad (4.14)$$

A Markov decision process (MDP) is typically used to formulate the RL problem. The objective of the MDP is to identify the policy that can map a given state to a given action while maximizing the expected rewards. In order to solve the MDP, Q-learning is a widely used RL approach. In Q-learning, the RL agent chooses an action from a particular state and observes its feedback. The Q-value can be derived using the Bellman equation (O'Donoghue *et al.*, 2018) as:

$$\mathrm{Q}^*(s_t, a_t) = \mathrm{Q}(s_t, a_t) + \gamma[K(t) + \mu \max_{a'} \mathrm{Q}(\mathrm{s}', a') - \mathrm{Q}(\mathrm{s}_t, a_t)] \quad (4.15)$$

In this case, γ and μ stand for the learning rate and discount factor, respectively. At time slot t, the agent should choose an action with the highest Q-value. The results of Q-learning are accumulated in the form of a table known as a Q-table, which is suitable for limited state-action pairings. Putting all Q-values into a computed Q-table will increase data size and becomes very challenging for the RL agent to extract all the state-action values in a reasonable time. In order to overcome this issue, a deep neural network (DNN) is implemented with Q-learning. DQN is a well-known method that represents the state-action space $Q(s_t, a_t) \approx Q^*(s_t, a_t: \theta)$ using a function approximation method instead of calculating all the Q-values into a Q-table. θ represent the weights and biases of the online neural network. A DQN can also evaluate network performance using the target Q-network $Q(s_t, a_t; \theta')$, as shown in Figure 4.3a. More specifically, the online Q-network can be trained at each time step t to minimize the loss function $L(\theta)$ to produce the actual value as:

$$L(\theta) = \left[\left(y^t - Q(s_t, a_t; \theta)\right)^2\right] \quad (4.16)$$

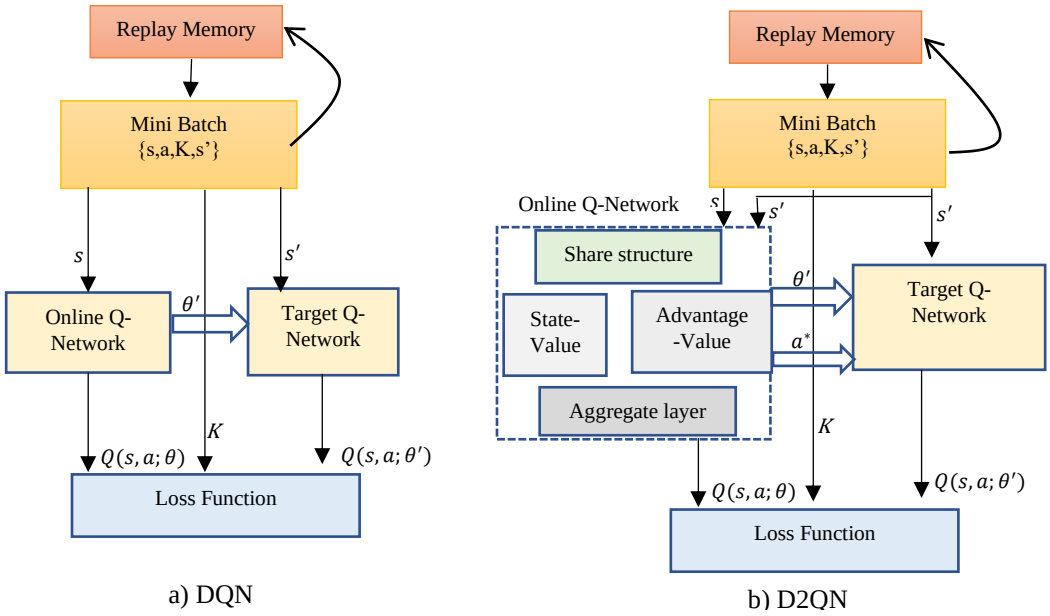


Figure 4.3: Network architecture of DQN and D2QN

whereas the target value y^t can be represented as:

$$y^t = K(t) + \mu \max_{a'} Q(s', a'; \theta') \tag{4.17}$$

The action can be picked from the online network by bypassing the online network parameter with the greedy policy 'ε.' The target network is then copied from the online network in some earlier iteration, reducing the correlation among the training samples (Mnih *et al.*, 2015). Experience replay further strengthens learning stability by storing the transition experiences $e_t = (s_t, a_t, K(t), s')$ into a finite-sized dataset $\mathcal{D}_t = \{e_1, e_2, \dots, e_t\}$, which is randomly sampled by the RL agent to train the neural network. The same max operator function is used to select and evaluate the action for the Q-value in Equation (4.15) and (4.17), causing an overestimation problem for the agent. Therefore, D2QN (Wang *et al.*, 2016) is proposed in highly dynamic scenarios to avoid this issue. The main motivation for using D2QN is to accurately determine which RRHs should be assigned to UEs based on the user's data rate demands. According to (Wang *et al.*, 2016), D2QN ensures higher performance

than conventional DQN methods. As shown in Figure 4.4b, the proposed D2QN approach entails two streams of layers, i.e., state value layer $V(s_t)$ and state-dependent action value layer $A(s_t, a_t)$. These two layers represents the relative advantages of action for a better result and can be combined as follows:

$$Q(s_t, a_t) = V(s_t; \theta, \vartheta) + A(s_t, a_t; \theta, \beta) \tag{4.18}$$

The value layer $V(s; \theta, \vartheta)$ is the scalar function that selects the best action from the given set of RRHs. The advantage layer $A(s_t, a_t; \theta, \beta)$ is the vector of $|A|$-dimensional values for the selected action. ϑ and β stand for the value function and advantage function parameters. The problem of unidentifiability in (Wang *et al.*, 2016) can be addressed by replacing Equation (4.18) as follows:

$$Q(s_t, a_t; \theta, \vartheta, \beta) = V(s_t; \theta, \vartheta) \ + \left(A(s_t, a_t; \theta, \beta) - \frac{1}{|A|} \sum_{a'} A(s_t, a'; \theta, \beta) \right) \tag{4.19}$$

Moreover, the advantage layer helps to improve the network stability by reducing the Q-value range and eliminating the excess degrees of freedom when the state is unchanged.

4.3.4 Power Allocation

As mentioned in Equation (4.5), the active power, sleep power, and transition power are composed of constant values, which can be calculated very easily. Therefore, to minimize the total power consumption $P_{total}(t)$, the selection of transmission power is accounted for each time slot t. Thus, Equation (4.5) can be reduced to a slot-by-slot optimization problem as:

$$\min_{\{w_{j,u}\}} p_{trans}(t) \tag{4.20}$$

$$\text{subject to } \partial_u(t) \geq \rho_u, \quad u \in \mathcal{U} \tag{4.20.1}$$

$$\sum_{u \in \mathcal{U}} |w_{j,u}(t)|^2 \leq P_J, \qquad j \in \mathbb{A} \tag{4.20.2}$$

P_J indicate maximum allowable RRH's transmit power. $\rho_u = \mathcal{I}_m\left(2^{D_u/B}\text{-}1\right)$; Constraints (4.20.1) denote that all user demands must be guaranteed, and constraints (4.20.2) indicate the RRH's transmit power limitations. In Equation (4.20), a convex optimization problem is derived, which can be modified to a second-order cone optimization problem (SOCP) (Wiesel, Eldar, and Shamai, 2006). The optimal solution for Equation (4.20) can be achieved using (Soma *et al.*, 1998). The optimization problem may have no feasible solution at the start of the learning process because there are not enough active RRHs to meet user demands. In this case, the agent becomes more aggressive in turning on more RRHs to satisfy the user demands. A summary of the optimal hyperparameters performance of the D2QN used in this work is shown on Table 4.3, whereas the detailed D2QN pseudocode is presented in Table 4.4 Algorithm 4. 2. Like chapter 3, the computational complexity for Equation (4.20), can be derived from, e.g., see (Ben-Tal, A. and Nemirovski, 2001, Chapter 6). Thus, the worst-case computational complexity of the proposed D2QN algorithm is $\mathcal{O}(J^{3.5}U^{3.5}E + H_z + \mathcal{D} + |\theta|)$, where E represents the number of episodes required to converge Algorithm 4.2. H_z, $\mathcal{D}$ and $|\theta|$ denotes the extracted CSI hash code, the number of experience samples in the replay buffer and the cardinal of weights, respectively. The proposed D2QN algorithm achieves the best network performance results compared to the nature DQN. However, the computational complexity of the proposed D2QN algorithm is higher than the DQN.

Table 4.3: Selection of hyperparameters values for D2QN

Hyperparameters	value
Learning rate $\curlyvee$	10^{-3}

Epoch	10
Training and testing episodes	1000 and 10
Activation function	ReLU
Optimizer	RMSProp (Zoph and Le, 2017)
Number of hidden layers	3
Number of neurons per layer	(32,32,64)
Mini-batch size	512
Discount factor μ	0.995
Experience memory $\mathcal{N}_{\mathcal{D}}$	10000

Table 4.4: Algorithm 4.2 D2QN Based Resource Allocation

1: Initialize Experience memory with a capacity $\mathcal{N}_{\mathcal{D}}$

2: Initialize the online network and target network with weights and biases θ and θ'

3: **for** each decision epoch t, **do:**

4: Received the initial observation of the state s_t

5: Determine the hash code using the general hash function $H_k(\bar{z})$.

6: Find the nearest anchor $(\widetilde{z^*_n})$ to $(\bar{z})$ with respect to the Hamming distance.

7: Feed the generated hash code to the state-space

8: **for** each time slot, **do**:

9: Select a random action with a probability **P**

10: **else:**

11: Select an action $a_t = argmaxQ(s_t, a_t; \theta)$

12: Obtain a set of RRH $\mathcal{J}$

13: Obtain the beamforming solution for a given action a_t

14: Calculate the reward and next state

15: Store $(s_t, a_t, K(t), s')$ into the experience replay buffer

16: Set the mini-batch sample from the Replay buffer

17: Calculate the Q-value for the D2QN

18: Calculate the target Q-value $Q(s_t, a_t; \vartheta, \beta)$

19: Update the main Q-network to minimize the loss function

20: Observe the reward $K(s_t, a_t)$ and next state s'

21: **end for:**

22: **end for**

4.4 RESULTS AND DISCUSSIONS

This section explicitly presents the simulation and performance results of the proposed D2QN algorithm. Mainly, three key performance metrics are used to evaluate the effectiveness of the proposed algorithm: i) convergence performance of the proposed algorithm, ii) considering the hash bits r_b and anchor node n effectiveness iii) performance of joint EE-SE with the satisfaction of UEs. To make a fair comparison for the proposed network performance, Table 4.5 summarizes the simulation parameters setting of the proposed work (Dai and Yu, 2016). In this chapter, the user demand is considered in the range of 10 Mbps to 60 Mbps. The proposed algorithm's performance is compared with Q-learning (Sun, Boateng, Ayepah-Mensah, *et al.*, 2019), without considering CSI generalization and myopic approach. At first, the RL agent was trained for 1000 training episodes to make it aware of the environment. Then, the performances are plotted for 100 testing episodes in the simulation environment with TensorFlow 1.14.0 and python 3.7.5.

Table 4.5: Simulation Parameters Setting

Symbol	Parameter	Value
$p_{j,act}$	Active power	6.8 W

$p_{j,slp}$	Sleep power	4.3 W
$p_{j,trans}$	Transmit power	1 W
$p_{j,tp}$	Transition power	3 W
B	Bandwidth	10 MHz
$\varsigma_{j,u}$	Shadowing coefficient	8 dB
η	Power amplifier efficiency	25 %
σ^2	Noise power	-102 dBm
$PL(d_{j,u})$	Pathloss with a distance of (km)	$148.1+37.6\log_2 d_{j,u} dB$
$\mathcal{I}_m$	Capacity gap	1
$\mathfrak{y}_{j,u}$	Small scale fading	$\mathcal{CN}(0,1)$
$\varphi_{j,u}$	Antenna Gain	9 dBi
L	Training sample	50000
n	anchors	10
Z	Nearest number of anchors to matrix	2
r_b	Length of hash bits	9

4.4.1 Convergence Performance

The weighted EE-SE convergence performance can be seen in Figure 4.4, and it can be observed that both DQN and D2QN have the same weighted EE-SE performance at the start of learning. The term “Proposed solution” will be used hereafter instead of the Proposed solution (D2QN CSI_Feature_Extractor). With the increasing number of episodes, the Proposed solution yields better weighted EE-SE performance than the DQN, as demonstrated in Figure 4.4. The DQN algorithm makes a slow adaptive control switching decision because the learning agent requires massive efforts to learn the optimal CSI feature for data exploration to approximate the Q-value function. In other words, the DQN

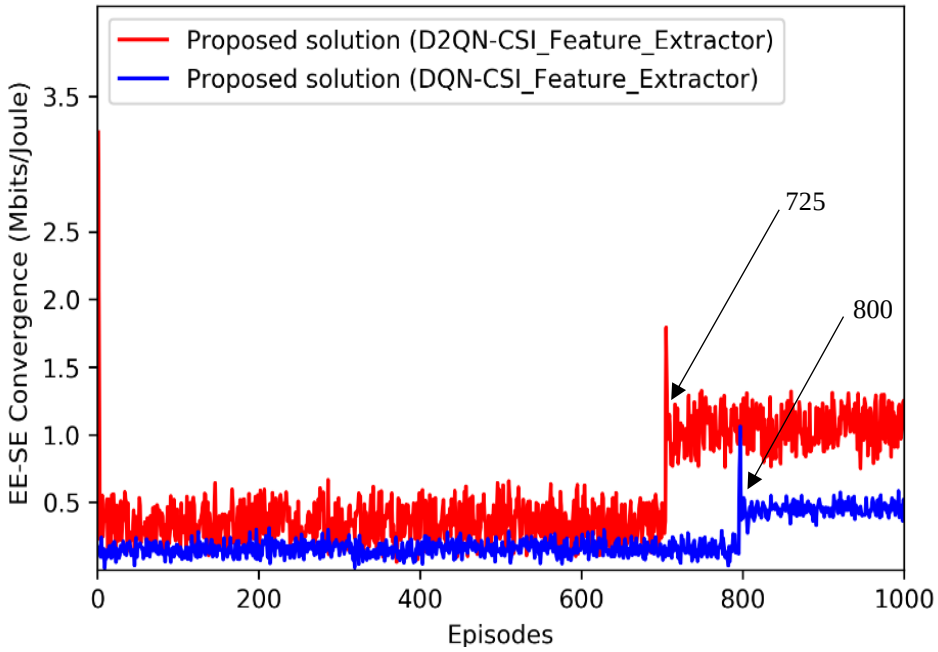


Figure 4.4: CSI Feature Extractor convergence performance for proposed DQN and D2QN

algorithm is not able to extract the optimal CSI feature due to a lack of data exploration. Therefore, the DQN starts convergence after reaching 800 episodes. Conversely, after every few episodes, the proposed D2QN algorithm takes greedy actions to determine the optimal CSI features to approximate the Q-value function. However, once the number of episodes approaches 725, the Proposed solution starts the convergence performance regarding the weighted EE-SE, as illustrated in Figure 4.4. It can be concluded from Figure 4.4 that the Proposed solution achieves faster convergence and improves the learning performance than the DQN.

4.4.2 Hash Bits and Anchors Effectiveness

The joint EE-SE performance against the growing user demand is plotted in Figure 4.5 for different anchor n values within the Hamming radius of 2. The joint EE-SE performance decreases significantly with increasing anchor n value, as shown in Figure 4.5. The larger the cluster radius, the wider the state-

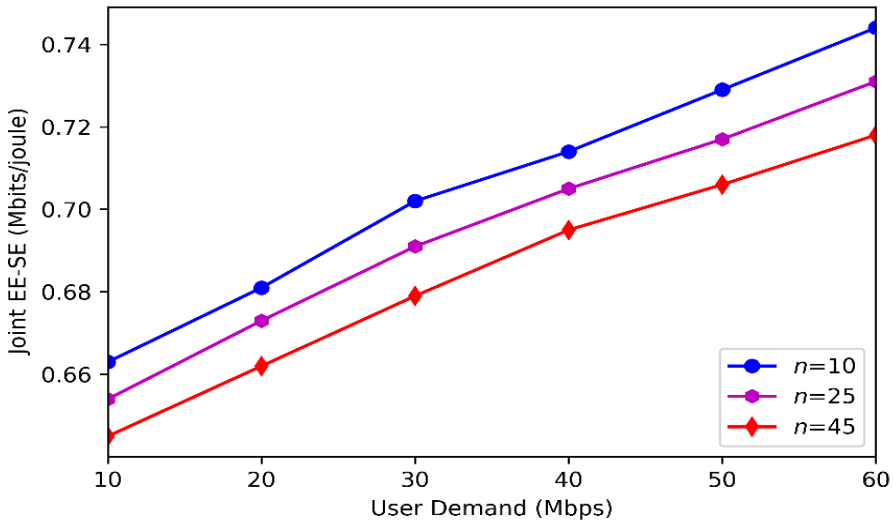


Figure 4.5: Effect of anchor n value on EE-SE performance against user demand

space becomes for Equation (4.12). It means that the agent has to exert more effort to handle such an enormous state space. From Figure 4.5, one can see that when $n = 10$, it provides 8-14% better EE-SE performance than $n = 25$. On the other hand, for $n = 25$, the joint EE-SE performance is 10-15% better than for $n = 50$. The impact of hash bits r_b on the average EE-SE performance against the user demands, is shown in Figure 4.6. From Figure 4.6, one can observe that larger values of n are associated with higher power consumption

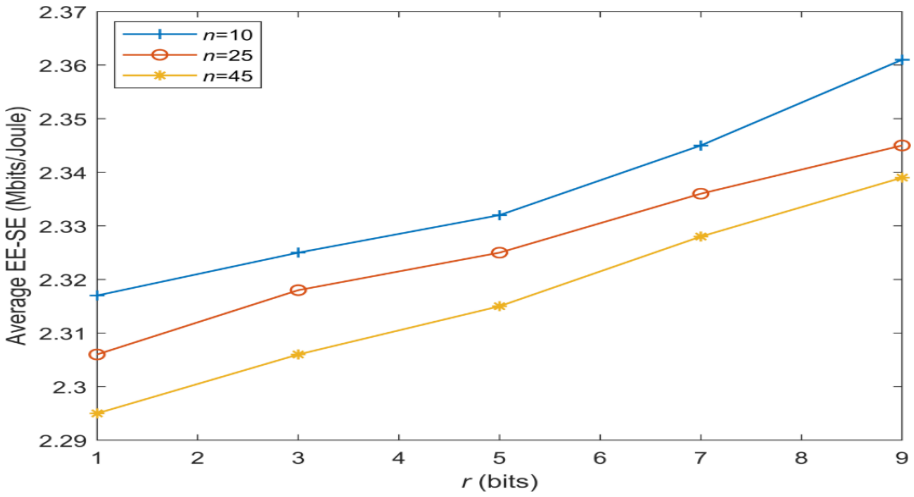


Figure 4.6: Effect of hash bits r_b value on EE-SE performance against user demand

due to the lengthening of the hash code for Equation (4.12). This means that the larger value of n has a direct relationship to increasing the values of r_b. The performance of the average EE-SE is linearly increasing with r_b. This is due to the smaller difference between the data sample z and n. The learning agent requires less effort to extract the required CSI feature. It is important to note that the value of r_b must be less than the value of n. It can be concluded from Figure 4.5 and Figure 4.6 that the inaccurate values of r_b and n will decrease the joint EE-SE performance. Therefore, it is important to find the optimal values of r_b and n to improve the network performance. In this chapter, these two values are considered as $r_b = 9$ and $n = 10$.

4.4.3 Joint Performance of Weighted EE-SE

The predicted values of n and r_b are then used to optimize the joint EE-SE performance. In this section, two scenarios are analyzed to examine the performance of joint EE-SE, i.e., 1) $J = 4,\ U = 2$ and 2) $J = 12\ , U = 4$. The proposed solution is compared with Q-Learning (Sun, Boateng, Ayepah-Mensah, *et al.*, 2019) without CSI generalization and myopic approach (Dai and Yu, 2016). In the myopic approach, the main focus is on the immediate reward value taken from the action and ignores its impact on future values. As a result, a channel is always selected in which the maximum immediate reward value is obtained. The myopic approach has no performance guarantee, especially if a channel becomes correlated. In order to maximize the long-term performance of the EE-SE, Q-learning is first considered. As shown in Figure 4.7, the performance of Q-learning is superior to the myopic approach. This is because Q-learning attempts to maximize the network's performance by using past

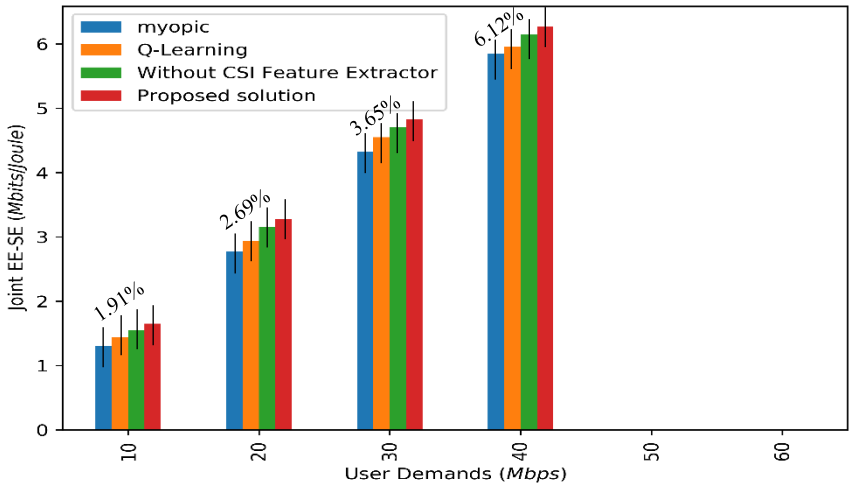


Figure 4.7: joint EE-SE performance vs user demand for $J = 4$, $U = 2$

learning experiences. However, such method becomes unstable, especially when the value of CSI is changing continuously in the state space. The D2QN method is then used to approximate the value of state-space and name it without a CSI Feature Extractor so that the agent tries to maximize its expected future reward. Ideally, the RL agent will achieve the optimal solution for a large state-space value. However, due to the continuous state-space explosion, the dimensionality problem occurs. The performance of joint EE-SE degrades if the state-space is not discretized, as shown in Figure 4.7. This motivates the need for to a D2QN based solution with extracted CSI generalization by using AGH approach. The Proposed solution outperforms the other three approaches as shown in Figure 4.7. However, no matter which method is utilized, the joint performance of EE-SE increases linearly as the user demand increases. Figure 4.7 shows that the Proposed solution achieves 6-10% better performance at each step increasing user demand. However, once the user demand exceeds 40Mbps, each of the four approaches becomes unsustainable. The reason is that all these approaches require more transmission power, which results in higher joint EE-

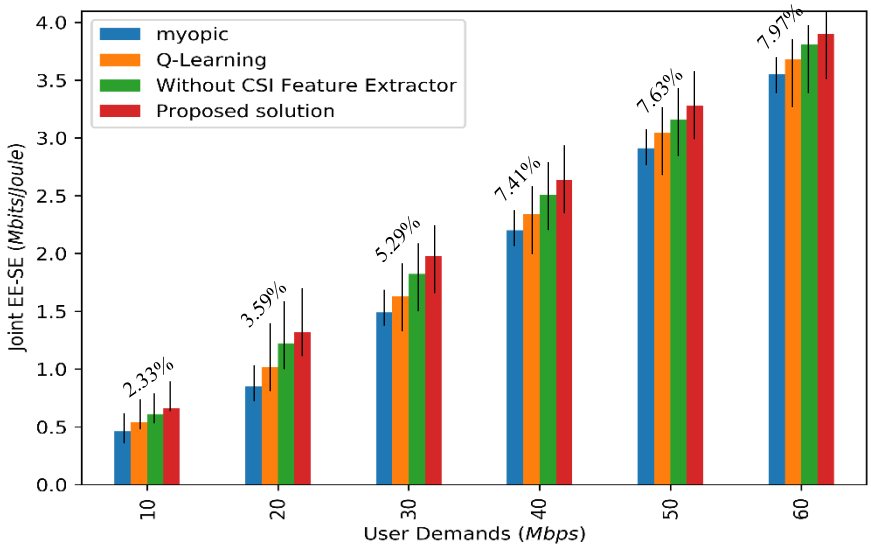


Figure 4.8: joint EE-SE performance vs user demand for $J = 12$, $U = 4$

SE performance. In order to avoid this issue, the number of RRHs and users are increased to $J = 12$ and UEs $U = 4$, as shown in Figure 4.8. Even with the increasing number of users and RRHs, the Proposed solution still achieves better performance by 5% -12% than other approaches. This improvement demonstrates that the CSI generalization is more effective than the baseline approaches in terms of system performance.

Lastly, the performance of the average EE-SE for varying values of α is presented in Figure 4.9. The performance of the average EE-SE decreases for all four approaches as the value of α increases. This is because the difference between EE-SE will be lessened as the α value increases. Compared to the D2QN without a CSI Feature Extractor, Q-Learning, and myopic approaches, the Proposed solution achieves 4.7%, 5.3%, and 6.4% better performance, respectively, by assuming $\alpha = 0$. Similar results are observed when $\alpha = 1$; the average EE-SE performance drops to 2.85%. Despite this drop, the Proposed

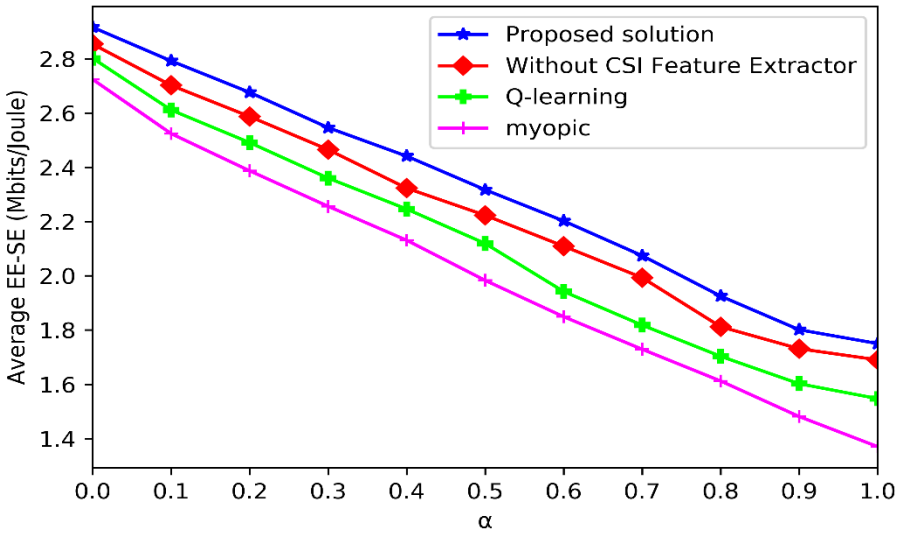


Figure 4.9: Tuning parameter effects on Average EE-SE performance

solution still outperforms the D2QN without CSI Feature Extractor, Q-Learning, and myopic approaches. This is because the average EE-SE performance reflects the fact that the total power of RRH increases as the α value increases

4.5 SUMMARY

In this chapter, the RA scheme based on the D2QN is proposed in order to maximize the joint tradeoff between EE-SE and satisfies the user's QoS requirements in CRAN. In particular, the machine learning technique based on AGH is used to limit the dynamic feature of CSI and then feed to the input of DRL. The near-optimal control strategy is used for turning on and off RRHs to maximize the joint EE-SE performance and meet the QoS requirement for users using a D2QN-based approach. Lastly, the D2QN-based AGH method is examined to improve the EE-SE performance by comparing without the CSI Feature Extractor, Q-learning (Sun, Boateng, Ayepah-Mensah, *et al.*, 2019), and myopic approach (Dai and Yu, 2016). It is shown that the proposed D2QN-

based AGH method improves the performance of joint EE-SE. Furthermore, based on simulation results, the proposed solution is more effective than baseline approaches to improve network performance, learning performance, and convergence speed.

CHAPTER 5

RESOURCE MANAGEMENT FOR CLOUD RAN USING CONVOLUTIONAL NEURAL NETWORKS BASED ON DEEP Q-NETWORK

In this chapter, the convolutional neural networks-based deep Q-networks (CNN-DQN) is introduced in order to balance the energy consumption and ensure the user's quality of service (QoS) demand in downlink cloud radio access networks (CRAN). After formulating the Markov decision process (MDP) for maintaining energy efficiency (EE), a three-layer CNN is proposed to represent the environment features as input state spaces. Deep Q-network (DQN) is then implemented to dynamically control the status of RRHs based on the user's QoS demand and energy consumption in the CRAN. Finally, the resource allocation (RA) problem is solved based on transmitted power and user demand constraints to fulfil the QoS demands and maximize EE. This chapter is concluded with a simulation study, demonstrating how the proposed scheme performs in terms of EE, power savings, and user satisfaction.

5.1 INTRODUCTION

As the number of mobile subscribers has grown exponentially over the past two decades, user data traffic has also grown exponentially. Cisco 2020 predicts that mobile subscribers will reach 5.7 billion by 2023 and that data traffic will reach 110 exabytes (EB) (Cisco, 2020). Therefore, it is necessary to install a large number of base stations (BSs) within the coverage area in order to fulfill the

above requirements but installing more BSs will increase infrastructure costs and energy consumption. About 60-75% of the total energy consumption in the cellular network is consumed by the BSs (O *et al.*, 2017). Therefore, when the number of users is low, the BSs need to be dynamically turned off to ensure low energy consumption.

Currently, the capacity of the existing radio access network (RAN) is limited by the remote resource management among BSs. Network densification is one way to increase the existing RAN framework's capacity. As a consequence of such processes, capital and operational costs are increased (CAPEX and OPEX), and existing RAN frameworks cannot support the ever-increasing user demands and mobile subscribers (Yadav and Dobre, 2018).

Using cloud radio access networks (CRANs) is a promising technology to address all of the above difficulties and provide fast, reliable, and scalable real-time communication for next-generation networks (Checko *et al.*, 2015). The main idea behind CRANs is to separate the BS functionality into distributed low-cost, low-power remote radio heads (RRHs) and a centralized baseband unit (BBU). The RRHs are responsible for transceiving the radio signal, and the BBU leads to the signal processing functions. As a result of centralized processing, the CRAN assigns radio resource knowledge to RRHs based on user demand and mobility. Although the CRAN has very significant implications for the upcoming wireless network era, adaptively solving the RA problem remains a topic of research.

The RA problem in the CRAN has been studied extensively from several perspectives, such as EE (Tham *et al.*, 2017), throughput (Ali *et al.*, 2017), and

transmission power (Dhif-Allah *et al.*, 2018). However, most of these studies follow the traditional model-based approach with a static network environment. This approach becomes impractical, especially if the network state is affected by user mobility at each time step t. Therefore, this chapter investigates a model-free approach to solving the RA problem throughout the entire operational period in real-time.

Reinforcement learning (RL) is a model-free machine-learning (ML) approach in which the learning agent interacts continuously with an unknown environment to apply its knowledge to a complex decision-making problem (Sutton and Barto, 2018). The learning agents select the actions from each state and then use the available data to train the model to make decisions at each time step t. Deep learning (DL) has been successfully applied in many fields recently, i.e., speech recognition, natural language processing, image processing, and computer vision (CV). DL has also been utilized in wireless communication to learn the sequential control task to aid the RL algorithm (Cheng *et al.*, 2017). Convolutional neural networks (CNN) advance the DL method that can extract more complex dynamic features in mobility scenarios (Lecun, Bengio, and Hinton, 2015). Many existing works define the state of the wireless network as the user demand and RRHs without considering their relationship with each other (Xu *et al.*, 2017) and (Zhao *et al.*, 2019). A major disadvantage of these works is the requirement that users report their information to their respective RRHs, thereby increasing the overhead associated with signaling. If such information exists between the users and RRHs, then RRHs should store all valid information. Thus, users need not provide any such information for signaling. This process reduces the signaling

burden in the network. Furthermore, the works discussed in (Xu *et al.*, 2017) and (Zhao *et al.*, 2019) exploit fully connected layers to train neural networks (NNs) as opposed to convolutional layers, thereby substantially increasing training parameters (Lee, Kim and Cho, 2018). The above drawback provides a reason for considering the relationship between RRHs and users as raw observations and proposing a three-layer CNN-based deep Q-Network (CNN-DQN) to capture the random state features in the environment. Furthermore, this chapter combines the CNN and DQN schemes for extracting information from users and RRHs in the input of the network state. According to this study, the CNN phase is responsible for feature extraction, while the DQN phase is responsible for dynamically turning on and off the RRHs.

The rest of the chapter is structured as follows. Section 5.2 illustrates the system model and the power consumption function. Section 5.3 describes the proposed method along with the 3-layer CNN phase. Section 5.4 presents simulation details and results, followed by a conclusion in Section 5.5. In Table 5.1, a list of the mathematic notations that are used in this chapter is presented.

Table 5.1: List of Key Notations

Notations	Description
$\mathcal{R}$	Set of RRHs
$\mathcal{U}$	Set of UEs
$\mathbb{T}$	Time-period
D_u	Data rate demand
PL	Path loss

$d_{j,u}$	Distance between RRHs and UEs
$\zeta_{r,u}$	Antenna gain
$\rho_{r,u}$	Shadowing coefficient
$\omega_{r,u}$	Small-scale fading
∂_u	Signal-to-interference-plus-noise ratio
h_u	Channel gain
w_u	Beamforming weight
W	Bandwidth
$\mathcal{J}_m$	Capacity gap
C_u	Data rate
δ^2	Noise power
$p_{r,T}$	Transmission power
τ	Power amplifier
$p_{r,A}$	Active power
$p_{r,S}$	Sleep power
$p_{r,\mathcal{G}}$	Transition power
$\mathcal{S}$	Set of sleep mode of RRHs
$\mathcal{A}$	Set of an active mode of RRHs
N	MDP tuple
S	Discrete state-space
A	Discrete action space

$P(s', k\|s, a)$	Transition probability
$K(s, a)$	Reward
$V_\pi(\mathrm{s})$	State value function
μ	Discount factor
$\mathcal{D}_t$	Experience Replay Memory
π	Policy
γ	Learning rate
M	Input matrix
O	Convolutional output filter
$\mathbb{I}$	Input-size
$\mathbb{K}$	Kernel-size
$\mathbb{P}$	Padding
$\mathbb{S}$	Stride
β	Dropout probability
$\mathcal{D}$	Experience replay

5.2 SYSTEM MODEL

Like previous chapters, the downlink CRAN is considered in this chapter which is composed of a single BBU, set of RRHs and set of UEs and denoted as $\mathcal{R} = \{1,2,\ldots R\}$, and $\mathcal{U} = \{1,2,\ldots U\}$, respectively as shown in Figure 5.1. A time period $\mathbb{T} = \{1,2,\ldots T\}$ is also assumed in this chapter. The UEs change their position randomly and report their user data rate demand $D_u \in [D_{min,} D_{max}]$ and

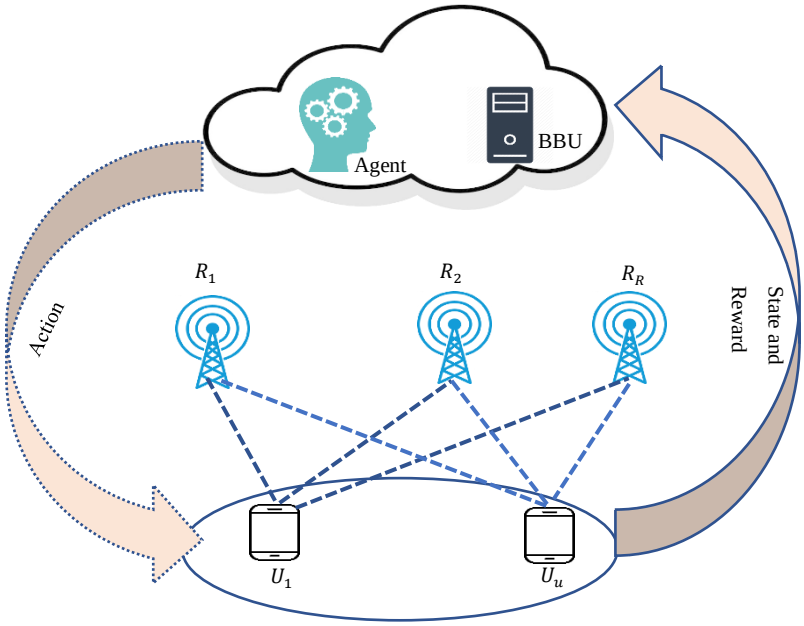


Figure 5.1: CRAN resource allocation under DRL framework

channel state information (CSI) to the BBU pool. The BBU pool is assumed to serve as the RL agent. Each RRH and the UE are equipped with a single antenna without loss of generality. Furthermore, it is also assumed that users can access all the RRHs and that the RRHs are connected to the centralized BBU pool. As a result, all the information is processed in a centralized manner. The path loss is then determined based on (Dai and Yu, 2016):

$$PL(d_{r,u}) = 148.1 + 37.6 \log_2 d_{r,u}\ dB \tag{5.1}$$

where the distance between RRHs and users is represented by $d_{r,u}$. The channel fading model is considered from the previous work (Shi, Zhang, and Letaief, 2015):

$$h_{r,u}(t) = 10^{-PL(d_{r,u})/20} \sqrt{\zeta_{r,u}\, \rho_{r,u}\, \omega_{r,u}} \tag{5.2}$$

The $\zeta_{r,u}$, $\omega_{r,u}$ and $\rho_{r,u}$ denotes the antenna gain, small-scale fading, and shadowing coefficient, respectively. The rayleigh channel fading model is considered in this work; where $\omega_{r,u}$ is the independent and identically

distributed (i.i.d) complex Gaussian random variable that captures the small-scale fading effects associated with a radio link between RRH and UE. The signal-to-interference plus noise (SINR) ∂_u received by the users u at time t can be expressed as follows:

$$\partial_u(t) = \frac{|h_u^H(t)w_u(t)|^2}{\delta^2 + \sum_{v \neq u}|h_v^H(t)w_u(t)|^2}, u \in \mathcal{U} \tag{5.3}$$

whereas δ^2 indicates the background noise. $h_u(t)$ and $w_u(t)$ represents the channel gain and beamforming weight between RRHs and users at time t and can be expressed as $h_u(t) = \left[h_{1,u}(t), h_{2,u}(t), \cdots h_{R,u}(t)\right]^T$ and $w_u(t) = \left[w_{1,u}(t), w_{2,u}(t) \cdots w_{R,u}(t)\right]^T$, respectively. Finally, the data rate achieved by the user at time t is given as:

$$C_u(t) = W\, log_2\left(1 + \frac{\partial_u(t)}{\mathcal{J}_m}\right), u \in \mathcal{U} \tag{5.4}$$

The W and $\mathcal{J}_m$ specifications describe the channel bandwidth and SNR gap, respectively. The $\mathcal{J}_m$ depends on the modulation scheme. In this chapter, $\mathcal{J}_m$ is assumed to be 1.

5.2.1 Power Consumption Model

According to (Auer *et al.*, 2012), the relationship between BS power consumption and transmit power can be approximated linearly. Therefore, for each RRH, a linear power model is applied as:

$$p_r = \begin{cases} \frac{1}{\tau}p_{r,T} + p_{r,A} & ; r \in \mathcal{A} \\ p_{r,S} & ; r \in \mathcal{S} \end{cases} \tag{5.5}$$

τ indicates the drain efficiency of the power amplifier. $p_{r,T} = \sum_{r \in \mathcal{A}} \sum_{u \in \mathcal{U}} \left|w_{r,u}\right|^2$ signifies the RRHs r transmit power; $p_{r,A}$ is the power

consumption of active RRH r without transmitting signals. When no transmission is necessary, the RRHs r can be set to sleep mode as $p_{r,s}$. $\mathcal{S}$ and $\mathcal{A}$ represent sleep and active modes of RRHs, respectively. Thus, one has $\mathcal{A}\ U\ \mathcal{S} = \mathcal{R}$.

Furthermore, most of the works, e.g. (Shi, Zhang and Letaief, 2015), (Dai and Yu, 2016), and (Gerasimenko *et al.*, 2015), have ignored the transition power to calculate the total power consumption, which is a change mode power of the RRH states. Transition power is essential to be considered in the power minimization framework, as shown in (Xu, Lin and Zhong, 2014) and (Xu *et al.*, 2015). Therefore, in this chapter, the transition power is also considered, denoted as $p_{r,\mathcal{G}}$. $\mathcal{G}$ stands for the set of transition mode of RRH. Therefore, the total power consumption $P_{total}(t)$ of all RRHs at time slot t can be expressed mathematically as:

$$P_{total}(t) = \underbrace{\sum_{r\in\mathcal{A}} p_{r,A} + \sum_{r\in\mathcal{S}} p_{r,S}}_{\text{State power}} + \underbrace{\sum_{r\in\mathcal{A}}\sum_{u\in\mathcal{U}} \frac{1}{\tau}\left|w_{r,u}\right|^2}_{\text{Transmit power}} + \underbrace{\sum_{r\in\mathcal{G}} p_{r,\mathcal{G}}}_{\text{Transition power}} \tag{5.6}$$

5.2.2 Problem Formulation

This chapter adjusts the transmission power per RRH and the user data rate to maximize the EE. The EE is defined as the ratio between the sum of throughput and total power consumption at time slot t. Therefore, the EE is treated as an objective function. The EE can be expressed mathematically as:

$$EE(t) = \frac{\sum_{u=1}^{U} C_u(t)}{P_{total}(t)} \tag{5.7}$$

Furthermore, the EE optimization problem is formulated as:

$$max \sum_{t=1}^{T} EE(t) \tag{5.8}$$

$$\text{subject to } D_u(t) \leq W \log_2\left(1 + \frac{\partial_u(t)}{\mathcal{J}_m}\right), \forall u \in \mathcal{U}, t \in \mathbb{T} \tag{5.8.1}$$

$$\sum_{\mathrm{u}\in\mathcal{U}} \left|\mathrm{w}_{\mathrm{r,u}}\right|^2 \leq P_{\mathrm{r}}, \qquad \forall r \in \mathcal{A},\ t \in \mathbb{T} \tag{5.8.2}$$

Constraint (5.8.1) states that each user's target data rate must be less than or equal to the achievable data rate. In contrast, constraint (5.8.2) indicates that the user's transmit power must not exceed the maximum transmit power.

5.3 CONVOLUTIONAL NEURAL NETWORK-BASED RESOURCE ALLOCATION OPTIMIZATION

This section is divided into the three-sub sections. First, the basics of RL are explained to understand the flow of the mechanism. Secondly, the network state feature is extracted by the CNN before feeding to the input of the DRL agent. Finally, the power allocation optimization problem is formulated.

5.3.1 Basic of Reinforcement Learning Components

RL is a powerful artificial intelligence (AI) technique in which an agent interacts solely with an unknown environment to monitor the current state and map the situation to maximize the reward value. RL generally follows the Markov decision process (MDP) framework to model the complex decision-making problem. The MDP is represented as a tuple of $N = \big(S, A, K(s,a), P(s', k|s,a)\big)$. The state and action space are represented by S and A, respectively. The reward function is designated by $K(s,a)$. Similarly, the agent moves from the current state $s \in S$ to the next state $s' \in S$ to execute a

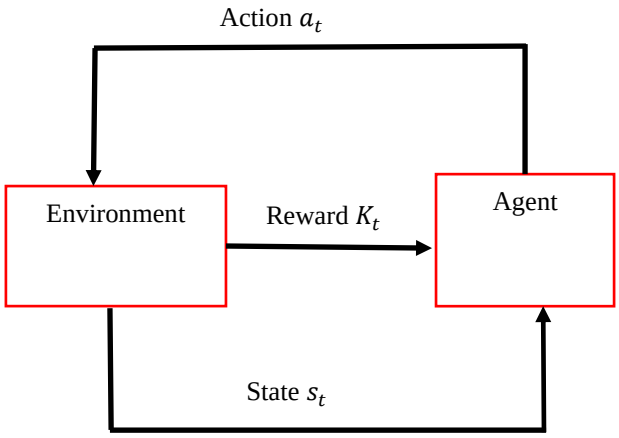


Figure 5.2: Reinforcement learning basic components and agent-environment interaction

cumulative reward with a certain probability known as transition probability $P(s', k|s, a)$. As shown in Figure 5.2, the agent observes the current state of the network at each time step t and executes an action. After executing the action, the feedback is obtained in the form of a scalar reward from the environment. The objective of the agent is to find the near-optimal control policy $a = \pi^*(s)$ that maximizes the reward function value over time. In order to calculate the average cumulative reward function, the state value function $V_\pi(\mathrm{s}) = \mathbb{E}_\pi\{\sum_{i=0}^{\infty} \mu^i k_{t+i+1}|s_t = s\}$ is introduced. The state value function $V_\pi(\mathrm{s})$ uses the recursive relationship based on the Bellman equation as follows:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',k} P(s', k|s, a)[K + \mu V_\pi(s')] \tag{5.9}$$

where μ denotes the importance of future rewards versus current rewards and is known as the discount factor. According to (Mnih *et al.*, 2015), dynamic programming and Q-learning are the two approaches to solving the MDP framework. Dynamic programming is the most commonly used method in the model-based approach since the state transition probability is already known. However, in a complex environment, i.e., 5G and beyond, the state transition probability will change at each time step t. Therefore, the dynamic

programming approach cannot be used to resolve the complex decision-making problem. To conclude the MDP framework, an unknown state transition probability is considered in this chapter. The unknown state transition probability is solved using the model-free Q-learning approach.

5.3.2 Q-Learning Approach

This sub-section describes how Q-learning can solve unknown state transition problems based on the temporal difference method. To learn about Q-learning, first, the concept of the Q-value function is examined, known as the state-action value function and represented as $Q(s,a) = \mathbb{E}_{\pi}\{\sum_{i}^{\infty} \mu^{i} K_{t+i+1} | s_t{=}s, a_t{=}a\}$. The numerical representation of the optimal Q-function is expressed as $Q^{*}(s,a) = max_{\pi}\, Q(s,a)$. The Bellman equation can be used for getting the optimal Q-function and can be expressed mathematically as:

$$Q^{*}(s,a) = \sum_{s',K} P(s',K|s,a)\left[K + \mu \max_{a'} Q^{*}(s',a')\right] \tag{5.10}$$

The action selection in Q-learning relies on ϵ-greedy exploration, where the agent chooses the random action with a probability of ϵ and the greedy action with a probability of $1-\epsilon$. Initially, the Q-value is initialized with the state and action values. It is then updated iteratively as the action selection is evolved as:

$$Q(s,a) \leftarrow Q(s,a) + \gamma\left[K + \mu \max_{a'} Q(s',a') - Q(s,a)\right] \tag{5.11}$$

where γ is the configurable hyperparameters known as learning rates and specifically used in the training of neural networks. The γ is usually in the range of 0.0 and 1.0. From Equation (5.11), it is clear that the state and action values are stored in the form of Q-tables, which work well for a limited state-action dimension. However, it can be a problem for Q-learning to keep all the state-

action values in the lookup table in a complex network (5G and beyond) since the state-action value increases exponentially.

5.3.3 Deep Q-Network Learning

To avoid the dimensionality problem, a linear function approximation method is proposed to approximate the Q-value function. However, such a method cannot estimate the Q-value function accurately. DRL then solves this problem with a neural network called a deep neural network (DNN). The main idea of DNN is to approximate the Q-value function using a non-linear function. Deep Q-network (DQN) is a widely used DRL algorithm proposed for various applications. A separate target network and an experience replay $\mathcal{D}$ function are added to the dataset in the DQN, which helps reduce the correlations between data and makes the system more stable (Fan *et al.*, 2019). In the DQN, the learning agent collects all the information and then applies this information to train the policy (offline) in its background. Thus, the DQN makes all the decisions efficiently and timely based on the already learned policy. In the DQN, the state-action value function $Q(s,a)$ can be expressed as $K + \mu Q^*(\mathrm{s}', a')$. The loss function is then calculated as:

$$L(\theta) = \mathbb{E}\left[\left(y^{target} - Q(s,a;\theta)\right)^2\right] \tag{5.12}$$

where

$$y^{target} = K + \mu \max_{a'}(s', a'; \theta') \tag{5.13}$$

θ and θ' indicate the weights of the evaluated and target networks, respectively. Then, these weights are optimized by using the stochastic gradient descent algorithm (Bottou, 2012).

5.3.4 Convolutional Neural Network-Based Proposed Scheme

In the presence of random user movements at each time step t, the state space dimension exponentially grows. The explosive growth of the state space makes it difficult for the DQN agent to render all the information in a reasonable time. Therefore, this chapter proposes a relational CNN-DQN algorithm, which breaks the state space dimensionality issue and results in the optimal control policy on the RRHs on/off switching. Three hidden convolutional layers are proposed based on the dynamic network state-space feature. Each hidden layer contains 32,32, and 64 convolution filters with a $M \times M$ input matrix, respectively. For each convolutional filter, Xavier's normal initializer is used to initialize the coefficient of channel gain (H. Ide and T. Kurita, 2017). The output of the convolution filter is calculated as follows:

$$O = \frac{\mathrm{I} - \mathbb{K} + 2\mathbb{P}}{\mathbb{S}} + 1 \tag{5.14}$$

where O is the output of the convolutional filter and I, $\mathbb{K}$, $\mathbb{P}$, and $\mathbb{S}$ are the input matrix size, kernel (filter size), padding, and stride, respectively. For simplicity, the kernel size of all hidden layers is assumed to be 2×2, and the padding value and stride value are 0 and 1, respectively. Moreover, the activation function is used as a rectified linear unit (ReLU) for all hidden layers. The proposed CNN comprises a convolutional layer, max-pooling layers, flatten layers, and fully connected layers, as shown in Figure 5.3. The environment state-space features are extracted from the convolutional layers. The pooling layers are used to down sample the extracted features. Finally, a max filter is applied to output the maximum value of a particular region. The output of the last max-pooling layer is dropped out with a probability of $\beta = 0.25$. The last max-pooling layer is

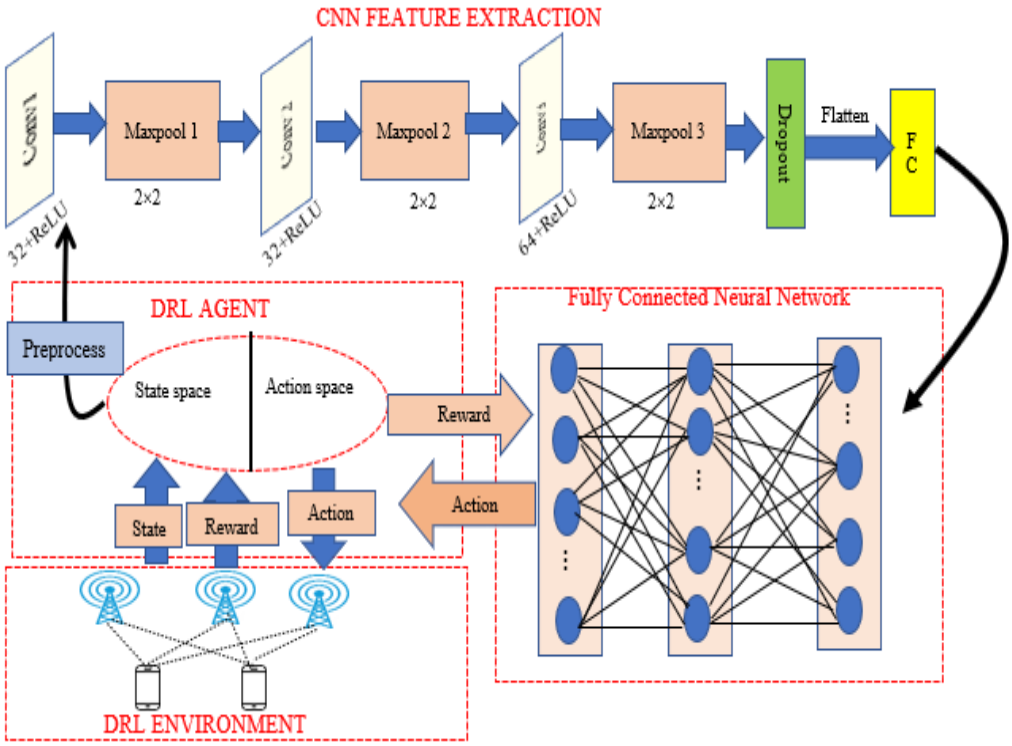


Figure 5.3: Proposed CNN based DQN Framework

flattened into a one-dimensional vector, which is then connected to 100×1 of fully connected (FC) neural networks (NN). Accordingly, the extracted state feature of CNN is fed into DQN to perform the on/off RRHs switch decision. The training process is then performed by the DQN algorithm, as shown in 5.3. In this chapter, the state space $s(t)$, the action space $a(t)$, and the reward function $K(t)$ are defined as follows:

- **State-space:**

During each time step t, the state features are captured, which contains user demand $D_u(t)$, the RRHs on/off state $v_r(t)$ and relational matrix between RRHs r and users u denoting as $H \in R^{U \times R}$. The relational matrix can be expressed mathematically as:

$$H(t) = \begin{bmatrix} h_{11} & \cdots & h_{1r} \\ \vdots & \ddots & \vdots \\ h_{u1} & \cdots & h_{UR} \end{bmatrix} \quad (5.15)$$

The h_{UR} represents the coefficient of channel gain between RRHs R and users U. Finally, these three features are concatenated, i.e., user demand, state of RRHs, and channel gain, into a single vector. Thus, the state-space becomes as:

$$s(t) = [D_u(t), v_r(t), H(t)]^T \tag{5.16}$$

- **Action space:**

The action is defined based on the on/off state of RRHs at each time slot t. The action space can be expressed as $a_r(t) \in \{0,1\}$. However, the RL agent is only allowed to choose the action based on the active set of RRHs $\mathcal{A}$ in this work.

- **Reward Function:**

According to the proposed framework model, the reward function $K(t)$ describes whether to punish or encourage the RL agent based on their behavior. The reward function is actually the objective function defined in Equation (5.7) that shows the improvement in EE and can be written as follows:

$$K(t) = EE(t) = \frac{\sum_{u=1}^{U} C_u(t)}{P_{total}(t)} \tag{5.17}$$

5.3.5 Resource Allocation Optimization

Referring to Equation (5.6), three kinds of power are considered, i.e., state power, transition power, and transmit power. The state power and transition power are based on the current state and action and can be easily calculated. To minimize the total power consumption of Equation (5.6), it is necessary to minimize the transmit power at each time step t. The transmit power relies on allocating beamforming weights in the active set of RRHs $\mathcal{A}$. Thus, the optimization problem is expressed as:

$$\min_{w_{r,u}} p_{r,T} \tag{5.18}$$

$$\text{subject to } SINR_u(t) \leq C_u(t)\ \forall u \in \mathcal{U} \tag{5.18.1}$$

$$\sum_{u \in \mathcal{U}} \left|w_{r,u}\right|^2 \leq P_{r,t}\ ,\ \forall\ r \in \mathcal{A} \tag{5.18.2}$$

The objective is to obtain a minimum total transmitting power based on the state of RRHs, and the user demands. Here user demand represents the requested transmission rate for each user. The variables $w_{r,u}$ are distributed weights corresponding to beamforming power. Furthermore, the $SINR_u(t) = \mathcal{J}_m(2^{\frac{D_u}{W}} - 1)$; whereas $P_{r,t}$ represents the constraint of maximum RRHs transmit power. Also, constraint (5.18.1) assures that all user's demands will be met, while constraint (5.18.2) limits the transmission power in each RRH. The problem explained in (5.18) belongs to the convex optimization problem and can be modified to become the second-order cone optimization problem (Wiesel, Eldar and Shamai, 2006). An iterative approach can be used to solve this problem (Soma *et al.*, 1998). There may be no feasible solutions at the start for the beamforming optimization, which means that more RRHs would need to be activated to meet the user demands. Thus, the RL agent would get negative rewards and would be out of the training loop. The detailed process is summarized in Table 5.2 Algorithm 5.1.

Table 5.2: Algorithm 5.1 CNN-Based DQN Framework

1: Initialize the experience memory $\mathcal{D}$ with the capacity
2: Initialize the weights and biases for the main and target network (θ and θ')
3:**for** each episode, **do**:
4: Observe the initial state $s(t)$
5: Extract the state feature $\emptyset(t)$ using CNN
6: Feed the extracted feature to the DRL agent

7: for each time slot t do:

8: Choose a probability p

9: if $\boldsymbol{\varepsilon} \geq$ p then:

10: Select a random action $a(t)$

11: else

12: Select a greedy action $a(t) = argmaxQ^*(\emptyset(t), a(t); \theta)$

13: end if

14: Solve Equation (5.18) to obtain the optimal beamforming solution based on an active set of RRHs $\mathcal{A}$.

15: Calculate reward $K(t)$ and successor state $s(t+1)$

16: Store the transition of $(s(t), a(t), K(t), s(t+1)$ into $\mathcal{D}$

17: Randomly sample mini-batch transition $(s(t), a(t), K(t), s(t+1)$ from $\mathcal{D}$

18: Extract the next-state feature $\emptyset'$ using CNN

19: Set target

$$y(t) = \begin{Bmatrix} K(t), & if\ episode\ terminate \\ K(t) + \mu maxQ(\emptyset', a'; \theta'), & elsewise \end{Bmatrix}$$

20: Train the network to minimize the loss function of Equation (5.12)

21: Perform the stochastic descent step on $\big(y(t) - Q(\emptyset(t), a(t); \theta)\big)$

22: **end for:**

23:**end for**

5.3.6 Computational Complexity

The computational complexity of the proposed CNN-DQN algorithm is derived from Equation (5.18). Since Equation (5.18) can be modified to second-order cone programming (SOCP), which can be solved in polynomial time by a standard interior-point method, e.g., (Ben-Tal, A. and Nemirovski, 2001). The total number of variables of Equation (5.18) is $R + U$, and a total number of constraints is $2R + 2U + 1$. Thus, the worst-case computational complexity per-episode is $\mathcal{O}(RU)^{3.5}$. Therefore, the overall computational complexity of

Algorithm 5.1 is $\mathcal{O}(R^{3.5}\ U^{3.5}\ K + \Psi.\Omega + \mathcal{D} + \left|G^{\theta}\right|)$, where K is the number of episodes required to converge Algorithm 5.1. $(\Psi.\Omega)$, $\mathcal{D}$ and G^{θ} specify the size of extracted channel gain, the number of experience samples from the replay buffer, and the number of hidden layers, respectively. Similarly, the computational complexity of (Xu *et al.*, 2017) is $\mathcal{O}(R^{3.5}\ U^{3.5}\ K + \mathcal{D} + \left|G^{\theta}\right|)$. The computational complexity of the proposed algorithm is much higher than that in (Xu *et al.*, 2017) since the proposed algorithm limits the size of the channel gain feature at the input of the network state. However, the signalling overhead of our proposed algorithm is much less than that in (Xu *et al.*, 2017) because users do not have to exchange their information to the respective RRHs. RRHs record all the information between RRHs and users. That reduces the signalling burden of the network.

Table 5.3: Simulation Parameters

Parameters	Symbol	Value
Transition power	$p_{r,\mathcal{G}}$	3 W
Transmit power	$p_{r,T}$	1 W
Sleep power	$p_{r,S}$	4.3 W
Active power	$p_{r,A}$	6.8 W
Bandwidth	W	10 MHz
Noise power	δ^2	-102 dBm
Power amplifier efficiency	τ	25 %
SNR gap	$\mathcal{J}_{\mathrm{m}}$	1
Antenna Gain	$\zeta_{r,u}$	9 dBi
Shadowing coefficient	$\rho_{r,u}$	8 dB
Small scale fading	$\omega_{r,u}$	$\mathcal{CN}(0,1)$

5.4 RESULTS AND DISCUSSIONS

This section analyzes the simulation setting and demonstrates the performance of the proposed CNN-DQN approach. The proposed approach is compared with the conventional DQN and the traditional approach. In order to simplify the traditional approach, the traditional approach is assumed to be a full coordinate association and denoted as FA. As for FA, the discount factor value is considered as $\mu = 0$, in which the DRL agent only learns the action that produces an immediate power consumption value and user QoS satisfaction. The user demand is fixed within a range of [10-60] Mbps with a step size of 10 Mbps. Additionally, two scenarios are examined to verify the effectiveness of increasing the number of RRHs when RL agents cannot satisfy the user's QoS demand. First, 1000 training episodes are considered to teach the DRL agent to recognize the behavior of the environment. Finally, the results are generated based on 100 testing episodes. To compare the proposed solution network performance with the FA approach (Dai and Yu, 2016), the optimal simulation parameters used in this work can be found in Table 5.3.

5.4.1 Effect of Hyperparameter

Learning rate (γ) is a crucial performance hyperparameter used in ML. This hyperparameter determines how to tune the neural network to achieve optimal performance. Therefore, a suitable value for γ must be selected. The larger the value of γ, the greater the chance there is to over-fit the model, but a higher value of γ increases the learning speed of the neural network. When the γ is small, it is easier to prevent the model from over-fitting. However, it requires tremendous computing power to train the neural network.

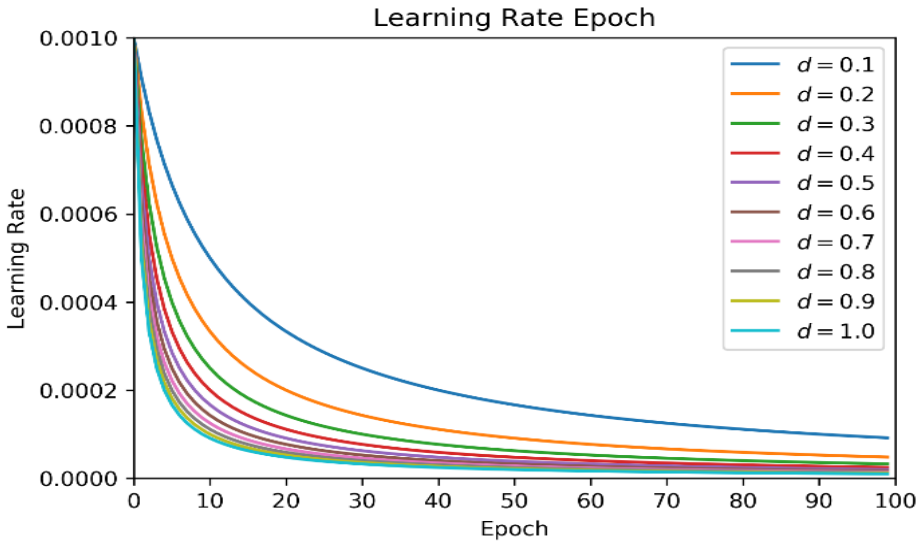


Figure 5.4: Effect of learning rate on the different decaying values with epoch

For epoch i, the γ is given by:

$$\gamma = \frac{\gamma_{init}}{1 + i \times d}, \quad 1 \leq i \leq 100 \tag{5.19}$$

The d and γ_{init} represent the positive integer and initial learning rate, respectively, which are used to control the decaying speed. In this chapter, the positive integer value is assumed as $d \in \{0.1,0.2,0.\ 3,\ldots 1.0\}$. Figure 5.4 shows that when $d = 0$, γ becomes constant for all the epoch values. However, as the d increases, the γ decreases sharply. In order to avoid the neural network from overfitting, these values are considered as $\gamma = 0.001 = 10^{-3}$and $d = 1$.

5.4.2 Power Allocation

In this section, the proposed CNN-DQN approach is demonstrated for computing power consumption performance on different values of the user demands. The proposed approach is then compared with the conventional DQN and FA, as shown in Figure 5.5. At first, the $R = 6$ and $U = 4$ are considered. Figure 5.5 shows that power consumption increases exponentially with all three approaches as the user demand increases. The proposed CNN-DQN approach can reduce power consumption by 5-10% at all the user demand points.

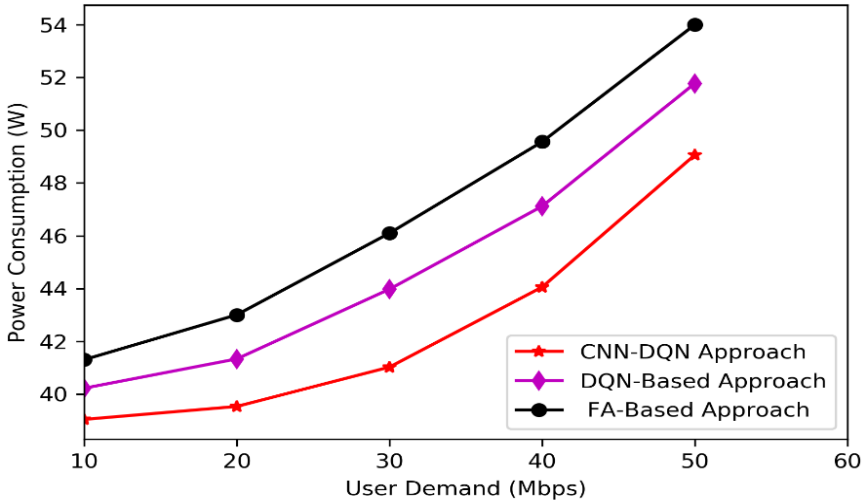


Figure 5.5: Comparison of the proposed algorithm with other algorithms for power saving on different user demand, $R = 6, U = 4$

Moreover, the proposed CNN-DQN approach and the DQN-based approach consistently outperform the FA-based approach. The reason comes from the learning of the environment. So, at each time step t, the learning agent takes the best possible action from the action space. The FA approach randomly chooses the action from the current action space and does not learn anything from the environment. However, all three approaches become infeasible to satisfy the user QoS demand after reaching 50Mbps. There are not enough active RRHs available to meet user QoS requirements. In order to avoid this problem, the number of RRHs is increased to $R = 8$ with $U = 4$, as shown in Figure 5.6. As a result, the infeasibility issue is solved, and the proposed CNN-DQN approach can significantly reduce power consumption and satisfy the user's QoS requirement, as shown in Figure 5.6. However, the increased RRHs will increase the power consumption of the system. Figure 5.6 illustrates that when the user demand is 50Mbps, the power consumed by the proposed CNN-DQN solution is 48.73W for $R = 6$, while at the same point, the power consumption

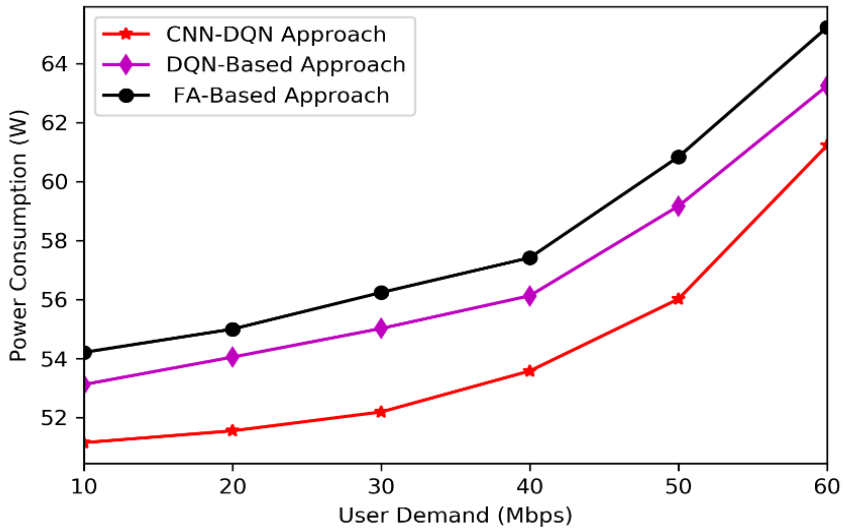


Figure 5.6: Comparison of the proposed algorithm with other algorithms for power saving on different user demand, $R = 8, U = 4$

is 55W for $R = 8$.

5.4.3 Energy Efficiency Maximization

The EE performance is plotted against different user demands for $R = 6, U = 4$, as shown in Figure 5.7. The EE is linearly increasing with the increasing user demand. As shown in Figure 5.7, the DRL approach outperforms the FA

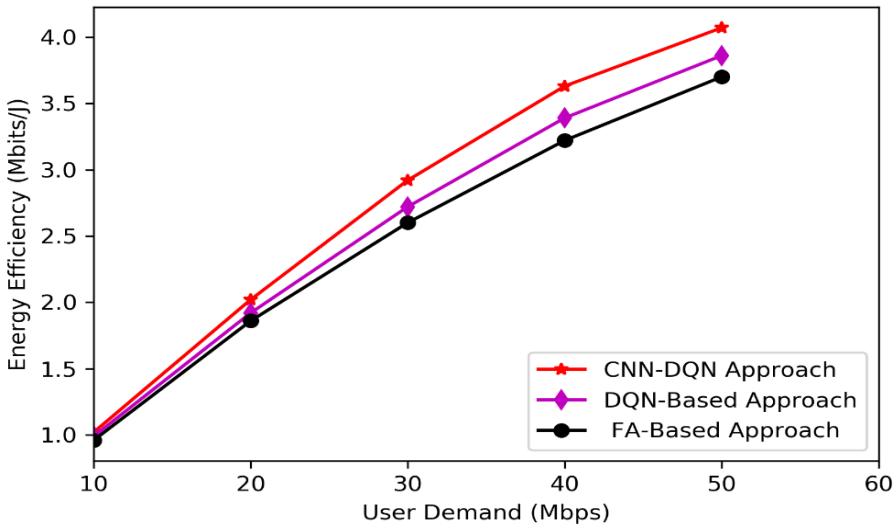


Figure 5.7: Comparison of proposed algorithm with other algorithms for energy efficiency maximization on different user demands $R = 6, U = 4$

approach. The FA approaches primarily focus on EE performance within the immediate network state, making decisions exclusively for the current action space. DQN-based approaches are designed to improve EE performance at each point of the user demand compared with the FA approaches. However, the DQN-based approach contains a large number of state-action pairs, which reduces the system performance and increases computational complexity. However, it still achieves $4\% - 8\,\%$ better performance over the FA-based approach. Figure 5.7 shows that the proposed CNN-DQN approach reduces the training parameters and outperforms the other two approaches for increasing user demand. The proposed CNN-DQN approach achieves $5\% - 12\,\%$ better performance than the other approaches. However, all three approaches diminish their performance as the user data rate grows beyond 50Mbps due to resource bottlenecks. In order to avoid the bottleneck problem, more RRHs should always be turned on when accommodating higher user demands, so the number of RRHs should be increased to $R = 8$ with the same user $U = 4$, as shown in

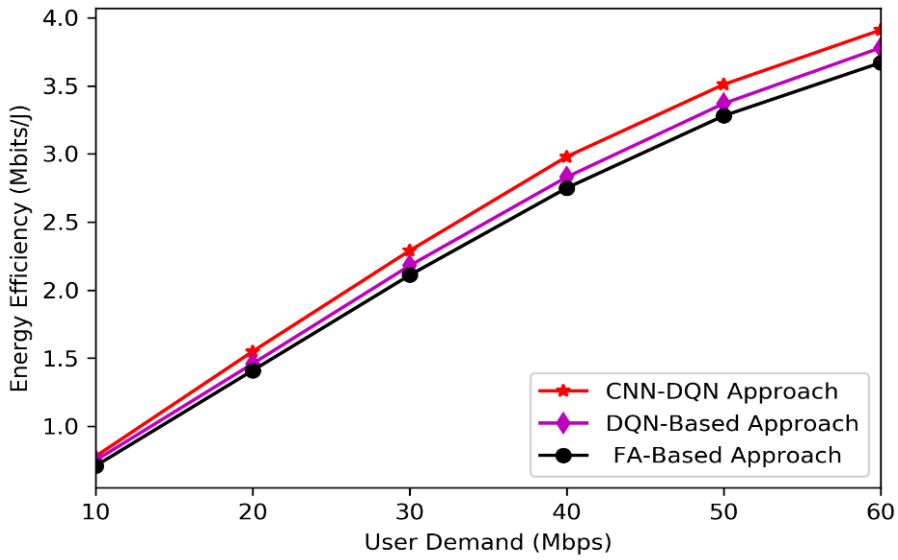


Figure 5.8: Comparison of proposed algorithm with other algorithms for energy efficiency maximization on different user demands $R = 8, U = 4$

Figure 5.8. However, the proposed CNN-DQN approach is still more efficient at all points of user demands, such as when the user demand is 60Mbps, the proposed CNN-DQN approach improves the EE by up to 8% more than the DQN. These performances are evidence of using a CNN-DQN approach for a high mobility scenario.

5.4.4 Relationships Between Energy Efficiency versus Power Consumption

Figure 5.9 shows the EE performance versus the power consumption for $R = 6$ and $U = 4$. From Figure 5.9, it can be seen that at the start, EE is slightly increased over a small increase in power for all the approaches. However, the EE starts to decline without further increasing after reaching to maximum value for all three approaches. This is due to the requirement that high transmit power is required to meet the users' QoS demands. From Figure 5.9, one can observe that the proposed CNN-DQN approach achieves a maximum EE of 4.10 Mbits/J with a power consumption of 48.73 W. Similarly, DQN and FA can achieve 3.92 Mbits/J with a power consumption of 51.95 W and

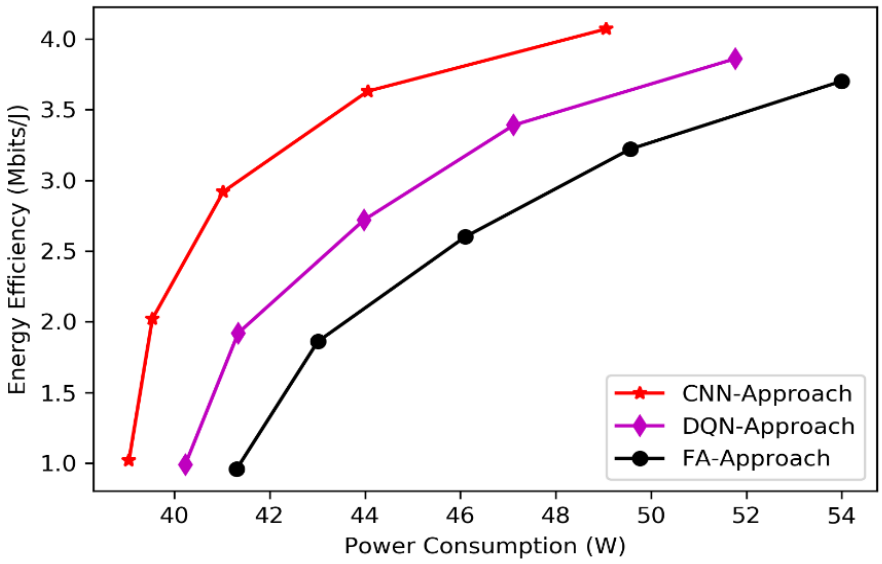


Figure 5.9: Comparison of proposed solution for Energy Efficiency maximization with power consumption for *R*=6 and *U*=4

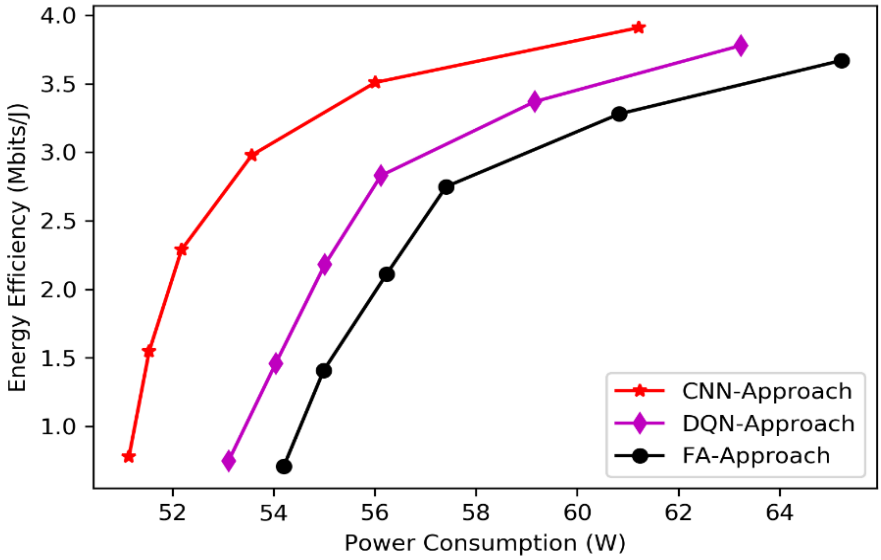


Figure 5.10: Comparison of proposed solution for Energy Efficiency maximization with power consumption for *R*=8 and *U*=4

3.61 Mbit/J with a power consumption of 54.07W, respectively. A similar trend has been applied to Figure 5.10 by increasing the number RRHs $R = 8$ with the same number of users as $U = 4$. As shown in Figure 5.10, the proposed CNN-DQN approach can achieve the EE of 3.95 Mbit/J, with a power consumption of 61.15 W, while the DQN-based and FA-based approaches can achieve the EE of 3.70 Mbits/J and 3.55 Mbits/J with a power consumption of 63W and 65W, respectively. These figures show the proposed method's effectiveness in achieving more EE with less power consumption.

5.4.5 Transmit Power Selection

Figure 5.11 shows the average EE performance of different transmission power levels. This figure shows that the proposed CNN-DQN approach consistently outperforms the DQN and FA. In the beginning, the transmission power of RRHs is very low; thus, all three approaches achieve roughly the same average EE performance. As the transmission power is increased, the overall EE

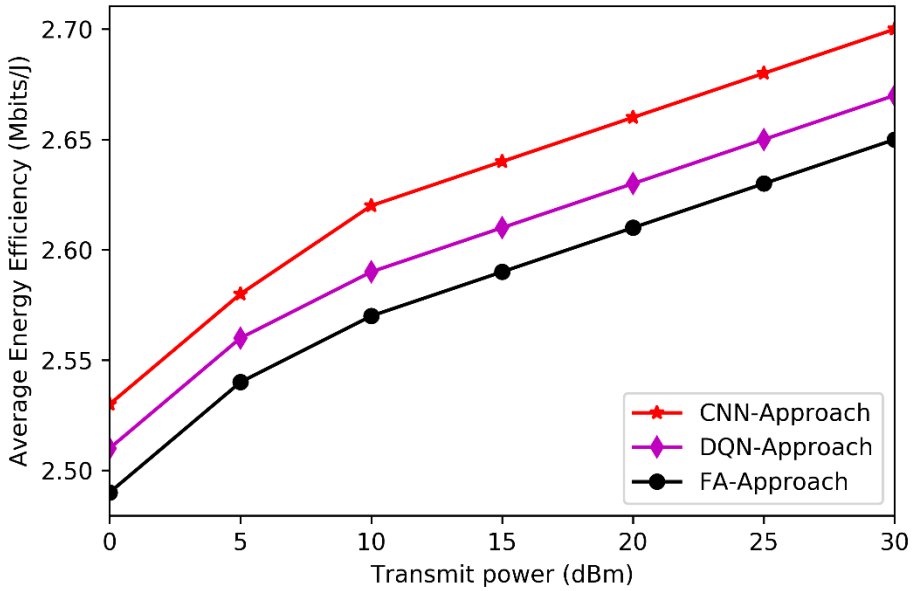


Figure 5.11: Average Energy Efficiency performance vs Transmit power

performance increases linearly. A higher value of average EE can be achieved with the proposed CNN-DQN approach on different transmit power levels, which shows how effective the proposed CNN-DQN approach is for various transmit power levels.

5.5 SUMMARY

This chapter combines the CNN approach with the DQN to balance the EE performance and simultaneously satisfy the user QoS demand in the downlink CRAN. The CNN phase is responsible for extracting the input channel state information. As a result, the extracted feature of CNN is fed to the input of DQN, which turns on/off RRHs in response to user demands. The RA scheme is then formulated as a convex optimization for balancing the performance of EE and meeting the user's QoS requirement. Finally, comprehensive simulation results for different scenarios demonstrate that the proposed CNN-DQN results in the best balances in terms of EE performance while meeting the user

requirements in the varying scenario.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1 CONCLUSIONS

5G and beyond networks are expected to face unprecedented challenges related to heterogeneity in terms of deployments, environments, and mobility scenarios. Cloud radio access network (CRAN) emerges as a promising candidate, which can meet these requirements by deploying low-cost, intelligent, and multiple distributed antennas called remote radio heads (RRHs). However, achieving the optimal resource allocation (RA) in terms of power minimization, maximizing EE and SE in CRANs using the conventional approach becomes infeasible for a large and complicated state, especially when the network performance changes with environmental changes.

Inspired by the success of DRL in solving complicated control problems, this thesis proposes three DRL-based RA algorithms that optimize the CRAN performance in terms of EE, SE, and total power consumption. The details of this thesis's main contribution are summarized as follows:

Firstly, a Double DQN-based RA framework in CRAN is proposed and presented in chapter 3, which maximizes the total EE within the constraints of per RRH transmission power selection and user rates. The channel state information (CSI) is considered at the input of the network state, which is updated continuously at each time step t, and generates a state space too large. Hence, the tabular methods (Q-learning) and conventional DQN methods

become insufficient and unsuitable to limit state space size. In order to address this shortcoming, a new approach based on the features of each state is adopted, known as function approximation. The aim of this approach is to use these features to generalize the CSI value estimation at the input of the network state with similar features. These estimated features are then passed on to Double DQN to find the optimal control policy in order to turn on/off the RRHs based on the user demand. Simulation results indicate that the proposed Double DQN based RA method saves 22% more power as compared with the conventional approach and improves EE's performance by 20% with a minimum requirement of user data rates.

Secondly, a Dueling DQN-based (D2QN) RA scheme is proposed and presented in chapter 4 in order to optimize the long-term tradeoff between EE-SE and satisfy the user QoS requirements in CRANs. Specifically, ML techniques known as Anchor Graph Hashing (AGH) is implemented to discretise the CSI generalized features before feeding them into the DRL input. In addition, the D2QN method is configured to learn the near-optimal switching strategy to turn on/off RRHs in order to maximize EE-SE performance under the varying channel gain while satisfying the user QoS requirements. Finally, the proposed D2QN based AGH method is compared with D2QN without CSI generalization, Q-learning, and myopic approach. The improved EE-SE performance is compared with the AGH-based D2QN method based on the simulation results.

Thirdly, a convolutional neural network (CNN)-based deep Q-network (CNN-DQN) approach is proposed in downlink CRANs and presented in chapter 5, which balances the EE performance and satisfies the user QoS demand. First,

the CNN approach is combined with DQN, where the CNN phase is responsible for extracting the input state information, which contains the CSI, user's demand and features of RRHs. The extracted feature of CNN is then fed to the input of DQN, which dynamically performs the switching decision of the RRHs based on the energy consumption of user demand. Then, the RA optimization scheme is formulated based on the user constraints and transmit power to balance the performance of EE and satisfy the user QoS requirements. The performance of the proposed CNN-DQN approach is compared with the traditional approach (Dai and Yu, 2016) and DQN approach (Xu *et al.*, 2017). The proposed CNN-DQN approach shows better EE performance and satisfies the users' QoS requirements in a different scenario.

In general, the network performance is determined after successfully compiling the simulation for 1000 training episodes and 100 testing episodes in the simulation environment with 16 GB RAM, intel core i3-7100 (3.90GHz), TensorFlow 1.14.0, and python 3.7.5.

With efficient optimization and adaptive schemes, the proposed DRL-based work can demonstrate better adaptability than the existing solution explained in the literature. The simulation results show the effectiveness of the proposed DRL-based method in terms of fast convergence speed, better learning performance, and improved network performance than the baseline approaches.

6.2 FUTURE WORKS

In this thesis, the channel state information's uncertainty has not been considered. This work can be extended by incorporating the uncertainty in the wireless propagation environment.

Different approaches are utilized to limit the channel state information features. However, this process requires a lot of computation power to make the channel state information linear. In the future, a deep deterministic policy gradient (DDPG) algorithm can be used, consisting of an actor-critic neural network and a classifier for mapping the continuous channel gain at the input of the network state for different RA decisions.

One major drawback of DRL is that it can end up having a local optimum instead of a global optimum. On the contrary, generative adversarial network (GAN) is a well-known technique for reaching global optimum because of its loss function (i.e., binary cross-entropy) used in training. Therefore, by combining GAN with RL, a GAN-RL might perform better compared to DRL. In the future, a GAN-RL based RA framework can be developed for CRANs.

LIST OF PUBLICATIONS

Journals

1. Iqbal, A., Tham, M.L. and Chang, Y.C., 2021. Double deep Q-network-based energy-efficient resource allocation in cloud radio access network. *IEEE Access, 9*, pp.20440-20449, **IF= 3.75**.
2. Iqbal, A., Tham, M.L. and Chang, Y.C., 2021. Resource allocation for joint energy and spectral efficiency in cloud radio access networks based on deep reinforcement learning. *Transactions on Emerging Telecommunications Technologies*, p.e12490, **IF=2.63.**
3. Iqbal, A., Tham, M.L. and Chang, Y.C., 2021. Convolutional Neural Network-Based Deep Q-Network (CNN-DQN) Resource Management in Cloud Radio Access Network. *China Communications*, **IF=2.68.**

Conferences

1. Tham, M.L., Iqbal, A. and Chang, Y.C., 2019, November. Deep reinforcement learning for resource allocation in 5G communications. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 1852-1855), IEEE.
2. Iqbal, A., Tham, M.L. and Chang, Y.C., 2020, August. Double deep Q-network for power allocation in cloud radio access network. In *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)* (pp. 272-277), IEEE.
3. Iqbal, A., Tham, M.L. and Chang, Y.C., 2021, June. Energy-and Spectral-Efficient Optimization in Cloud RAN based on Dueling Double Deep Q-Network. In *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)* (pp. 311-316), IEEE.

REFERENCES

Ahmad, A. A., Dahrouj, H., Chaaban, A., Sezgin, A., Al-Naffouri, T. Y. and Alouini, M. S. (2020) 'Power minimization via rate splitting in downlink cloud-radio access networks', *2020 IEEE International Conference on Communications Workshops, ICC Workshops 2020 - Proceedings*, pp. 0–5. doi: 10.1109/ICCWorkshops49005.2020.9145363.

Ali, M., Rabbani, Q., Naeem, M., Qaisar, S. and Qamar, F. (2017) 'Joint User Association, Power Allocation, and Throughput Maximization in 5G H-CRAN Networks', *IEEE Transactions on Vehicular Technology*, 66(10), pp. 9254–9262. doi: 10.1109/TVT.2017.2715229.

AlQerm, I. and Shihada, B. (2018) 'Sophisticated Online Learning Scheme for Green Resource Allocation in 5G Heterogeneous Cloud Radio Access Networks', *IEEE Transactions on Mobile Computing*, 17(10), pp. 2423–2437. doi: 10.1109/TMC.2018.2797166.

Aqeeli, E., Moubayed, A. and Shami, A. (2018) 'Power-Aware Optimized RRH to BBU Allocation in C-RAN', *IEEE Transactions on Wireless Communications*, 17(2), pp. 1311–1322. doi: 10.1109/TWC.2017.2777825.

Ari, A. A. A., Gueroui, A., Titouna, C., Thiare, O. and Aliouat, Z. (2019) 'Resource allocation scheme for 5G C-RAN: a Swarm Intelligence based approach', *Computer Networks*, 165. doi: 10.1016/j.comnet.2019.106957.

Asheralieva, A. (2017) 'Bayesian Reinforcement Learning-Based Coalition Formation for Distributed Resource Sharing by Device-to-Device Users in Heterogeneous Cellular Networks', *IEEE Transactions on Wireless Communications*, 16(8), pp. 5016–5032. doi: 10.1109/TWC.2017.2705039.

Auer, G., Giannini, V., GÓdor, I., Blume, O., Fehske, A., Rubio, J. A., Frenger, P., Olsson, M., Sabella, D., Gonzalez, M. J., Imran, M. A. and Desset, C. (2012) 'How much energy is needed to run a wireless network?', *Green Radio Communication Networks*, 9781107017, pp. 359–384. doi: 10.1017/CBO9781139084284.017.

Ben-Tal, A. and Nemirovski, A. (2001) *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*.

Bottou, L. (2012) 'Stochastic gradient descent tricks', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTU(1), pp. 421–436. doi: 10.1007/978-3-642-35289-8_25.

Chai, G., Wu, W., Yang, Q., Liu, R., Qin, M. and Kwak, K. S. (2021) 'Energy-efficient resource allocation for multi-RAT networks under time average QoS constraint', *Wireless Networks*, 27(1), pp. 323–338. doi: 10.1007/s11276-020-02456-3.

Chang, Y., Yuan, X., Li, B., Niyato, D. and Al-Dhahir, N. (2018) 'A joint unsupervised learning and genetic algorithm approach for topology control in energy-efficient ultra-dense wireless sensor networks', *IEEE Communications*

Letters. IEEE, 22(11), pp. 2370–2373. doi: 10.1109/LCOMM.2018.2870886.

Checko, A., Berger, M. S., Kardaras, G., Dittmann, L. and Christiansen, H. L. (2016) 'Cloud Radio Access Network architecture . Towards 5G mobile networks', *Technical university of denmark*. Available at: http://orbit.dtu.dk/files/123876544/ACheckoPhDThesisCRANFinalRC.pdf.

Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S. and Dittmann, L. (2015) 'Cloud RAN for Mobile Networks - A Technology Overview', *IEEE Communications Surveys and Tutorials*, 17(1), pp. 405–426. doi: 10.1109/COMST.2014.2355255.

Chen, L., Nguyen, T. M. T., Yang, D., Nogueira, M., Wang, C. and Zhang, D. (2021) 'Data-Driven C-RAN Optimization Exploiting Traffic and Mobility Dynamics of Mobile Users', *IEEE Transactions on Mobile Computing*, 20(5), pp. 1773–1788. doi: 10.1109/TMC.2020.2971470.

Cheng, X., Fang, L., Yang, L. and Cui, S. (2017) 'Mobile Big Data: The Fuel for Data-Driven Wireless', *IEEE Internet of Things Journal*, 4(5), pp. 1489–1516. doi: 10.1109/JIOT.2017.2714189.

Chih-Lin, I., Li, H., Korhonen, J., Huang, J. and Han, L. (2018) 'RAN Revolution with NGFI (xhaul) for 5G', *Journal of Lightwave Technology*. IEEE, 36(2), pp. 541–550. doi: 10.1109/JLT.2017.2764924.

Chughtai, N. A., Ali, M., Qaisar, S., Imran, M., Naeem, M. and Qamar, F. (2018) 'Energy Efficient Resource Allocation for Energy Harvesting Aided H-CRAN', *IEEE Access*, 6, pp. 43990–44001. doi: 10.1109/ACCESS.2018.2862920.

Cisco (2020) 'Cisco: 2020 CISO Benchmark Report', *Computer Fraud & Security*, 2020(3), pp. 4–4. doi: 10.1016/s1361-3723(20)30026-9.

Coskun, C. C. and Ayanoglu, E. (2017) 'Energy-spectral efficiency tradeoff for heterogeneous networks with QoS constraints', *IEEE International Conference on Communications*, pp. 1–7. doi: 10.1109/ICC.2017.7997007.

Dai, B. and Yu, W. (2016) 'Energy Efficiency of Downlink Transmission Strategies for Cloud Radio Access Networks', *IEEE Journal on Selected Areas in Communications*, 34(4), pp. 1037–1050. doi: 10.1109/JSAC.2016.2544459.

Dhif-Allah, O., Dahrouj, H., Al-Naffouri, T. Y. and Alouini, M. S. (2018) 'Distributed Robust Power Minimization for the Downlink of Multi-Cloud Radio Access Networks', *IEEE Transactions on Green Communications and Networking*, 2(2), pp. 327–335. doi: 10.1109/TGCN.2017.2780449.

Dinh, T. H. L., Kaneko, M., Fukuda, E. H. and Boukhatem, L. (2021) 'Energy Efficient Resource Allocation Optimization in Fog Radio Access Networks with Outdated Channel Knowledge', *IEEE Transactions on Green Communications and Networking*, 5(1), pp. 146–159. doi: 10.1109/TGCN.2020.3034638.

Eisen, M., Zhang, C., Chamon, L. F. O., Lee, D. D. and Ribeiro, A. (2019) 'Learning Optimal Resource Allocations in Wireless Systems', *IEEE Transactions on Signal Processing*. IEEE, 67(10), pp. 2775–2790. doi:

10.1109/TSP.2019.2908906.

Fan, J., Wang, Z., Xie, Y. and Yang, Z. (2019) 'A Theoretical Analysis of Deep Q-Learning', 120(1995), pp. 1–4. Available at: http://arxiv.org/abs/1901.00137.

Fan, Y. and Li, H. (2017) 'Distributed Approximating Global Optimality with Local Reinforcement Learning in HetNets', *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*, 2018-Janua, pp. 1–7. doi: 10.1109/GLOCOM.2017.8254853.

Farhadi Zavleh, A. and Bakhshi, H. (2021) 'Resource allocation in sparse code multiple access-based systems for cloud-radio access network in 5G networks', *Transactions on Emerging Telecommunications Technologies*, 32(1), pp. 1–20. doi: 10.1002/ett.4153.

Gao, Z., Zhang, J., Yan, S., Xiao, Y., Simeonidou, D. and Ji, Y. (2019) 'Deep Reinforcement Learning for BBU Placement and Routing in C-RAN', in *Optical Fiber Communications Conference and Exhibition (OFC)*.

Geramifard, A., Walsh, T. J., Tellex, S., Chowdhary, G., Roy, N. and How, J. P. (2013) 'A tutorial on linear function approximators for dynamic programming and reinforcement learning', *Foundations and Trends in Machine Learning*, 6(4), pp. 375–454. doi: 10.1561/2200000042.

Gerasimenko, M., Moltchanov, D., Florea, R., Andreev, S., Koucheryavy, Y., Himayat, N., Yeh, S. P. and Talwar, S. (2015) 'Cooperative radio resource management in heterogeneous cloud radio access networks', *IEEE Access*. IEEE, 3, pp. 397–406. doi: 10.1109/ACCESS.2015.2422266.

Gholipoor, N., Nouruzi, A., Salarhosseini, S., Javan, M. R., Mokari, N. and Jorswieck, E. A. (2021) 'Learning based E2E Energy Efficient in Joint Radio and NFV Resource Allocation for 5G and Beyond Networks', pp. 1–14. Available at: http://arxiv.org/abs/2107.05991.

Gressling, T. (2020) *84 Automated machine learning, Data Science in Chemistry*. doi: 10.1515/9783110629453-084.

Hado van Hasselt , Arthur Guez, D. S. (2016) 'Deep Reinforcement Learning with Double Q-Learning', *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1). doi: 10.1109/ICRA40945.2020.9196684.

Hassani, S. El, Haidine, A. and Jebbar, H. (2020) 'Road to 5G : Key Enabling Technologies', 14(11), pp. 1034–1048. doi: 10.12720/jcm.14.11.1034-1048.

Hou, K., Xu, Q., Zhang, X., Huang, Y. and Yang, L. (2021) 'User association and power allocation based on unsupervised graph model in ultra-dense network', *IEEE Wireless Communications and Networking Conference, WCNC*, 2021-March, pp. 0–5. doi: 10.1109/WCNC49053.2021.9417279.

Hsieh, C. K., Chan, K. L. and Chien, F. T. (2021) 'Energy-efficient power allocation and user association in heterogeneous networks with deep reinforcement learning', *Applied Sciences (Switzerland)*, 11(9). doi: 10.3390/app11094135.

Huang, X., Fan, W., Chen, Q. and Zhang, J. (2020) 'Energy-Efficient Resource Allocation in Fog Computing Networks with the Candidate Mechanism', *IEEE*

Internet of Things Journal, 7(9), pp. 8502–8512. doi: 10.1109/JIOT.2020.2991481.

Ian Goodfellow, Yoshua Bengio, A. C. (2016) 'Deep learning', *Nature*, 29(7553), pp. 1–73.

Jiang, Q. Y. and Li, W. J. (2015) 'Scalable graph hashing with feature transformation', *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai), pp. 2248–2254.

Kai Arulkumaran, M. P. D. and Miles Brundage, and A. A. B. (2017) 'Deep Reinforcement Learning: A Brief Survey', *IEEE Signal Processing Magazine*, 34(6), pp. 26–38.

Karunakaran, D., Worrall, S. and Nebot, E. (2020) 'Efficient statistical validation with edge cases to evaluate Highly Automated Vehicles', *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*. doi: 10.1109/ITSC45102.2020.9294590.

Khan, M. I., Reggiani, L., Alam, M. M., Moullec, Y. Le, Sharma, N., Yaacoub, E. and Magarini, M. (2020) 'Q-learning based joint energy-spectral efficiency optimization in multi-hop device-to-device communication', *Sensors (Switzerland)*, 20(22), pp. 1–23. doi: 10.3390/s20226692.

Kim, J., Park, J., Noh, J. and Cho, S. (2020) 'Autonomous Power Allocation Based on Distributed Deep Learning for Device-to-Device Communication Underlaying Cellular Network', *IEEE Access*, 8, pp. 107853–107864. doi: 10.1109/ACCESS.2020.3000350.

Kubat, M. (2017) *An Introduction to Machine Learning, An Introduction to Machine Learning*. doi: 10.1007/978-3-319-63913-0.

Labana, M. and Hamouda, W. (2020) 'Joint User Association and Resource Allocation in CoMP-Enabled Heterogeneous CRAN', *2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings*, pp. 1–6. doi: 10.1109/GLOBECOM42002.2020.9322501.

Lecun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436–444. doi: 10.1038/nature14539.

Lee, W., Kim, M. and Cho, D. H. (2018) 'Deep Power Control: Transmit Power Control Scheme Based on Convolutional Neural Network', *IEEE Communications Letters*, 22(6), pp. 1276–1279. doi: 10.1109/LCOMM.2018.2825444.

Li, D., Xu, S. and Li, P. (2021) 'Deep reinforcement learning-empowered resource allocation for mobile edge computing in cellular v2x networks', *Sensors (Switzerland)*, 21(2), pp. 1–18. doi: 10.3390/s21020372.

Li, H., Gao, H., Lv, T. and Lu, Y. (2018) 'Deep Q-Learning based dynamic resource allocation for self-powered ultra-dense networks', *2018 IEEE International Conference on Communications Workshops, ICC Workshops 2018 - Proceedings*, pp. 1–6. doi: 10.1109/ICCW.2018.8403505.

Li, J., Gao, H., Lv, T. and Lu, Y. (2018) 'Deep reinforcement learning based computation offloading and resource allocation for MEC', *IEEE Wireless*

Communications and Networking Conference, WCNC, 2018-April, pp. 1–6. doi: 10.1109/WCNC.2018.8377343.

Li, M., Meng, Y., Liu, J., Zhu, H., Liang, X., Liu, Y. and Ruan, N. (2016) 'When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals', *Proceedings of the ACM Conference on Computer and Communications Security*, 24-28-Octo, pp. 1068–1079. doi: 10.1145/2976749.2978397.

Li, W., Ni, W., Tian, H. and Hua, M. (2021) 'Deep reinforcement learning for energy-efficient beamforming design in cell-free networks', *2021 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2021*. doi: 10.1109/WCNCW49093.2021.9420002.

Lin, Z. and Liu, Y. (2019) 'Joint uplink and downlink transmissions in user-centric OFDMA cloud-RAN', *IEEE Transactions on Vehicular Technology*, 68(8), pp. 7776–7788. doi: 10.1109/TVT.2019.2924437.

Liu, W., Wang, J., Kumar, S. and Chang, S. F. (2011) 'Hashing with graphs', *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 1–8.

Liu, Z., Chen, X., Chen, Y. and Li, Z. (2019) 'Deep reinforcement learning based dynamic resource allocation in 5G ultra-dense networks', *Proceedings - 2019 IEEE International Conference on Smart Internet of Things, SmartIoT 2019*, pp. 168–174. doi: 10.1109/SmartIoT.2019.00034.

Liu, Z., Yu, P., Zhou, F., Feng, L. and Li, W. (2021) 'Intelligent and Energy-efficient Distributed Resource Allocation for 5G Cloud Radio Access Networks', *Proceedings of the 2021 17th International Conference on Network and Service Management: Smart Management for Future Networks and Services, CNSM 2021*, pp. 70–76. doi: 10.23919/CNSM52442.2021.9615594.

Luo, C., Ji, J., Wang, Q., Chen, X. and Li, P. (2020) 'Channel State Information Prediction for 5G Wireless Communications: A Deep Learning Approach', *IEEE Transactions on Network Science and Engineering*. IEEE, 7(1), pp. 227–236. doi: 10.1109/TNSE.2018.2848960.

Luo, J., Chen, Q. and Tang, L. (2018) 'Reducing Power Consumption by Joint Sleeping Strategy and Power Control in Delay-Aware C-RAN', *IEEE Access*, 6, pp. 14655–14667. doi: 10.1109/ACCESS.2018.2810896.

Luo, Y., Yang, J., Xu, W., Wang, K. and Renzo, M. Di (2020) 'Power Consumption Optimization Using Gradient Boosting Aided Deep Q-Network in C-RANs', *IEEE Access*. IEEE, 8, pp. 46811–46823. doi: 10.1109/ACCESS.2020.2978935.

Luong, P., Gagnon, F., Tran, L. N. and Labeau, F. (2021) 'Deep reinforcement learning-based resource allocation in cooperative UAV-assisted wireless networks', *IEEE Transactions on Wireless Communications*. IEEE, 20(11), pp. 7610–7625. doi: 10.1109/TWC.2021.3086503.

Meng, F., Chen, P., Wu, L. and Cheng, J. (2020) 'Power Allocation in Multi-User Cellular Networks: Deep Reinforcement Learning Approaches', *IEEE Transactions on Wireless Communications*, 19(10), pp. 6255–6267. doi:

10.1109/TWC.2020.3001736.

Mesodiakaki, A., Adelantado, F., Alonso, L. and Verikoukis, C. (2014) ‘Energy-efficient context-aware user association for outdoor small cell heterogeneous networks’, *2014 IEEE International Conference on Communications, ICC 2014*, pp. 1614–1619. doi: 10.1109/ICC.2014.6883553.

Michael W. Berry, A. M. (2020) ‘Supervised and unsupervised learning’, *Biomedical Signal Processing and Artificial Intelligence in Healthcare*, (January), pp. 51–89. doi: 10.1016/b978-0-12-818946-7.00003-2.

Miozzo, M., Giupponi, L., Rossi, M. and Dini, P. (2015) ‘Distributed Q-learning for energy harvesting Heterogeneous Networks’, *2015 IEEE International Conference on Communication Workshop, ICCW 2015*, pp. 2006–2011. doi: 10.1109/ICCW.2015.7247475.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015) ‘Human-level control through deep reinforcement learning’, *Nature*, 518(7540), pp. 529–533. doi: 10.1038/nature14236.

Nguyen, K. K., Duong, T. Q., Vien, N. A., Le-Khac, N. A. and Nguyen, L. D. (2019) ‘Distributed Deep Deterministic Policy Gradient for Power Allocation Control in D2D-Based V2V Communications’, *IEEE Access*. IEEE, 7, pp. 164533–164543. doi: 10.1109/ACCESS.2019.2952411.

Nikbakht, R., Jonsson, A. and Lozano, A. (2021) ‘Unsupervised Learning for C-RAN Power Control and Power Allocation’, *IEEE Communications Letters*, 25(3), pp. 687–691. doi: 10.1109/LCOMM.2020.3027991.

O’Donoghue, B., Osband, I., Munos, R. and Mnih, V. (2018) ‘The uncertainty Bellman equation and exploration’, *35th International Conference on Machine Learning, ICML 2018*, 9, pp. 6154–6173.

O, A., S, A., M, D. and A, S. (2017) ‘T ime -Dependent Energy and R esource M anagement in M obility -Aware D2D-Empowered 5G Systems’, (August), pp. 14–22.

Peesapati, S. K. G., Olsson, M., Masoudi, M., Andersson, S. and Cavdar, C. (2021) ‘Q-learning based Radio Resource Adaptation for Improved Energy Performance of 5G Base Stations’, *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2021-Septe, pp. 979–984. doi: 10.1109/PIMRC50174.2021.9569420.

Peng, M., Yu, Y., Xiang, H. and Poor, H. V. (2016) ‘Energy-Efficient Resource Allocation Optimization for Multimedia Heterogeneous Cloud Radio Access Networks’, *IEEE Transactions on Multimedia*, 18(5), pp. 879–892. doi: 10.1109/TMM.2016.2535722.

Rony, R. I., Jain, A., Lopez-Aguilera, E., Garcia-Villegas, E. and Demirkol, I. (2017) ‘Joint access-backhaul perspective on mobility management in 5G networks’, *2017 IEEE Conference on Standards for Communications and Networking, CSCN 2017*, pp. 115–120. doi: 10.1109/CSCN.2017.8088608.

Shams, F., Bacci, G. and Luise, M. (2015) 'Energy-Efficient Power Control for Multiple-Relay Cooperative Networks Using Q-Learning', *IEEE Transactions on Wireless Communications*, 14(3), pp. 1567–1580. doi: 10.1109/TWC.2014.2370046.

Shi, Y., Zhang, J. and Letaief, K. B. (2015) 'Robust Group Sparse Beamforming for Multicast Green Cloud-RAN with Imperfect CSI', *IEEE Transactions on Signal Processing*, 63(17), pp. 4647–4659. doi: 10.1109/TSP.2015.2442957.

Soma, M., Vandenberghe, L., Boyd, S. and Lebret, H. (1998) 'Applications of second-Order cone', 284.

Sun, G., Ayepah-Mensah, D., Budkevich, A., Liu, G. and Jiang, W. (2020) 'Autonomous cell activation for energy saving in cloud-RANs based on dueling deep Q-network', *Knowledge-Based Systems*, 192. doi: 10.1016/j.knosys.2019.105347.

Sun, G., Ayepah-Mensah, D., Xu, R., Boateng, G. O. and Liu, G. (2020) 'End-to-end CNN-based dueling deep Q-Network for autonomous cell activation in Cloud-RANs', *Journal of Network and Computer Applications*. Elsevier Ltd, 169(August), p. 102757. doi: 10.1016/j.jnca.2020.102757.

Sun, G., Boateng, G. O., Ayepah-Mensah, D. and Liu, G. (2019) 'Relational Reinforcement Learning Based Autonomous Cell Activation in Cloud-RANs', *IEEE Access*. IEEE, 7(December), pp. 63588–63604. doi: 10.1109/ACCESS.2019.2916470.

Sun, G., Boateng, G. O., Huang, H. and Jiang, W. (2019) 'A reinforcement learning framework for autonomous cell activation and customized energy-efficient resource allocation in C-RANs', *KSII Transactions on Internet and Information Systems*, 13(8), pp. 3821–3841. doi: 10.3837/tiis.2019.08.001.

Sutton, R. S. and Barto, A. G. (2012) 'Reinforcement learning: An Introduction Second edition', *Learning*, 3(9), p. 322. Available at: https://books.google.com/books?id=CAFR6IBF4xYC&pgis=1%5Cnhttp://incompleteideas.net/sutton/book/the-book.html%5Cnhttps://www.dropbox.com/s/f4tnuhipchpkgoj/book2012.pdf.

Sutton, R. S. and Barto, A. G. (2018) 'Reinforcement learning: An Introduction Second edition', *Learning*, 3(9), p. 322. Available at: https://books.google.com/books?id=CAFR6IBF4xYC&pgis=1%5Cnhttp://incompleteideas.net/sutton/book/the-book.html%5Cnhttps://www.dropbox.com/s/f4tnuhipchpkgoj/book2012.pdf.

Tan, F., Chen, H., Zhao, F. and Li, X. (2018) 'Energy-efficient power allocation for massive MIMO-enabled multi-way AF relay networks with channel aging', *Eurasip Journal on Wireless Communications and Networking*, 2018(1). doi: 10.1186/s13638-018-1222-2.

Tang, J., So, D. K. C., Alsusa, E. and Hamdi, K. A. (2014) 'Resource efficiency: A new paradigm on energy efficiency and spectral efficiency tradeoff', *IEEE Transactions on Wireless Communications*. IEEE, 13(8), pp. 4656–4669. doi: 10.1109/TWC.2014.2316791.

Tasnim Rodoshi, R., Kim, T. and Choi, W. (2020) 'Deep Reinforcement

Learning Based Dynamic Resource Allocation in Cloud Radio Access Networks', *International Conference on ICT Convergence*, 2020-Octob(October), pp. 618–623. doi: 10.1109/ICTC49870.2020.9289530.

Tham, M. L., Chien, S. F., Holtby, D. W. and Alimov, S. (2017) 'Energy-efficient power allocation for distributed antenna systems with proportional fairness', *IEEE Transactions on Green Communications and Networking*, 1(2), pp. 145–157. doi: 10.1109/TGCN.2017.2697452.

Tullberg, H., Popovski, P., Li, Z., Uusitalo, M. A., Höglund, A., Bulakci, Ö., Fallgren, M. and Monserrat, J. F. (2016) 'The METIS 5G System Concept: Meeting the 5G Requirements', *IEEE Communications Magazine*, 54(12), pp. 132–139. doi: 10.1109/MCOM.2016.1500799CM.

Vu, T. T., Ngo, D. T., Dao, M. N., Durrani, S., Nguyen, D. H. N. and Middleton, R. H. (2018) 'Energy Efficiency Maximization for Downlink Cloud Radio Access Networks with Data Sharing and Data Compression', *IEEE Transactions on Wireless Communications*. IEEE, 17(8), pp. 4955–4970. doi: 10.1109/TWC.2018.2834370.

Vu, T. X., Vu, T. A., Chatzinotas, S. and Ottersten, B. (2017) 'Spectral-efficient model for multiuser massive MIMO: Exploiting user velocity', *IEEE International Conference on Communications*. doi: 10.1109/ICC.2017.7997444.

Wang et al (2015) 'ACQUISITION OF CHANNEL STATE INFORMATION IN HETEROGENEOUS CLOUD RADIO ACCESS NETWORKS : CHALLENGES AND RESEARCH DIRECTIONS', 22(3), pp. 100–107.

Wang, K., Zhou, W. and Mao, S. (2016) 'Energy efficient joint resource scheduling for delay-aware traffic in cloud-RAN', *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*. doi: 10.1109/GLOCOM.2016.7841793.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M. and De Frcitas, N. (2016) 'Dueling Network Architectures for Deep Reinforcement Learning', *33rd International Conference on Machine Learning, ICML 2016*, 4(9), pp. 2939–2947.

Wei, Y., Yu, F. R., Song, M. and Han, Z. (2018) 'User Scheduling and Resource Allocation in HetNets with Hybrid Energy Supply: An Actor-Critic Reinforcement Learning Approach', *IEEE Transactions on Wireless Communications*, 17(1), pp. 680–692. doi: 10.1109/TWC.2017.2769644.

Wiesel, A., Eldar, Y. C. and Shamai, S. (2006) 'Linear precoding via conic optimization for fixed MIMO receivers', *IEEE Transactions on Signal Processing*, 54(1), pp. 161–176. doi: 10.1109/TSP.2005.861073.

Wu, S., Zeng, Z. and Xia, H. (2017) 'Load-aware energy efficiency optimization in dense small cell networks', *IEEE Communications Letters*, 21(2), pp. 366–369. doi: 10.1109/LCOMM.2016.2620173.

Xiao, L., Jiang, D., Chen, Y., Su, W. and Tang, Y. (2020) 'Reinforcement-Learning-Based Relay Mobility and Power Allocation for Underwater Sensor Networks against Jamming', *IEEE Journal of Oceanic Engineering*. IEEE,

45(3), pp. 1148–1156. doi: 10.1109/JOE.2019.2910938.

Xu, C., Lin, F. X., Wang, Y. and Zhong, L. (2015) 'Automated OS-level device runtime power management', *ACM SIGPLAN Notices*, 50(4), pp. 239–252. doi: 10.1145/2694344.2694360.

Xu, C., Lin, F. X. and Zhong, L. (2014) 'Device drivers should not do power management', *Proceedings of 5th Asia-Pacific Workshop on Systems, APSYS 2014*. doi: 10.1145/2637166.2637233.

Xu, S., Li, R. and Yang, Q. (2018) 'Improved genetic algorithm based intelligent resource allocation in 5G Ultra Dense networks', *IEEE Wireless Communications and Networking Conference, WCNC*. IEEE, 2018-April, pp. 1–6. doi: 10.1109/WCNC.2018.8377114.

Xu, Y. H., Yang, C. C., Hua, M. and Zhou, W. (2020) 'Deep deterministic policy gradient (DDPG)-Based resource allocation scheme for NOMA vehicular communications', *IEEE Access*, 8, pp. 18797–18807. doi: 10.1109/ACCESS.2020.2968595.

Xu, Z., Wang, Y., Tang, J., Wang, J. and Gursoy, M. C. (2017) 'A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs', *IEEE International Conference on Communications*. doi: 10.1109/ICC.2017.7997286.

Yadav, A. and Dobre, O. A. (2018) 'All Technologies Work Together for Good: A Glance at Future Mobile Networks', *IEEE Wireless Communications*, 25(4), pp. 10–16. doi: 10.1109/MWC.2018.1700404.

Ye, H., Li, G. Y. and Juang, B. H. F. (2019) 'Deep Reinforcement Learning Based Resource Allocation for V2V Communications', *IEEE Transactions on Vehicular Technology*. IEEE, 68(4), pp. 3163–3173. doi: 10.1109/TVT.2019.2897134.

Yu, N., Miao, Y., Mu, L., Du, H., Huang, H. and Jia, X. (2016) 'Minimizing Energy Cost by Dynamic Switching ON/OFF Base Stations in Cellular Networks', *IEEE Transactions on Wireless Communications*, 15(11), pp. 7457–7469. doi: 10.1109/TWC.2016.2602824.

Yuan, S., Zhang, Y., Qie, W., Ma, T. and Li, S. (2021) 'Deep reinforcement learning for resource allocation with network slicing in cognitive radio network', *Computer Science and Information Systems*, 18(3), pp. 979–999. doi: 10.2298/CSIS200710055Y.

Zappone, A., Technologies, H. and Labs, O. (2018) 'ONLINE ENERGY-EFFICIENT POWER CONTROL IN WIRELESS NETWORKS BY DEEP NEURAL NETWORKS 1 : Large Networks and Systems Group (LANEAS), Laboratoire des Signaux et Systmes 2 : Mathematical and Algorithmic Sciences Laboratory ', pp. 1–5.

Zeng, D., Zhang, J., Gu, L., Guo, S. and Luo, J. (2018) 'Energy-efficient coordinated multipoint scheduling in green cloud radio access network', *IEEE Transactions on Vehicular Technology*, 67(10), pp. 9922–9930. doi: 10.1109/TVT.2018.2863246.

Zhang, K., Wen, X., Chen, Y. and Lu, Z. (2020) ‘Deep Reinforcement Learning for Energy Saving in Radio Access Network’, *2020 IEEE/CIC International Conference on Communications in China, ICCC Workshops 2020*, pp. 35–40. doi: 10.1109/ICCCWorkshops49972.2020.9209916.

Zhang, T., Zhu, K. and Wang, J. (2021) ‘Energy-Efficient Mode Selection and Resource Allocation for D2D-Enabled Heterogeneous Networks: A Deep Reinforcement Learning Approach’, *IEEE Transactions on Wireless Communications*, 20(2), pp. 1175–1187. doi: 10.1109/TWC.2020.3031436.

Zhang, X., Jia, M., Gu, X. and Guo, Q. (2019) ‘An energy efficient resource allocation scheme based on cloud-computing in H-CRAN’, *IEEE Internet of Things Journal*. IEEE, 6(3), pp. 4968–4976. doi: 10.1109/JIOT.2019.2894000.

Zhang, Z., Zhang, D. and Qiu, R. C. (2020) ‘Deep reinforcement learning for power system applications: An overview’, *CSEE Journal of Power and Energy Systems*, 6(1), pp. 213–225. doi: 10.17775/CSEEJPES.2019.00920.

Zhao, D., Qin, H., Song, B., Zhang, Y., Du, X. and Guizani, M. (2020) ‘A Reinforcement Learning Method for Joint Mode Selection and Power Adaptation in the V2V Communication Network in 5G’, *IEEE Transactions on Cognitive Communications and Networking*, 6(2), pp. 452–463. doi: 10.1109/TCCN.2020.2983170.

Zhao, N., Liang, Y. C., Niyato, D., Pei, Y., Wu, M. and Jiang, Y. (2019) ‘Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks’, *IEEE Transactions on Wireless Communications*, 18(11), pp. 5141–5152. doi: 10.1109/TWC.2019.2933417.

Zoph, B. and Le, Q. V. (2017) ‘Neural architecture search with reinforcement learning’, *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.