Game Theoretic and Machine Learning Techniques for Efficient Resource Allocation in Next Generation Wireless Networks

by
Neetu Raveendran

A Dissertation submitted to the Department of
Electrical and Computer Engineering, Cullen College of Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Electrical and Computer Engineering

Chair of Committee: Dr. Zhu Han

Committee Member: Dr. Miao Pan

Committee Member: Dr. Hien Nguyen Van

Committee Member: Dr. Srinivas Shakkottai, Texas A&M University

Committee Member: Dr. Dusit Niyato, Nanyang Technological University

University of Houston
December 2019

ACKNOWLEDGMENTS

First and foremost, I would like to express my utmost gratitude to my advisor, Dr. Zhu Han, for motivating me to pursue Doctoral studies in the first place, and for offering me a position as Research Assistant in his Wireless Networking, Signal Processing and Security Lab. He is extremely supportive, be it in encouraging innovative thinking in his students, or in understanding the issues we face during our research. It is my great pleasure and privilege to have worked under the guidance of someone so accomplished, and yet so down to earth as Dr. Han.

I would like to extend my sincere gratitude to my dissertation committee members: Dr. Miao Pan, Dr. Hien Nguyen Van, Dr. Srinivas Shakkottai, and Dr. Dusit Niyato for their support and valuable suggestions throughout the course of my dissertation. I would also like to thank the current and past members of my lab and the C3 lab, especially, Dr. Sai Mounika Errapotu, Dr. Huaqing Zhang, Dr. Yunan Gu, Dr. Radwa Aly, Dr. Jingyi Wang, Saeed Ahmadian, Xunsheng Du, Mohamed Elmossalamy, Kuo Chun Tsai, Dawei Chen, Reginald Banez, Qiuyang Shen, Debing Wei, Xinyue Zhang, Jiahao Ding, Dian Shi and Pavana Prakash, for all their help and support. I would like to thank all my friends for their amazing support and encouragement, especially my dear friends in Houston, without whom I would not have done this.

My family is my biggest source of strength and motivation. I would like to thank my dearest Papa, for teaching me to dream and for letting me follow my own rather than his. I would like to thank my dearest Amma, for this powerhouse of a person she is, and for the million things she has done for me. I would like to thank my sister Neethi, for her unconditional love and friendship all these years. I would also like to thank my in-laws, for their wonderful support. My thank you also goes to this family back in India, which has taught me and given me a lot, growing up.

Most importantly, I would like to thank my dearest husband, Pradeep. Thank you for believing that I can do this, even when I did not. Thank you for the remarkable support throughout these three years, for cheering me up during tough times, for keeping me company during long work nights, and so much more. I would not have decided to do this if it was not for you. Thank you for being such an incredible partner.

ABSTRACT

The rationale behind the next generation wireless networks is the handling of the recent massive surge in wireless traffic, especially due to the advent of the Internet of Things (IoT) ecosystem. Tremendously high data rates, extremely low latency, and significantly high Quality of Service (QoS) are among the key objectives of the forthcoming fifth generation (5G) standard. Some of the concepts which act as the driving forces behind realizing these goals are network virtualization, fog computing, heterogeneous networks, and spectrum sharing. Taking these into account, a few efficient resource allocation frameworks for these techniques are proposed in this dissertation. Considering the distributed behaviors of the different sets of entities involved and their interrelationships, we incorporate the potentials of game theory and Machine Learning (ML) as powerful mathematical tools for strategic decision making. Firstly, two resource allocation frameworks for network virtualization based on matching theory are proposed: a three-sided matching based model involving radio resources, physical infrastructure, and mobile users for wireless network virtualization, and a similar model involving Tracking Areas (TAs), Virtual Network Function (VNF) instances, and Cloud Networks (CNs) for Network Function Virtualization (NFV). Secondly, an Equilibrium Problem with Equilibrium Constraints (EPEC) and a many-to-many matching based framework is proposed for NFV integrated IoT fog computing: a large-scale model for the optimization of resource pricing for the Data Service Operators (DSOs), as well as for the optimization of resource allocation from the Fog Nodes (FNs) as per the requirements of the Authorized Data Service Subscribers (ADSSs). Thirdly, a resource allocation framework for heterogeneous networks based on Reinforcement Learning (RL) and EPEC is proposed: a multi-hop data transmission route determination model for an indoor Visible Light Communication (VLC) and Device-to-Device (D2D) heterogeneous network. Finally, a framework to enhance the

spectrum utilization of a Cognitive Radio Network (CRN) is proposed: a classification approach to detect Primary User Emulation (PUE) attacks using Generative Adversarial Networks (GANs), which are effective ML models to train classifiers in a semi-supervised manner. In this dissertation, a comprehensive discussion of these frameworks is performed, followed by the validation of their effectiveness through extensive simulations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Next Generation Wireless Networks

The Internet of Things (IoT) is a paradigm that connects people, things, data, and processes together in this information-centric era [1]. An increase in the number of smart devices connected to the internet leads to the expansion of the IoT. The number of Machine-to-Machine (M2M) connections is expected to grow from just below a billion in 2017 to 3.9 billion by 2022 globally, which is a four-fold growth in five years. The monthly global mobile data traffic is expected to increase to 77 exabytes by 2022, and the annual traffic is predicted to reach an epic amount of one zettabyte [1].

The evolution of wireless devices used in different aspects of our daily life, which depend on mobile networks for communication worldwide, has fueled the huge growth in mobile data traffic. Moreover, due to the increasing popularity and usage of video streaming platforms, more than half of the mobile data traffic is accounted for by video [1]. This tremendous growth of mobile broadband has resulted in a Radio Frequency (RF) spectrum crunch in wireless communications across the globe. The next generation of mobile networks is designed with the intention of handling the current and expected mobile traffic surge and the resultant crunch in available wireless resources.

The fifth generation (5G) of mobile networks which started getting deployed by the end of 2018, is a revolution in wireless technology. 5G is expected to be the backbone of a vast IoT ecosystem that can meet the communication demands of billions of connected devices [2]. The key specification requirements of 5G are shown in Fig. 1.1. 5G aims at delivering a peak data rate of 10 Gbps, which is a 10×

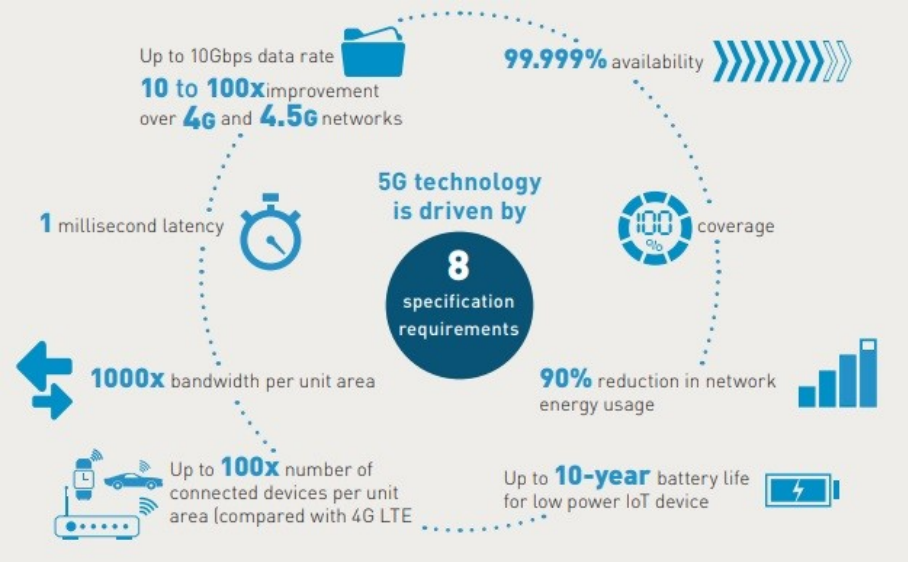


Figure 1.1: Objectives of 5G mobile networks [2].

improvement over the fourth generation (4G) networks. Latency as low as 1 ms, and 100× the number of connected devices per unit area as compared to 4G Long-Term Evolution (LTE) networks are among the other major highlights.

The use cases of 5G networks span from Vehicle-to-Everything (V2X) and Virtual Reality (VR) applications which demand ultra low latencies, to cloud based *big data* processing applications which demand extremely high data rates [2]. In order to meet these requirements and to ensure Quality-of-Experience (QoE) to the users in next generation networks, the integration and smooth co-existence of a variety of technologies is inevitable. A multi-service architecture based on *network slicing* and *edge computing* has been proposed to improve network flexibility and reduce service latency, as well as the Device-to-Device (D2D) technology to extend coverage [3]. In addition, dynamic spectrum management techniques and enhanced energy efficiency optimization can help better achieve the objectives of next generation networks.

As 5G deployments have started gaining momentum worldwide, efficient resource allocation algorithms are indispensable. However, considering the diversity in use cases and applications, as well as the variety of devices and entities in the IoT ecosystem, distributed algorithms are more desirable than centralized approaches. In order to design such algorithms, we need powerful mathematical tools that can model the strategic decision making of different sets of autonomous entities. To that end, this dissertation draws from the potentials of *game theory* and Machine Learning (ML), which have attracted a lot of attention in the recent past, as two promising mathematical frameworks that can model strategic decision making scenarios.

1.2 Game Theory for Next Generation Wireless Networks

Game theory consists of a set of tools that help us analyze the rational objectives of decision makers while strategically considering the behaviors of other decision makers [4]. It came into existence as a separate field when John von Neumann published his paper on game theory in 1928. This was followed by his book in 1944, co-authored with Oskar Morgenstern, which dealt with finding solutions for two-person zero-sum games. In 1950, John Nash developed the *Nash equilibrium*, a criterion for the mutual consistency of the strategies of players. He proved that for every non-zero-sum non-cooperative finite n-player game, an equilibrium solution concept called the Nash equilibrium exists in mixed strategies [5].

Game theory flourished as a field in the following years, which saw extensive development in the theoretical aspects as well as in its application in various fields. A standard example of a game analyzed in game theory is the *prisoner's dilemma*, which examines the conflict of interest between two rational individuals. In this classical example, a good decision which is acceptable to both individuals, requires coopera-

tion between them [5]. A lot of real world situations can be modeled by the prisoner's dilemma, and many other games like Bayesian games, differential games, evolutionary games, etc. Game theoretic frameworks help represent many real life situations due to their potential in modeling logical decision making scenarios, and hence, find applications in numerous fields like politics, economics, psychology, sociology, and computer science, to name a few.

The global reach of Internet for communication and the seamless availability of mobile broadband in different parts of the world have triggered the development of distributed and heterogeneous communication networks across the globe. This decentralized nature of communication systems has resulted in the emergence of a large number of autonomous entities with their own selfish and diverse objectives. In this light, research on the scope of game theory in modeling wireless communication scenarios has seen a massive surge in the past years.

The scope and applications of game theory in wireless communications and networking have been discussed comprehensively in [5]. Modeling the power control problem in mobile networks using a non-cooperative game and solving it by finding the Nash equilibrium, is an example. A lot of similar and other kinds of issues in wireless networks, which involves strategic interactions between decision making entities, can be modeled using game theory. Considering the need for efficient distributed frameworks for modeling the interactions between different sets of entities in the next generation networks, we utilize the potentials of game theoretic models in this dissertation.

1.3 Machine Learning for Next Generation Wireless Networks

Another set of powerful mathematical tools which has attracted a lot of attention in the past decade is ML. ML is the science of programming computers to perform tasks based on the patterns from past experience or sample data [6]. The pioneering work on ML set off in the 1950s, and the term *Machine Learning* was coined in 1959 by Arthur Samuel. The first *learning machine* that was capable of learning and becoming artificially intelligent was proposed in 1950 by Alan Turing.

The following years saw a monumental growth in the development of ML concepts and algorithms like Recurrent Neural Networks (RNNs), Reinforcement Learning (RL), and Support Vector Machines (SVMs), to list a few. ML algorithms are of different types based on their learning approach like *supervised learning*, *unsupervised learning*, *self learning*, *feature learning*, etc. ML has significant overlap with *optimization*, in the sense that many learning problems are modeled as the minimization problem of a *loss function*. ML also shares some aspects with *data mining*, which deals with finding unknown properties in the data, in the sense that data mining methods are employed in ML as unsupervised learning.

The applications of ML are ubiquitous in the current big data era, ranging from *computer vision* and *natural language processing*, to user behavior analytics and brain-machine interfaces (BMI). The applications and scope of ML in wireless communications is underlined in [7], specifically for the next generation networks. The role of supervised and unsupervised learning in Cognitive Radio Networks (CRNs), that of unsupervised learning in heterogeneous networks, and that of RL in D2D communication have been discussed in detail. Inspired from the myriad of possibilities presented by ML models as emphasized in [7], this dissertation exploits the potential of ML as a powerful tool for decision making from data or experience.

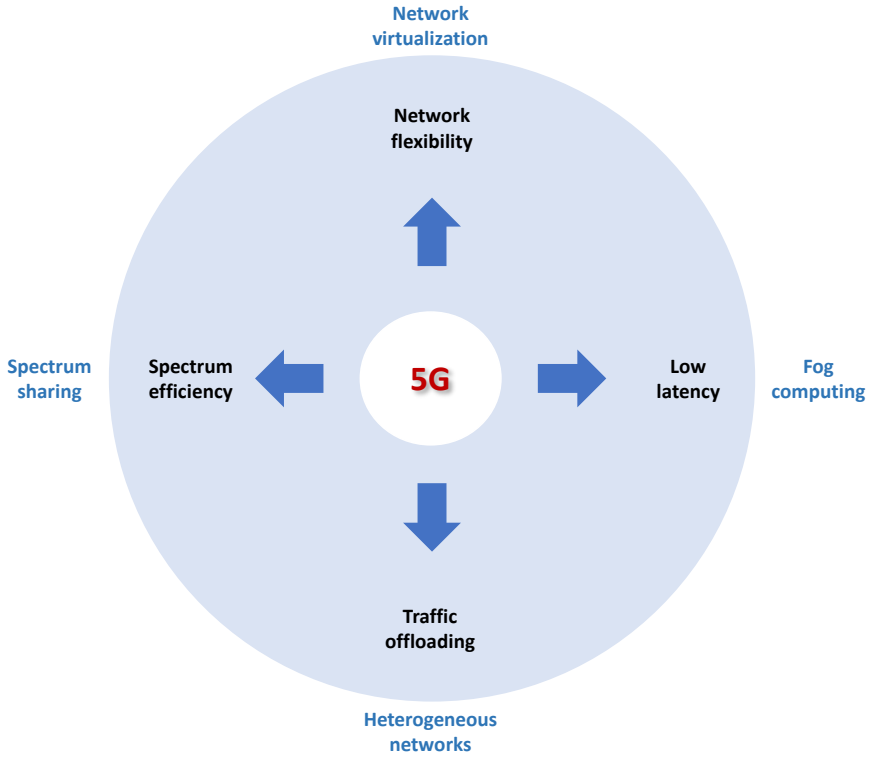


Figure 1.2: Dissertation contribution areas in 5G networks research.

1.4 Game Theory and Machine Learning for Resource Allocation

Keeping in mind the objectives of 5G networks as mentioned in Section 1.1, and the possibilities of game theory and ML as discussed in Section 1.2 and Section 1.3, respectively, this dissertation aims to contribute some efficient algorithms for wireless resource allocation. Specifically, this dissertation focuses on some pivotal concepts mentioned in Section 1.1, and employs the attributes of game theory and ML to the dimensions of next generation networks shown in Fig. 1.2, as follows:

- *Network virtualization*: A number of virtual networks exist on a single physical network, and leads to better flexibility and utilization through network slicing. The interrelationships between different slices can be modeled using game

theory, by considering their localized conditions.

- *Fog computing*: A number of small-scale but flexible computing devices are deployed close to the users, providing low latency and transmission costs. Game theoretic models can be used to represent the strategic interactions between the computing devices and the users, and to achieve distributed allocation of the computing resources.

- *Heterogeneous networks*: The traffic congestion in wireless networks is minimized by offloading some traffic to other kinds of networks. The potential of ML can be employed for learning routing optimization in dynamic heterogeneous network scenarios.

- *Spectrum sharing*: The spectrum usage efficiency is improved by the dynamic management and allocation of spectrum resources. ML based classification approaches can be employed to identify spectrum usage inefficiencies.

1.5 Dissertation Contributions and Organization

This dissertation makes use of the virtues of the game theoretic and ML techniques for achieving efficient resource allocation solutions for some of the key aspects of next generation networks, as mentioned in the previous section. The main contributions of this dissertation can be summarized as follows:

- *A distributed resource allocation solution for wireless network virtualization*: A three-sided resource allocation approach for wireless network virtualization is proposed by considering the user requirements and the virtual network slices.

- *A distributed resource allocation solution for Network Function Virtualization (NFV) in 5G networks*: A three-sided resource allocation approach for NFV is

proposed by considering the user demands of various network functionalities as well as their deployments in 5G networks.

- *Large-scale IoT fog computing pricing and revenue optimization while considering NFV resource requirements*: A large-scale optimization of resource pricing for operators and resource purchasing for users is performed, along with the optimization of fog resource allocation, as per the user demands of network functionalities.

- *Heterogeneous network routing optimization while considering revenue optimization for D2D users*: Data transmission routing optimization for a heterogeneous network involving D2D communication is performed, while simultaneously optimizing the revenue of the mobile users.

- *Enhancing spectrum efficiency of Dynamic Spectrum Access (DSA) by identifying attackers pretending as legitimate users of the spectrum*: An attack detection model for DSA is proposed, to identify malicious users seizing spectrum opportunities from legitimate users, thus improving the spectrum usage efficiency.

Specifically, in Chapter 2, a matching theory based wireless network virtualization resource allocation mechanism is proposed, which is a distributed three-sided matching between radio spectrum, physical infrastructure, and mobile users. A Restricted Three-sided Matching with Size and Cyclic preference (R-TMSC) model is implemented, and is solved by the proposed spectrum-oriented and user-oriented R-TMSC algorithms. The effectiveness of the proposed algorithm is further validated through simulations. Simulation results show that the proposed spectrum-oriented and user-oriented algorithms outperform the traditional resource allocation schemes. The spectrum-oriented algorithm enhances the user throughput and satisfaction within a smaller algorithm run time. Furthermore, as the number of users increases, the proposed algorithms serve more users than traditional methods.

In Chapter 3, we propose a three-sided matching based framework for resource allocation in NFV. We utilize the R-TMSC problem to model the relationships between the Tracking Areas (TAs), Virtual Network Functions (VNFs), and Cloud Networks (CNs) in a 5G Core network. The simulation results clearly demonstrate the superior performance of the proposed framework in terms of the data rates and user satisfaction, compared to a centralized random allocation approach.

In Chapter 4, the competitions between the Data Service Operators (DSOs) and the Authorized Data Service Subscribers (ADSSs) in an IoT fog computing scenario is modeled as an Equilibrium Problem with Equilibrium Constraints (EPEC). As the size of a typical fog computing network is large, the Alternating Direction Method of Multipliers (ADMM) algorithm which has been recognized as a large-scale optimization tool, is invoked to solve the EPEC. This results in the optimization of resource pricing for the DSOs as well as the amount of resources to be purchased by the ADSSs. Further, the resource requirements for different VNFs to be deployed to serve the ADSSs are obtained, and are utilized in a matching theory based approach to allocate the fog computing resources, in a distributed manner. The effectiveness of the proposed framework is then demonstrated through simulations. Simulation results show that the optimization of the utility functions of DSOs and ADSSs can be achieved within a few iterations of the ADMM. The simulations also show that the matching based approach minimizes the cost of fog resource allocation compared to a centralized approach.

In Chapter 5, an RL based approach to determine multi-hop data transmission routes in an indoor Visible Light Communication (VLC)-D2D heterogeneous network is proposed. The rewards for the RL based method is obtained dynamically, by formulating the interactions between the D2D users as an EPEC and using ADMM to solve it. The proposed technique can achieve optimal data transmission routes in a distributed manner. Simulation results demonstrate the effectiveness of the proposed

approach, showing that transmission routes with low delays and high capacities can be achieved through the application of RL and EPEC.

In Chapter 6, the competition between Secondary Users (SUs) and Primary User Emulation (PUE) attackers in a CRN is modeled as a minimax zero-sum game. Thereafter, a Wasserstein Generative Adversarial Network (WGAN) based framework is employed for efficient PUE attack detection in a CRN. The proposed approach determines the cyclostationary features of the signals sensed by SUs, which are input into a Convolutional Neural Network (CNN) based GAN for classification. The performance of the proposed framework is evaluated through simulations. The convergence of the proposed classification model highlights the potential of GAN for the detection of PUE attacks in CRNs.

Finally, the conclusions and some possible directions for future research are discussed in Chapter 7.

Chapter 2

Cyclic Three-Sided Matching Game Inspired Wireless Network Virtualization

2.1 Introduction

Virtualization is becoming an increasingly popular concept, applied in many areas such as virtual memory, virtual machines, and virtual data centers [8]. Network virtualization is the technology in which there exists a number of virtual networks, each of which is a partition or aggregation of the underlying physical substrate network [9]. It involves the abstraction, isolation, and sharing of resources among different entities. This enables supporting heterogeneous applications, without having to modify the existing fundamental architecture. As a result, network virtualization offers great network flexibility, maximizes network utilization, and inspires innovation in products and services [10].

The implementation of virtualization in wired networks, such as in virtual private networks, has prevailed for decades. With the current tremendous growth in mobile wireless traffic, due to the massive user numbers and diverse communication content, it is reasonable to extend virtualization to wireless networks. In wireless networks, virtualization involves the sharing of both infrastructure and spectrum resources [11]. Multiple virtual networks can dynamically share the physical substrate networks, leading to better management of resources and lower operational expenses. This paradigm is commonly referred to as wireless network virtualization [8].

Wireless network virtualization decouples the functionalities in networks by separating the roles of infrastructure and service, thus improving the network utilization. In addition, since resource allocation and management are flexible and more dynamic with virtual resources than physical resources, new network technologies can be de-

ployed easily. However, in spite of the vast potential of wireless network virtualization, several design challenges remain to be addressed, which include the isolation, discovery and allocation of resources, mobility and network management, security and so on [8]. In particular, the resource allocation challenge calls for comprehensive efforts, as it decides how the virtual networks are embedded on top of the physical networks, and thus, directly affecting the network utilization.

A popular way of defining the different roles in wireless network virtualization is by classifying them into Infrastructure Providers (InPs), Mobile Virtual Network Operators (MVNOs), Service Providers (SPs), and end users. Even though [8] discusses the role of Mobile Virtual Network Providers (MVNPs), which lease the physical network resources from the InPs and create virtual resources (and may possess spectrum resources as well), along with the MVNOs who operate and assign these virtual resources to the SPs, *MVNOs* has been discussed as a term used collectively to include both MVNPs and MVNOs. Hence, for brevity and to avoid any confusion, we have used the term MVNOs in the latter sense. That is to say, the InPs own the infrastructure resources, while the MVNOs own the spectrum resources and are responsible for creating and managing the virtual resources (including both infrastructure and radio resources). The SPs then rent/purchase virtual resources from the MVNOs in a wholesale way, and provide specific services such as VoIP, video streaming, etc., to the end users. In short, virtual resources which exist on physical network infrastructures owned by InPs, are created and managed by MVNOs, and are requested by SPs to serve end users.

A traditional resource allocation solution in wireless network virtualization is to configure the virtual resource/service packages first and then offer the off-the-rack services to the users [12]. Such an approach decouples the virtual service generation procedure, which is accomplished by the MVNOs, from the user service management procedure, which is accomplished by the SPs. A *wireless network virtualization con-*

troller acts as a centralized entity through which the MVNOs manage the virtual resources. Henceforth, we will refer to the wireless network virtualization controller as the *wireless network controller* for brevity, as this work deals only with wireless network virtualization. The centralized allocation of the virtual resources using the wireless network controller lacks the flexibility needed to meet user specific requirements and user mobility. Furthermore, resource allocation solutions are moving from the traditional centralized approaches to more distributed methods, considering the high density, mobility, and self-organizing features of next generation wireless networks like device-to-device (D2D) communication, Long-Term Evolution-Unlicensed (LTE-U), and so on. Traditional centralized optimization [13] results in high computational complexity and communication overhead, and hence, results in the need for less complex and distributed solutions.

Matching theory has emerged as a promising approach for future wireless resource allocation, by overcoming some limitations of optimization and game theory [14–17]. The major advantages of matching theory are that we are able to consider individual utilities for the users and the SPs, and that it provides a distributed solution while considering the localized preferences of all the entities [17]. Reference [14] also emphasizes on how the users have preferences on resources and vice versa based on local information, and how the distributed nature of matching takes this into account. It is also highlighted how for every resource allocation problem, there exists at least one stable matching (determined using the *Gale-Shapley* algorithm) due to the *deferred acceptance* method [18]. Reference [15] highlights the stability aspect of matching theory for a non-regulated scenario, and also how it provides a stable resource allocation compared to competitive methods based on game theory.

Matching is a framework that is based on the formation of mutually beneficial relationships between two sets of entities [19–21], and provides mathematically yielding solutions based on the preferences of these entities. The advantages of matching

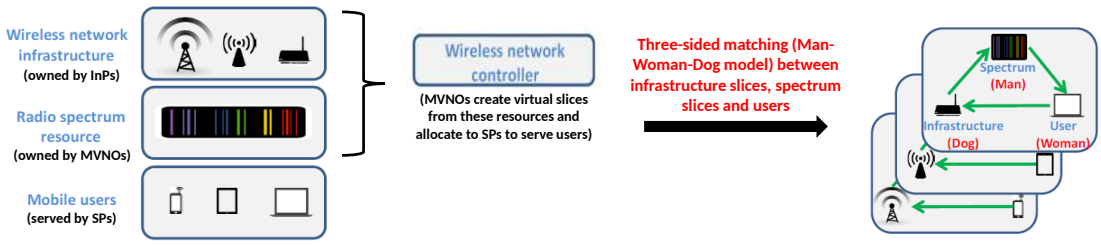


Figure 2.1: System model for wireless network virtualization.

theory in wireless resource allocation have been discussed in detail in [22], which include characterizing the behavior of heterogeneous nodes by suitable models, defining general preferences that can manage Quality-of-Service (QoS) related considerations, obtaining stable and optimal solutions satisfying the system objectives, and implementing efficient algorithms at a faster rate.

In this chapter, we make use of these advantages of matching theory to integrate the dynamics between all the three elements of abstraction in wireless network virtualization, unlike most of the previous works which dealt mainly with SPs and InPs [23]. Accordingly, we propose a matching-based resource allocation framework for wireless network virtualization, which matches three network elements: spectrum, infrastructure, and end users, simultaneously. The *three-sided matching* framework and the corresponding matching-based solution, which have been proposed in this chapter, have the following advantages: (a) the conventional centralized resource allocation decouples the virtual service generation procedure by the MVNOs from the user service management procedure by the SPs, which can yield non-optimal results compared to our coupled three-sided matching framework, where all three entities are considered simultaneously; (b) the time-varying nature of spectrum behavior and the changing user requirements demand continuous adjustments in resource allocation, which can be efficiently achieved by the distributed nature of the matching algorithm. The major contributions of this chapter are briefly summarized as follows:

- We propose a distributed resource allocation framework for wireless network virtualization, in which unlike the conventional decoupled virtual service generation and user service management, we tackle the problem by modeling it as a three-sided matching between radio spectrum, physical infrastructure, and mobile users.

- With joint consideration of user satisfaction, SP revenue, and system cost-performance, we formulate the three-sided matching as an optimization problem, which is *nondeterministic polynomial time* (NP)-hard. Consequently, we model the optimization problem by exploiting the Three-Dimensional Stable Marriage model with Cyclic Preferences (3DSM-CYC), in which each type of agent ranks the other type of agent in its order of preference, and such three preference lists form a cycle[1].

- In order to accommodate virtualization, we consider a variant of the 3DSM-CYC model, the Three-sided Matching with Size and Cyclic preference problem (TMSC), as it allows each agent to have multiple partners. However, since the process of determining whether a stable matching exists for a TMSC model itself is NP-complete, we transform it into a Restricted Three-sided Matching with Size and Cyclic preference problem (R-TMSC) by adding a few plausible restrictions. The R-TMSC model can be solved by the proposed spectrum-oriented and user-oriented R-TMSC algorithms, and a stable solution is always guaranteed. The effectiveness of the proposed algorithm is validated through simulations.

The rest of this chapter is organized as follows. We discuss some of the important previous work relevant to our research in Section 2.2. In Section 2.3, we present the system framework and assumptions for addressing the resource allocation

[1]For example, spectrum ranks only user, user ranks only infrastructure and infrastructure ranks only spectrum.

problem in wireless network virtualization. Here, two important performance metrics are discussed in Section 2.3.1 and Section 2.3.2. Then, in Section 2.4, we formulate the proposed model as an optimization problem, with the objective of maximizing the system cost-performance. The three-sided matching-based approach to solve the optimization problem in a distributed way is explained in Section 2.5. In this section, we discuss the concept of stability in Section 2.5.1, the TMSC model in Section 2.5.2, the R-TMSC model in Section 2.5.3, the spectrum-oriented R-TMSC model in Section 2.5.4, and the user-oriented R-TMSC model in Section 2.5.5. We discuss the performance of the proposed algorithm through simulation results in Section 2.6. Finally, conclusions are drawn in Section 2.7.

2.2 Related Work

Since wireless network virtualization is considered to be a feasible method to achieve better spectrum efficiency, higher data rate, and lesser cost per bit in fifth generation (5G) networks [13, 24], a great amount of related research on resource allocation has been going on [25–33]. A Software Defined Networking (SDN) based framework for resource allocation in wireless network virtualization is proposed in [34], where the allocated resources are adjusted dynamically according to the service requirement and network status variations. A Network Virtualization Substrate (NVS) for optimal virtualization of wireless resources in cellular networks is designed and implemented in [35]. In [36], network slicing in 5G is discussed, where the issue of network resource allocation is dealt with using an algorithm for handling network slice requests. References [37], [38] and [39] also discuss network slicing in 5G, focusing on enabling end-to-end network slicing, and on an auction based model for maximizing the network revenue, and on dynamic allocation of network resources to different slices, respectively. Network slicing for Content Delivery Networks (CDN)

is discussed in [40].

Due to the tremendous potential of matching theory in wireless resource allocation scenarios as discussed in [22], methods to attain optimal resource allocation in wireless network virtualization using matching theory have been prevalent. A novel two-level hierarchical matching algorithm to separately achieve revenue maximization for the InPs and MVNOs has been proposed in [41], by formulating service selection and resource purchasing as a combinatorial optimization problem. The associations between users and Base Stations (BSs) have been formulated as a one-to-many matching game, and a distributed algorithm has been proposed, that results in stable user-BS matchings [42]. Reference [43] also proposes a matching game based resource allocation scheme, simultaneously taking into account the objectives of the InPs and the multiple network operators. In [44], the stable marriage model was employed in the resource allocation problem, to attain matchings between multiple repeaters and vehicle antennas. Reference [45] utilizes matching theory to arrive at stable two-sided matchings between different kinds of files generated by source nodes and relay nodes that forward these files, in Delay Tolerant Networks (DTN). A framework was proposed to find stable matchings of users and resources based on the channel and context aware preference lists in [14]. A route level resource allocation algorithm was proposed for dynamic topology, through a stable and fair allocation utilizing the stable matching algorithm [46]. A framework utilizing matching theory, for Cognitive Radio Networks (CRNs), was proposed in [47], for content-caching was proposed in [48], and for LTE-U was proposed in [49].

Even though all of the above mentioned works discussed the application of matching theory in wireless network virtualization resource allocation, the optimization for only two sets of entities have been considered at a time for resource allocation: InPs and MVNOs, users and BSs, and so on. Therefore, in this chapter, we are motivated to address resource allocation in wireless network virtualization by taking into

account the three entities of abstraction: radio spectrum, physical infrastructure, and mobile users. This calls for a three-sided matching model, unlike the two-sided matching game that has been exploited mostly in the literature. Consequently, we will be able to couple the virtual service generation by the MVNOs with the user service management by the SPs.

Three-sided matching succeeds in modeling many real life situations like the supplier-firm-buyer model [50] etc. A cyclic three-sided stable matching approach for networking services has been discussed for the first time in [51], where a three-sided matching problem has been formulated, by considering the cyclic three-sided preferences in computer networking systems. The NP-completeness of determining the existence of stable matching has been proved, and a restricted version of the three-sided matching algorithm has been designed. This restricted algorithm has been proved to arrive at stable matchings, and the effectiveness of the algorithm has been shown through simulations in [51].

As discussed above, many of the existing research works have considered two-sided matching games to achieve optimal resource allocation in wireless network virtualization. However, according to my knowledge, a three-sided matching based approach considering the preferences of three sets of entities has not been considered for resource allocation in wireless network virtualization. Therefore, along the lines of the cyclic three-sided matching discussed in [51], we model the typical wireless network virtualization scenario as a three-sided matching game between radio spectrum, physical infrastructure, and mobile users. We propose a restricted three-sided formulation in order to always achieve stable results, and propose spectrum-oriented and user-oriented algorithms to arrive at stable matchings.

2.3 System Model

As already discussed, in wireless network virtualization, the MVNOs create virtual resources from the physical infrastructure and radio spectrum resources, which are then allocated to the SPs to serve the end users. The infrastructure and radio resources are abstracted and split into *slices* by the MVNOs to facilitate virtualization [11]. These slices are then utilized to serve the users by the SPs, thus ensuring isolation from the underlying physical networks. Traditionally, the virtual resource allocation and management are centrally handled by the wireless network controller, as mentioned in Section 2.1, which decouples the virtual service generation from the user resource management. Therefore, we propose a distributed resource allocation framework which addresses this issue by modeling the virtual resource allocation as a three-sided matching between the radio spectrum slices, physical infrastructure slices, and mobile users. Even though the proposed approach gives a distributed solution by considering the localized preferences of the parties involved, the wireless network controller can still act as the entity to run the three-sided matching algorithm once the preferences are collected, thus managing the virtual resources.

To this end, we consider a wireless network virtualization scenario as shown in Fig. 2.1, with a set of K spectrum band slices, $\mathcal{S} = \{s_1, s_2, ..., s_K\}$, and a set of N physical infrastructure slices, $\mathcal{B} = \{b_1, b_2, ..., b_N\}$. For brevity, the spectrum band slices will be referred to as spectrum bands and the physical infrastructure slices will be referred to as infrastructures henceforth. All the spectrum bands are assumed to have identical bandwidth, and the infrastructures include BSs, access points, core network elements and so on. The set of subscribed mobile users is represented by $\mathcal{U} = \{u_1, u_2, ..., u_M\}$, where M is the number of all users subscribed to one particular SP. The three-sided matching between $\mathcal{S}$, $\mathcal{B}$ and $\mathcal{U}$ can be represented by $\mathcal{M} \subseteq \mathcal{S} \times \mathcal{B} \times \mathcal{U}$. Henceforth, we call $\mathcal{S}$, $\mathcal{B}$ and $\mathcal{U}$, the matching agents.

Each spectrum band can be shared between multiple infrastructures, and is limited by its maximum capacity, $q^{s_k} = q^s, \forall k \in \{1, 2, \ldots, K\}$ in its allocation to the users. On the other hand, each infrastructure is shared between multiple spectrum bands, and is limited by its maximum capacity, $q^{b_j} = q^b, \forall j \in \{1, 2, \ldots, N\}$. In addition, any particular spectrum band assigned to any particular infrastructure can be shared between multiple users. In other words, the matching between the spectrum bands, $\mathcal{S}$, and the infrastructures, $\mathcal{B}$, is a many-to-many matching, while the matching between the (spectrum band ($\mathcal{S}$), infrastructure ($\mathcal{B}$)) pairs and the users, $\mathcal{U}$, is a one-to-many matching.

We begin by defining the performance metrics from the following two perspectives: user experience and SP revenue.

2.3.1 User Experience

One of the most important aspects of wireless services, which the SPs are concerned about is the user satisfaction or user experience. In order to enhance user satisfaction, we can consider the users' Signal to Interference Noise Ratio (SINR) as the key metric, as it decides the bounds of the channel capacity, and hence, the quality of the wireless service. Since the channel condition primarily depends on the transmitter and the receiver rather than the characteristics of the utilized frequency band, we define user experience as the SINR between the user and the infrastructure. In this chapter, we specifically deal with the uplink transmission from the user to the infrastructure. Hence, the SINR will be that received at the infrastructure. It can be represented as

$$\Gamma_{i,j} = \frac{P_{i,j} g_{i,j}}{\sigma_I^2 + \sigma_N^2}, \tag{2.1}$$

$\forall i \in \{1, 2, \ldots, M\}$, and $\forall j \in \{1, 2, \ldots, N\}$, where $\Gamma_{i,j}$ is the received SINR of infras-

tructure b_j from user u_i. $P_{i,j}$ and $g_{i,j}$ are the transmitted power and the channel gain between u_i and b_j, respectively. σ_I^2 represents the channel interference from the other mobile users due to channel reuse, and σ_N^2 represents the channel noise.

2.3.2 SP Revenue

Another factor that we use to measure the system performance is the revenue that the SPs earn from the users. The mandatory revenue is the incentive that motivates SPs to provide better service to their subscribed users. We assume that each user offers a price based on its desired rate and requirements. Hence, SPs would naturally prefer serving the users with higher offers. We define the SP's revenue, R_{SP}, as the summation of prices offered by the matched users minus the summation of the costs paid to the MVNOs for the matched spectrum resources, which can be represented as

$$R_{SP} = \sum_{u_i \in \mathcal{U}} O_i - \sum_{s_k \in \mathcal{S}} C_k = \sum_{u_i \in \mathcal{U}} \alpha r_i - \sum_{s_k \in \mathcal{S}} C_k, \tag{2.2}$$

$\forall i \in \{1, 2, \ldots, M\}$, and $\forall k \in \{1, 2, \ldots, K\}$, where O_i is the price that user u_i offers to all spectrum bands, based on its desired transmission rate r_i, α is the price per Mbps, C_k is the price paid to the MVNO for spectrum band s_k.

2.4 Problem Formulation

In the previous section, we discussed two performance metrics, which are both essential for a good resource allocation scheme in wireless virtual networks. The system objective in this chapter is designed as a combination of both performance metrics. We define our system objective as the *cost-performance* under the three-sided matching, CP_{sys}, which is represented as

$$CP_{sys} = \frac{\sum CP(i)}{M}, \tag{2.3}$$

$\forall i \in \{1, 2, \ldots, M\}$. Here, the cost-performance of the system, CP_{sys}, is the average of the cost-performance values of all the users, where the cost-performance value of user u_i, $CP(i)$, is given by

$$CP(i) = \frac{\sum \rho_{i,j,k} s_k \log(1 + \Gamma^k_{i,j})}{O_i}, \tag{2.4}$$

$\forall i \in \{1, 2, \ldots, M\}$, $\forall j \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, K\}$. Here $\rho_{i,j,k}$ is a binary value, which is equal to 1, if user u_i is utilizing frequency band s_k for its downlink transmission through infrastructure b_j, and 0, otherwise. $\Gamma^k_{i,j}$ represents the actual SINR of user u_i, if matched with infrastructure b_j and spectrum s_k (also considering the interference from other users that share the same s_k and b_j), which is represented as

$$\Gamma^k_{i,j} = \frac{P_{i,j} g_{i,j}}{\sigma_I^2 + \sigma_N^2} = \frac{P_{i,j} g_{i,j}}{\sum_{i' \neq i} \rho_{i',j,k} P_{i',j} g_{i',j} + \sigma_N^2}. \tag{2.5}$$

Taking (2.3), (2.4) and (2.5) into consideration, we formulate the optimization problem for our scenario, which is expressed as

$$\max_{\rho_{i,j,k}} \; CP_{sys}, \tag{2.6}$$

$$\text{s.t.} \; \sum_{i,j} \rho_{i,j,k} \leq q^s, \tag{2.7}$$

$$\sum_{i,k} \rho_{i,j,k} \leq q^b, \tag{2.8}$$

$$\Gamma_{i,j} \geq \Gamma_{min}, \text{ and} \tag{2.9}$$

$$\rho_{i,j,k} \in \{0, 1\}, \tag{2.10}$$

$\forall i \in \{1, 2, \ldots, M\}$, $\forall j \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, K\}$. Here (2.6) is the system objective, which aims at maximizing the overall cost performance of the system, which is equivalent to the data rate attained per unit price paid by user u_i. (2.7) and (2.8) satisfy the capacity constraints for spectrum s_k and infrastructure b_j, respectively, where $\rho_{i,j,k}$ is the binary value indicating downlink transmission, and q^s and q^b are the maximum capacities of each spectrum and each infrastructure, respectively. (2.9) states the minimum SINR requirement for the selection of infrastructure b_j by user u_i, where Γ_{min} is the minimum SINR threshold.

Obviously, this optimization problem is a Mixed Integer Non-Linear Programming (MINLP) problem[2], which is generally NP-hard to solve [52]. This motivates us to adopt a feasible suboptimal solution. Therefore, we introduce the matching-theory based distributed approach, the 3DSM model, which will be discussed in the next section.

2.5 Three-sided Stable Matching Game

Three-sided relationships are very common in the social and economic domains, e.g., the supplier-firm-buyer relationship, the kidney exchange problem, and so on. Generally, the three-sided matching can be treated as the three-dimensional generalization of the Stable Marriage (SM) model [20], where the three types of matching agents can be considered as men, women and dogs. This three-dimensional variant of SM is usually referred to as a 3DSM problem. The 3DSM problem, also referred to as the Three Gender Stable Marriage problem, was introduced by Knuth [53].

Primarily, there are two models of the 3DSM problem, depending on the nature of the agents' preference lists. For the first model, each agent might rank in the order of preference, the pairs of other agents that they are ready to form triples with. In

[2]The nonlinearity is caused by $\Gamma^k_{i,j}$ in the system objective.

the second model, the preference lists of each type of agents include only one type of agents (e.g., men rank only women in the order of preference, women's lists contain only dogs, and dogs rank only men), and is referred to as the 3DSM-CYC problem.

The 3DSM-CYC model was introduced by Ng and Hirschberg [54], as a restriction on the 3DSM model. As an intriguing variant of 3DSM, the 3DSM-CYC problem refers to the case in which the matching agents' preference lists comprise of only one type of agents (instead of pairs of agents). However, the problem of determining whether a given instance of 3DSM-CYC admits a strongly stable matching is NP-complete as studied by [55].

2.5.1 Stability

Consider the matching $\mathcal{M}$, as mentioned in Section 2.3. Let $\mathcal{T} = \mathcal{S} \times \mathcal{B} \times \mathcal{U}$ denote the set of all possible triples. Hence, the matching $\mathcal{M} \subseteq \mathcal{T}$, is a set of triples from $\mathcal{T}$. In order to understand the concept of stability for a three-sided matching, we need to understand the idea of a *blocking triple*, which is as given in Definition 3.1.

Definition 2.1. ***Blocking Triple in 3DSM:*** *A triple* $(s_k, u_i, b_j) \notin \mathcal{M}$*, but* $(s_k, u_i, b_j) \in \mathcal{T}$*, in which each of* s_k*,* u_i*, and* b_j*, prefers triple* (s_k, u_i, b_j) *to at least one of their current matched partners.*

To elaborate, a blocking triple consists of a spectrum, a user and an infrastructure, each of which has the desire to get matched with each other as a triple, instead of staying with the current matched partners in $\mathcal{M}$. A matching $\mathcal{M}$ is said to be *stable* if there exists no blocking triple for $\mathcal{M}$ [51].

2.5.2 TMSC Model

In [51], Cui and Jia studied an interesting variant of the 3DSM-CYC model, the TMSC problem for three-sided networks. TMSC is different from traditional three-sided matching problems, in that it allows each agent to have multiple partners.

We use our spectrum-user-infrastructure instance to explain the TMSC model. In this instance, we assume that spectrums only rank users, users only rank infrastructures, and infrastructures only rank spectrums in their orders of preferences. Each agent can be matched up to a limited number of the other type of agents, that it ranks in the order of preference. The detailed definition of the TMSC model is given in Definition 2.2.

Definition 2.2. ***Three-sided Matching with Size and Cyclic Preference Problem:*** *The three-sided matching problem of TMSC is to find a matching* $\mathcal{M} = \{(s_k, u_i, b_j)\}$ *with the maximum cardinality:*

$$\boldsymbol{max}\ |\mathcal{M}|, \tag{2.11}$$

$$\boldsymbol{s.t.}\ \mathcal{N}(\mathcal{M}, s_k) \leq q^s, \tag{2.12}$$

$$\mathcal{N}(\mathcal{M}, u_i) \leq q^u,\ \textit{and} \tag{2.13}$$

$$\mathcal{N}(\mathcal{M}, b_j) \leq q^b, \tag{2.14}$$

$\forall i \in \{1, 2, \ldots, M\}$, $\forall j \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, K\}$, where $\mathcal{N}(\mathcal{M}, x)$ represents the number of partners that x has in the matching $\mathcal{M}$[3]. (2.11) represents the cardinality of the matching $\mathcal{M}$ (the number of (s_k, u_i, b_j) triples in the matching). (2.12), (2.13) and (2.14) represent the constraints due to the maximum capacities of spectrum, user and infrastructure, s_k, u_i and b_j, respectively. Here, q^s and q^b are as mentioned in Section 2.3. q^u can be considered as the maximum budget of each user, to purchase services from the SPs.

[3]Here partner refers to an agent of the type of agents in x's preference list.

TMSC is however, NP-hard [51]. Biro and McDermid studied in [55], that the problem of deciding whether a stable matching exists in an instance of the Cyclic 3DSM problem with Incomplete lists (Cyclic 3DSMI) is NP-complete. TMSC is a generalization of the 3DSMI problem according to [51], and hence, the same applies to TMSC.

2.5.3 R-TMSC Model

As discussed above, even the process of determining whether a stable matching exists for a TMSC model is NP-complete. Hence, we consider techniques to refine the TMSC model to make it easily solvable. Therefore, we add a few reasonable restrictions as given below, and transform the TMSC problem into an R-TMSC problem: (1) The preference lists of spectrums are derived from a master preference list. This master list is the set of all users in strict order (e.g., according to the prices offered), and the preference lists of all spectrums are derived from this master list, including all or just part of it; (2) The infrastructures are indifferent with the spectrums, i.e., for each infrastructure, the spectrums in its preference list form one tie. We refer to this model, satisfying both (1) and (2), as the R-TMSC model. This model will be discussed and modified to be implemented in our wireless network virtualization resource allocation problem. Finding the maximum cardinality matching of the R-TMSC problem is still NP-hard as proved in [51].

Taking the above mentioned restrictions into consideration, we build the R-TMSC model for our scenario. Firstly, we construct the preference lists for each spectrum, user and infrastructure. As mentioned before, in the cyclic preference problem, the preference lists of each type of agents include only one type of agents. Therefore, the preference lists of spectrums consist of only users, users' preference lists contain only infrastructures, and infrastructures' lists are comprised of only spectrums, all in the order of preference.

The preference list of each spectrum over the users is derived from a master list, that ranks the users according to their offer prices, O_i, in descending order[4]. The users who demand higher data rates will offer higher prices, and are more preferred by the spectrums. All spectrums' preference lists are derived from the master list, and in our case, all spectrums create identical preference lists (we assume that all users are acceptable by all spectrums) as

$$PL_s(k, i) = O_i, \tag{2.15}$$

$\forall i \in \{1, 2, \ldots, M\}$, and $\forall k \in \{1, 2, \ldots, K\}$.

On the other hand, the users rank the acceptable infrastructures according to the service quality (the acceptable set is generated by applying (2.9)), which is measured by SINR $\Gamma_{i,j}$[5]. The SINR in turn decides the data rates for the wireless service, and thus, the users indirectly choose the infrastructures according to the expected data rates. We denote the preference lists for users as

$$PL_u(i, j) = \Gamma_{i,j}, \tag{2.16}$$

$\forall i \in \{1, 2, \ldots, M\}$, and $\forall j \in \{1, 2, \ldots, N\}$.

According to the R-TMSC model, the infrastructures are indifferent with the spectrums. In other words, the preference list of any infrastructure consists of a tie, with all spectrums ranked the same, which can be represented as

$$PL_b(j, k) = 1, \tag{2.17}$$

$\forall j \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, K\}$.

[4]Here, since it is the SPs who provide services using the purchased spectrum bands, it is basically the SPs that rank the users.

[5]We assume the interference, $\sigma_I^2 = 0$ in building the preference lists, since the matching actions of other users are not known in advance to any user.

2.5.4 Spectrum-oriented R-TMSC

After finishing the generation of all the agents' preference lists, we propose our spectrum-oriented R-TMSC algorithm. Slightly different from the R-TMSC algorithm discussed in [51], we tailor it to fit our problem setting. Before moving on to the algorithm, we define the following sets for an instance of R-TMSC and matching $\mathcal{M}$:

$$A^{+1}(\mathcal{M}, s_k) = \{u_i | u_i \succ_{s_k} \mathcal{M}(s_k), u_i \in PL_s\} \tag{2.18}$$

denotes the set of all users that spectrum s_k prefers to its current partner $\mathcal{M}(s_k)$,

$$A^{+1}(\mathcal{M}, u_i) = \{b_j | b_j \succ_{u_i} \mathcal{M}(u_i), b_j \in PL_u\} \tag{2.19}$$

denotes the set of all infrastructures that user u_i prefers to its current partner $\mathcal{M}(u_i)$,

$$A^{-1}(\mathcal{M}, s_k) = \{b_j | b_j \in \mathcal{B}, s_k \in PL_b, \mathcal{N}(\mathcal{M}, b_j) < q^b\} \tag{2.20}$$

represents the set of all infrastructures that still have capacity to accept spectrum s_k, and

$$A^{-2}(\mathcal{M}, s_k) = \{u_i | A^{+1}(\mathcal{M}, u_i) \cap A^{-1}(\mathcal{M}, s_k) \neq \emptyset, u_i \in \mathcal{U}\} \tag{2.21}$$

represents the set of all users, such that there exists an infrastructure b_j that user u_i prefers to its current partner $\mathcal{M}(u_i)$, and infrastructure b_j still has capacity to accept spectrum s_k.

Also, let $SL_u \subseteq PL_u$, $SL_b \subseteq PL_b$, and $SL_s \subseteq PL_s$, respectively, be sub-lists of agents from the preference lists. We define $Head(SL_u, u_i)$ as the elements (infrastructures) in SL_u with the highest priority. Similarly, $Head(SL_b, b_j)$ and $Head(SL_s, s_k)$ represent the spectrums in SL_b and users in SL_s with the highest priority, respectively.

In light of these definitions, the basic idea of the spectrum-oriented R-TMSC algorithm is to search for the "best" triple and add this triple to the matching $\mathcal{M}$ each

time, which starts from an empty set. Each "best" triple (in the form of (u_i, b_j, s_k)) is generated by first selecting a spectrum satisfying certain requirements, and then this selected spectrum chooses the best user that meets its requirements, and finally this selected user picks the most eligible infrastructure. The detailed procedure is described in Algorithm 2.1.

Algorithm 2.1 Spectrum-oriented R-TMSC Algorithm

Input: $\mathcal{U}$, $\mathcal{B}$, $\mathcal{S}$
Output: $\mathcal{M}$

1: **Initialization**;
2: Construct the preference lists PL_u, PL_b, and PL_s;
3: $\mathcal{M} = \emptyset$, $flag = 1$;
4: **while** $flag == 1$ **do**
5: $\quad flag = 0$;
6: $\quad$ **for** each $s_k \in \mathcal{S}$ **do**
7: $\quad\quad \mathcal{U}' = A^{+1}(\mathcal{M}, s_k) \cap A^{-2}(\mathcal{M}, s_k)$;
8: $\quad\quad$ **if** $\mathcal{U}' \neq \emptyset$ **then**
9: $\quad\quad\quad u_i = Head(\mathcal{U}', s_k)$;
10: $\quad\quad\quad \mathcal{B}' = A^{+1}(\mathcal{M}, u_i) \cap A^{-1}(\mathcal{M}, s_k)$;
11: $\quad\quad\quad b_j = Head(\mathcal{B}', u_i)$;
12: $\quad\quad\quad$ **if** $\mathcal{N}(\mathcal{M}, s_k) == 1$ **then**
13: $\quad\quad\quad\quad \mathcal{M} = \mathcal{M} \backslash \{\mathcal{M}(s_k), \mathcal{M}(\mathcal{M}(s_k)), s_k\}$;
14: $\quad\quad\quad\quad flag = 1$;
15: $\quad\quad\quad$ **end if**
16: $\quad\quad\quad$ **if** $\mathcal{N}(\mathcal{M}, u_i) == 1$ **then**
17: $\quad\quad\quad\quad \mathcal{M} = \mathcal{M} \backslash \{u_i, \mathcal{M}(u_i), *\}$;
18: $\quad\quad\quad\quad flag = 1$;
19: $\quad\quad\quad$ **end if**
20: $\quad\quad\quad \mathcal{M} = \mathcal{M} \cup \{u_i, b_j, s_k\}$;
21: $\quad\quad$ **end if**
22: $\quad$ **end for**
23: **end while**
24: Output stable matching $\mathcal{M}$;

Algorithm 2.1 starts with an empty matching $\mathcal{M}$. $\mathcal{U}' = A^{+1}(\mathcal{M}, s_k) \cap A^{-2}(\mathcal{M}, s_k)$, as in line 7, searches for a better triple to improve $\mathcal{M}$. If the *if* statement in line 8 holds true, then the lines till 21 are executed to update $\mathcal{M}$. This is done by selecting a more preferred partner (user) for spectrum s_k as in line 9, and then, selecting a more preferred partner (infrastructure) for that user u_i as in line 11. Finally, this better triple is added to the matching $\mathcal{M}$, as shown in line 20, and this is repeated till we obtain the best triples. This algorithm is called spectrum-oriented R-TMSC matching, since we choose a spectrum first to begin with, and then this spectrum

chooses from its list of preferred users, and the users in turn select their preferred infrastructures.

Theorem 2.3. *The spectrum-oriented R-TMSC algorithm will stop and output a stable matching after a finite number of steps.*

Proof. Algorithm 2.1 refines a list of triples in each iteration and proceeds by adding the best triple to an initially empty matching $\mathcal{M}$, as in line 20. The *while* loop goes on till the $flag$ drops to 0. During each iteration, a user u_i is assigned to a better infrastructure b_j in its preference list. Let spectrum s_k be matched to a user, say u_x, but while doing these operations, this $u_x = \mathcal{M}(s_k)$ will be unmatched. Necessarily, u_i is better than u_x for s_k, i.e., $u_i \succ_{s_k} u_x$. Thus, a higher priority user must be matched to a better infrastructure, whenever a matched user becomes unmatched. As the number of users, and the number of infrastructures in each user's preference list is limited, the algorithm will stop after a finite number of steps. To prove the stability, let us suppose that the output matching $\mathcal{M}$ from Algorithm 2.1 is unstable. This implies that there must be a blocking triple (u_i, b_j, s_k) such that: $s_k \in PL_b$, $\mathcal{N}(\mathcal{M}, b_j) < q^b$, $b_j \succ_{u_i} \mathcal{M}(u_i)$, and $u_i \succ_{s_k} \mathcal{M}(s_k)$. So $u_i \in A^{+1}(\mathcal{M}, s_k)$, $b_j \in A^{+1}(\mathcal{M}, u_i)$, $b_j \in A^{-1}(\mathcal{M}, s_k)$, and $u_i \in A^{-2}(\mathcal{M}, s_k)$. Then $A^{+1}(\mathcal{M}, s_k) \cap A^{-2}(\mathcal{M}, s_k) \neq \emptyset$ and $A^{+1}(\mathcal{M}, u_i) \cap A^{-1}(\mathcal{M}, s_k) \neq \emptyset$. The algorithm will not stop in such a case, and hence, this is a contradiction. Therefore, the output matching $\mathcal{M}$ from Algorithm 2.1 is stable [51]. □

Theorem 2.4. *The spectrum-oriented R-TMSC algorithm can always find a stable matching in* $O(K \sum_{u_i \in \mathcal{U}} |PL_u|)$ *iterations.*

Proof. During each iteration of Algorithm 2.1, at least one user will be assigned to its most preferred infrastructure, if each infrastructure has a large capacity q^b. Hence, the maximum time required for this is decided by the total number of spectrum bands, K, and the total number of users, M, which gives a time complexity of $O(KM)$. However,

when q^b is small, at least one user is assigned to a better infrastructure in its preference list during each for loop till the $flag$ becomes 0 and the algorithm terminates. Even in the worst case, each user is pre-matched to all the infrastructures in its preference list, in the order of preference, while the algorithm runs. As a result, instead of M, the lengths of the preference lists of the users decide the maximum time required, resulting in a time complexity equal to $O(K\sum_{u_i\in\mathcal{U}}|PL_u|)$, where $K\sum_{u_i\in\mathcal{U}}|PL_u| \leq |\mathcal{T}|$ ($|\mathcal{T}|$ is the total number of possible triples) [51]. □

Thus, when the number of entities is finite, we can see that the proposed spectrum-oriented R-TMSC algorithm always arrives at a stable matching in a finite number of steps, which is decided by the number of spectrum bands and the lengths of preference lists of the users, as proved in Theorem 2.3 and Theorem 2.4. The obtained stable matching implies that none of the spectrum-user-infrastructure triples have entities that prefer other partners to the currently matched partners. This in turn implies that the spectrums (SPs) have been matched to users according to their preferred offer prices, and the users have been matched to infrastructures according to their preferred QoS, in a stable manner (the infrastructures are indifferent with the spectrums, as discussed before). This demonstrates the existence of a feasible algorithm considering the network slices (spectrum and infrastructure resources) as well as the users, simultaneously, with an emphasis on the SP (spectrum) perspective.

2.5.5 User-oriented R-TMSC

As in the case of spectrum-oriented R-TMSC, we define the following sets for an instance of user-oriented R-TMSC and matching $\mathcal{M}$:

$$A^{+1}(\mathcal{M}, u_i) = \{b_j | b_j \succ_{u_i} \mathcal{M}(u_i), b_j \in PL_u\} \tag{2.22}$$

denotes the set of all infrastructures that user u_i prefers to its current partner $\mathcal{M}(u_i)$,

$$A^{+1}(\mathcal{M}, b_j) = \{s_k | s_k \in PL_b, \mathcal{N}(\mathcal{M}, b_j) < q^b\} \tag{2.23}$$

denotes the set of all spectrums in the preference list of infrastructure b_j,

$$A^{-1}(\mathcal{M}, u_i) = \{s_k | s_k \in \mathcal{S}, u_i \in PL_s, \mathcal{N}(\mathcal{M}, s_k) < q^s\} \tag{2.24}$$

represents the set of all spectrums that still have capacity to accept user u_i,

$$A^{-2}(\mathcal{M}, u_i) = \{b_j | A^{+1}(\mathcal{M}, b_j) \cap A^{-1}(\mathcal{M}, u_i) \neq \emptyset, b_j \in \mathcal{B}\} \tag{2.25}$$

represents the set of all infrastructures, such that there exists a spectrum s_k in the preference list of infrastructure b_j, and spectrum s_k still has capacity to accept user u_i.

The other definitions are the same as those in the case of spectrum-oriented R-TMSC. The objective of the user-oriented R-TMSC algorithm is also to search for the "best" triple and add this triple to the matching each time, which starts from an empty set. Each "best" triple (in the form of (u_i, b_j, s_k)) is generated by first selecting a user satisfying certain requirements, and then this selected user chooses the best infrastructure that meets its requirements, and finally, this selected infrastructure picks an arbitrary spectrum from its preference list (since we assume plausibly that the infrastructures are indifferent with the spectrums). The detailed procedure is as described in Algorithm 2.2.

Similar to the spectrum-oriented R-TMSC algorithm, Algorithm 2.2 starts with an empty matching $\mathcal{M}$. $\mathcal{B}' = A^{+1}(\mathcal{M}, u_i) \cap A^{-2}(\mathcal{M}, u_i)$ as in line 7 searches for a better triple to improve $\mathcal{M}$. If this set $\mathcal{B}' \neq \emptyset$, then the *for* loop continues to update $\mathcal{M}$. Since we focus on the users to begin with the process, it is called a user-oriented R-TMSC matching. These users then choose from their lists of preferred infrastructures, and the infrastructures in turn select arbitrary spectrums from their preference lists, as they are indifferent with spectrums in R-TMSC.

Algorithm 2.2 User-oriented R-TMSC Algorithm

Input: $\mathcal{U}$, $\mathcal{B}$, $\mathcal{S}$
Output: $\mathcal{M}$

Initialization;
Construct the preference lists PL_u, PL_b, and PL_s;
$\mathcal{M} = \emptyset$, $flag = 1$;
while $flag == 1$ **do**
    $flag = 0$;
    for each $u_i \in \mathcal{U}$ **do**
        $\mathcal{B}' = A^{+1}(\mathcal{M}, u_i) \cap A^{-2}(\mathcal{M}, u_i)$;
        if $\mathcal{B}' \neq \emptyset$ **then**
            $b_j = Head(\mathcal{B}', u_i)$;
            $\mathcal{S}' = A^{+1}(\mathcal{M}, b_j) \cap A^{-1}(\mathcal{M}, u_i)$;
            Select arbitrary s_k from $\mathcal{S}'$;
            if $\mathcal{N}(\mathcal{M}, u_i) == 1$ **then**
                $\mathcal{M} = \mathcal{M} \backslash \{u_i, \mathcal{M}(u_i), \mathcal{M}(\mathcal{M}(u_i))\}$;
                $flag = 1$;
            end if
            if $\mathcal{N}(\mathcal{M}, b_j) == 1$ **then**
                $\mathcal{M} = \mathcal{M} \backslash \{*, b_j, \mathcal{M}(b_j)\}$;
                $flag = 1$;
            end if
            $\mathcal{M} = \mathcal{M} \cup \{u_i, b_j, s_k\}$;
        end if
    end for
end while
Output stable matching $\mathcal{M}$;

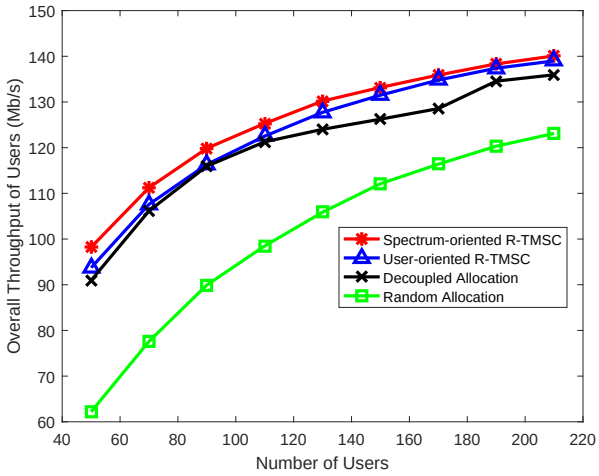
As in the case of spectrum-oriented R-TMSC, the following can be easily proved for user-oriented R-TMSC:

- The user-oriented R-TMSC algorithm will stop and output a stable matching after a finite number of steps.
- The user-oriented R-TMSC algorithm can always find a stable matching in $O(M \sum_{b_j \in \mathcal{B}} |PL_b|)$ iterations.

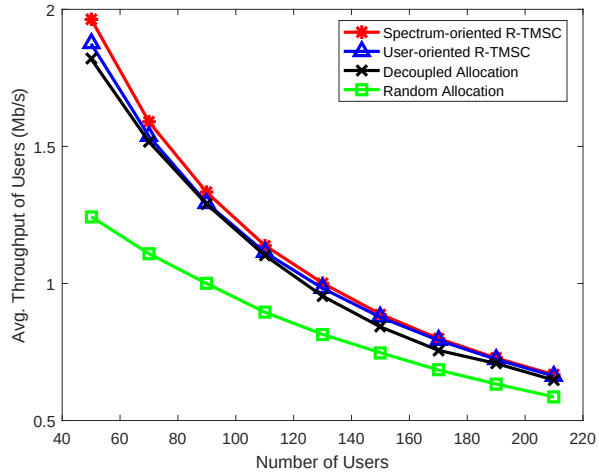
Similar to the spectrum-oriented R-TMSC algorithm, the proposed user-oriented R-TMSC algorithm also always arrives at a stable matching in a finite number of steps, which is decided by the number of users and the lengths of preference lists of the infrastructures. The obtained stable matching implies that the users have been matched to infrastructures according to their preferred QoS, and the spectrums have been matched to users according to their preferred offer prices, in a stable manner. This presents another feasible algorithm from the perspective of the users instead of the SPs.

2.5.6 Convergence

Even though the proposed R-TMSC algorithms are distributed approaches, once the preference lists are created, the matching algorithm can be run offline at an entity like the wireless network controller, and is not iterative. Also, since each entity needs to rank only a few entities which are accessible, the preference lists would not be too long, and the algorithm can converge [51] in a few ms on a large-scale processor (given that the algorithm converged for around 200 users in almost 800 ms in our small-scale processor). However, user mobility can lead to changes in preference lists of different entities. We can either run the three-sided matching algorithm repeatedly, or use algorithms such as the Roth-Vande Vate (RVV) algorithm, which can transform a



(a) Overall user throughput.



(b) Average user throughput.

Figure 2.2: User throughput analysis.

random matching into a stable matching [49], for dynamically adapting to the changes due to user mobility in our future work.

2.6 Performance Evaluation

In this section, we evaluate the proposed spectrum-oriented R-TMSC algorithm by comparing it with the user-oriented R-TMSC algorithm, a *decoupled* allocation, as well as a random allocation, through MATLAB simulations.

The spectrum-oriented R-TMSC algorithm operates by adding a triple to the matching each time, while the triple is generated by finding a qualified spectrum first, and then the best qualified user for this spectrum, and finally the best qualified infrastructure for this user. Similarly, the user-oriented R-TMSC operates by adding one triple each time, but the triple is generated from one qualified user, followed by finding the best qualified infrastructure for this user, and a random qualified spectrum for this infrastructure. We compare the performance of the proposed algorithms with that of a decoupled allocation scheme, which decouples the virtual service generation from the user service management, emulating the traditional centralized allocation by the wireless network controller. For simplicity, we follow the assumption that the infrastructures are indifferent with the spectrums as considered in the R-TMSC scheme, to form spectrum-infrastructure pairs. These resource pairs are then matched with the users using the Gale-Shapley algorithm, which is used to find a stable solution for two-sided matching problems [20]. For comparison purposes, we also consider a random allocation approach, which randomly matches users to spectrum-infrastructure pairs.

We assume a circular cellular network with a radius of $R = 800$ m, consisting of $M \in [50, 210]$ mobile users, $N = 5$ infrastructures and $K = 20$ spectrum bands. The bandwidth of each spectrum band is set to be 5 MHz. The capacity of each infrastructure is 44 Mbps, while the capacity of each frequency band is 11 Mbps. The minimum SINR requirements for all mobile users are set at an identical value of 25 dB. For the propagation gain $g = C\beta\zeta d^{-\alpha}$, we set the path loss constant C as 10^{-2}, the multipath fading gain β as the exponential distribution with unit mean, and the shadowing gain ζ as the log-normal distribution with 4 dB deviation and the path loss exponent α as 4.

In Fig. 2.2a and Fig. 2.2b, the overall and average throughput of users are evaluated. We increase the user numbers from 50 to 210 by a step size of 20. As shown

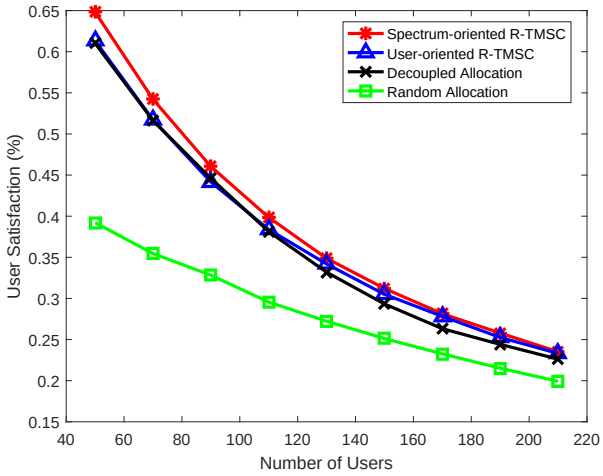


Figure 2.3: User satisfaction.

in Fig. 2.2a, the network throughput increases under all four schemes as more users join the network. It is reasonable, since spectrum is reused between users who share the same infrastructure and spectrum, which improves the spectrum efficiency. On the other hand, Fig. 2.2b shows that the average user throughput decreases as more users get matched to the available spectrum and infrastructure resources. It is due to the interference caused by the users who share the same resources. We can also observe from Fig. 2.2a and Fig. 2.2b that spectrum-oriented R-TMSC outperforms user-oriented R-TMSC slightly, and both outperform the decoupled and random allocations.

Fig. 2.3 gives another insight on the system performance from the perspective of user satisfaction. In this chapter, we consider the user satisfaction percentage as the ratio between the actual transmission rate and the expected transmission rate. As discussed in Section 2.1, users make offers to the SPs according to the expected rates. As a result, the users who have higher rate demands will offer higher prices, and thus, are more preferred by the SPs and are better served by allocating resources. It is obvious that as more users join, the user satisfaction decreases. With more users sharing the same radio and infrastructure resources, the interference grows,

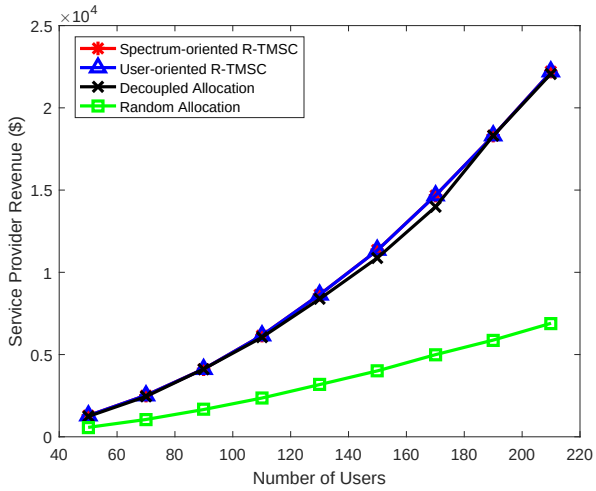


Figure 2.4: SP revenue.

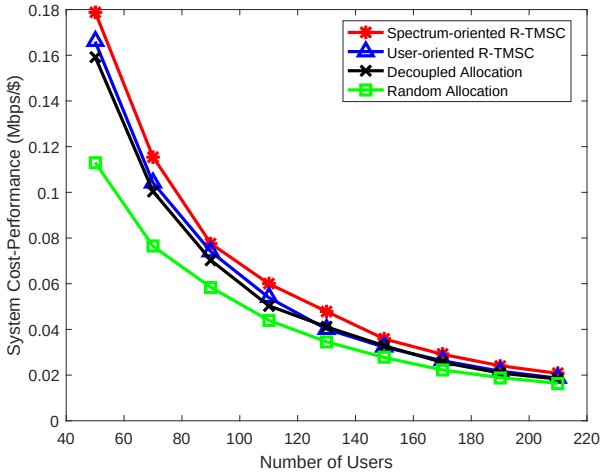


Figure 2.5: System cost-performance.

leading to a performance degradation. However, the spectrum-oriented algorithm still outperforms the user-oriented one, and both demonstrate better results than the decoupled and random allocations.

Fig. 2.4 compares the four methods in terms of the revenue of the SPs. The SP revenue is calculated as the total income obtained by providing service to matched users using the purchased spectrum resources. Accordingly, more users, more overall revenue. We can see that, apart from random allocation, the other three algorithms achieve more or less the same SP revenue.

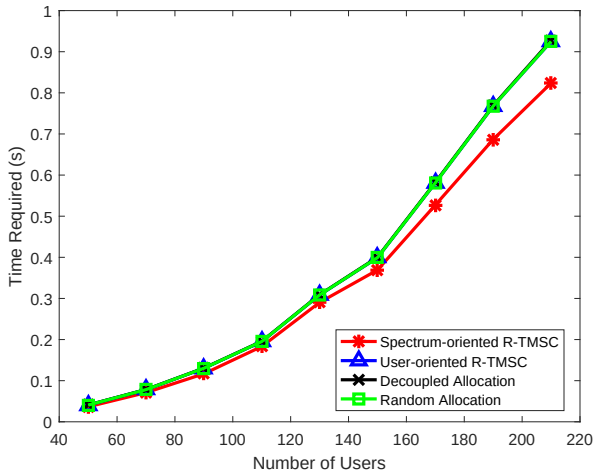


Figure 2.6: Algorithm run time.

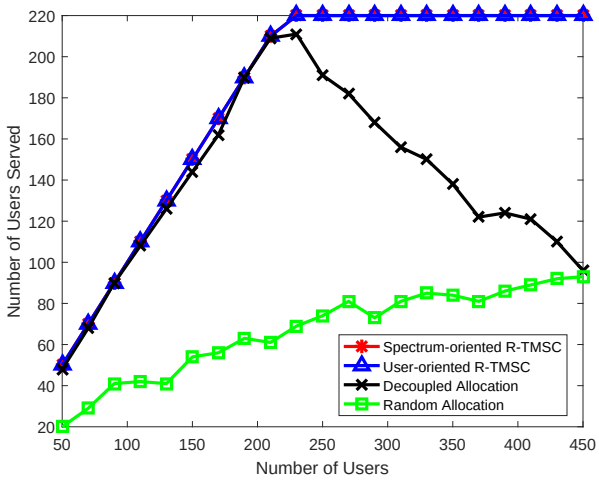


Figure 2.7: Cardinality of output matching.

In Fig. 2.5, we analyze the cost-performance of the system. As defined in Section 2.4, the system objective is to optimize the system cost-performance, which is the actual transmission data rate of each user over its offer price, averaged over all users (average data rate/unit price). The cost-performance metric not only indicates how good the users are performing, but also conveys the benefits earned by the SPs. As can be seen from the figure, it decreases as more users join. This is caused by the average user throughput decrease as indicated in both Fig. 2.2b and Fig. 2.3. The spectrum-oriented algorithm again proves itself to be better than the user-oriented

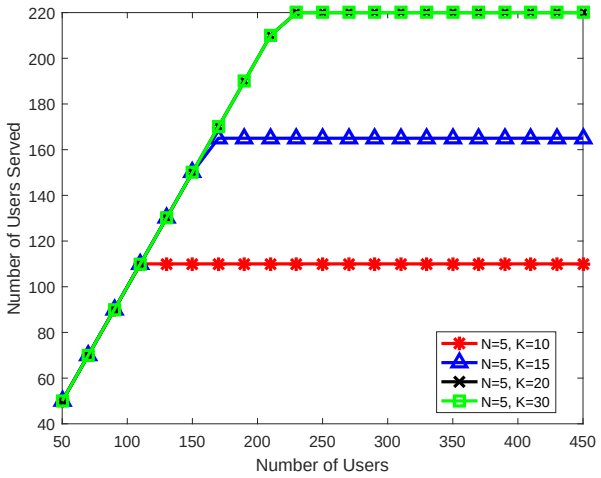


Figure 2.8: Cardinality of spectrum-oriented R-TMSC.

algorithm, and both matching algorithms beat the decoupled and random allocations.

Fig. 2.6 compares the run times of the four algorithms for $M \in [50, 210]$ mobile users, $N = 5$ infrastructures and $K = 20$ spectrum bands. Undoubtedly, the execution time increases as the number of users increases. The difference in run times between the spectrum-oriented algorithm and the other schemes also grows with the number of users. Besides, spectrum-oriented R-TMSC takes 100 ms less than the other algorithms to finish, which is a huge margin in the wireless communication scenario.

Fig. 2.7 illustrates the cardinality of the output matching on the number of users, which indicates the number of users served. We increase the user numbers from 50 to 450 by a step size of 20. We can observe from the figure that the spectrum-oriented R-TMSC, user-oriented R-TMSC and the decoupled methods serve all the users, till the number of users is almost 200. Thereafter, the spectrum-oriented and user-oriented algorithms level off at serving around 220 users, as the number of users increases further. This is due to the limited spectrum and infrastructure resources available. The decoupled allocation has a falloff after around 220 users. The random allocation performs poorly throughout.

Fig. 2.8 shows the cardinality (number of users served) of the spectrum-oriented R-TMSC algorithm on the number of users for four different cases: $K = 10, K = 15, K = 20$ and $K = 30$. $N = 5$ in all of these cases. It can be noticed from the figure that all the users are served in all four cases, until the number of users reaches a particular value. For the $K = 10$ case, the maximum number of users served is 110, whereas for $K = 15$, it is 165, and it is 220 for the $K = 20$ and $K = 30$ cases. Evidently, for the given number of infrastructures ($N = 5$), as the number of spectrum bands increases, more number of users can be served.

2.7 Conclusion

In this chapter, we propose a matching-based framework for resource allocation in wireless network virtualization. Utilizing a variant of the 3DSM model, the R-TMSC model, we formulate the relations between the radio resources, physical infrastructure and mobile users. The proposed spectrum-oriented and user-oriented R-TMSC algorithms are proved to always generate stable matching results in a finite number of steps. Simulation results validate the effectiveness of the proposed matching-based approaches compared to the traditional centralized methods. The spectrum-oriented R-TMSC algorithm enhances the user throughput and satisfaction, as well as the system cost-performance. It also runs faster than traditional methods, with the run time margin increasing along with the number of users. Moreover, for a given amount of resources, the proposed algorithms serve more number of users than the traditional decoupled and random allocations.

Chapter 3

Virtual Core Network Resource Allocation in 5G Systems using Three-Sided Matching

3.1 Introduction

One of the implementations of virtualization in next generation mobile networks is the paradigm known as Network Function Virtualization (NFV). NFV has been coined as a technique to decouple physical network infrastructure from the network functions that run on top of it [56]. NFV allows various mobile network components to be virtualized and placed as software components on top of a virtualization platform [57]. As a result, NFV facilitates flexible provisioning of network functionalities, thus maximizing the network resource utilization and minimizing the service costs [58].

NFV is among the prime enablers of 5G networks, where a particular service can be disintegrated into a set of Virtual Network Functions (VNFs), which can be implemented in software hosted on Cloud Networks (CNs). Virtualization in 5G systems will cover the Radio Access Network (RAN) and the Core Network (Evolved Packet Core (EPC) and 5G Core). The EPC components which will be virtualized are Mobility Management Entity (MME), Home Subscriber Sub-system (HSS), Serving Gateway (SGW), and Packet Data Network Gateway (PGW). The 5G Core components which are going to be virtualized include Access and Mobility Management Function (AMF), Session Management Function (SMF), Authentication Server Function (AUSF), and User Plane Functions (UPF). Virtual instances of these components will be initiated and hosted in federated cloud networks, which form the virtual EPC (vEPC)/5G Core network [57].

The RAN of next generation networks consists of a set of Tracking Areas (TAs) as per Release 8 of the 3GPP mobile network specifications [57], which are groups

of cells that are under the coverage of a set of eNodeBs (BSs). The MVNO communicates its resource needs based on user traffic requirements in the TAs to the NFV Orchestrator (NFVO), which then converts these resource needs into software component requirements to be initiated in the network. The software components are Virtual Machines (VMs) in the CNs, which host the required instances of VNFs according to the needs of the user traffic in the TAs.

As discussed in Chapter 2, the centralized allocation of virtual resources using the NFVO does not have the flexibility demanded by user specific requirements and user mobility. Moreover, resource allocation solutions are moving towards more distributed methods due to the high computational complexity and communication overhead of centralized allocations. Among these methods, a popular approach is matching theory, which provides distributed solutions while considering the localized preferences of all the entities involved [17]. Accordingly, we propose a resource allocation framework for NFV based on three-sided matching, by considering the interrelationships between the three sets of entities: TAs, VNF instances, and CNs. The contributions of this chapter can be summarized as below:

- A distributed resource allocation framework is proposed for NFV in 5G vEPC, by modeling the interactions between the TAs, VNF instances, and CNs using three-sided matching.
- Similar to Chapter 2, the TMSC model is considered here, as it allows each agent to have multiple partners. However, since the process of determining whether a stable matching exists for a TMSC model itself is NP-complete, we transform it into an R-TMSC problem by adding a few plausible restrictions.
- The R-TMSC model can be solved by the proposed algorithm, and a stable solution is always guaranteed. The effectiveness of the proposed algorithm is verified via simulations.

The rest of this chapter is organized as follows. Some of the relevant previous works are discussed in Section 3.2. The system model is presented in Section 3.3, and the problem is formulated in Section 3.4. We perform the algorithm analysis in Section 3.5, where firstly, we discuss the three-sided stable matching game for the considered scenario in Section 3.5.1. Secondly, we introduce the TMSC model for the NFV resource allocation scenario in Section 3.5.2, and then, discuss the proposed R-TMSC model in Section 3.5.3. Finally, we perform a detailed analysis of the algorithm in Section 3.5.4. Thereafter, we demonstrate the performance of our proposed algorithm through simulation results in Section 3.6. The chapter is concluded in Section 3.7.

3.2 Related Work

Resource allocation in wireless network virtualization is a popular area among researchers. An information-centric wireless network virtualization architecture for 5G mobile wireless networks is proposed in [59]. The application of network virtualization in smart cities by enabling the use of 5G is discussed in [60], and [61] discusses a user mobility and service usage oriented approach for wireless virtual networks. An auction based model for maximizing the network revenue, and dynamic allocation of resources to different network slices are discussed in [38] and [39], respectively.

A lot of prominent research has been going on with regard to NFV as well. Reference [62] discusses autonomic placement of VMs according to the policies of the CN providers, and [63, 64] propose solutions for cloud federation formation. Some of the research works on the placement of VNFs for the vEPC/5G Core are [65–67], which focus mainly on SGW placement, PGW placement, and load-aware dynamic placement of SGW/PGW, respectively. The aforementioned works deal with either determining the optimal number of VNFs or VNF placement in the CNs. Reference [57] proposes a solution to address both by firstly, determining the optimal number

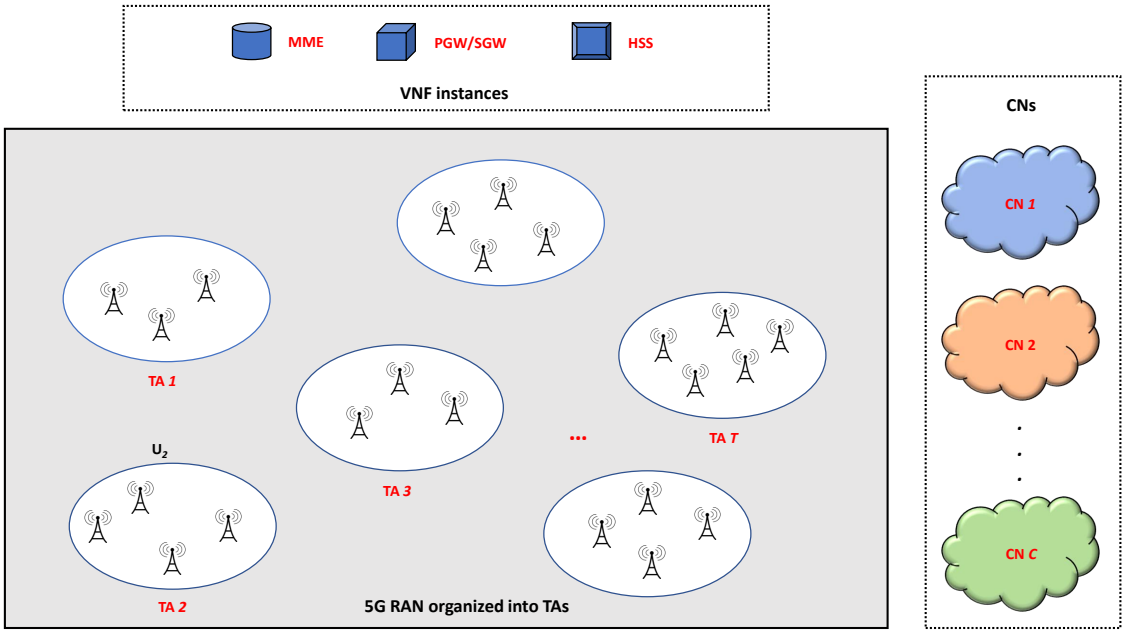


Figure 3.1: 5G network before EPC virtualization.

of VM instances for each VNF, and then, placing these VNF instances on CNs.

However, as per my knowledge, a framework to perform resource allocation in NFV by jointly considering both the user traffic requirements for VNF instances, and their placements in different CNs, has not been proposed. Here, we integrate these two aspects by harnessing the advantages of matching theory, which is a framework that provides feasible solutions based on the mutually beneficial relationships between different sets of entities. Accordingly, in this chapter, we propose a three-sided matching based framework for resource allocation in NFV, which considers three important sets of entities in a vEPC/5G Core network: TAs, VNF instances, and CNs. Even though my focus here will be on vEPC, the proposed solution can be easily extended to 5G Core VNFs.

3.3 System Model

We consider a wireless network virtualization scenario in which the BSs in the RAN are organized in TAs [57]. As mentioned before, the EPC components of 5G will be virtualized and hosted in a federated cloud network. Therefore, the initial step in resource allocation is determining the optimal number of instances for each VNF: MME, HSS, SGW, and PGW, and then, deploying them according to the requirements of the TAs.

Here, we consider a network with U users, T TAs, V total instances of all the VNFs deployed at a given time, and C CNs that belong to different cloud network operators, as shown in Fig. 3.1. As in any wireless scenario, the main objective is to enhance the user experience by keeping the overall costs at a minimum. We can consider the users' SINR as the key metric for user satisfaction, since it determines the channel capacity bounds. This in turn decides the data rates received by the users, and hence, the overall QoS.

Hence, the price paid by user l for the services received in TA i is given by

$$R_i^l = t_i^l r_i^l, \tag{3.1}$$

where t_i^l is the price per $Mbps$ in TA i for user l, and r_i^l is the data rate (indicative of the SINR and thus, the QoS) in $Mbps$ in TA i for user l.

The service requirements of the traffic generated by the users in a TA are addressed by the different instances of the initiated VNFs. Hence, the utility of TA i can be represented as the revenue obtained from the users minus the price paid for the service of the VNFs, as

$$U_{TA}^i = \sum_{l=1}^{U_i} t_i^l r_i^l - (\alpha_i m_j + \beta_i s_j + \gamma_i p_j + \delta_i h_j), \tag{3.2}$$

where U_i is the total number of users in TA i, m_j, s_j, p_j, and h_j are the service prices

of the VNF instances (MME, SGW, PGW, and HSS, respectively) that serve TA i, and α_i, β_i, γ_i, and δ_i are scalars which indicate what part of the services of these VNF instances are occupied by the traffic requirements from TA i (multiple TAs can be served by the same instance of a VNF). If TA i doesn't use the service of any one or more of these VNF instances, the scalars associated with those instances are set to 0.

Subsequently, these VNF instances are hosted in CNs managed by different operators. The amount of virtual resources (CPU, memory, storage etc.) allocated to each instance of each VNF is called *flavor*, and each flavor is dedicated to a specific VM in the CN. Thus, the utility of VNF instance j can be expressed as the difference between the revenue obtained from the TAs and the price paid for the assigned flavors in CN k, as

$$U_{VNF}^{j} = \sum_{i=1}^{T} \kappa_i v_j - f_k^j c_k^j, \tag{3.3}$$

where $\kappa_i v_j = \alpha_i m_j$ or $\beta_i s_j$ or $\gamma_i p_j$ or $\delta_i h_j$, when the VNF instance is an MME, SGW, PGW or HSS, respectively. $f_k^j = \theta \Gamma_k^j$ is the number of flavors assigned to VNF instance j, which is proportional to the average SINR (QoS) between CN k and the users in TAs served by VNF instance j, Γ_k^j, and θ is a scalar. c_k^j is the cost/flavor for VNF instance j in CN k.

Finally, the utility of CN k can be described as the revenue from the VNF instances minus the operating costs of the cloud network, and can be written as

$$U_{CN}^{k} = \sum_{j=1}^{V} (f_k^j c_k^j - o_k^j - d_k^j), \tag{3.4}$$

where $o_k^j = f_k^j \theta_k^j$ denotes the operating costs of CN k in hosting VNF instance j, where θ_k^j denotes the operating costs per flavor of CN k in hosting VNF instance j. $d_k^j = n_k^j + q_k^j$ denotes the costs of service delay, where n_k^j is the cost due to network

delay, and q_k^j is the cost due to queuing delay at the CN servers.

Considering (3.2), (3.3), and (3.4), and summing up over all the T TAs, V VNF instances, and C CNs will give us the total utility of the virtual network as

$$U_{VN} = \sum_{i=1}^{T} U_{TA}^i + \sum_{j=1}^{V} U_{VNF}^j + \sum_{k=1}^{C} U_{CN}^k = \sum_{i=1}^{T} \sum_{l=1}^{U_i} t_i^l r_i^l - \sum_{k=1}^{C} \sum_{j=1}^{V} (o_k^j + d_k^j), \quad (3.5)$$

where the first term indicates the total service price paid by all the users in all the TAs in the RAN and the second term indicates the total overall costs of the federated cloud network.

3.4 Problem Formulation

As mentioned before, the main objective from the perspective of the network is to maintain the user experience at a good level, while simultaneously keeping the overall costs at a minimum. In this regard, we express the optimization problem of the virtual network as

$$\begin{aligned} &\max_{r_i^l} U_{VN} = \sum_{i=1}^{T} \sum_{l=1}^{U_i} t_i^l r_i^l - \sum_{k=1}^{C} \sum_{j=1}^{V} (o_k^j + d_k^j), \\ &s.t. \begin{cases} t_i^l \leq t_i^{th}, \\ d_k^j \leq d_k^{th}, \\ o_k^j \leq o_k^{th}, \end{cases} \end{aligned} \quad (3.6)$$

where the first constraint is to keep the price per Mbps in TA i for each user below the threshold value for TA i, t_i^{th}. The second and third constraints are to keep the costs of service delay and operating costs for each instance, d_k^j and o_k^j, below the respective threshold values, d_k^{th} and o_k^{th}, in CN k.

In order to achieve this, the TAs need to be assigned to the VNF instances according to the user traffic requirements in the TAs. Subsequently, the VNF instances need to be deployed in the CNs according to the flavor requirements of the instances.

In order to achieve this, we need a distributed algorithm that takes into account the requirements of all the three entities involved. Hence, we introduce the three-sided matching based approach in the next section.

3.5 Algorithm Analysis

Here, we discuss the proposed algorithm in detail, where we introduce the three-sided stable matching game for the considered scenario in Section 3.5.1. Then, we discuss the TMSC and R-TMSC models in Section 3.5.2 and Section 3.5.3, respectively, and finally, perform an analysis of the proposed algorithm in Section 3.5.4.

3.5.1 Three-sided Stable Matching Game

In this chapter, let the three-sided matching between the T TAs, V VNF instances and C CNs be denoted by $\mathcal{M} \subseteq \mathcal{T} \times \mathcal{V} \times \mathcal{C}$, where $\mathcal{T}$, $\mathcal{V}$, and $\mathcal{C}$ are the sets of TAs, VNFs instances, and CNs considered at a given time, respectively. Here, $\mathcal{M}$ is a set of triples from $\mathcal{T} \times \mathcal{V} \times \mathcal{C}$. For the considered NFV scenario, a blocking triple consists of a TA, a VNF instance and a CN, each of which has the desire to get matched with each other as a triple, instead of staying with the current matched partners in $\mathcal{M}$. A matching $\mathcal{M}$ is said to be stable if there exists no blocking triple for $\mathcal{M}$ [51].

3.5.2 TMSC Model

Chapter 2 discussed the TMSC problem in detail, which allows each agent to have multiple partners. In order to explain the TMSC model for the NFV resource allocation scenario, we consider the triple (TA i, VNF instance j, CN k), and denote it as (T_i, V_j, C_k). Here, we assume that TAs only rank VNF instances, VNF instances only rank CNs, and CNs only rank TAs in their orders of preferences, and that each

agent can be matched to a limited number of the other type of agents which it ranks. The detailed definition is given as follows.

Definition 3.1. ***Three-sided Matching with Size and Cyclic Preference:*** *The three-sided matching problem of TMSC is to find a matching* $\mathcal{M} = \{(T_i, V_j, C_k)\}$ *with the maximum cardinality:*

$$\begin{aligned} &\max |\mathcal{M}|, \\ &s.t. \begin{cases} \mathcal{N}(\mathcal{M}, T_i) \leq q^{T_i}, \\ \mathcal{N}(\mathcal{M}, V_j) \leq q^{V_j}, \\ \mathcal{N}(\mathcal{M}, C_k) \leq q^{C_k}, \end{cases} \end{aligned} \tag{3.7}$$

$\forall i \in \{1, 2, \ldots, T\}$, $\forall j \in \{1, 2, \ldots, V\}$, and $\forall k \in \{1, 2, \ldots, C\}$, where $\mathcal{N}(\mathcal{M}, x)$ represents the number of partners that x has in the matching $\mathcal{M}$. $|\mathcal{M}|$ represents the cardinality of the matching $\mathcal{M}$, which is the number of (T_i, V_j, C_k) triples in the matching. The constraints represent the capacity limitations of TA T_i, VNF instance V_j, and CN C_k, respectively. The problem of deciding whether a stable matching exists in an instance of TMSC is NP-hard [51], as discussed in Chapter 2.

3.5.3 R-TMSC Model

Due to the aforementioned NP-completeness of the TMSC model, we need to perform some refinement in order to make it easily solvable. Accordingly, a few plausible restrictions are added to the TMSC problem to transform it into an R-TMSC problem. The assumptions are: (1) the preference lists of TAs are derived from a master preference list, which is the set of all VNF instances in strict order; (2) the CNs are indifferent with the TAs (for each CN, the TAs in its preference list form one tie). Considering these restrictions, the R-TMSC model for the considered scenario is built.

Firstly, we create the preference lists for each of the TAs, VNF instances and CNs. Since this is a cyclic preference problem, the preference lists of TAs consist

of only VNF instances, VNF instances' preference lists contain only CNs, and CNs' lists are comprised of only TAs, all in the order of preference. The preference list of each TA over the VNF instances is derived from a master list that ranks the instances according to their service costs, v_j, in ascending order. Each TA creates its preference list from this master list according to the user traffic requirements for the VNF instances (v_j can be m_j, s_j, p_j, and/or h_j, accordingly), and preferring the instances offering cheaper service costs as

$$PL_T(i, j) = v_j. \tag{3.8}$$

Similarly, the VNF instances create their preference lists by ranking the CNs in the descending order of the offered QoS (SINR), which is decided by the flavors assigned by the CNs [57] (the better the SINR, the better they can serve the users in the TAs) as

$$PL_V(j, k) = \Gamma_k. \tag{3.9}$$

According to our assumption of the R-TMSC model, the CNs are indifferent with the TAs. In other words, the preference list of any CN consists of a tie with all TAs ranked the same, which can be represented as

$$PL_C(k, i) = 1. \tag{3.10}$$

3.5.4 Algorithm Analysis

Once the preference lists are created, the R-TMSC algorithm for the three-sided matching of the TAs, VNF instances, and CNs can be executed. The R-TMSC algorithm discussed in [51] is modified according to our scenario. The following sets

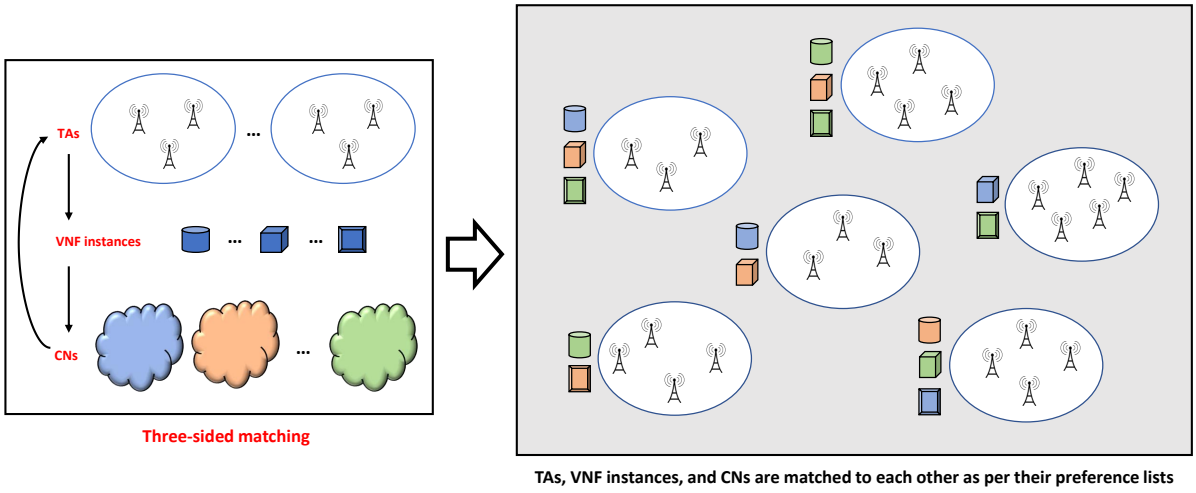


Figure 3.2: 5G network after EPC virtualization.

for matching $\mathcal{M}$ and an instance of R-TMSC are defined:

$$A^{+1}(\mathcal{M}, T_i) = \{V_j | V_j \succ_{T_i} \mathcal{M}(T_i), V_j \in PL_T\} \tag{3.11}$$

denotes the set of all VNF instances that TA T_i prefers to its current partner $\mathcal{M}(T_i)$,

$$A^{+1}(\mathcal{M}, V_j) = \{C_k | C_k \succ_{V_j} \mathcal{M}(V_j), C_k \in PL_V\} \tag{3.12}$$

denotes the set of all CNs that VNF instance V_j prefers to its current partner $\mathcal{M}(V_j)$,

$$A^{-1}(\mathcal{M}, T_i) = \{C_k | C_k \in \mathcal{C}, T_i \in PL_C, \mathcal{N}(\mathcal{M}, C_k) < q^{C_k}\} \tag{3.13}$$

represents the set of all CNs that still have capacity to accept TA T_i,

$$A^{-2}(\mathcal{M}, T_i) = \{V_j | A^{+1}(\mathcal{M}, V_j) \cap A^{-1}(\mathcal{M}, T_i) \neq \emptyset, V_j \in \mathcal{V}\} \tag{3.14}$$

represents the set of all VNF instances, such that there exists a CN C_k that VNF instance V_j prefers to its current partner $\mathcal{M}(V_j)$, and CN C_k still has capacity to accept TA T_i.

Additionally, let the sub-lists of agents from the preference lists be denoted as $SL_T \subseteq PL_T$, $SL_V \subseteq PL_V$, and $SL_C \subseteq PL_C$, respectively. We define $Head(SL_V, V_j)$

as the elements (CNs) in SL_V with the highest priority. Similarly, $Head(SL_C, C_k)$ and $Head(SL_T, T_i)$ represent the TAs in SL_C and VNF instances in SL_T with the highest priority, respectively. Given these quantities, the algorithm basically searches for the "best" triple and adds this triple to the matching $\mathcal{M}$ (starts from an empty set) each time. Each "best" triple, in the form of (V_j, C_k, T_i), is generated by initially selecting a TA satisfying certain requirements, followed by this selected TA choosing the best VNF instance that meets its requirements, and finally this selected VNF instance choosing the most eligible CN. The detailed procedure is described in Algorithm 3.1. The network after the implementation of the proposed three-sided matching algorithm is shown in Fig. 3.2.

Algorithm 3.1 R-TMSC Matching Algorithm for 5G vEPC

Input: $\mathcal{V}$, $\mathcal{C}$, $\mathcal{T}$
Output: $\mathcal{M}$

Initialization;
Construct the preference lists PL_V, PL_C, and PL_T;
$\mathcal{M} = \emptyset$, $flag = 1$;
while $flag == 1$ **do**
    $flag = 0$;
    for each $T_i \in \mathcal{T}$ **do**
        $\mathcal{V}' = A^{+1}(\mathcal{M}, T_i) \cap A^{-2}(\mathcal{M}, T_i)$;
        if $\mathcal{V}' \neq \emptyset$ **then**
            $V_j = Head(\mathcal{V}', T_i)$;
            $\mathcal{C}' = A^{+1}(\mathcal{M}, V_j) \cap A^{-1}(\mathcal{M}, T_i)$;
            $C_k = Head(\mathcal{C}', V_j)$;
            if $\mathcal{N}(\mathcal{M}, T_i) == 1$ **then**
                $\mathcal{M} = \mathcal{M} \backslash \{\mathcal{M}(T_i), \mathcal{M}(\mathcal{M}(T_i)), T_i\}$;
                $flag = 1$;
            end if
            if $\mathcal{N}(\mathcal{M}, V_j) == 1$ **then**
                $\mathcal{M} = \mathcal{M} \backslash \{V_j, \mathcal{M}(V_j), *\}$;
                $flag = 1$;
            end if
            $\mathcal{M} = \mathcal{M} \cup \{V_j, C_k, T_i\}$;
        end if
    end for
end while
Output stable matching $\mathcal{M}$;

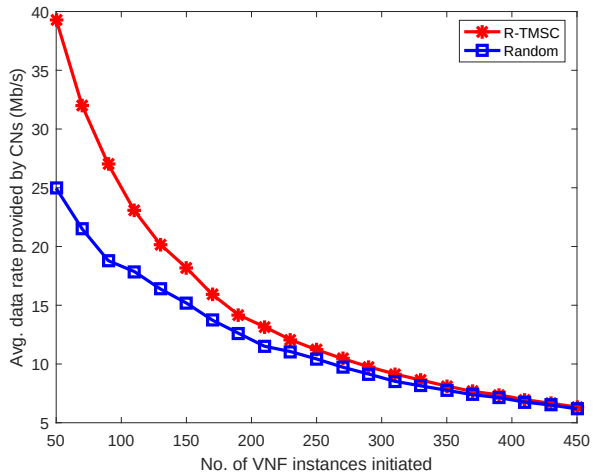


Figure 3.3: Average data rate provided by CNs.

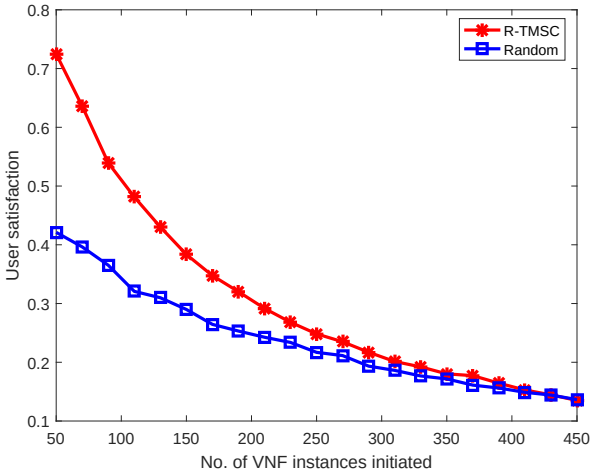


Figure 3.4: User satisfaction.

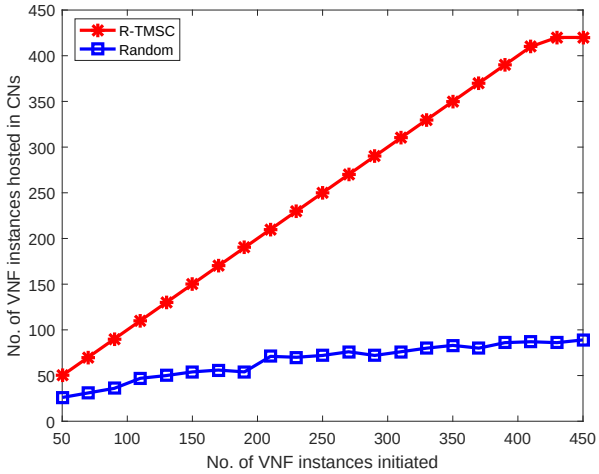


Figure 3.5: Cardinality of output matching.

3.6 Simulation Results

In this section, we evaluate the proposed R-TMSC algorithm by comparing it with a centralized random allocation of the available sets of entities, through MATLAB simulations. The comparison with a centralized random allocation is plausible to draw a contrast with the proposed distributed approach, and due to the unavailability of comparable benchmarks in literature. We consider part of a RAN with a radius of $R = 1$ km, consisting of $T = 20$ TAs, $C = 5$ CNs and $V \in [50, 450]$ VNF instances. The bandwidth of each frequency band used is set to 100 MHz. The capacity of each CN is set to 400 Mbps, while the capacity of each TA is set to a value of 100 Mbps. The minimum SINR requirement for the mobile users in all the TAs is set at a uniform value of 25 dB.

Fig. 3.3 shows that the average data rate offered by the CNs decreases as more VNF instances get matched to the available TAs and CNs. We can also observe that the proposed R-TMSC algorithm outperforms the random allocation, as it takes into account the localized preferences of all the entities. However, as the number of instances increases, the performance of the proposed algorithm becomes comparable

to that of the random allocation due to the capacity limits of the considered entities.

Fig. 3.4 demonstrates the user satisfaction, which is defined here as the ratio between the actual data rate and the expected data rate from the CNs to the VNF instances, which in turn, is for serving the user traffic requirements. We can see that the user satisfaction drops as more number of VNF instances get initiated. Also, the performances of both algorithms become comparable with increase in the number of instances due to the considered capacity limits, similar to Fig. 3.3.

Fig. 3.5 illustrates the cardinality of the output matching, which indicates the number of initiated VNF instances that are hosted in CNs. We can observe that the R-TMSC algorithm matches all the VNF instances to CNs, till the number of instances is almost 425. Thereafter, as the number of instances increases further, the algorithm levels off at serving around 425 users, as there are only limited numbers of available TAs and CNs. The R-TMSC algorithm outperforms the random allocation by a huge margin.

3.7 Conclusion

In this chapter, we propose a three-sided matching based framework for resource allocation in NFV. The relationships between the TAs, VNF instances, and CNs in a vEPC/5G Core network are modeled by utilizing a variant of the 3DSM problem called the R-TMSC problem. The proposed R-TMSC algorithm will always generate stable matching results in a finite number of steps. The simulation results illustrate the effectiveness of the proposed framework compared to a centralized random allocation approach. The proposed method improves the average capacity provided by the CNs as well as the user satisfaction. Additionally, more number of VNF instances are hosted in CNs through the R-TMSC algorithm.

Chapter 4

Pricing and Resource Allocation Optimization for IoT Fog Computing and NFV: An EPEC and Matching Based Perspective

4.1 Introduction

The advent of the Internet of Things (IoT) brings about massive internetworking of devices that we use in our everyday lives. This gives rise to the need for storing and processing tremendous amounts of data efficiently [68]. The handling of such a large amount of data can be realized by *cloud computing* [69], by providing the required resources for the users to access various applications on demand. Additionally, the heterogeneity of applications in IoT calls for the virtualization of wireless networks, which leads to better flexibility and management through the abstraction and sharing of resources [10]. As mentioned in previous chapters, wireless network virtualization involves the sharing of the physical substrate network by multiple virtual networks, and to facilitate this, both spectrum and infrastructure resources are isolated and split into slices [8, 11].

Virtualization is one of the key driving forces of 5G mobile networks, which aims at delivering extremely high capacity, low latency, and high device density per area. Specifically, NFV is a paradigm which decouples the physical network infrastructure from the network functions that run on it [56], as seen in Chapter 3. The different services are disintegrated into VNFs, and are placed on top of a virtualization platform as software components on CNs [57, 58]. The increase in the demand for storage and processing capabilities due to the provisioning of 5G services can be addressed by similar convergence of network and cloud infrastructures [70].

However, the processing requirements vary according to the applications, with

some applications demanding faster processing speeds than others [71]. Traditionally, the large-scale data centers built by the Data Service Operators (DSOs) to meet the processing needs of the Authorized Data Service Subscribers (ADSSs) are far from the ADSSs. In the light of the fast processing demands of next generation networks, computation resources are being moved to the edge of the network. This concept is known as *fog computing* [72], in which a number of small-scale but flexible computing devices called Fog Nodes (FNs) are deployed close to the ADSSs. These micro clouds are also known as *edge clouds* or *cloudlets*, as they have lesser computing resources compared to data center based clouds and are deployed at the network edge [73]. Due to their proximity to the ADSSs, the FNs can provide data services with low latency and low transmission costs [74].

Considering the heterogeneity and complexity of IoT applications, an integration of the fog computing technology with NFV is inevitable for rendering computation flexibility and scalability in next generation networks. As a result, an efficient resource allocation solution for a fog enabled NFV platform needs to effectively model the interactions between the different sets of entities: DSOs, ADSSs, and FNs, as well as enable the DSOs to allocate resources from the FNs as per the VNF requirements of the ADSSs.

The DSOs and ADSSs in a typical fog computing scenario are autonomous entities that try to maximize their own profits. The DSOs try to allocate resources from the FNs at prices which favor them, and the ADSSs purchase these resources according to their own benefits. However, the maximization of profit for one DSO might affect the profits of other DSOs. Also, if one ADSS tries to maximize the amount of purchased resources at a given price, it might affect the amount of computing resources available to the other ADSSs. Therefore, in order to reach a stable and social optimum, we need to model the competition among them, and find an equilibrium solution.

Modeling this competition results in an Equilibrium Problem with Equilibrium Constraints (EPEC), which is a hierarchical optimization problem with equilibria at two levels [75]. Due to the conflicts between them and amongst themselves, there exist equilibrium criteria at both the level of the DSOs and at the level of the ADSSs. In order to balance these conflicting objectives, we use an incentive mechanism as in [76], and then perform the optimization of their utilities. The Alternating Direction Method of Multipliers (ADMM), which is considered an efficient tool for large-scale optimization is adopted here, due to its decomposition and fast convergence properties [77].

A large-scale fog computing optimization framework to achieve this is proposed in [78], which formulates the interactions between the DSOs and ADSSs as an EPEC. It is then solved using an ADMM based algorithm, which provides the optimal values of the resource prices to be set by the DSOs, and the optimal amount of resources to be purchased by the ADSSs, resulting in the simultaneous optimization of profits for both sets of entities. However, a fog computing resource allocation framework for the next generation networks is incomplete without the dimension of virtualization. We need to efficiently allocate the resources from FNs as per the resource needs of various VNFs initiated at the ADSSs, by considering the localized conditions.

Accordingly, in this chapter, we extend the resource allocation framework proposed in [78] to integrate NFV, by proposing a matching theory based algorithm for the DSOs to allocate resources from the FNs as per the VNF requirements of the ADSSs. Taking all of this into account, we state the objectives and contributions of this chapter as below:

- We model the competitions between the DSOs and ADSSs in an IoT fog computing scenario as an EPEC, for which we use an incentive mechanism, in order to balance the conflicting objectives of both sets of entities.

- An ADMM based algorithm is invoked to solve the EPEC and obtain the optimal values of the resource prices to be set by the DSOs, and the optimal amount of resources to be purchased by the ADSSs, resulting in profit optimization for both.

- Further, we obtain the resource requirements for different VNFs to be deployed from the resource requirements of the ADSSs. This is finally utilized in a *many-to-many matching* algorithm which allocates the computing resources of the FNs according to the resource requirements of the VNFs, in a distributed manner.

- The effectiveness of the proposed framework is then demonstrated through simulations. The simulation results show that the proposed ADMM based EPEC algorithm converges within a few iterations to give optimum results. It is also observed that the proposed many-to-many matching algorithm outperforms the centralized approach in terms of the cost of the FN resources.

The remainder of this chapter is arranged as follows. We discuss some of the relevant previous works in Section 4.2. In Section 4.3, we introduce the system model, and formulate the problem in Section 4.4. We analyze the proposed framework in Section 4.5, where firstly, we introduce the concept of ADMM in Section 4.5.1. Secondly, we discuss the design of the incentive function in Section 4.5.2, and then the ADMM based EPEC algorithm is discussed in detail in Section 4.5.3. The many-to-many matching algorithm for VNF resource allocation is discussed in detail in Section 4.5.4. We discuss the performance of our model through simulation results in Section 4.6. Finally, the chapter is concluded in Section 4.7.

4.2 Related Work

Wireless network virtualization resource allocation for next generation networks has been widely discussed in the literature [79–81]. Reference [82] proposes a three-sided matching based framework for wireless network virtualization resource allocation considering the spectrum and infrastructure resources and mobile users. The research works in NFV generally deal with either determining the optimal number of required VNFs or the placement of VNFs in CNs. Reference [83] proposes a matching based framework for NFV resource allocation, by jointly considering both the user requirements for VNFs as well as their placements in different CNs.

The management of resources in fog computing is challenging due to a large number of FN deployments, and is extensively discussed in research areas [74]. A multi-dimensional framework has been proposed in [84], where a QoS consistent contract providing a comprehensive payment plan to the FNs, revenue maximization of the network operators, and incentives to the FNs has been evaluated. A mathematical framework for service-oriented heterogeneous resource sharing has been proposed in [85], and [86] proposes a Distributed Dataflow (DDF) programming model that coordinates the resources distributed across hosts in fog computing. Reference [87] proposes a matching game based joint radio and computational resource allocation problem for optimizing system performance and improving user satisfaction in IoT fog computing. Reference [88] investigates the formation of stable coalitions among Fog Infrastructure Providers (FIPs), and proposes a mathematical model for profit maximization in order to allocate IoT applications to sets of FIPs.

There are a few recent works that integrate fog computing and virtualization in IoT. Reference [89] proposes Virtual Fog, which is a complete layered framework for IoT and connects the layers from fog computing through virtualization. A dynamic resource allocation framework for NFV enabled Mobile edge-cloud (MEC) is

discussed in [73], in which both low latency requirements and MEC cost efficiency are addressed. Reference [70] demonstrates three use cases of an integrated cloud/fog and heterogeneous networks orchestration through a 5G NFV experimental platform, and performs testing of end-to-end IoT and mobile services. Even though these works perform the indispensable integration of NFV and fog computing for IoT, a resource allocation framework for IoT fog computing and NFV, by taking into consideration the DSOs, ADSSs, and FNs, and also the resource requirements of the various VNFs, has not been proposed.

The fog computing scenario considered in [78] addresses the competition among multiple DSOs and multiple ADSSs, as opposed to related previous works. The optimization of the resulting EPEC scenario with a large number of entities is handled by the convergence properties of ADMM [90,91], which is a powerful large-scale optimization tool. The simulation results demonstrate that the optimization of the utility functions of DSOs can be achieved while simultaneously performing the optimization of the utility functions of ADSSs.

In order to bring in the NFV perspective here, we have to determine the resource requirements for various VNFs that serve the ADSSs, which are to be deployed in the FNs operated by different DSOs. To that end, we need to model the localized preferences of the FNs and the VNF resource requirements in a distributed manner. As seen from previous chapters, a promising candidate here is matching theory, which has gained popularity in recent years as an efficient distributed framework [14]. The formation of mutually beneficial relationships between different sets of entities forms the basis for matching theory, and it is known to have overcome certain limitations of optimization and game theory [92].

Considering a large number of FN deployments and VNF initiations, and also the fact that different instances of the same VNF initiated to serve different ADSSs,

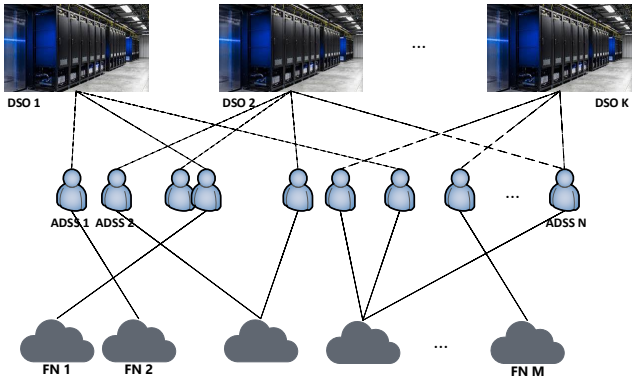


Figure 4.1: System architecture for IoT fog computing.

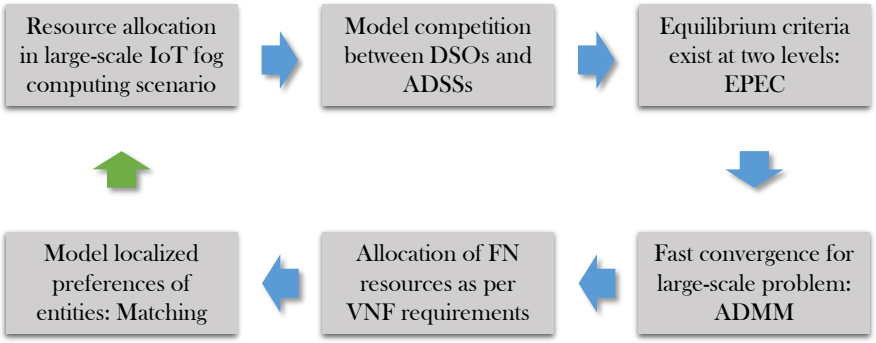


Figure 4.2: Proposed NFV resource allocation framework for fog computing in IoT.

can be deployed in the same FN for ease of management (and also that many FNs can together host a single VNF instance), in this chapter, we propose a many-to-many matching [93] based algorithm for NFV enabled IoT fog computing, to perform VNF resource allocation in FNs.

4.3 System Model

We consider an IoT fog computing scenario consisting of multiple DSOs, ADSSs, and FNs, as shown in Fig. 4.1. As mentioned before, unlike the massive data centers which are usually located far from the ADSSs, the FNs are deployed closer to the

ADSSs, which helps to reduce the service latency and congestion by computation offloading. The ADSSs request for computing resources from the DSOs, and the DSOs serve the ADSSs by allocating computing resources from the FNs operated by them. At the same time, the traditional practice in an NFV scenario is that the MVNO communicates the resource needs based on the requirements of the users to the NFVO. Then, the NFVO translates these resource needs into software component requirements that need to be initiated in the network [83].

In an NFV enabled IoT fog computing scenario, these software components would be the VM resources available in the distributed network of FNs. Once the VNF requirements of each ADSS are known to the DSOs, they can allocate VM resources from the FNs operated by them. The VM resources in the FNs can then host the allocated instances of VNFs. Fig. 4.2 summarizes the proposed framework. In this chapter, we consider the NFVO to be a centralized entity that coordinates the proposed distributed resource allocation schemes.

Here, a network with K DSOs, N ADSSs, and M FNs is considered. We denote the computing resources from the FNs in terms of Computing Resource Blocks (CRBs). The price for one CRB set by DSO i for ADSS j is denoted by $\{\theta_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$, and $\boldsymbol{\theta_i}$ is the pricing profile for DSO i. The number of CRBs purchased from DSO i by ADSS j is denoted by $\{x_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$.

Therefore, according to the profits and costs of DSOs, we express the utility function of DSO i, $\forall i \in \{1, 2, \ldots, K\}$, as

$$P_i(\boldsymbol{\theta_i}) = \sum_{j=1}^{N} U_{i,j}(\theta_{i,j}), \tag{4.1}$$

where

$$U_{i,j}(\theta_{i,j}) = R_{i,j} - D_{i,j} - O_{i,j}, \tag{4.2}$$

and

$$R_{i,j}(\theta_{i,j}) = x_{i,j}\theta_{i,j} \tag{4.3}$$

is the revenue from the computing resources provided to ADSS j by DSO i, where $x_{i,j}$ is the number of CRBs purchased from DSO i by ADSS j, and $\theta_{i,j}$ is the price for one CRB purchased by ADSS j from DSO i.

$$D_{i,j} = n_{i,j} + q_{i,j} \tag{4.4}$$

is the cost due to service delay. Here,

$$n_{i,j} = \gamma d_{i,j} \tag{4.5}$$

is the cost incurred by the network delay, which is the delay from the physical service provider (i.e., FN) to ADSS j. Consider its value to be a linear function of the distance from the physical service provider to ADSS j, $d_{i,j}$, and γ is the cost per unit distance. We consider the workload of ADSS j to follow a Poisson process with a workload rate of w_j, and also assume the queue length as the workload rate/computing service rate, and as a result, we get the cost incurred by the queuing delay at the servers, $q_{i,j}$ as

$$q_{i,j} = \kappa \frac{w_j}{\mu x_{i,j}}, \tag{4.6}$$

where κ is the cost per unit queue length, and each CRB can provide computing service at the rate of μ.

$$O_{i,j} = x_{i,j}\eta_{i,j} \tag{4.7}$$

is the operational and measurement cost for the resources provided by the FNs to ADSS j, thus helping DSO i in offloading. Here, $\eta_{i,j}$ is the price set by the FNs helping DSO i serve ADSS j.

As mentioned before, the number of CRBs purchased from DSO i by ADSS j is denoted by $\{x_{i,j}|i = 1, 2, ..., K; j = 1, 2, ..., N\}$, and $\mathbf{x}_j$ is the CRB purchase profile of ADSS j. According to the profits and costs of the ADSSs, we express the utility function of ADSS j, $\forall j \in \{1, 2, \ldots, N\}$, as

$$Q_j(\mathbf{x}_j) = \sum_{i=1}^{K} W_{i,j}(x_{i,j}), \tag{4.8}$$

where

$$W_{i,j}(x_{i,j}) = T_{i,j} - D_{i,j} - R_{i,j}(x_{i,j}). \tag{4.9}$$

Here,

$$T_{i,j} = \beta_j w_j \tag{4.10}$$

is the revenue obtained by ADSS j from its workload data, where β_j is the revenue obtained by ADSS j per unit workload rate. $D_{i,j}$ is the cost due to service delay, similar to the case of the DSOs.

$$R_{i,j}(x_{i,j}) = x_{i,j}\theta_{i,j} \tag{4.11}$$

is the cost of the computing resources provided to ADSS j by DSO i, which is the same as the revenue obtained by DSO i from the computing resources provided to ADSS j, and hence, we have used the same notation, $R_{i,j}$.

4.4 Problem Formulation

As mentioned before, the DSOs and the ADSSs are assumed to be autonomous entities, that aim to maximize their own profits. However, the maximization of $P_i(\boldsymbol{\theta_i})$ for one DSO may affect the utilities of other DSOs and the ADSSs, and similarly, the maximization of $Q_j(\mathbf{x_j})$ for one ADSS may affect the utilities of other ADSSs and the DSOs. Also, the optimization of the utility functions of the ADSSs should be performed in such a way that the optimization of the utility functions of the DSOs

are not affected. When the number of DSOs and ADSSs are large as in a typical IoT fog computing scenario, a centralized optimization of the utilities of all the DSOs as in (4.1), and those of all the ADSSs as in (4.8) simultaneously, is a difficult task.

From the utility function discussed in the previous section, we can express the optimization problem of DSO i as

$$\max_{\theta_\mathbf{i}} P_i = \sum_{j=1}^{N} [R_{i,j}(\theta_{i,j}) - D_{i,j} - O_{i,j}] \tag{4.12}$$

$$s.t. \begin{cases} \sum\limits_{j=1}^{N} A_{i,j}\theta_{i,j} = B_i, \\ D_{i,j} \leq D_{th}, \end{cases} \tag{4.13}$$

where $\boldsymbol{\theta_i} = (\theta_{i,1}, \theta_{i,2}, ..., \theta_{i,N})$ is the row vector that represents the prices set by DSO i for each of the N ADSSs. The first linear constraint for DSO i in (4.13) indicates the limit on the total price per CRB offered by DSO i, where all $\{A_{i,j}|j = 1, 2, ..., N\}$ and B_i are real, scalar constants. D_{th} in the second constraint denotes the upper bound for the cost of service delay between DSO i and ADSS j.

We can also express the optimization problem of ADSS j as

$$\max_{\mathbf{x_j}} Q_j = \sum_{i=1}^{K} [T_{i,j} - D_{i,j} - R_{i,j}(x_{i,j})] \tag{4.14}$$

$$s.t. \begin{cases} \sum\limits_{i=1}^{K} X_{i,j}x_{i,j} = Y_j, \\ D_{i,j} \leq D_{th}, \end{cases} \tag{4.15}$$

where $\mathbf{x_j} = [x_{1,j}, x_{2,j}, ..., x_{K,j}]^T$ is the vector that represents the resources purchased by ADSS j from each of the K DSOs. The first linear constraint for ADSS j in (4.15) indicates the total resource requirement of ADSS j in terms of the number of CRBs purchased from all the DSOs, where all $\{X_{i,j}|i = 1, 2, ..., K\}$ and Y_j are real, scalar

constants. The second constraint is similar to the one in the optimization problem of DSO i.

Since $x_{i,j}$ denotes the number of CRBs purchased from DSO i by ADSS j, the values in $\{x_{i,j}|i = 1, 2, ..., K; j = 1, 2, ..., N\}$ are decided by the ADSSs. Hence, this matrix would consist of values which are the optimal values of $x_{i,j}$, so as to maximize the utilities of the ADSSs as in (4.14), rather than the optimal values of $x_{i,j}$, so as to maximize the utilities of the DSOs as in (4.12). Therefore the DSOs need to provide incentives to the ADSSs, in order to make the ADSSs choose values in $\{x_{i,j}|i = 1, 2, ..., K; j = 1, 2, ..., N\}$ favoring the DSOs. To this end, we can formulate the problem as an incentive mechanism design, which can lead to an optimum result as per the utilities of the DSOs in (4.12), while simultaneously considering the utilities of the ADSSs in (4.14).

Here, we can consider $\theta_{i,j}$ to be the incentive factor provided by DSO i to ADSS j, as the DSO can influence the value of $x_{i,j}$ by setting the price at a certain $\theta_{i,j}$. By controlling the incentive factor $\{\theta_{i,j}|i = 1, 2, ..., K; j = 1, 2, ..., N\}$, DSO i can get each of the ADSSs to choose the values of $x_{i,j}$ such that its profit, $P_i(\theta_{i,*})$ is maximized. $\boldsymbol{\theta_j} = (\theta_{1,j}, \theta_{2,j}, ..., \theta_{K,j})^T$ is the vector of incentive factors for ADSS j. We can use this to design an incentive function $\Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j})$, which indicates the interactions between the DSOs and ADSS j.

In summary, the DSOs' optimization problem can be formulated as

$$\begin{aligned}
&\max_{\theta_\mathbf{i}} P_i = \sum_{j=1}^{N} [R_{i,j}(\theta_{i,j}) - D_{i,j} - O_{i,j}] \\
&s.t. \begin{cases} \sum\limits_{j=1}^{N} A_{i,j}\theta_{i,j} = B_i, \\ D_{i,j} \leq D_{th}, \\ \mathbf{x_j} = \arg\max \Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j}), \\ \qquad s.t. \begin{cases} \sum\limits_{i=1}^{K} X_{i,j}x_{i,j} = Y_j, \\ D_{i,j} \leq D_{th}, \end{cases} \end{cases}
\end{aligned} \tag{4.16}$$

$\forall i \in \{1, 2, \ldots, K\}$, and $\forall j \in \{1, 2, \ldots, N\}$.

This is an example of an EPEC, which is a hierarchical optimization problem that contains equilibrium problems at both the upper and lower levels [75]. That is to say, there exist equilibrium criteria at the upper level as well, rather than just minimizing the real-valued functions subject to equilibrium constraints. In our scenario, both the DSOs as well as the ADSSs have a set of equilibrium constraints, as shown in (4.16). As there are two levels of entities with equilibrium constraints, a centralized solution that is feasible for everyone is difficult. Here, the DSOs are at advantage, as they make the first move by declaring the prices for the computing resources they provide. They can predict the amount of resources going to be purchased by the ADSSs and reach an optimal price to maximize their utilities. However, we need a solution that can optimize the utilities of the DSOs, while simultaneously considering the utilities of the ADSSs.

The ADSSs can only control the values of $x_{i,j}$, the number of CRBs purchased from the DSOs, and the DSOs can only decide the values of $\theta_{i,j}$, the incentive factors provided to the ADSSs. As the DSOs can predict the amount of resources going to be purchased, they can use the incentive factors to control the resources purchased by the ADSSs. Even though mechanisms like Stackelberg games [5] can be applied here, they work well only in scenarios with one leader and multiple followers. In our case, the coordination of multiple conflicting utilities might demand high complexity to give an optimal result. Also, the network size can practically be very large. Therefore, we need an algorithm that would converge regardless of the network size. These requirements point us to the ADMM for the above optimization problem in an IoT fog computing network. In the next section, the detailed analysis of ADMM for EPEC is considered.

4.5 Algorithm Analysis

Here, we firstly discuss the basic concept of the ADMM in Section 4.5.1. After that, we move on to the design of the incentive function in Section 4.5.2. That is followed in Section 4.5.3 by the detailed explanation of the ADMM based EPEC algorithm used to optimize the profits of both DSOs and ADSSs in IoT fog computing. Finally, the many-to-many matching algorithm for the allocation of FN resources as per the VNF requirements of the ADSSs is discussed in detail in Section 4.5.4.

4.5.1 Alternating Direction Method of Multipliers

To understand the working of the ADMM, let us consider a network with one service provider and N users, where the provider wants to maximize its utility as

$$\begin{aligned} &\max H(\mathbf{y_j}) = \sum_{j=1}^{N} h_j(y_j) \\ &s.t. \sum_{j=1}^{N} C_j y_j - D = 0, \end{aligned} \tag{4.17}$$

where each $h_j(y_j)$ is a strongly convex function, y_j is a real, scalar variable, and C_j and D are given real, scalar constants [76].

Here, the values of $\mathbf{y_j}$ can be updated by the provider as

$$\mathbf{y_j}(t+1) = \arg\max \left(H\left(y_j\right)\right) + \sum_{j=1}^{N} \lambda_j\left(t\right) C_j y_j + \Psi, \tag{4.18}$$

where

$$\Psi = \frac{\rho}{2} \sum_{j=1}^{N} \|C_j y_j - D\|_2^2. \tag{4.19}$$

Here, $\|.\|_2$ denotes the *Frobenius norm*, $\rho > 0$ is a damping factor, and t is the iteration

step index [76]. λ is the dual variable, and it is updated as

$$\lambda_j\left(t+1\right)=\lambda_j\left(t\right)+\rho\left(\sum_{j=1}^{N}C_jy_j\left(t+1\right)-D\right). \tag{4.20}$$

When each $h_j(y_j)$ is *strongly convex*, it has been proved that the ADMM converges quickly [76]. Hence, it is used for large-scale optimization problems in big networks.

4.5.2 Incentive Function Design

DSO i wants to maximize its profit, P_i, by providing certain incentives to the ADSSs. Here, as the DSOs set the prices, $\theta_{i,j}$ per unit of computing resource that the ADSSs purchase, the incentive factor can be assumed to be a discount from the initial prices set by the DSOs. Let ${\theta_{i,j}}^{(p)}$ denote the price set by the DSOs at the beginning of the p^{th} iteration. Let ${\theta_{i,j}}^{\prime(p)}$ denote the value of price that the DSOs have evaluated at the end of the p^{th} iteration. Then the incentive factor can be expressed as

$$\delta_{i,j}={\theta_{i,j}}^{(p)}-{\theta_{i,j}}^{\prime(p)}. \tag{4.21}$$

This would result in an incentive function expressed as

$$\Phi_j(Q_j(\mathbf{x_j}),\theta_\mathbf{j})=\Delta\delta_{i,j}, \tag{4.22}$$

where Δ is a positive scalar value, which can be the same or different for each DSO.

4.5.3 ADMM based EPEC in IoT Fog Computing

As mentioned before, the DSOs initially announce the prices for the CRBs that they provide. This announced set consists of prices, $\{\theta_{i,j}|i=1,2,...,K;j=$

$1, 2, ..., N\}$ set by each DSO i for each ADSS j, that maximize the profit, $P_i(\boldsymbol{\theta_i})$ for each DSO i.

Next, we explain the ADMM based EPEC method, which is an iterative process. Each iteration of the ADMM can be explained as a two-step process as given below:

1. **Optimization Problem of the ADSSs**: Each ADSS j uses the announced prices at the start of each iteration p, ${\theta_{i,j}}^{(p)}$ to calculate the values of $\{{x_{i,j}}^{(p)} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$, the number of CRBs to be purchased from each DSO i, to maximize its profit $Q_j(\mathbf{x_j})$. Here, the superscript (p) denotes the value at the p^{th} iteration of the method. This is the *inner loop* of the ADMM. t is the iteration step index of the inner loop.

   We described $\mathbf{x_j}$ in (4.16) as $\mathbf{x_j} = \arg\max \Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j})$, where $\Phi_j(Q_j(\mathbf{x_j}), \theta_\mathbf{j})$ is the incentive function as described above. For each ADSS j, maximizing the incentive function is equivalent to maximizing its profit, $Q_j(\mathbf{x_j})$ to form a set of values, $\mathbf{x_j}$ which can in turn maximize the incentives provided by the DSOs. Hence, the value of $\mathbf{x_j}$ is updated at each iteration of the inner loop by ADSS j as

$$\mathbf{x_j}^{(p)}(t+1) = \arg_{\mathbf{x}_j} \max \left(Q_j\left(\mathbf{x_j}\right)\right) + \sum_{i=1}^{K} {\lambda_i}^{(p)}(t) X_{i,j} x_{i,j} + \Psi, \tag{4.23}$$

   where

$$\Psi = \frac{\rho}{2} \sum_{i=1}^{K} \left\| \sum_{m=1, m\neq j}^{N} X_{i,m} {x_{i,m}}^{(p)}(\tau) + X_{i,j} x_{i,j} - Y_j \right\|_2^2 \tag{4.24}$$

   and $\tau = t + 1$ if $m < j$, $\tau = t$ if $m > j$. Here, $\rho > 0$ is a damping factor as mentioned above, and λ is the dual variable which is updated as

$${\lambda_i}^{(p)}(t+1) = {\lambda_i}^{(p)}(t) + \rho \left(\sum_{i=1}^{K} X_{i,j} {x_{i,j}}^{(p)}(t+1) - Y_j \right). \tag{4.25}$$

At the end of the inner loop during each iteration p of the outer loop, the ADSSs arrive at a set of values, $\mathbf{x_j}$, which maximizes their profits. At the same time, these values are predicted by the DSOs, and are used to update the values of $\theta_{i,j}$.

2. **Optimization Problem of the DSOs**: The DSOs are able to predict the behaviors of ADSSs and the values of $x_{i,j}$. The DSOs then invoke ADMM as

$$\boldsymbol{\theta_i}'^{(p)}(t+1) = \arg_{\theta_i} \max \left(P_i \left(\boldsymbol{\theta_i}\right)\right) + \sum_{j=1}^{N} \lambda_j{}^{(p)}\left(t\right) A_{i,j}\theta_{i,j} + \Psi, \tag{4.26}$$

where

$$\Psi = \frac{\rho}{2} \sum_{j=1}^{N} \left\| \sum_{m=1,m\neq i}^{K} A_{m,j}\theta_{m,j}{}^{(p)}\left(\tau\right) + A_{i,j}\theta_{i,j} - B_i \right\|_2^2 \tag{4.27}$$

and $\tau = t+1$ if $m < i$, $\tau = t$ if $m > i$. Here, $\rho > 0$ is the damping factor, and λ is the dual variable which is updated as

$$\lambda_j{}^{(p)}\left(t+1\right) = \lambda_j{}^{(p)}\left(t\right) + \rho \left(\sum_{j=1}^{N} A_{i,j}\theta_{i,j}{}^{(p)}\left(t+1\right) - B_i \right). \tag{4.28}$$

Thus, the DSOs recalculate the values of $\theta_{i,j}$ that maximize their profits. This would result in an updated set of values for the price, $\boldsymbol{\theta_i}'^{(p)}$. As discussed in Section 4.5.2, $\delta_{i,j}$ denotes the difference between the updated values of price, $\theta_{i,j}{}'^{(p)}$ and $\theta_{i,j}{}^{(p)}$, the price at the start of iteration p. Based on this difference in prices, the DSOs calculate the incentive factor as $\Delta\delta_{i,j}$, which is a discount in the announced prices. Here, Δ is a positive scalar value, which can be the same or different for each DSO. This would result in the values of $\theta_{i,j}$ for the next iteration as

$$\theta_{i,j}{}^{(p+1)} = \theta_{i,j}{}^{(p)} \pm \Delta\delta_{i,j}. \tag{4.29}$$

Algorithm 4.1 ADMM based EPEC in IoT Fog Computing

Input: $\{\theta_{i,j} | i = 1, 2, ..., K; j = 1, 2, ..., N\}$, $p = 1$
Output: ${\theta_{i,j}}^{(opt)}, {x_{i,j}}^{(opt)}, i = 1, 2, ..., K, j = 1, 2, ..., N$
while $\left\| \sum_{i=1}^{K} P_i({\boldsymbol{\theta_i}}'^{(p)}) - \sum_{i=1}^{K} P_i({\boldsymbol{\theta_i}}'^{(p-1)}) \right\| \geq \varepsilon$ **do**
(1) Optimization for ADSSs using ADMM (*inner loop*):
ADSSs use the announced prices, ${\theta_{i,j}}^{(p)}$ to evaluate $x_{i,j}$ values, and their maximum profits, $Q_j(\mathbf{x_j})$;
(2) Optimization for DSOs using ADMM (*outer loop*):
DSOs predict the behavior of ADSSs and $x_{i,j}$ values, invoke ADMM to perform maximization of profits, resulting in new prices, ${\theta_{i,j}}'^{(p)}$, and update the prices to ${\theta_{i,j}}^{(p+1)}$ by evaluating incentives;
(3) $p = p + 1$;
end while
Result: Optimal values of CRBs purchased, $\mathbf{x}^{(opt)} = \mathbf{x}^{(p)}$
Optimal values of price, $\boldsymbol{\theta}^{(opt)} = \boldsymbol{\theta}^{(p)}$

The updated values, ${\theta_{i,j}}^{(p+1)}$ are then provided to the ADSSs for the $(p+1)^{th}$ iteration. This is the *outer loop* of the ADMM. The outer loop terminates when

$$\left\| \sum_{i=1}^{K} P_i({\boldsymbol{\theta_i}}'^{(p)}) - \sum_{i=1}^{K} P_i({\boldsymbol{\theta_i}}'^{(p-1)}) \right\| < \varepsilon, \tag{4.30}$$

where ε is a pre-determined small-valued threshold. The ADMM algorithm is shown in detail in Algorithm 4.1.

The utility function of DSO i as in (4.12), and the utility function of ADSS j as in (4.14) are convex. As per [76], if the utility functions of the DSOs and the ADSSs are strictly convex, ADMM can converge to optimal values, $\mathbf{x}^{(opt)}$ and $\boldsymbol{\theta}^{(opt)}$. For the theoretical proof of convergence, the readers are referred to [76]. The application of ADMM in the case of non-convex objective functions is discussed in [90].

4.5.4 Many-to-Many Matching Algorithm for VNF Resource Allocation

After the execution of the ADMM based EPEC algorithm, once the optimal values for the CRBs to be purchased by the ADSSs, and the price offered by the DSOs are obtained, the next step is the allocation of the required CRBs from the FNs, as per the VNF requirements of the ADSSs. In this chapter, we assume that only one VNF instance is initiated to serve an ADSS at a given time, and that the CRB requirement of an ADSS at a given time is the amount of VM resources required by that VNF instance. Also, as already mentioned, different instances of the same VNF initiated to serve different ADSSs, can be deployed in the same FN for ease of management, as well as many FNs can together host a single VNF instance.

In an IoT fog computing scenario, the DSOs might have different preferences on the FNs based on the resource prices set by the FNs. The DSOs would naturally prefer the FNs offering them the lowest price. Accordingly, the DSOs create their preference lists by arranging the FNs in the ascending order of their prices as

$$PL_{DSO}(i) = \eta_{k,i}, \tag{4.31}$$

$\forall i \in \{1, 2, \ldots, K\}$, and $\forall k \in \{1, 2, \ldots, M\}$.

Similarly, the FNs have preferences on the DSOs based on the CRB (VM resource) requirements of the ADSSs served by the DSOs. That is, the FNs consider VNF instances which require more VM resources (imply needing faster computation) as of having higher priority. Accordingly, the FNs arrange each (DSO,ADSS) pair in the descending order of the CRB (VM resource) requirement to form their preference lists as

$$PL_{FN}(k) = x_{i,j}, \tag{4.32}$$

$\forall i \in \{1, 2, \ldots, K\}$, $\forall j \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, M\}$. Hence, the preference

Algorithm 4.2 Many-to-Many Matching Algorithm for VNF Resource Allocation

for FN k **do**
  Construct the preference list $PL_{FN}(k)$ on all (DSO,ADSS) according to (4.32);
  One pointer is set as the indicator pointing at the first (DSO,ADSS) in the preference list.
end for
for DSO i **do**
  Construct the preference list $PL_{DSO}(i)$ on all FNs according to (4.31);
end for
We set a $flag_k$, $\forall k \in \{1, 2, \ldots, M\}$, as an indicator to show if the CRBs of FN k were selected by the (DSO,ADSS) in the previous round, but not in the current round. The initial value of $flag_k = 1$;
while the pointers of all FNs have not pointed at all the (DSO,ADSS) in their preference list **do**
  FNs propose to (DSO,ADSS) with their prices;
  for FN k who still has available CRBs **do**
    if $flag_k = 1$ **then**
      The pointer stays at the current position in the list;
    else
      The pointer jumps to the next position in the list;
    end if
    The FN proposes to the pointed (DSO,ADSS) in its preference list with its available CRBs;
    We set $flag_k = 0$;
  end for
  (DSO,ADSS) determine which FNs to select;
  for (DSO,ADSS) $x_{i,j}$ **do**
    if The total available number of CRBs proposed by the FNs exceed its requirements **then**
      (DSO,ADSS) $x_{i,j}$ selects the required number of CRBs from the FNs, and rejects the rest;
      For CRBs of the FN k which are selected by the (DSO,ADSS) in the last round, but not in the current round, we set $flag_k = 1$;
    end if
  end for
end while

list of each FN has $K \times N$ entries.

Once the preference lists are generated, a many-to-many matching can be generated between the two sets of entities. A many-to-many matching is a matching problem in which entities from a set can be assigned to multiple entities in the other set, and vice versa, based on their capacity constraints [20]. The many-to-many matching algorithm proposed for our NFV integrated IoT fog computing scenario is described in detail in Algorithm 4.2.

Table 4.1: Parameter settings for simulations

Parameter	Value
K	5
N	20
M	25
d_{ij}	$U(0,1)$ km
D_{th}	0.1
γ	10^{-5}
κ	10
w_j	*Poisson* distributed with mean = 1000 s^{-1}
μ	500 K s^{-1}
η	$U(0, 10^{-3})$
β	$U(0, 10^{-1})$
Δ	0.5
ε for ADMM	10^{-3}
ρ for ADMM	1.5

4.6 Simulation Results

This section evaluates the performance of the proposed ADMM for EPEC and many-to-many matching based framework with MATLAB. The values of the different parameters used for the simulations are shown in Table 4.1. The number of ADSSs, N, and FNs, M are varied in certain cases; if either of N or M values is not shown to change in the simulation figure, then it has the value as shown in Table 4.1. The notations in the $U(a, b)$ format in Table 4.1 denote random values from a continuous *uniform* distribution in the interval (a, b).

Fig. 4.3 demonstrates the convergence of the ADMM based EPEC algorithm. It shows how the total profit of the DSOs, $\sum_{i=1}^{K} P_i(\boldsymbol{\theta}_i)$, behaves during the optimization using ADMM. For an error threshold of $\varepsilon = 10^{-3}$, it takes only $p = 4$ iterations for the ADMM to converge. Hence, the conflicting utilities of the DSOs have been optimized in just a few iterations. It also shows how the error value of the ADMM converges to

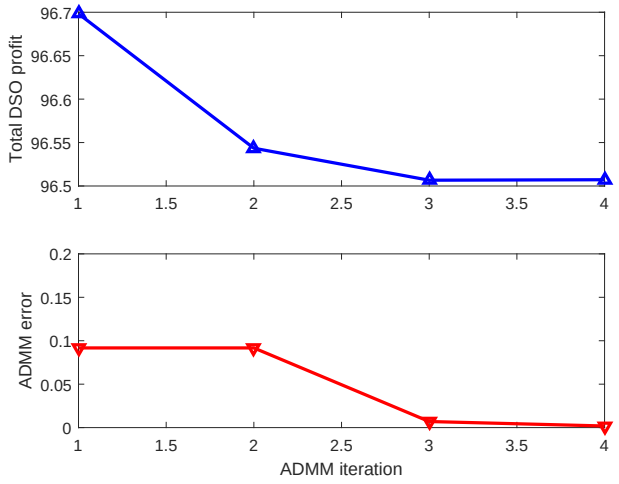


Figure 4.3: Total DSO profit and ADMM error vs. number of ADMM iterations.

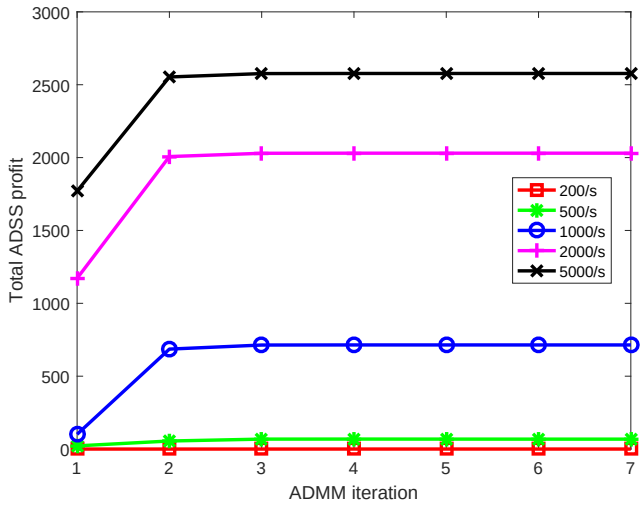


Figure 4.4: Total ADSS profit vs. mean of workload rate, w_j.

the threshold value in those few iterations.

Fig. 4.4 shows the relation between the total profit of the ADSSs, $\sum_{j=1}^{N} Q_j(\mathbf{x_j})$, and the mean of the workload arrival rate of ADSSs, w_j for five cases: 200 s^{-1}, 500

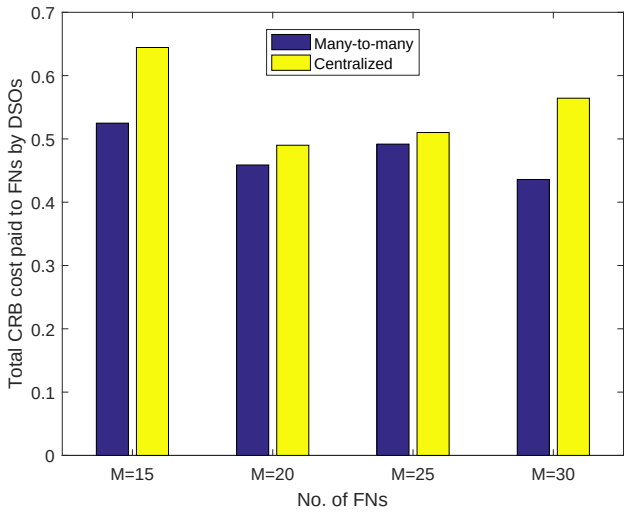


Figure 4.5: The total cost for the CRBs paid to FNs by the DSOs.

s^{-1}, 1000 s^{-1}, 2000 s^{-1}, and 5000 s^{-1}. It can be seen that as the mean value of w_j increases from $200s^{-1}$ to $5000s^{-1}$, the total profit of the ADSSs increases.

Fig. 4.5 to Fig. 4.8 compare the performance of the proposed many-to-many matching algorithm with a centralized algorithm. The centralized algorithm is an approach in which the NFVO, which acts as a centralized entity in NFV enabled IoT fog computing, performs the resource allocation itself. The NFVO allocates the resources from the FNs according to the VNF requirements of the CRBs. Similar to the case of the many-to-many matching algorithm, we assume that the largest CRB requirement, i.e., the largest value of $x_{i,j}$ maps to the VNF instance with the highest priority. However, as opposed to the distributed approach with preference lists for the two sets of entities, in the centralized approach, the NFVO arranges the VNF instances as per their priorities (CRB requirements or $x_{i,j}$ values), and allocates them to the FNs as per their available resources. Each comparison has been executed for 50 times, and the average values are plotted here.

Fig. 4.5 compares the total cost for the CRBs paid to FNs by the DSOs, between

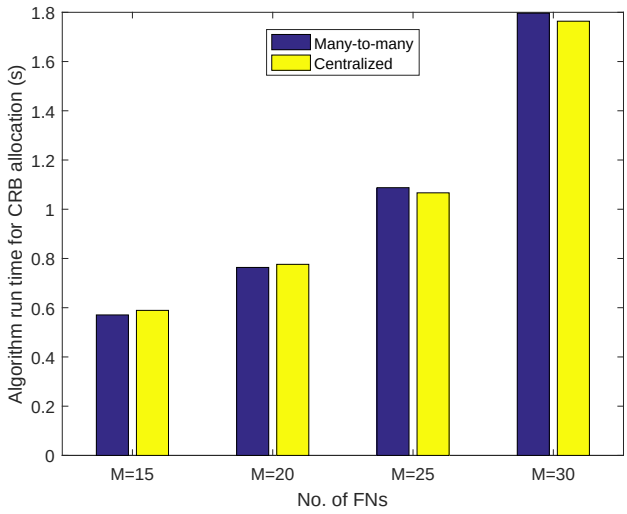


Figure 4.6: Algorithm run time for the allocation of CRBs.

the proposed many-to-many matching algorithm and the centralized algorithm. The comparison is performed for four cases: $M = 15$, $M = 20$, $M = 25$, and $M = 30$. It can be observed that in all the four cases, the total CRB cost is lesser in the proposed approach than in the centralized approach. It can also be noted that the difference is more in the first and last cases. When $M = 20$ and $M = 25$, the number of FNs is comparable to that of the ADSSs, $N = 20$, and hence, the resource allocation might be comparable in both the approaches.

Fig. 4.6 compares the run times of the many-to-many matching and the centralized algorithms. Again, the comparison is performed for four cases: $M = 15$, $M = 20$, $M = 25$, and $M = 30$. It is obvious that the algorithm run times increase with the number of entities, as can be observed. The proposed algorithm has larger run times when the number of FNs increases, which is reasonable as it is a distributed approach.

Fig. 4.7 analyzes the total cost for the CRBs paid to FNs by the DSOs for

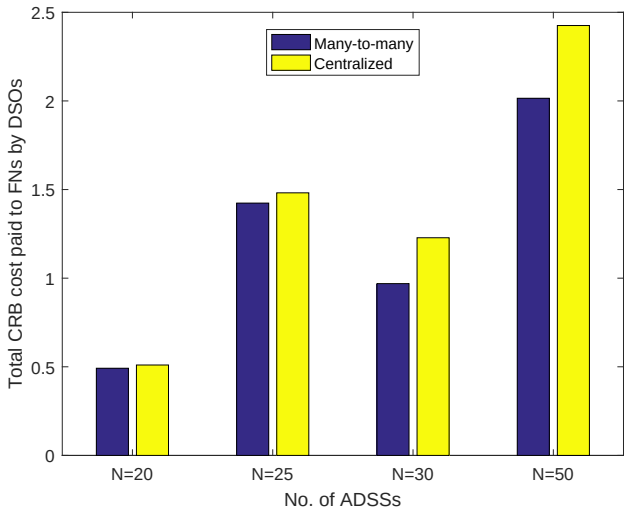


Figure 4.7: The total cost for the CRBs paid to FNs by the DSOs.

the two approaches, similar to Fig. 4.5. However, the number of ADSSs is varied here to study four cases: $N = 20$, $N = 25$, $N = 30$, and $N = 50$. It can be observed again that the proposed many-to-many matching approach outperforms the centralized approach. It can also be noted that the difference is small in the first and second cases, since the number of FNs, $M = 25$, might be comparable to that of the ADSSs. However, the difference increases as the number of ADSSs grows larger.

Fig. 4.8 compares the algorithm run times similar to Fig. 4.6. Here, the number of ADSSs is varied to study four cases: $N = 20$, $N = 25$, $N = 30$, and $N = 50$. Intuitively, the algorithm run times increase with the number of entities. Again, the proposed algorithm has slightly larger run times when the number of ADSSs increases, which is reasonable as it is a distributed approach.

The system used for executing the simulations has a small-scale Intel(R) Core(TM) $i7-7500U$ CPU with a 16 GB RAM. Therefore, the algorithm run time values shown in Fig. 4.6 and Fig. 4.8 are in hundreds of ms. In a practical IoT fog computing

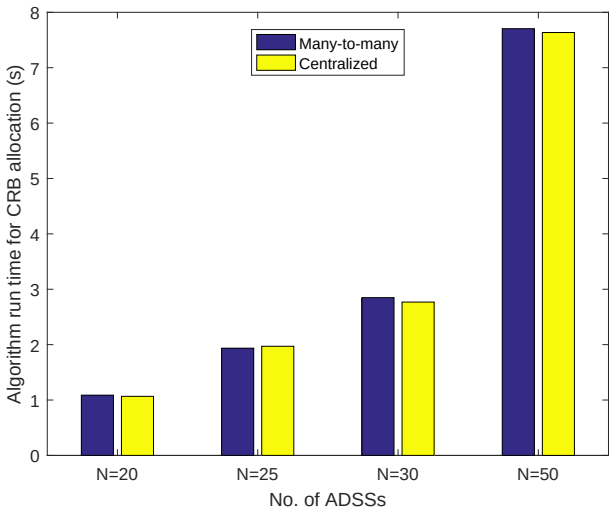


Figure 4.8: Algorithm run time for the allocation of CRBs.

network, the proposed ADMM based EPEC and many-to-many matching algorithms can be executed at the NFVO. For the many-to-many matching algorithm, once the preference lists are generated, it can be executed by a centralized entity. With a large-scale processor in a practical network, the algorithm run times will decrease tremendously.

4.7 Conclusion

In this chapter, we propose a distributed resource allocation framework for an NFV integrated IoT fog computing scenario. Initially, an ADMM based EPEC algorithm is proposed to model the competitions between the DSOs and the ADSSs, which provides the optimal values of the amount of resources to be purchased by the ADSSs, and the optimal values of the resource prices to be set by the DSOs. Thereafter, a many-to-many matching based algorithm is invoked to allocate the computing resources of the FNs according to the VNF resource requirements of the

ADSSs. The simulation results demonstrate that the ADMM based EPEC algorithm converges quickly to give optimum results. It is also observed from the simulation results that the proposed many-to-many matching algorithm outperforms the centralized approach in terms of the cost of the FN resources. The proposed resource allocation model combining EPEC and matching can be efficiently used in NFV enabled IoT fog computing scenarios.

Chapter 5

VLC and D2D Heterogeneous Network Optimization: A Reinforcement Learning Approach Based on Equilibrium Problems with Equilibrium Constraints

5.1 Introduction

The current RF spectrum crunch in wireless communications has made exploring and exploiting alternative sources of bandwidth inevitable. With a vast bandwidth of approximately 300 THz, visible light spectrum can facilitate high data rate communication, called Visible Light Communication (VLC). Apart from the immense bandwidth, some of the advantages of using visible light for communication are: (i) it is unlicensed and hence, provides free spectrum, (ii) it is secure as its propagation is limited, and (iii) VLC is easy to be implemented through already existing ubiquitous and inexpensive visible light sources such as Light Emitting Diodes (LEDs) [94, 95]. LEDs are expected to be the major sources of illumination, and also the transmitters for VLC, using which data rates in the range of hundreds of Mbps can be achieved [96].

Although VLC ensures high data transmission rates, it is hindered from serving users in strong sunlight areas or shaded areas, resulting in limited coverage. Taking this into consideration, combining cellular communication with VLC has been proposed. A heterogeneous network integrating cellular and VLC communications guarantees good coverage from the cellular network and high capacity from the VLC network [97]. In such a heterogeneous network, the traffic congestion in the cellular network is minimized by offloading some of the traffic to the VLC network. This improves the spectrum reuse in the heterogeneous network, by utilizing both the licensed and the visible spectrums for communication [98]. In these networks, the

mobile users which are able to access the VLC network can relay the data to the mobile users which cannot be served by the VLC network. This can be realized using Device-to-Device (D2D) communication.

D2D communication enables mobile devices to transmit data directly between each other, without relaying it through a base station, thus improving the spectral and energy efficiencies, and minimizing the latency [99]. D2D communication can be integrated with VLC to serve mobile devices that are inaccessible by the VLC transmitters [98]. The mobile users being served can pay the mobile users which act as relays, for their services. However, there are many issues in such a VLC-D2D heterogeneous network. One of the most important issues is the determination of data transmission routes from the VLC transmitters to the end mobile devices. Traditionally, the VLC Service Provider (VLCSP) can determine this data transmission route based on maximizing its revenue and minimizing the overall latency, in a centralized manner [98].

However, the data transmission environment between the VLC transmitters and the mobile devices, and amongst the mobile devices themselves, in a VLC-D2D heterogeneous network, is highly dynamic. There can also exist a competition among the mobile devices accessible by the VLC transmitters, to relay the data to the mobile devices inaccessible by the VLC transmitters, and obtain revenue. A centralized solution available for the VLCSP and the mobile devices is difficult in such cases. Additionally, due to the random nature of the positions and data requirements of the mobile devices, the parameters in the network cannot be fully predicted. This calls for techniques to determine optimal distributed data transmission routes for a stochastic communication environment, which indeed is very challenging.

Recently, Machine Learning (ML) paradigms have gained wide popularity in several decision making scenarios, and Reinforcement Learning (RL) is a general

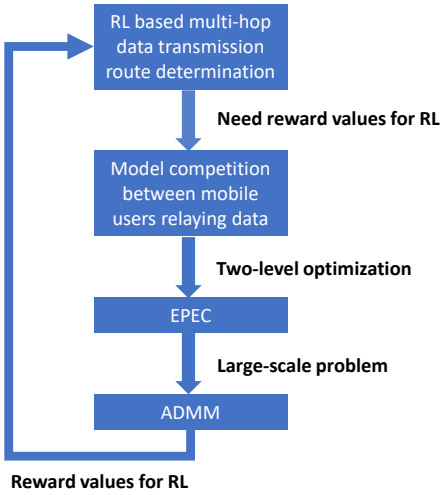


Figure 5.1: Proposed multi-hop route selection algorithm for VLC-D2D heterogeneous network.

framework that helps to learn behavior through trial-and-error interactions with a dynamic environment [100–102]. Consequently, taking into account the dynamic and unpredictable nature of our considered communication environment, in this chapter, we propose an RL based method to determine data transmission routes in the VLC-D2D heterogeneous network.

Being a learning method that does not require a model of the considered environment makes RL suitable for our scenario, using which we can deduce optimal data transmission routes through trial-and-error. In a typical RL scenario, a learning agent which is capable of taking *actions*, observes the environment after each of its actions [103]. Each action influences the *state* reached by the agent in the future. The success in RL scenarios is measured through total *rewards*, and hence, the agent basically aims to maximize its accumulated rewards through its actions. Therefore, in order to perform learning in our heterogeneous network scenario, we need a method to calculate the rewards.

Accordingly, we consider a network where the VLCSP, the Cellular Service Provider (CSP), and the mobile users, are independent entities. In the heterogeneous

network environment under consideration, the independent entities can use their experiences from interacting with the environment, to improve their behaviors [104]. In order to obtain the rewards, we also model the competition among the mobile devices accessible by the VLC transmitters in relaying the data. Subsequently, we propose an ADMM based EPEC approach for determining the optimal rewards dynamically, which is discussed in detail later. The interconnection between RL, EPEC, and ADMM in the proposed algorithm is shown in Fig. 5.1. The major contributions of this chapter are summarized as follows:

- We propose a multi-hop data transmission route determination method for an indoor VLC-D2D heterogeneous network, utilizing an RL based technique. This is a distributed method that can determine the route based on local information, unlike the conventional, centralized method where the VLCSP determines the route based on data transmission path delays.

- The proposed RL method utilizes *Q-learning*, which is a direct RL technique, and works by continuously improving the knowledge of the consequences of certain actions at certain states. In our scenario, it enables the transmitted data to learn from its interactions with the environment to find an optimal route in a distributed fashion.

- In order to compute the rewards dynamically for the RL based route determination method, we formulate the interactions between the mobile users which relay data using D2D communication, as an EPEC optimization problem. An EPEC is a hierarchical optimization problem that contains equilibrium constraints at two levels. Here, the utilities of the mobile users which send and receive the relayed data are simultaneously optimized by using EPEC.

- Then, we utilize the properties of ADMM as a large-scale optimization tool to solve the EPEC problem, given the large number of entities (mobile users),

in the considered scenario. Using ADMM, we achieve optimal solutions for the EPEC, which contribute to the rewards for the Q-learning based route determination approach.

- The effectiveness of the proposed algorithm is then validated through simulations, which emphasize the effect of the RL method on transmission capacity and latency. The simulation results highlight the effect of the number of learning steps and the importance of future rewards, on the data transmission rates and path delays.

The rest of this chapter is organized as follows. Some of the relevant previous works are discussed in Section 5.2. In Section 5.3, we present the system architecture and the key parameters used in our VLC-D2D heterogeneous network optimization model. Then, in Section 5.4, we formulate the optimization problem of the proposed model, and introduce the RL and EPEC formulations, which are used in our proposed algorithms. The proposed algorithms and their detailed analyses are provided in Section 5.5. Here, we discuss the Q-learning based algorithm for optimal data transmission route determination in Section 5.5.1, and the ADMM based EPEC algorithm for the determination of rewards in Section 5.5.2. We present the performance of the proposed Q-learning based optimal route determination method, which also incorporates the ADMM based EPEC technique, in Section 5.6, where we firstly discuss the simulation results in Section 5.6.1, and then, discuss a few aspects of the results in Section 5.6.2. Finally, conclusions are drawn in Section 5.7.

5.2 Related Work

Since the deployment of heterogeneous networks is a promising solution to the wireless capacity crunch issue, a great deal of research is prevalent in this area. Reference [105] shows 90% offloading from the macrocell base station by using heteroge-

neous small cell-based networks. A survey on the state-of-the-art and challenges of Long-Term Evolution-Advanced (LTE-A) heterogeneous networks is given in [106], where the elements of LTE-A heterogeneous networks introduced in different LTE releases of 3GPP are also summarized, among which D2D communication is introduced in Release 12. Reference [107] utilizes game-theoretic approaches to provide distributed solutions to the resource allocation issues in D2D communication underlaying cellular networks, where the complex strategies of the D2D and cellular users to maximize their own utilities are modeled using the tools from game theory. Improving the coverage for mobile users at the cell edge is of key importance in wireless communication, and is facilitated using D2D range extension in [108]. Reference [109] deals with the energy efficiency maximization in a D2D-assisted heterogeneous network, and considers optimal power allocation, along with user equipment association.

Owing to the remarkable wireless traffic offloading potential of VLC, the research on inclusion of VLC in heterogeneous networks is growing tremendously [110–114]. Reference [115] elaborates on user-centric VLC heterogeneous networks from the signal coverage, system control, and service provision perspectives, and discusses a few open challenges. An indoor heterogeneous network combining VLC and RF have been proposed in [116], where a new VLC frame, multiple access mechanism, and a novel handover scheme have been introduced, and the capacity performance of the network has been improved compared to an RF only system. Reference [117] proposes an indoor hybrid system integrating WiFi and VLC, which utilizes the bandwidth benefits of VLC, and proposes the optimal resource allocation among users. An indoor VLC and RF heterogeneous network is investigated from the energy efficiency point of view in [118]. Reference [119] considers an indoor VLC attocell and RF femtocell network, where the optimal resource allocation is achieved through a distributed algorithm.

In order to implement such complex heterogeneous networks and to ensure

proper allocation of resources, we need efficient algorithms that can handle complicated scenarios. Research on ML algorithms is quite prominent in this regard, of which RL research has gained a lot of momentum in the recent years. Since it is a learning method that does not need a model of the environment, RL is well suited for scenarios where the environment changes during learning [120]. It has been discussed widely in areas like machine learning, neural networks, operations research, control theory, and so on [121]. RL is the learning algorithm behind AlphaGo, the first computer program to defeat a professional player at Go, one of the most challenging classical games [122]. Reference [102] performs a basic survey of RL, and also discusses a few classic model-free algorithms. A survey of RL in robotics is given in [123], through behavior generation in robots. A comprehensive survey on the recent developments in RL with function approximation, and a comparison of the performance of different RL algorithms is provided in [121]. Reference [124] demonstrates a simple two-player soccer game using RL techniques, and a model for a route planning system based on multi-agent RL is proposed in [125].

Due to the flexibility and effectiveness of RL in learning, it has been applied to solve many issues in wireless communication. Reference [126] proposes an RL based sub-band selection policy to anti-jamming communications with Wideband Autonomous Cognitive Radios (WACRs) in a scenario with multiple policy-learning agents. The sub-band selection problem in wideband Cognitive Radio (CR) is dealt with in [127], using an extension of the Q-learning algorithm, called the replicated Q-learning. Some other applications of RL can be found in [128–135], which are in wireless sensor networks, LTE-A networks, cognitive network and radio resource management, wireless communication in healthcare, and energy-efficient wireless communication.

In order to harness the benefits of heterogeneous networks as well as the potentials of the VLC and D2D technologies, our scenario of interest is an indoor VLC-D2D

heterogeneous network. An indoor VLC-D2D heterogeneous network scenario has been considered in [98], where the mobile devices accessible by the VLC transmitters, relay the data to the mobile devices which are not accessible by the VLC transmitters. Reference [98] proposes a hierarchical game, where the interactions between the VLCSP and the CSP, the CSP and the mobile users, and between the VLCSP and the mobile users are modeled as Stackelberg games. The network arrives at the solution when the mobile users achieve a Nash equilibrium among them, and the above mentioned Stackelberg games achieve Stackelberg equilibriums. Nevertheless, the data transmission route is determined by the VLCSP, by considering only the transmission path delays.

In the case of the indoor VLC-D2D heterogeneous network considered in this chapter, we adopt an RL based route determination method, as discussed in the previous section. To my best knowledge, an RL based method to determine data transmission routes in a dynamic VLC-D2D heterogeneous network environment has not been employed in any previous work. Also, determining the rewards for RL dynamically during the learning process has not been considered before. Hence, in this chapter, we utilize the potential of RL as a powerful learning algorithm for a stochastic environment. The rewards are computed during the RL process using EPEC, which is solved by making use of the capacity of ADMM as an efficient large-scale optimization tool.

5.3 System Model

An indoor downlink scenario consisting of K VLC transmitters of a VLCSP, a CSP, and T mobile users is considered, following the settings in [98]. Out of the T mobile users, there are M Mobile Users in Coverage area (MUiCs) of the VLC transmitters and N Mobile Users in Darkness (MUiDs). As shown in Fig. 5.2, MUiCs

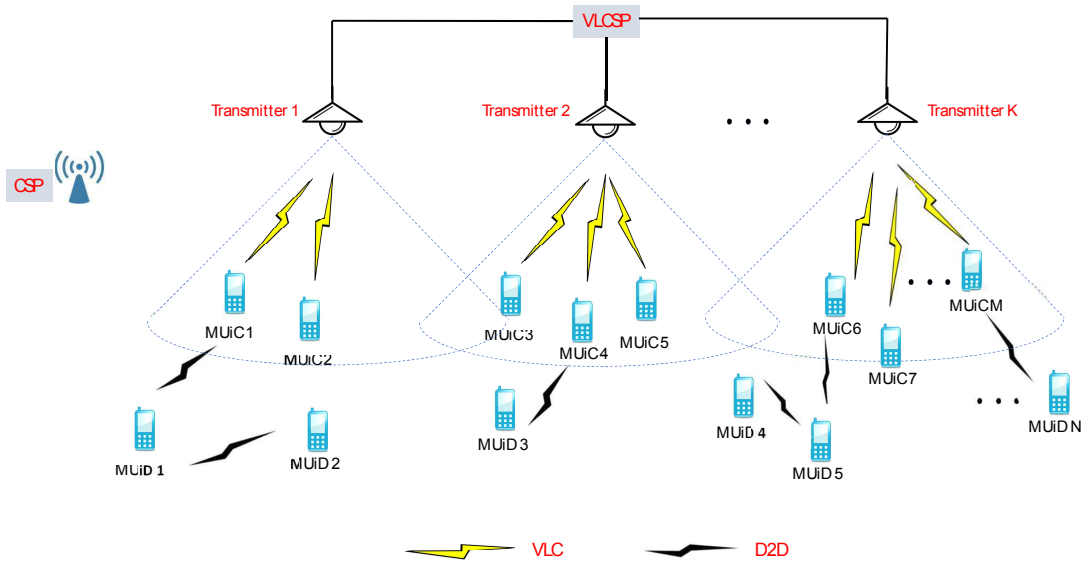


Figure 5.2: System architecture of VLC-D2D heterogeneous network.

are the mobile users which are under the visible light cones of the VLC transmitters, and MUiDs are the mobile users which are not directly under the visible light coverage of the VLC transmitters. For data transmission, as the MUiDs can only receive weak signals from the cellular base stations, D2D communication is adopted in which some MUiCs and MUiDs can act as relays for the end MUiDs [98]. The MUiCs and MUiDs that relay the service from the VLC transmitter to the end MUiD are called Mobile Users as Relays (MUaRs). As shown in Fig. 5.2, MUiC 1 is able to receive traffic from the VLC transmitter. Then, MUiC 1 acts as a relay for MUiD 1 which also acts as a relay for the end MUiD 2.

For the visible light communication between VLC transmitter κ, for $\kappa \in \{1, 2, \ldots, K\}$, and MUiC m, for $m \in \{1, 2, \ldots, M\}$, the transmission rate can be defined as

$$C_{\kappa m} = S_{\kappa} \log_2 \left(1 + \frac{P_{\kappa} G_{\kappa m}}{\sigma_I^2 + \sigma_N^2}\right), \tag{5.1}$$

where S_{κ} is the amount of spectrum for each of the K VLC transmitters, P_{κ} is the

transmit power of each of the K VLC transmitters, $G_{\kappa m}$ is the channel gain between VLC transmitter κ and MUiC m, σ_I^2 is the interference from other visible light sources, and σ_N^2 is the channel noise.

For the D2D communication between MUaR i, for $i \in \{1, 2, \ldots, T\}$, and MUaR j, for $j \in \{1, 2, \ldots, T\}$, $i \neq j$ (or between MUaR i, for $i \in \{1, 2, \ldots, T\}$, and end MUiD j, for $j \in \{1, 2, \ldots, N\}$), the transmission rate can be defined as

$$C_{ij} = S_{ij} \log_2 \left(1 + \frac{P_{ij} G_{ij}}{I_c + N_0}\right), \tag{5.2}$$

where S_{ij} is the amount of allocated wireless spectrum for MUaR i to transmit data to MUaR j (or for MUaR i to transmit to the end MUiD), P_{ij} is the transmit power from MUaR i to MUaR j (or from MUaR i to the end MUiD), G_{ij} is the channel gain between MUaR i and MUaR j (or between MUaR i and the end MUiD), I_c is the interference from cellular uplink, and N_0 is the additive noise.

We consider the VLC transmitters, the MUaRs and the end MUiDs in our network as nodes in a graph. For the multihop wireless data transmission from a VLC transmitter to an end MUiD, the data packet size is denoted by $\mathcal{M}$. The penalty due to the delay between node i and node j can be expressed as [98]

$$D_{ij} = \alpha \frac{\mathcal{M}}{C_{ij}}, \tag{5.3}$$

where α denotes the penalty of unit service delay. The penalty due to the total delay is the sum of the penalties between all the nodes in the data transmission service route.

Accordingly, we consider an L-hop route, $\{h_1, h_2, \ldots, h_L\}$, for the data transmission from VLC transmitter κ, for $\kappa \in \{1, 2, \ldots, K\}$, to the end MUiD, where node h_l, $\forall l \in \{1, 2, \ldots, L-1\}$, is an MUaR, and node h_L is the end MUiD.

The utility for node h_l in the transmission route, is given by the revenue received by the node minus the delay penalty and the cost of the wireless spectrum. The utility function can be expressed as

$$U_l^{MUaR} = \beta\mathcal{M} - \gamma D_l - r_l S_l + r_{l-1} S_{l-1}, \tag{5.4}$$

where β and γ are the weight factors. D_l has the same physical meaning as D_{ij} from (5.3), which is the penalty of delay from node h_l to node h_{l+1} in the transmission route, r_l is the price per unit of the wireless spectrum resources, and S_l has the same physical meaning as S_{ij} from (5.2), which is the amount of allocated wireless spectrum for node h_l. Thus, $\beta\mathcal{M}$ is the revenue obtained by node h_l by relaying a data packet of size $\mathcal{M}$, γD_l is the delay penalty, $r_l S_l$ is the price paid for the spectrum, and $r_{l-1} S_{l-1}$ is the revenue obtained by providing spectrum to the upstream node h_{l-1}.

In this chapter, we propose to determine the L-hop data transmission route, as shown in Fig. 5.3, from a VLC transmitter to the end MUiD, utilizing an RL technique, which is discussed in detail in the next section. In order to formulate the model for the RL technique, let us consider the transmitted data from a particular VLC transmitter to a particular end MUiD as the learning *agent*, the current location (node) of the transmitted data as the *state*, and the transmission direction of the agent from the current node to the next accessible node as an *action*. Here, as there can be different data transmitted from different VLC transmitters to the same end MUiD or different end MUiDs at the same time, we consider a scenario wherein a number of single agent RL tasks take place simultaneously.

Another key factor in this learning technique is *reward*, which is simply the payoff obtained by the agent, by performing an action at a given state. In RL, the actions may not only affect the immediate reward, but also the rewards obtained in future [103]. In our scenario, the transmission data rate can be considered as the reward. As such, we define the reward matrix $\mathbf{R}$, which contains the rewards obtained

by the agent, by selecting every possible action (transmitting to all accessible nodes), from every state (current position of transmitted data). We express the reward matrix for our single agent RL scenario as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & . & . & . & . & . & \mathbf{R}_{1(K+T)} \\ . & . & . & . & . & . & . \\ \mathbf{R}_{K1} & . & . & . & . & . & \mathbf{R}_{K(K+T)} \\ . & . & . & . & . & . & . \\ \mathbf{R}_{(K+M)1} & . & . & . & . & . & \mathbf{R}_{(K+M)(K+T)} \\ . & . & . & . & . & . & . \\ \mathbf{R}_{(K+T)1} & . & . & . & . & . & \mathbf{R}_{(K+T)(K+T)} \end{bmatrix},$$

where $T = M + N$ is the total number of mobile devices in our indoor scenario. Here, the rows indicate the states, and the columns indicate the actions. For example, $\mathbf{R}_{1(K+1)}$ is the reward obtained by the agent, by taking the action of transmitting data to state $K + 1$ (MUiC 1) from state 1 (VLC transmitter 1), $\mathbf{R}_{(K+M+1)(K+T)}$ is the reward obtained by the agent, by taking the action of transmitting data to state $K + T$ (MUiD N) from state $K + M + 1$ (MUiD 1), and so on.

Here, $\mathbf{R}_{11}, \mathbf{R}_{22}, \ldots, \mathbf{R}_{(K+T)(K+T)}$ can be taken as 0, since none of the nodes transmits data to themselves. $\mathbf{R}_{11}, \ldots, \mathbf{R}_{1K}, \ldots, \mathbf{R}_{K1}, \ldots, \mathbf{R}_{KK}$ will be 0, as the VLC transmitters will not transmit data to each other. $\mathbf{R}_{(K+1)1}, \ldots, \mathbf{R}_{(K+1)K}, \ldots, \mathbf{R}_{(K+T)1}$, $\ldots, \mathbf{R}_{(K+T)K}$ will also be 0, as the mobile devices do not transmit data to the VLC transmitters. Additionally, as the VLC transmitters do not send data to the N MUiDs, $\mathbf{R}_{1(K+M+1)}, \ldots, \mathbf{R}_{1(K+T)}, \ldots, \mathbf{R}_{K(K+M+1)}, \ldots, \mathbf{R}_{K(K+T)}$ will be 0. In addition, since an MUiC will not transmit data to another MUiC, columns $K+1$ through $K + M$ of rows $K + 1$ through $K + M$ will be 0. Finally, since an MUiD will not transmit data to any MUiC, columns $K + 1$ through $K + M$ of rows $K + M + 1$ through $K + T$ will also be 0.

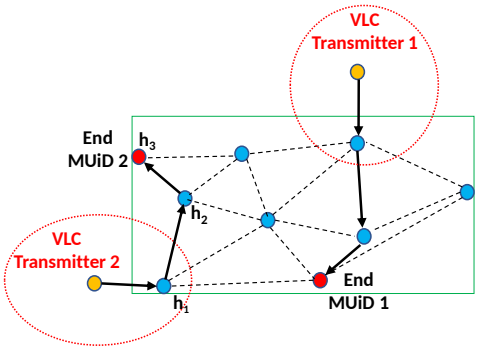


Figure 5.3: L-hop data transmission route.

5.4 Problem Formulation

In our considered scenario, the initial problem for VLC transmitter κ, for $\kappa \in \{1, 2, \ldots, K\}$, is determining an optimal route, $\{h_1, h_2, \ldots, h_L\}$, from the VLC transmitter to the end MUiD. Clearly, this can be considered as a single agent RL scenario, where the agent needs to learn the environment through its experience and take actions to maximize its rewards. In order to realize this, the agent has to *exploit* what is already known, and simultaneously, has to *explore* to be able to choose better actions in the future [103].

Q-learning is a model-free RL technique [136], in the sense that the agent can directly learn about its optimal policies, i.e., a mapping from a state to an action of the agent, without knowing the future rewards [120]. Q-learning works by learning an *action-value* function that constructs the optimal *policy*, by selecting the action with the highest *value*. This optimal policy gives the maximum achievable expected value of the total reward. In this chapter, we utilize single agent Q-learning to decide the optimal route from each VLC transmitter to the end MUiD. The detailed analysis of the Q-learning algorithm is included in the next section.

For utilizing Q-learning to decide the data transmission route for VLC transmitter κ, for $\kappa \in \{1, 2, \ldots, K\}$, we need to generate the reward matrix, $\mathbf{R}$, as shown in

the previous section. This is performed as an iterative update process, while modeling firstly, the interactions between the VLC transmitters and the MUaRs, and secondly, the interactions between the MUaRs and the end MUiDs.

Initially, we model the interactions between each of the K VLC transmitters and the set of M MUiCs, which is the set of candidates for the first hop, h_1, using (5.1). This generates some of the rewards in the $\mathbf{R}$ matrix, which are the reward values associated with all the possible first hops (MUiCs) for the agent, from each of the VLC transmitters.

In our scenario, we consider the first hops, h_1 to be MUiCs, and the rest of the hops in the L-hop route to be MUiDs, including the MUiDs acting as MUaRs, and the end MUiDs. Hence, we generalize and formulate the utility function of the MUaRs as

$$U_t^{MUaR} = \beta\mathcal{M} - \gamma D_t - r_t S_t + r_{t-1} S_{t-1}, \tag{5.5}$$

where all the symbols have the same physical meanings as given in (5.4), except that the subscript t denotes any MUaR.

When the current MUaR node transmits data to the next MUaR nodes, we consider S_t to be the spectrum allocated to the current node by the next nodes. An optimal value of S_t ensures a successful transmission from the current node to the next nodes. Hence, we express the utility function of the current MUaR node, with respect to the allocated spectrum resource, as

$$U_t^{MUaR}(S_t) = \beta\mathcal{M} - \gamma D_t - r_t S_t. \tag{5.6}$$

Accordingly, the utility of the next MUaR nodes is the revenue obtained by allocating the spectrum to the current MUaR node. Hence, we formulate the utility function of the next MUaR nodes, with respect to the price per unit of the spectrum

resources, r_t, as

$$U_{t+1}^{MUaR}(r_t) = r_t S_t. \tag{5.7}$$

We model the interactions between the MUaRs as an optimization problem, in order to maximize the utilities of the next MUaR nodes as in (5.7), while maximizing the utility of the current node as in (5.6). The optimization problem is expressed as

$$\begin{aligned} &\max_{r_t} U_{t+1}^{MUaR}(r_t) = r_t S_t \\ &s.t. \begin{cases} r_t > 0, \\ \mathbf{S}_t = \arg\max \left(U_t^{MUaR}(\mathbf{S}_t) = \beta \mathcal{M} - \gamma D_t - r_t S_t \right), \\ s.t.\ S_t > 0, \end{cases} \end{aligned} \tag{5.8}$$

for $t \in \{1, 2, \ldots, T\}$, where $\mathbf{S}_t$ is the vector containing the optimal values of S_t for the current MUaR to transmit data to the next MUaRs.

The values in $\mathbf{S}_t$ form the rest of the rewards in the $\mathbf{R}$ matrix, which are the reward values associated with all the possible next hops (MUaRs) for all the possible first hops (MUiCs), and then, the reward values associated with all the possible hops after that in the L-hop route, and so on. This helps generate the values in the $\mathbf{Q}$ matrix, which is the core of the Q-learning algorithm for the L-hop data transmission route calculation, and is discussed in detail in the next section.

Here, (5.8) is a two-level optimization problem. Such a hierarchical optimization problem that contains equilibrium criteria at two levels is called an EPEC [75,137], as already discussed in Chapter 4. (5.8) is an optimization problem, where the MUaRs at two levels have their sets of equilibrium constraints. A centralized solution for all parties is difficult in such cases. We need a solution that can maximize the utility of the current MUaR node, while maximizing the utilities of the next MUaR nodes.

Stackelberg games can be employed in similar cases [5]. However, they work well only in scenarios with one leader and multiple followers. If we need to coordinate

multiple conflicting utilities at both the levels, it might demand high complexity to provide optimal results. In addition, if we consider scenarios with a large number of entities, we need an algorithm that can converge for large networks. On that account, we consider ADMM for the optimization problem discussed above, for the VLC-D2D scenario. ADMM for EPEC is discussed in detail in the next section.

5.5 Algorithm Analysis

As mentioned before, Q-learning is a model-free technique used in RL scenarios, where the environment behavior is not fully known. In our scenario, the positions and data requirements of the mobile devices are random in nature and cannot be fully predicted. Therefore, Q-learning is a suitable tool to perform learning. Specifically, we propose to utilize the Q-learning technique to determine the L-hop data transmission routes from the VLC transmitters to the end MUiDs.

Here, the Q-learning based algorithm for optimal data transmission route determination is discussed in Section 5.5.1, and the ADMM based EPEC algorithm for the determination of rewards is discussed in Section 5.5.2.

5.5.1 Q-learning for Route Selection

The general Q-learning algorithm works by evaluating $\mathbf{Q}$ values for different $(state, action)$ pairs, denoted as $\mathbf{Q}(S, A)$. In order to implement the Q-learning algorithm, we need the immediate reward values associated with various actions from different states, i.e., $\mathbf{R}$ values for different $(state, action)$ pairs, denoted as $\mathbf{R}(S, A)$. For recording these reward values, we create a rewards matrix, $\mathbf{R}$, as shown in Section 5.3.

However, the reward values are not known beforehand in the considered scenario. We compute the reward values dynamically during Q-learning, and record

Algorithm 5.1 RL based multi-hop route selection algorithm in VLC-D2D heterogeneous network

Initialization:
i) Initialize the reward and Q-learning matrices:
$\mathbf{R} = \mathbf{0}$, and $\mathbf{Q} = \mathbf{0}$.
ii) Set the η parameter, the maximum number of learning steps, L_{max}, the maximum number of hops, L, and the minimum threshold for the D2D transmission data rate C_{th}.
iii) Set the current number of learning steps, $s = 0$, and the current number of hops, $l = 0$.
Generating the Q matrix:
while $s \leq L_{max}$ **do**
  Select an initial state for each agent as one of the K VLC transmitters.
  while current state $\neq$ end MUiD **do**
    if current state = VLC transmitter **then**
      Compute $\mathbf{R}(S, A)$ using Algorithm 5.2 and update.
    else
      Compute $\mathbf{R}(S, A)$ using Algorithm 5.3 and update.
    end if
    Select one of all possible actions for the current state.
    Using this possible action, go to the next state.
    Get the maximum $\mathbf{Q}$ value for this next state based on the stored $\mathbf{Q}$ values for all possible actions.
    Compute $\mathbf{Q}(S, A)$ for the current state using (5.9) and update.
    Update current state = next state.
  end while
  $s = s + 1$.
end while
Utilizing the Q matrix to find the best route:
Set current state = initial state.
From current state, find the action with the highest $\mathbf{Q}$ value.
$l = l + 1$.
if $l > L$ **or** $C_{ij} < C_{th}$ **then**
  Routing failure, no available route.
end if
Set current state = next state.
Repeat steps 21 to 26 until current state = end MUiD.
Output L-hop transmission route.

them in the $\mathbf{R}$ matrix. In this scenario, the VLC controller can implement the Q-learning algorithm, since it can act as an entity to co-ordinate and update the values of $\mathbf{R}(S, A)$ provided by the other entities.

As our Q-learning route selection algorithm runs, depending on the current state of the agent, there can be two different scenarios.

- **Current state = VLC transmitter** At the start of the algorithm, the transmitted data will be at a VLC transmitter, and hence, the next state will be an

MUiC. Therefore, we can compute the reward values using (5.1), which gives the capacity for VLC transmission between the VLC transmitter and the MUiC.

- **Current state = MUaR** In our considered L-hop route, the first hops are MUiCs, and the rest of the hops are MUiDs acting as MUaRs. These interactions between the MUaRs are modeled in Section 5.4, using the EPEC in (5.8). In this case, we invoke the ADMM technique to solve this EPEC, which is discussed in detail in Section 5.5.2. Thus, for the current state and action as MUaRs, we obtain and record the reward values.

Now, we add a similar matrix $\mathbf{Q}$, to represent the agent's learning through experience. As in the case of the $\mathbf{R}$ matrix, the rows of the $\mathbf{Q}$ matrix represent the current state of the agent, and the columns represent the actions. The agent begins the learning by knowing nothing, and hence, the $\mathbf{Q}$ matrix is initialized to $\mathbf{0}$.

The Q-learning route selection algorithm implemented by the VLC controller is described in detail in Algorithm 5.1. The algorithm has two main parts: generating the $\mathbf{Q}$ matrix, and utilizing the $\mathbf{Q}$ matrix to find the best data transmission route.

5.5.1.1 Generating the Q matrix

The generation of the $\mathbf{Q}$ matrix is performed for a number of iterations, which we set as the maximum number of learning steps, as in the *while* loop from steps 3-18. Initially, one of the K VLC transmitters is chosen to be the initial state of the agent. The *while* loop in steps 5-16 runs for as long as the current state of the agent becomes the end MUiD. Within each learning step, the agent continues the learning until it reaches the end MUiD within the maximum number of hops.

If the current state of the agent is a VLC transmitter, then the VLC transmitter computes the $\mathbf{R}(S, A)$ values for the $\mathbf{R}$ matrix using Algorithm 5.2. If the current state of the agent is not a VLC transmitter, then the current state of the agent will

be an MUaR, which results in an EPEC as in (5.8). The corresponding $\mathbf{R}(S, A)$ values are evaluated by the MUaRs, using the ADMM for EPEC algorithm, which is discussed in detail in Algorithm 5.3.

Once the $\mathbf{R}(S, A)$ values are evaluated, steps 11-15 are executed, which form the core of the learning part. The determination of the **Q**-value for each $(state, action)$ pair, $\mathbf{Q}(S, A)$, for a particular $\mathbf{R}(S, A)$ in the reward matrix, is given by

$$\mathbf{Q}(S, A) = \mathbf{R}(S, A) + \eta * \max[\mathbf{Q}(S', \mathbf{A}')], \tag{5.9}$$

where S is the current state of the agent, A is the selected action for the current state, S' is the next state for state S, and $\mathbf{A}'$ is the action space of the agent in state S'. Here, η is the *discount factor*, and is the weight given to the future rewards. An η value closer to 0 indicates more preference given to immediate rewards, and if η is closer to 1, future rewards are considered with greater weight.

5.5.1.2 Utilizing the Q matrix to find the best route

Once the agent learns from its experience and obtains an optimized **Q** matrix from the above mentioned steps, we put this learning to our use. The remaining steps in the algorithm explain how the agent navigates through the obtained **Q** matrix, to find an optimum route. It starts from the initial state, which is a VLC transmitter, and finds the action with the highest **Q**-value for this initial state, to be the next state. Now, for this next state, the action with the highest **Q**-value is found, to be the next state. This process is continued until the current state of the agent becomes the end MUiD. Each time when the next state is updated, and thus, one more hop is added to the route, it is checked if the number of hops, l has not exceeded the maximum number of hops, L. This procedure results in an L-hop data transmission route from the VLC transmitter to the end MUiD.

Algorithm 5.2 Computation of rewards for VLC

1: For VLC transmitter κ, for $\kappa \in \{1, 2, \ldots, K\}$, and MUiC m, for $m \in \{1, 2, \ldots, M\}$, compute the reward, $\mathbf{R}(S, A)$ as
$C_{\kappa m} = S_{\kappa} \log_2 \left(1 + \frac{P_{\kappa} G_{\kappa m}}{\sigma_I^2 + \sigma_N^2}\right)$, as in (5.1).
2: Output $\mathbf{R}(S, A)$.

Here, an important issue is regarding the choice of actions during learning. The agent can either choose an action to maximize its current **Q**-value, or choose an action randomly from among all of its possible actions. The approach in which the agent chooses to maximize its current **Q**-value is called the *greedy* approach [120]. The agent can be trapped in a local optimum in this case, and the solution is for the agent to explore other possible actions. In this chapter, we face this issue by employing an ϵ-greedy method, where ϵ is a probability factor. The agent chooses the action with the maximum **Q**-value with a probability of $1 - \epsilon$, and a random action with a probability of ϵ.

Next, we discuss the reward calculation procedure using the ADMM for EPEC algorithm.

5.5.2 ADMM for EPEC

In this subsection, we firstly discuss the concept of ADMM as applied to the considered scenario [77]. Then, we explain the iterated process of the ADMM based EPEC algorithm used to model the interactions between the MUaRs. Finally, we discuss the convergence of the proposed ADMM based EPEC algorithm.

5.5.2.1 ADMM

To demonstrate the ADMM, let us consider an example scenario with one current MUaR and T' next MUaRs, where the current MUaR wants to maximize its

utility as

$$\begin{aligned} &\max U_t^{MUaR}(\mathbf{S}_t) = \beta\mathcal{M} - \gamma D_t - r_t S_t, \\ &s.t. S_t > 0, \end{aligned} \tag{5.10}$$

where S_t is a real, scalar variable.

Here, the values of $\mathbf{S}_t$ can be updated by the current MUaR as

$$\mathbf{S}_{ti}(t+1) = \arg\max \left(U_t^{MUaR}(\mathbf{S}_{ti})\right) + \lambda_i(t)\mathbf{S}_{ti} + \Psi, \tag{5.11}$$

$\forall i \in \{1, 2, \ldots, T'\}$, and

$$\Psi = \frac{\rho}{2}\|\mathbf{S}_{ti}\|_2^2. \tag{5.12}$$

Here, $\rho > 0$ is a damping factor, t is the iteration step index, and $\|\cdot\|_2$ denotes the Frobenius norm [76]. λ is the dual variable, and it is updated as

$$\lambda_i(t+1) = \lambda_i(t) + \rho\left(\mathbf{S}_{ti}(t+1)\right). \tag{5.13}$$

Due to its quick convergence property, ADMM is used for large-scale optimization problems in large networks [76], as discussed before.

5.5.2.2 ADMM for EPEC to obtain $\mathbf{R}(S, A)$ for MUaRs

In the Q-learning based route selection algorithm, if the current state of the agent is an MUaR, modeling the interaction between the current state (MUaR) and the next hop (MUaR) results in an EPEC as in (5.8). Each of the accessible next MUaRs provides spectrum, S_t, to the current MUaR, for which the current MUaR pays a price, r_t to the next MUaR node which it transmits to. The determination of the optimal values of $\mathbf{S}_t$ and r_t by the current and next MUaRs, respectively, forms the core of the ADMM based EPEC method in our scenario. It is an iterative process as shown in Algorithm 5.3, in which each iteration can be explained in two steps as given below:

5.5.2.2.1 Optimization problem of the current MUaR node Initially, the next MUaR nodes announce the prices for the bandwidth that they are providing. The current MUaR uses the announced prices at the start of each iteration, p, to calculate the values in $\mathbf{S}_t$, the amount of spectrum to be purchased from each accessible next MUaR, to maximize its profit $U_t^{MUaR}(\mathbf{S}_t)$. Here, the superscript p denotes the value at the p^{th} iteration of the method, and the superscript $p-1$ denotes the value at the $p-1^{th}$ iteration of the method. This is the *inner loop* of ADMM. t is the iteration step index of the inner loop.

We described $\mathbf{S}_t$ in (5.8) as $\mathbf{S}_t = \arg\max\left(U_t^{MUaR}(\mathbf{S}_t)\right)$. For the current MUaR, maximizing its utility, $\left(U_t^{MUaR}(\mathbf{S}_t)\right)$ forms a set of values, $\mathbf{S}_t$. Hence, the values in $\mathbf{S}_t$ are updated at each iteration of the inner loop by the current MUaR as

$$\mathbf{S}_{ti}^{(p)}(t+1) = \arg\max\left(U_t^{MUaR}(\mathbf{S}_{ti})\right) + \lambda_i{}^{(p)}(t)\mathbf{S}_{ti} + \Psi, \tag{5.14}$$

where

$$\Psi = \frac{\rho}{2}\|\mathbf{S}_{ti}\|_2^2, \tag{5.15}$$

where $\rho > 0$ is a damping factor as mentioned above, and λ is the dual variable, which is updated as

$$\lambda_i{}^{(p)}(t+1) = \lambda_i{}^{(p)}(t) + \rho\left(\mathbf{S}_{ti}^{(p)}(t+1)\right). \tag{5.16}$$

At the end of the inner loop during each iteration, p, of the outer loop, the current MUaR arrives at a vector of S_t values, $\mathbf{S}_t$, which maximizes its utility. At the same time, these values are predicted by the next MUaRs, and are used to update the values of r_t.

5.5.2.2.2 Optimization problem of the next MUaR nodes The next MUaRs are able to predict the behavior of the current MUaR and the values in $\mathbf{S}_t$. The next

MUaRs then execute ADMM as

$$r_t(i)^{(p)}(t+1) = \arg\max \left(U_{t+1}^{MUaR}\left(r_t(i)\right)\right) + \lambda^{(p)}(t) r_t(i) + \Psi, \tag{5.17}$$

where

$$\Psi = \frac{\rho}{2}\|r_t(i)\|_2^2. \tag{5.18}$$

Here, $\rho > 0$ is the damping factor, and λ is the dual variable, which is updated as

$$\lambda^{(p)}(t+1) = \lambda^{(p)}(t) + \rho\left(r_t(i)^{(p)}(t+1)\right). \tag{5.19}$$

Thus, the next MUaRs recalculate the values of r_t that maximize their profits.

The updated values, $r_t{}^{(p+1)}$, are then provided to the current MUaR for the $(p+1)^{th}$ iteration. This is the *outer loop* of ADMM. The outer loop terminates when

$$\left\| \sum_{t=1}^{T_1} U_{t+1}^{MUaR}(r_t^{(p)}) - \sum_{t=1}^{T_1} U_{t+1}^{MUaR}(r_t^{(p-1)}) \right\| < \varepsilon, \tag{5.20}$$

where T_1 is the number of next MUaR nodes, and ε is a pre-determined threshold. ADMM algorithm for EPEC is shown in Algorithm 5.3.

5.5.2.3 Convergence

ADMM is a widely used large-scale optimization tool for both convex and nonconvex objective functions. From (5.8), the utility functions of the current and next MUaR nodes in our considered scenario, $U_t^{MUaR}(\mathbf{S}_t)$ and $U_{t+1}^{MUaR}(r_t)$, respectively, are linear. The convergence of ADMM in the case of nonconvex objective functions is discussed in [90]. ADMM is guaranteed to converge to the set of stationary solutions, in the case of nonconvex objective functions, if the penalty parameter, ρ, is chosen to be sufficiently large. In our scenario, ADMM converges to the optimal values of $\mathbf{S}_t$ and r_t, which simultaneously maximize the utilities for the current and next MUaR nodes, respectively. Reference [90] provides the detailed convergence analysis of nonconvex objective functions.

Algorithm 5.3 Computation of rewards for D2D using ADMM for EPEC

1: **Initialization:**
Set the ε parameter, and number of iterations, $p = 0$.
2: **ADMM for EPEC between MUaRs:**
3: **while** $\left\| \sum_{t=1}^{T_1} U_{t+1}^{MUaR}(r_t^{(p)}) - \sum_{t=1}^{T_1} U_{t+1}^{MUaR}(r_t^{(p-1)}) \right\| \geq \varepsilon$ **do**
4: Optimization for the current MUaR node (*inner loop*):
The current MUaR node uses r_t to evaluate optimal $\mathbf{S}_t$, such that its utility is maximized.
5: Optimization for the next MUaR nodes (*outer loop*):
The next MUaR nodes predict the behavior of the current MUaR node, i.e., they deduce the optimal $\mathbf{S}_t$ obtained by the current MUaR node, and evaluate optimal r_t, such that their utilities are maximized.
6: $p = p + 1$.
7: **end while**
8: For MUaR i, for $i \in \{1, 2, \ldots, T\}$, and MUaR j, for $j \in \{1, 2, \ldots, T\}$, $i \neq j$ (or between MUaR i, for $i \in \{1, 2, \ldots, T\}$, and end MUiD j, for $j \in \{1, 2, \ldots, N\}$), compute the reward, $\mathbf{R}(S, A)$ as
$C_{ij} = S_{ij} \log_2 \left(1 + \frac{P_{ij} G_{ij}}{I_c + N_0}\right)$, as in (5.2), using optimal values from $\mathbf{S}_t$ as S_{ij}.
9: Output $\mathbf{R}(S, A)$.

Lemma 5.1. *The L-hop route for data transmission from the VLC transmitter to the end MUiD, obtained through the proposed RL based method is optimal.*

Proof. In the Q-learning algorithm, the reward for the first hop in the L-hop route is the data rate for the communication between the VLC transmitter and the MUiC, as determined using Algorithm 5.2. According to the received $\mathcal{M}$ and the known values of r_t, the current MUaR calculates the optimal $\mathbf{S}_t$ that maximizes its utility, which is predicted by the next MUaRs, and is utilized in calculating the optimal values of r_t that maximize their utilities. This process between the MUaRs is repeated until the ADMM converges, as shown in Algorithm 5.3. The utilities of the MUaRs computed using the optimal values of $\mathbf{S}_t$ and r_t form the rewards for the next hops in the L-hop route. Since the utilities of the MUaRs in the L-hop route are obtained using the optimal values of $\mathbf{S}_t$ and r_t, none of the MUaRs can deviate from these optimal values for better utilities. Clearly, the L-hop route deduced by the Q-learning algorithm which consists of these MUaRs is optimal. □

5.6 Performance Evaluation

Here, we discuss the simulation results in detail in Section 5.6.1 and discuss a few important aspects of the results in Section 5.6.2.

5.6.1 Simulation Results

In this section, we evaluate the performance of the proposed route selection algorithm using MATLAB. We consider a 5 m $\times$ 5 m room for the indoor VLC-D2D scenario. The number of VLC transmitters, K, and the number of mobile users, T, are varied to study different cases for evaluation purposes. The mobile users are placed randomly in the room, making some of them MUiCs and some of them MUiDs. The VLC transmitter(s) are placed equidistant along a non-diagonal line through the center of the room (if there is only one VLC transmitter, it is placed at the exact center of the room). Here, the spectrum for each VLC transmitter, S_κ is set as 1 GHz. The transmit power of each VLC transmitter, P_κ is set as 1 W, the typical output power of a white LED. Both the interference and noise for the VLC transmission, σ_I^2 and σ_N^2 are set to -20 dB. The interference and noise for the D2D transmission, I_c and N_0, are also set to -20 dB. The transmit power of each mobile user, P_{ij} is set as 300 mW. The data packet size, $\mathcal{M}$ is set as 1 Kb, and α is set to 1 s^{-1}. The values of β and γ are set to 10^6 b^{-1} and 10^6, respectively. We set the maximum number of hops for the data transmission route, L as 3. For Q-learning, we vary L_{max}, the maximum number of steps between 1 and 100, to study the effect of learning on different parameters. We set the default value of η as 0.8, to emphasize future rewards, and the greedy factor, ϵ is set as 0.1. The values of ε and ρ for ADMM are set as 10^{-6} and 1.5, respectively.

Fig. 5.4 shows the effect of Q-learning on the VLC transmission data rate for two cases: 1 VLC transmitter and 5 mobile users, and 2 VLC transmitters and 10

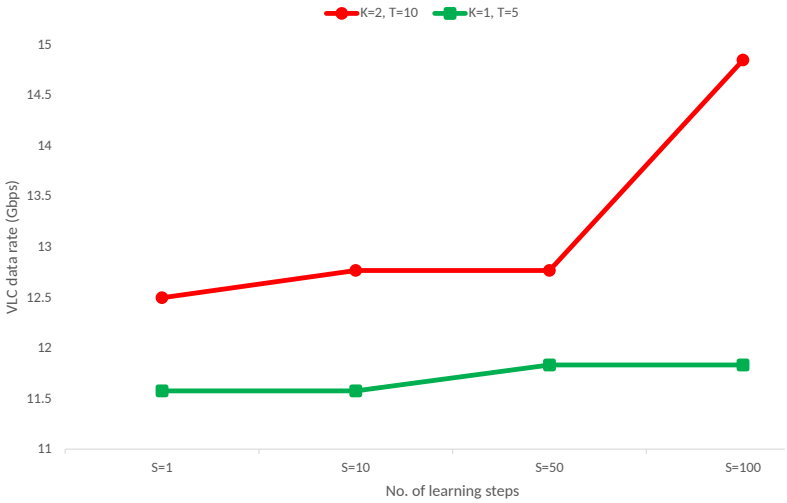


Figure 5.4: Effect of Q-learning on VLC transmission data rate.

mobile users. The figure shows the transmission capacity for the VLC communication between the VLC transmitter and the MUiC. We can see that as the number of learning steps increases from 1 to 100, the VLC transmission (first hop) capacity from the VLC transmitter to the MUiC increases, which demonstrates the impact of RL on improving the VLC transmission capacity. Specifically, we observe that as the number of learning steps doubles from 50 to 100, the VLC data rate improves by around 16% for the case with 2 VLC transmitters and 10 mobile users. This is due to a better $\mathbf{Q}$ matrix achieved via better learning through more interactions with the environment.

Fig. 5.5 shows the effect of Q-learning on the delay in VLC transmission for two cases: 1 VLC transmitter and 5 mobile users, and 2 VLC transmitters and 10 mobile users. The figure shows the delay for the VLC communication between the VLC transmitter and the MUiC. We can see that the VLC transmission (first hop) delay from the VLC transmitter to the MUiC decreases, as the number of learning steps increases from 1 to 100. This shows how RL helps in minimizing the VLC transmission delay in a VLC-D2D heterogeneous network. Here, we can see in particular that as

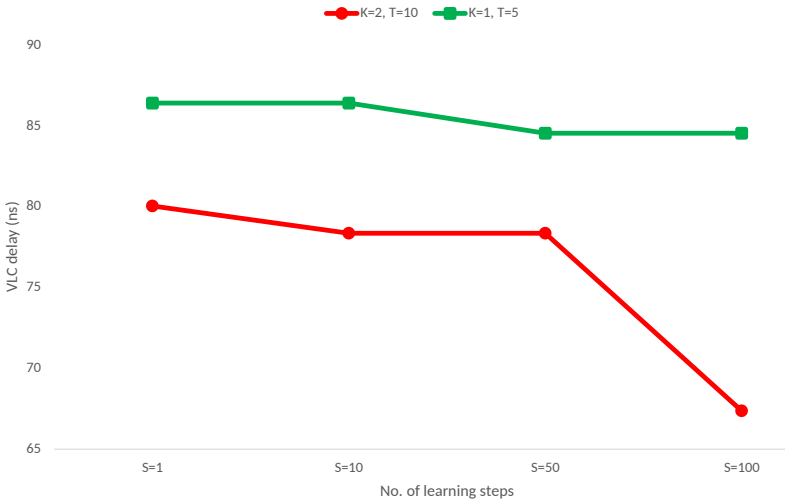


Figure 5.5: Effect of Q-learning on VLC transmission delay.

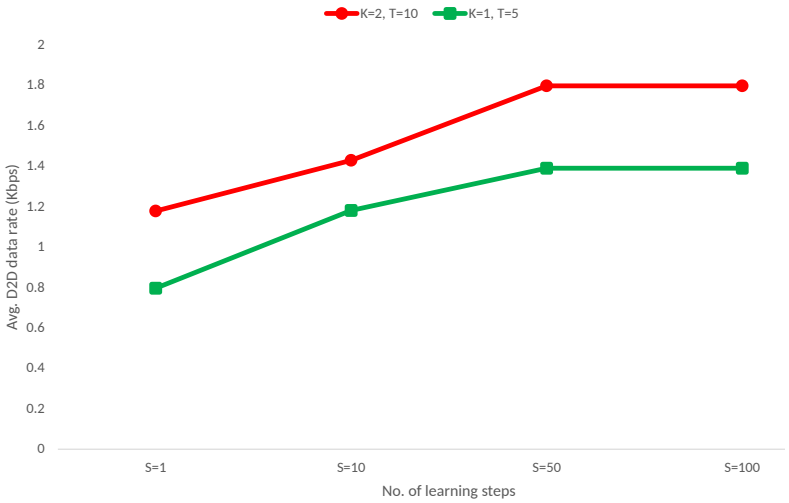


Figure 5.6: Effect of Q-learning on D2D transmission data rate.

the number of learning steps doubles from 50 to 100, the VLC delay decreases by around 14% for the case with 2 VLC transmitters and 10 mobile users, which is also the result of better learning.

Fig. 5.6 shows the effect of Q-learning on the average D2D communication data rate for two cases: 1 VLC transmitter and 5 mobile users, and 2 VLC transmitters and 10 mobile users. The figure shows the average transmission capacity for the D2D

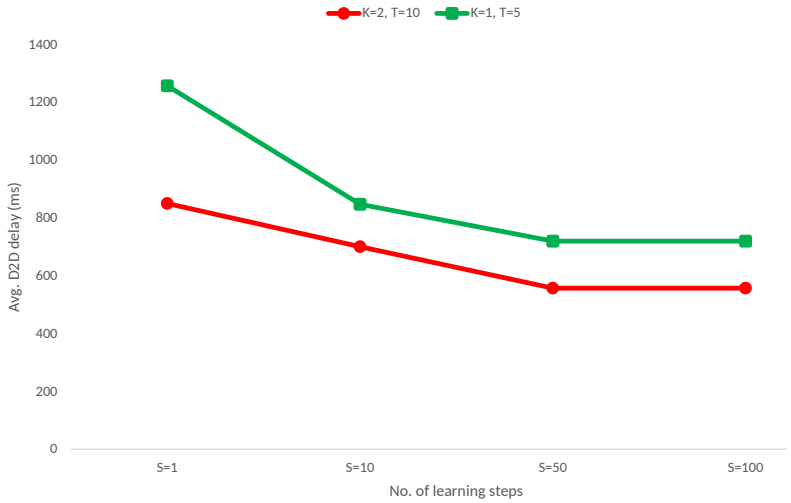


Figure 5.7: Effect of Q-learning on D2D transmission delay.

communication between the MUaRs. We can see that as the number of learning steps increases from 1 to 100, the average D2D communication (second and third hops) data rate between the MUaRs increases. There is a significant increase in data rate (48%), when the number of learning steps increases from 1 to 10 for the case with 1 VLC transmitter and 5 mobile users. This shows the impact of RL on improving the D2D communication data rate in a VLC-D2D heterogeneous network.

Fig. 5.7 shows the effect of Q-learning on the average delay in D2D communication for two cases: 1 VLC transmitter and 5 mobile users, and 2 VLC transmitters and 10 mobile users. The figure shows the average delay for the D2D communication between the MUaRs. We can see that the average D2D communication (second and third hops) delay between the MUaRs decreases, as the number of learning steps increases from 1 to 100. The average D2D delay is decreased by around 33% in the case with 1 VLC transmitter and 5 mobile users, for an increase in the number of learning steps from 1 to 10. This shows the impact of RL in minimizing the D2D communication delay in a VLC-D2D heterogeneous network.

The effects of the discount factor for Q-learning, η, on the average D2D commu-

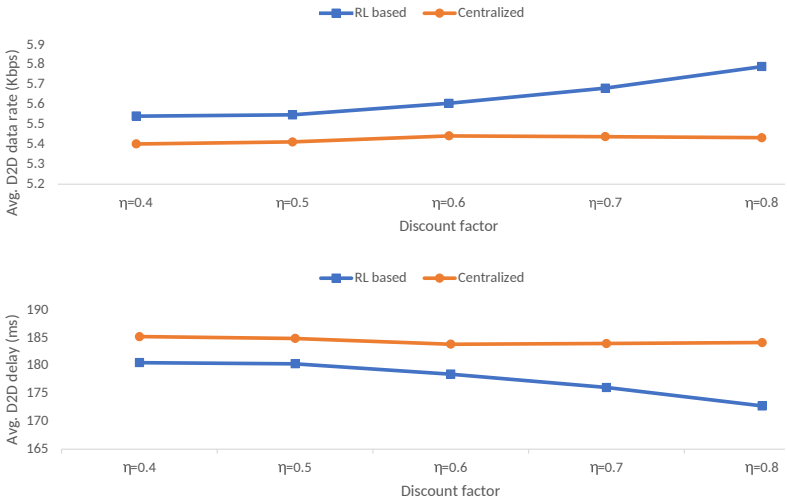


Figure 5.8: Effect of discount factor on D2D parameters.

nication data rate and delay are shown in Fig. 5.8. The proposed method is compared with a centralized method where the data transmission route is determined by the VLCSP based only on the transmission path delays. The average D2D data rate and delay does not change much for the centralized allocation. It can be observed that the average capacity of D2D communication increases, and the average delay decreases as η increases for the proposed method. This indicates better performance as the emphasis on future rewards grows, which in turn underlines the benefits of learning from experience.

The effects of Q-learning on the overall algorithm run time are shown in Fig. 5.9, for two cases: 1 VLC transmitter and 5 mobile users, and 2 VLC transmitters and 10 mobile users. It is obvious that as the number of learning steps increases, or as the number of entities increases, the algorithm run time will increase. Specifically, we can observe that the run time increases proportionally with the number of learning steps as well as the number of entities. Hence, the key observation here is that the time complexity of the proposed RL based multi-hop route determination algorithm using ADMM for EPEC, is linear.

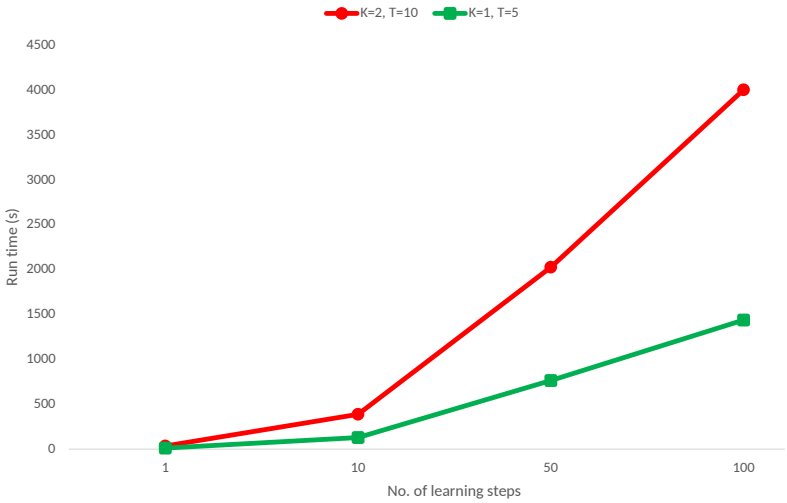


Figure 5.9: Effect of Q-learning on overall algorithm run time.

5.6.2 Discussion

Here, as we consider an indoor communication scenario, the mobility speed of the cellular users is significantly lesser compared to the learning speed of the RL algorithm. Thousands of data packets are transmitted in each second, and this enables the RL algorithm to perform repeated interactions and arrive at an optimal L-hop route at a rate much faster than the changes in locations of the mobile users. The RL algorithm can be run to capture any changes in the network topology.

Also, the run times in Fig. 9 are obtained using a small-scale Intel(R) Core(TM) $i7-7500U$ CPU with a 16 GB RAM. The algorithm's run times can be improved tremendously by using a better processor, which would be the case in a practical VLC-D2D heterogeneous network. Accordingly, the algorithm can be run every few minutes or seconds in order for the RL part to accommodate the changes in users' locations, and optimal multi-hop routes can be determined dynamically. Also, in the considered indoor scenario, L can be as small as 2, with just an MUiC relaying the data from the VLC transmitter to the end MUiD. In this case, the RL algorithm for route determination converges even faster, dismissing the effects of user mobility.

5.7 Conclusion

In this chapter, we propose an RL based data transmission route determination method for an indoor VLC-D2D heterogeneous network. We utilize the model-free Q-learning technique to determine L-hop data transmission routes from the VLC transmitters to the end MUiDs. We determine the rewards for the Q-learning method dynamically during each learning step, by formulating the interactions between the MUaRs as an EPEC optimization problem, and then, solving it using ADMM. We evaluate the performance of the proposed algorithm through MATLAB simulations. It can be observed from the simulations that RL improves the parameters of a typical VLC-D2D heterogeneous indoor downlink scenario. As the number of learning steps increases, the agent learns more through its interactions with the environment, and hence, the data transmission rate is improved, and the delay is minimized. We can observe from the simulation results that when the number of learning steps increases from 1 to 100, the VLC and D2D data rates are increased and the delays are reduced. It can also be noticed that the D2D data rate is improved and the delay is minimized as the discount factor for Q-learning increases, which highlights the importance of future rewards. The simulation results also demonstrate the time complexity of the proposed algorithm to be linear.

Chapter 6

Defending Primary User Emulation Attacks in Cognitive Radio Networks by Generative Adversarial Networks

6.1 Introduction

CR is a state-of-the-art technology that will facilitate better and efficient spectrum utilization in the future wireless communication paradigm [138]. The radio spectrum required for wireless communications is a scarce resource, and hence, the Dynamic Spectrum Access (DSA) technique has been proposed to put the wireless spectrum to efficient use [139, 140]. It serves as the inverse of the static spectrum management practiced currently, and enhances the spectrum efficiency by introducing more flexibility [140]. DSA is facilitated by means of a CR, which is a context-aware radio that can adapt according to the communication environment. Accordingly, the dynamic management of spectrum is enabled through a CRN, an intelligent network in which one can dynamically configure transmission parameters like communication protocol, operating frequency band and modulation scheme [141].

Licensed and unlicensed users in a CRN are called Primary Users (PUs) and Secondary Users (SUs), respectively. DSA works firstly by the *spectrum sensing* performed by SUs [142], which is the process of identifying unoccupied parts of the licensed spectrum called *white spaces* [143]. This is followed by the opportunistic utilization of these white spaces for communication by SUs, without interfering with the operations of PUs [144]. Thus, the radio spectrum can be shared by PUs and SUs without interference, and the spectral efficiency can be enhanced tremendously.

Nonetheless, spectrum sensing is faced with a few technical challenges, of which a crucial one is distinguishing PU signals from SU signals. This puts the security

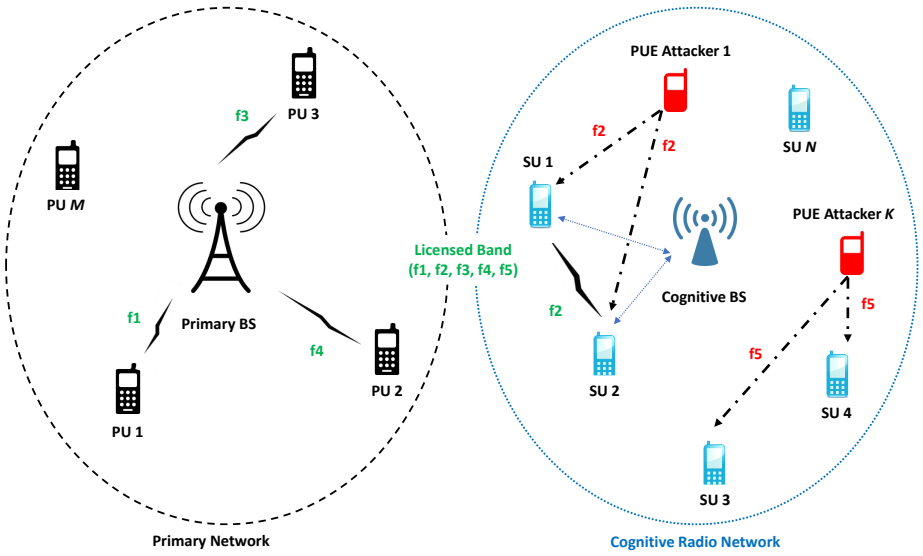


Figure 6.1: Illustration of PUE attacks in a CRN.

of CRNs at risk, as SUs lack information about the network spectrum usage by PUs [138]. This leads to one of the major security issues specific to CRNs, known as Primary User Emulation (PUE) attacks [145]. In a PUE attack, a PUE attacker emulates the transmission of PUs, and deceives legitimate SUs into clearing the spectrum for PUs, as is the inherent courtesy in a CRN. In this way, PUE attacks result in inefficient spectrum utilization of CRNs, thus defying their very purpose.

A great deal of research has been carried out on PUE attacks, given their impact on CRN security, and many different approaches have been proposed for PUE attack detection [146,147]. Considering the tremendous applications of machine learning techniques in wireless communication networks including CRNs [7,148], these techniques are gaining popularity in PUE attack classification as well. Specifically, deep learning based approaches have also been attaining prominence recently [149, 150].

Deep learning is a part of machine learning where computers learn from experience, through models which are many layers deep [152–154]. Additionally, Artificial Neural Networks (ANNs) are models that mimic the biological neural networks in

the human brain and enable complex problem solving through deep learning [151]. Recently, Generative Adversarial Networks (GANs) have emerged as an efficient and effective method to train classifiers in a semi-supervised manner [155]. GANs consist of two ANNs, a generative model and a discriminative model, pitted against each other in a minimax zero-sum game [5, 92].

GAN is considered to be a superior deep learning tool when available labeled data samples are limited [155]. The data augmentation capabilities of GAN can be of assistance in our PUE attack classification scenario, as the available data samples are limited here. However, in order to overcome the training instabilities associated with GAN, we consider a variant of the basic GAN called Wasserstein GAN (WGAN). Accordingly, in this chapter, we propose a WGAN based framework for the detection of PUE attacks in a typical CRN. The major contributions of this chapter are summarized as follows:

- Firstly, energy detection is performed to locate the frequency of the user. Then, the sensed signal is observed for a time period, and its cyclostationary features are computed, which are its statistical properties varying periodically with time. These features are later used to distinguish between PU and PUE attacker signals.

- The computed cyclostationary features are input as data value matrices into the GAN framework, in which both the generator and discriminator networks are Convolutional Neural Networks (CNNs) [156].

- Finally, the GAN framework is used to classify the sensed signal into PU signal or PUE attacker signal based on the cyclostationary feature value inputs.

- The proposed GAN framework is trained and tested using datasets generated in MATLAB. The performance of the model in successfully detecting PUE attacks

is demonstrated through simulations.

The rest of this chapter is organized as follows. We discuss previous work related to this chapter in Section 6.2. The system model and the problem formulation are discussed in Section 6.3. The application of the proposed framework is discussed in Section 6.4, where initially, we discuss the computation of cyclostationary features in Section 6.4.1, and then, we discuss the concept of GANs in Section 6.4.2. Next, we discuss the concept of WGAN in Section 6.4.3. Finally, the proposed GAN based detection of PUE attacks in CRNs is discussed in detail in Section 6.4.4. The performance of our framework is demonstrated through simulation results in Section 6.5. The chapter is concluded in Section 6.6.

6.2 Related Work

Approaches based on cyclostationary features have been immensely employed in CRNs for spectrum sensing and modulation classification. A signal classification algorithm using cyclostationarity is proposed in [157], which provides necessary information about the radio environment to CRs. Reference [158] proposes an approach for the detection and analysis of cyclostationary signatures, and presents an implementation of these signatures on a real cognitive radio test platform. The importance of cyclostationary feature detection in spectrum sensing due to its ability to differentiate modulated signals is highlighted in [159].

Following its successful applications in different areas of wireless communications, researchers have been exploring the possibilities of deep learning in CRNs. References [160], [161], and [162] demonstrate a few applications of ANNs in CRNs. A deep learning based fingerprinting model for detecting cognitive devices is proposed in [163]. Reference [164] proposes a deep learning based signal modulation classification method, and [165] employs a CNN to detect CR waveforms. Reference [166]

proposes adversarial machine learning to design intelligent jamming attacks and defense strategies against the attacks in CRNs. A similar approach with a GAN assisted jammer is demonstrated in [167]. Reference [168] adopts a GAN based spectrum sensing technique for CRNs, showing superior results due to training data augmentation. Automated modulation classification using data augmentation by GANs is discussed in [169].

PUE attacks and defense mechanisms in CRNs have been extensively investigated in recent research [170–173]. Reference [170] proposes a mean field game approach which helps SUs take strategic defense decisions when multiple attackers are present, and claims a detection accuracy of 89% with a false alarm probability of 0.09. Detecting PUE attacks based on a transmitter verification scheme based on the Time-Difference-of-Arrival (TDOA) localization method is proposed in [171]. An algorithm to differentiate PU signals from PUE attacker signals based on energy-efficient localization and variance of channel parameter is discussed in [172]. Reference [173] proposes an adaptive method for PUE attack detection, which is enabled through a Received Signal Strength (RSS) based hypothesis detection system.

Deep learning based methods have been employed for waveform detection, primary user sensing, device fingerprinting, and for PUE attack detection in CRNs. The application of deep learning to detect PUE attacks is realized in [174], which discusses two broad categories of PUE attack detection: fingerprint based detection and activity pattern based detection. A Recurrent Neural Network (RNN) based PUE attack detection method is proposed here, which falls under the activity pattern based detection category. This method is based on the observation of signal activity patterns in a CRN through intermittent spectrum sensing. Further, the gradient vanishing problem of RNNs is resolved through a multi-layer Long Short-Term Memory (LSTM) network, which achieves an average loss value of 0.0003.

Reference [144] also proposes a deep learning based PUE attack detection framework. Here, energy detection is used to locate the users in a frequency band, and then, a cyclostationarity based method is used to extract the signal features. Finally, a Multilayer Perceptron (MLP) ANN is used for the classification of PUE attacks, which demonstrates a detection accuracy of around 98% in hardware implementations.

Even though both of the aforementioned works employ deep learning networks for PUE attack detection, the following aspects set this work apart.

- This chapter proposes a PUE attack classification model based on deep learning, similar to [174] and [144]. However, the core of the detection framework is a GAN consisting of two competing networks, which recently became popular as a powerful model for scenarios with limited labeled data. Also, CNNs are used as the generator and discriminator in the proposed GAN model.

- As opposed to the activity pattern based detection method proposed in [174], this is a fingerprint based PUE attack detection method, as it is based on cyclostationary features, which are dependent on the transmitter's intrinsic features. Hence, we need not model the signal patterns of the normal traffic in a CRN for attack detection.

- Even though cyclostationarity based models are vastly applied in CRNs for modulation classification and spectrum sensing, research on applying such models in PUE attack classification is limited. This chapter utilizes the cyclostationary features of the sensed signals as input in the proposed GAN framework for attack detection.

According to the authors' knowledge, a GAN based framework for the detection of PUE attacks in CRNs has not been proposed in the literature. Here, we

convert the cyclostationary features that can help discriminate between PU and PUE attacker signals, into suitable inputs for the GAN framework. Then, by exploiting the minimax game model of GAN, we train the network using labeled datasets, to be able to successfully detect PUE attacks in CRNs. The simulation results show the convergence of the GAN model, and thus, emphasize the effectiveness of our proposed approach.

6.3 System Model and Problem Formulation

In this section, we firstly introduce the system model in Section 6.3.1, and then formulate the problem in Section 6.3.2.

6.3.1 System Model

As discussed in Section 6.1, the SUs perform spectrum sensing initially, which is followed by the opportunistic utilization of the spectral white spaces. A preferable spectrum sensing method is the *cooperative spectrum sensing*, which mitigates the hidden PU problem, in which a SU is unable to sense an active PU due to signal fading or due to the PU signal being out of range [175]. The SUs share their sensing results to gain better knowledge of the spectrum environment in cooperative sensing. In such scenarios, a *fusion center* acts as a centralized controller that collects information from all the SUs sensing a particular frequency band, and arrives at a final decision [176].

Here, we consider a typical CRN scenario consisting of a Primary Base Station (BS), a cognitive BS, which behaves like a centralized controller similar to a fusion center, and multiple PUs and SUs, as shown in Fig. 6.1. As mentioned before, the SUs will be continuously sensing the spectrum. During this process, when a spectral white space is detected, the SU utilizes that part of the spectrum for its transmission. However, if/when the SU detects the presence of a PU in that part of the spectrum,

the SU leaves the spectrum band or moves to another spectral white space for its communication. In this way, the spectrum is accessed dynamically, and thus, better spectral efficiency is achieved.

In order to understand how PUE attacks are carried out, let us consider a scenario as shown in Fig. 6.1, consisting of M PUs, N SUs, and K PUE attackers. The PUs and the primary BS communicate with each other using a licensed frequency band, which has the channels $f1$, $f2$, $f3$, $f4$, and $f5$. The cognitive BS performs spectrum sensing continuously, and allocates the unused licensed frequency band channels to the SUs for their communication with each other. From Fig. 6.1, we can see that the channels, $f1$, $f3$, and $f4$ are being used for the communication between the primary BS and PUs 1, 3, and 2, respectively. The cognitive BS senses the available licensed frequency band channels, $f2$ and $f5$, and is able to allocate these channels to the SUs for communication. Taking this into consideration, we highlight two cases of PUE attacks below.

- *Selfish PUE attack:* Channel $f2$ which is unoccupied by the PUs is sensed by the cognitive BS, and is allocated to SUs 1 and 2 for communication. PUE attacker 1 emulates the primary signals in channel $f2$, and attacks SUs 1 and 2 in service. Thus, PUE attacker 1 captures channel $f2$, resulting in a service interruption between SUs 1 and 2.

- *Malicious PUE attack:* Channel $f5$ is unoccupied by the PUs, and should be sensed by the cognitive BS, normally. However, PUE attacker K emulates the primary signals in channel $f5$, and if PUE attacker K cannot be identified correctly, the cognitive BS can sense it as a PU and mistake channel $f5$ to be occupied. Thus, PUE attacker K captures channel $f5$ which is unoccupied by the PUs, and wastes the communication opportunity of SUs 3 and 4.

6.3.2 Problem Formulation

As already discussed, the PUE attackers can jeopardize the spectrum opportunities of the SUs in a CRN by emulating the signals of the PUs. Such a competition between the SUs and the PUE attackers can be considered as a noncooperative game [5,92]. This is possible because the SUs and the PUE attackers have conflicting interests in terms of wireless spectrum access. Additionally, a spectrum access gain by the PUE attackers implies the loss of spectrum access to the SUs, and a gain of spectrum access by the SUs implies the loss of a PUE attack opportunity. Therefore, to be specific, such a noncooperative game is a two-player zero-sum game. Here, one player acts as a maximizer who tries to maximize its gain (loss to the other player), and the other player acts as a minimizer who tries to minimize its loss (gain to the other player), and it is impossible for both of them to win [5, 92].

In a noncooperative game, a solution concept is the *Nash equilibrium*, in which no player can gain anything by deviating from its current strategy [177]. For a zero-sum game, the *Nash equilibrium* solution is the *minimax* solution, which is the saddle point in mixed (selecting each strategy with a certain probability) strategies [178].

In our scenario, let us consider the maximum payoff when either of the SU or the PUE attacker wins spectrum access as a standard *value*, U, which can be considered as the transmission data rate. Hence, $\forall n \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, K\}$, the *minimax* solution for our scenario can be expressed as

$$\max_{S_{PUE}(k)} U_{SU}(n) = U, \text{ and} \tag{6.1}$$

$$\max_{S_{SU}(n)} U_{PUE}(k) = -U. \tag{6.2}$$

Here, (6.1) says that the maximum possible payoff, $U_{SU}(n)$, for SU n (player 1), given PUE attacker k (player 2)'s strategy, $S_{PUE}(k)$, is U. Similarly, (6.2) says that the

maximum possible payoff, $U_{PUE}(k)$, for PUE attacker k (player 1), given SU n (player 2)'s strategy, $S_{SU}(n)$, is $-U$.

In order to maximize U and let SU n always win against PUE attacker k, $\forall n \in \{1, 2, \ldots, N\}$, and $\forall k \in \{1, 2, \ldots, K\}$, we need to successfully distinguish PUE attacker signals from actual PU signals. However, we only have a limited set of signal characteristics (for example, modulation scheme) to learn from, to be able to achieve this goal efficiently. Taking this into account, and also the zero-sum game nature of the competition between the SUs and the PUE attackers, a potential candidate for a learning algorithm is GAN.

As discussed before, GANs consist of two ANNs, which are pitted against each other in a minimax zero-sum game. The minimax solution obtained for the GAN classification framework, as discussed in the next section, will in turn result in achieving the maximization of payoffs for the SUs, as shown in (6.1). Considering the great capabilities of GAN for semi-supervised learning and classification in a setting with minimal labeled data, in this chapter, we employ a GAN based framework in the next section, for the detection of PUE attacks in our considered CRN scenario.

6.4 Algorithm Analysis

In this section, we discuss our proposed GAN based PUE attack detection method in detail. Firstly, in Section 6.4.1, we discuss the computation of cyclostationary features of all the user signals, which will serve as the input to the GAN architecture. Then, we explain the basic idea of a GAN network in Section 6.4.2, followed by which we explain the working of our proposed algorithm in Section 6.4.4. Finally, we discuss the execution aspects of the algorithm in Section 6.4.5.

Before moving to the cyclostationary feature calculation, we list the assumptions in our proposed framework, similar to [144], as below.

- All the PUs, SUs and PUE attackers operate in the same frequency band.
- Only one user transmits during one time period.
- The transmitted signal power is much higher than the noise power in the channel.
- The modulation schemes of the PUs are already known, and has different signal features compared to the PUE attackers.

6.4.1 Computation of Cyclostationary Features

Simply put, if a signal has statistical properties varying periodically with time, such a signal is called *cyclostationary* [179]. Consider a cyclostationary signal $y(t)$. Due to the mentioned periodic property, we can represent this signal using its Spectral Correlation Function (SCF) as [144]

$$S_Y^{\alpha}(f) = \lim_{T\to\infty} \lim_{\Delta t\to\infty} \frac{1}{\Delta t} \int_{-\Delta t/2}^{\Delta t/2} \frac{1}{T} Y_T(t, f + \frac{\alpha}{2}) Y_T^*(t, f - \frac{\alpha}{2}) dt, \tag{6.3}$$

where $\{\alpha\}$ is the set of Fourier components. Here, $Y_T(t, f)$ is the time varying Fourier transform of $y(t)$, defined as

$$Y_T(t, f) = \int_{t-T/2}^{t+T/2} y(u) e^{j2\pi f u} du. \tag{6.4}$$

The spectral components of white noise are uncorrelated, and hence, for all $\alpha \neq 0$, the additive white noise does not affect the SCF, or in other words, SCF is robust to additive noise. Now, we define Spectral Coherence (SC), which is a

Algorithm 6.1 Computation of Cyclostationary Features

1: Split the sensed modulated signal into N_f frames. Each frame consists of $T = \frac{\Delta t}{N_f}$ samples, given the signal has Δt samples overall.
2: Compute the Fourier transform coefficients for each frame.
3: Shift the Fourier transform of each frame by $\alpha/2$ and $-\alpha/2$ and multiply them.
4: Compute the average value over N_f frames.
5: Perform frequency smoothening using a moving average filter.
6: Repeat steps 2-5 for each value of α to obtain the SCF as in (6.3).
7: Normalize the SCF to obtain the SC as in (6.5).
8: Obtain the CDP from the peak values of the SC as in (6.6).

normalized form of the SCF as [144]

$$C_Y^{\alpha}(f) = \frac{S_Y^{\alpha}(f)}{[S_Y^0(f + \frac{\alpha}{2})S_Y^0(f - \frac{\alpha}{2})]^{1/2}}. \tag{6.5}$$

The SC shows the spectral presence of the signal at different cyclic frequencies. The SC has values ranging from 0 to 1, and has non-zero components at different cyclic frequencies for different modulation schemes, which acts as the signature for identifying different modulation schemes.

In this chapter, we utilize only the maximum value of SC for each value of α, called the Cyclic Domain Profile (CDP) or α-profile, which can be expressed as [180]

$$P(\alpha) = \max_f |C_Y^{\alpha}(f)|, \tag{6.6}$$

where $|C_Y^{\alpha}(f)|$ is the absolute value of the SC of the received signal. The CDPs of the signals sensed by the SUs can be examined to map the spectral components to a particular modulation scheme, the features (carrier frequency, phase shift etc.) for which are already known for the PUs but not known for the PUE attackers. For this purpose, the CDP of the received signal is determined [180], which is explained in detail in Algorithm 6.1. The resulting CDP values are saved as matrices, and the CDP value matrices serve as input to the GAN framework for learning and classification.

6.4.2 Generative Adversarial Networks

GANs consist of a generative network and a discriminative network functioning as adversaries to each other. The generative model can be thought of as a counterfeiter trying to produce fake currencies, and the discriminative model is analogous to the police trying to discover the fake currencies [155]. The generative model can be called a *generator*, G, and the discriminative model can be called a *discriminator*, D. Here, G tries to generate samples that are difficult for D to discriminate from the actual data, while D tries to discriminate between samples from the actual data and those from G. Such a competition drives both networks into improving their methods, until the real samples become indistinguishable from the fake samples. This competition between the G and D networks can be modeled as a minimax two-player game [155].

Let $\mathbf{y}$ be a set of data samples, $p_{data}(\mathbf{y})$ be its distribution, and $p_{\mathbf{z}}(\mathbf{z})$ be a prior on input noise variables. G is a differentiable function representing the generator, and the generator's distribution, p_g, over the data $\mathbf{y}$ is given by the mapping $G(\mathbf{z})$. Let $D(\mathbf{y})$ represent the discriminator, and it gives the probability that $\mathbf{y}$ is from the actual data and not the generator.

The discriminator is trained to correctly label the data samples and the generator samples, while the generator is trained to keep the discriminator from discriminating the samples correctly. Consequently, we can represent the objective function for the minimax two-player game between G and D, with a *value* function of $V(G, D)$ as

$$\min_G \max_D V(G, D) = \mathbf{E}_{\mathbf{y}\sim p_{data}(\mathbf{y})}[\log D(\mathbf{y})] + \mathbf{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{6.7}$$

Here, we train D to maximize $\log D(\mathbf{y})$, and G to minimize $\log(1 - D(G(\mathbf{z})))$. However, in practice, G is trained to maximize $\log D(G(\mathbf{z}))$, rather than minimizing $\log(1 - D(G(\mathbf{z})))$, since $\log(1 - D(G(\mathbf{z})))$ saturates initially when D can discriminate

clearly but G is poor [155]. Thus, GANs are trained based on the competition between the generator and discriminator networks, of which the former tries to generate samples similar to the ones from the actual data, and the latter tries to distinguish if the samples belong to the data distribution (actual data) or the model distribution (generated by G).

Even though GANs are promising generative models, they suffer from a few issues: imbalance between both the networks causing overfitting, high sensitivity to hyperparameter selections, and mode collapse of the generator resulting in lesser diversity of generated samples. In addition, the parameters of the model keep oscillating, and this keeps GAN from always converging. Therefore, GANs are very hard to train, and a lot of research is being conducted on consistently training them in a stable manner.

6.4.3 Wasserstein GAN

The aforementioned issues in the training of GANs have led to the development of different variants of the basic GAN. Wasserstein-GAN (WGAN) is a form of GAN which overcomes some of the training hurdles associated with the basic GAN model [181, 182]. Reference [181] attributes the training difficulty of GANs to the aspect that the divergences which the GAN framework minimize are not continuous with respect to the parameters of the generator network. Therefore, the usage of the *Earth Mover (EM)* distance is proposed, which is also known as the *Wasserstein-1* distance $W(x, x')$, and is described as the minimum cost to transport mass for transforming the distribution x to the distribution x'. As opposed to the divergences in the basic GAN, the Wasserstein-1 distance is continuous everywhere [182].

The value function in (6.7) can be modified for the WGAN as

$$\min_{G} \max_{D \in \mathcal{D}} V(G, D) = \mathbf{E}_{\mathbf{y} \sim p_{data}(\mathbf{y})}[D(\mathbf{y})] - \mathbf{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))], \tag{6.8}$$

where $\mathcal{D}$ is the set of *1-Lipschitz* functions. Minimizing the value function as shown in (6.8) with respect to the parameters of G under an optimal D, minimizes the Wasserstein-1 distance [182], which can be denoted for our scenario as $W(p_{data}(\mathbf{y}), p_{\mathbf{z}}(\mathbf{z}))$. Also, as opposed to the basic GAN, the value function of the WGAN correlates with the quality of the samples.

6.4.4 WGAN based PUE Attack Detection in CRNs

In our scenario, the CDP matrices obtained using Algorithm 6.1 serve as input to the GAN framework. Therefore, (6.8) can be modified for our scenario as

$$\min_{G} \max_{D \in \mathcal{D}} V(G, D) = \mathbf{E}_{P(\alpha) \sim p_{data}(P(\alpha))}[D(P(\alpha))] - \mathbf{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))]. \tag{6.9}$$

The next major question is what kind of network to choose for the generator and the discriminator. We need an easy to train network that can work with the available limited data, and most importantly, work with the CDP value matrices. CNNs are an ideal choice, considering they have been used successfully for classifying millions of images into thousands of classes [184, 185].

Subsequently, in our GAN based network, we utilize CNNs as the generator and the discriminator, the architecture for which is shown in detail in Fig. 6.2 (ReLU stands for Rectified Linear Unit). The optimizer used in the training of the network is *RMSProp*, which is an optimization method that involves dividing the learning rate for a particular weight by a running average of the values of the recent gradients for that weight [183].

Now, we explore the training of the WGAN network using RMSProp optimization [181], as shown in Algorithm 6.2. Keeping all of this in mind, we elucidate our proposed PUE attack detection framework based on WGAN in Algorithm 6.3.

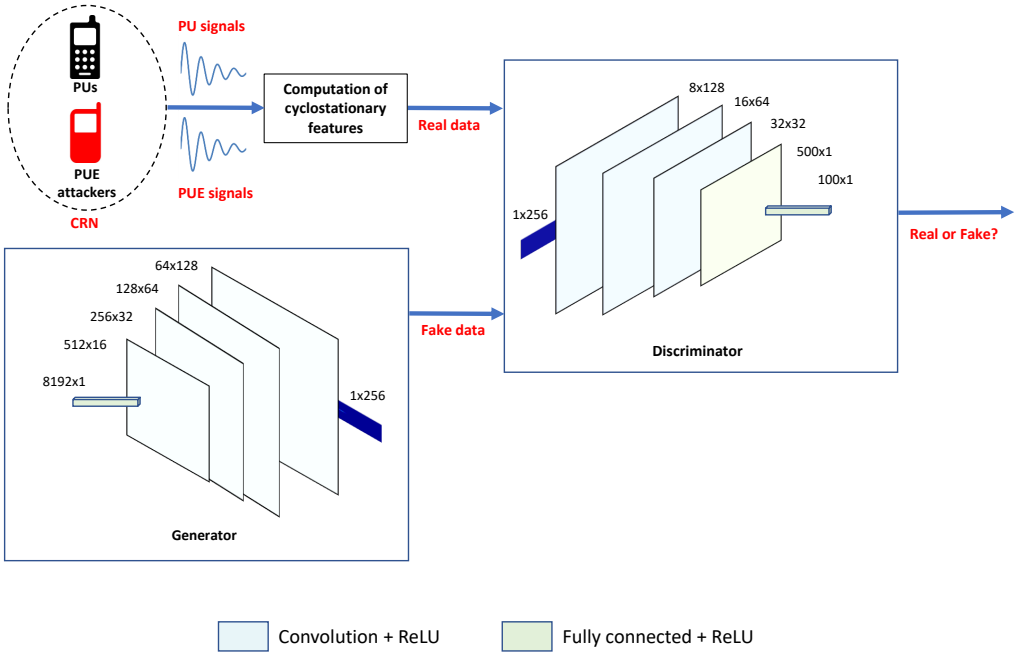


Figure 6.2: Proposed GAN architecture.

6.4.5 Discussion

The GAN based network for PUE attack detection can be deployed at the cognitive BS/fusion center. This is because it acts as a centralized controller to combine the sensing results from all the SUs. Another reason is that the SUs are low computation power devices, and will not be able to handle such complex algorithms. Most importantly, the generator and discriminator networks will be owned by a single entity, and one network can be used to enhance the performance of the other network and vice versa, thus resulting in a high-performance GAN classifier.

Initially, the GAN network at the cognitive BS/fusion center needs to be trained using some previous signal activity data from the CRN including those of PUs and PUE attackers. However, this training can be done offline, and it makes the model ready for PUE attack detection in real-time. Once the model is deployed in a real

Algorithm 6.2 WGAN Training Algorithm

Input: The learning rate β, the clipping parameter c, the batch size b, the number of epochs n_e, the number of iterations of the discriminator per generator iteration n_d.
for epoch number n_e **do**
  for $t = 0, \ldots, n_d$ **do**
    From noise prior $p_g(\mathbf{z})$, extract a minibatch of b noise samples, $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(b)}\}$.
    From the data distribution $p_{data}(\mathbf{y})$, extract a minibatch of b data samples, $\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(b)}\}$.
    $D \leftarrow \nabla_w \frac{1}{b} \sum_{l=1}^{b} [D(P(\alpha)^{(l)}) - D(G(\mathbf{z}^{(l)}))]$
    $w \leftarrow w + \beta.\text{RMSProp}(w, D)$
    $w \leftarrow clip(w, -c, c)$
  end for
  From noise prior $p_g(\mathbf{z})$, extract a minibatch of b noise samples, $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(b)}\}$.
  $G \leftarrow -\nabla_\theta \frac{1}{b} \sum_{l=1}^{b} D(G(\mathbf{z}^{(l)}))$
  $\theta \leftarrow \theta - \beta.\text{RMSProp}(\theta, G)$
end for

network, firstly, the SUs can sense the spectrum and perform the computation of cyclostationary features as shown in Algorithm 6.1. Then, the obtained CDP values, which may include those of PUs and PUE attackers, can be sent to the cognitive BS for classification.

The cognitive BS can make scheduling decisions for SUs based on the classification results from the GAN network. If a PUE attacker is detected in a frequency band, the cognitive BS can communicate it to the SUs performing spectrum sensing in that frequency band, and also take necessary actions to ensure proper spectrum usage.

6.5 Simulation Results

We implement Algorithm 6.1 to generate the CDP value matrices using MATLAB. The system used for this process has an Intel Core i7−7500U CPU with a 16 GB RAM. The simulation setup consists of a transmitter, an Additive White Gaussian

Algorithm 6.3 WGAN based PUE Attack Detection in CRNs

1: Perform energy detection to locate the frequency of the user.
2: **if** no user detected **then**
3: Go to step 1.
4: **else**
5: Observe the sensed signal for a time T.
6: Compute the cyclostationary features of the sensed signal using Algorithm 6.1.
7: Input the obtained CDP values for classification into the GAN framework, which has been trained using Algorithm 6.2.
8: **if** CDP values are similar to those of the PUs **then**
9: Classify as PU.
10: **else**
11: Classify as PUE attacker.
12: **end if**
13: **end if**

Noise Channel (AWGN) channel, and a receiver. The transmitter includes a random number generator, which generates random data bits, a baseband modulator, which then modulates it using one of BPSK, QPSK or FSK modulation schemes, and a raised-cosine filter, which minimizes intersymbol interference through pulse-shaping. Both the PUs and the PUE attackers are assumed to utilize the BPSK, QPSK or FSK techniques for baseband modulation. However, the phase offset is assumed to be 0 for the PUs in BPSK and $\pi/4$ for the PUs in QPSK, and the modulation order is assumed to be 8 for the PUs in FSK. Even though the baseband modulation schemes are assumed to be the same for the PUE attackers, the above mentioned parameters are different, and act as a set of fingerprints for identifying the attackers.

The number of data symbols is taken as 1000. The sampling frequency is considered to be 8× the symbol rate. The value of $EbNo$ (ratio of bit energy to noise power spectral density) is considered to be 10 dB, which reflects on the Signal-to-Noise Ratio (SNR) due to the AWGN channel. The value of N_f is taken as 10. The CDP values obtained through the simulations have been saved as 1×256 matrices to be fed as inputs to the GAN.

The GAN training as seen in Algorithm 6.2, and the PUE attack classification as seen in Algorithm 6.3 have been implemented in Python with TensorFlow as backend.

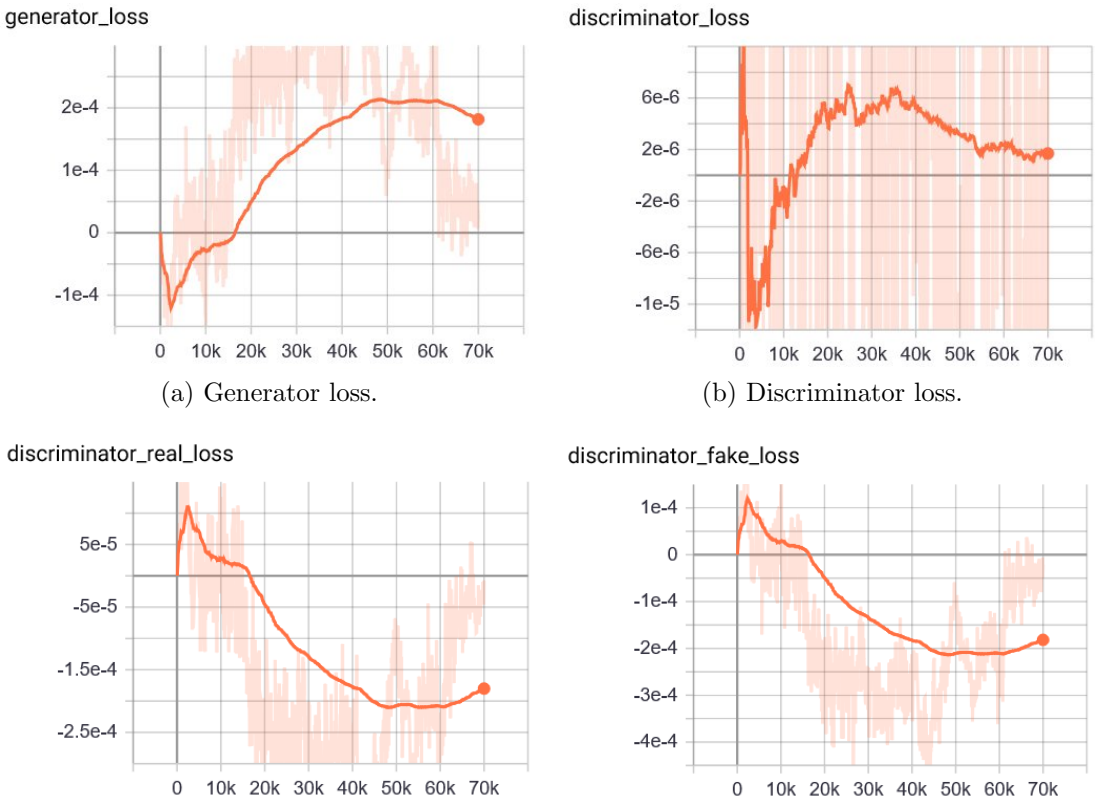
The system used for training and testing the GAN network has an Intel Core i7 − 8750H CPU with a 16 GB RAM, and a GeForce GTX 1070 GPU.

We used 22000 data value matrices for training and 2000 data value matrices for testing the network. We ran $n_e = 70$ epochs with a batch size of $b = 22$ data value matrices, which can be considered as 1000 iterations in each epoch to input all the 22000 matrices. Hence, the total number of steps = number of epochs × number of iterations = $70 \times 1000 = 70000$. The other hyperparameters of the WGAN model are set to the following values: $\beta = 0.00005$, $c = 0.01$, and $n_d = 5$, similar to [181].

Generally, loss functions help guide the training of a deep learning network, and represent the cost of inaccuracy in classification predictions. Therefore, we consider the variations in loss function values of the generator and the discriminator as the parameters for performance evaluation.

Fig. 6.3 shows the loss function values for the generator and the discriminator for the training dataset containing 22000 samples. Specifically, Fig. 6.3a shows the generator loss and Fig. 6.3b shows the discriminator loss. Fig. 6.3c and Fig. 6.3d show the discriminator loss values when it is fed with samples from the actual data, and with samples from the generator generated data, respectively. It can be observed from the figures that the average loss value is converging to 0 within the total number of training steps, even though there are slight oscillations in the beginning. The convergence of the WGAN model demonstrates that the training has been effective.

Fig. 6.4 shows the loss function values for the generator and the discriminator for the testing dataset containing 2000 samples. The generator loss is shown in Fig. 6.4a, and the discriminator loss is shown in Fig. 6.4b. Similar to the training losses shown before, Fig. 6.4c and Fig. 6.4d illustrate the discriminator loss values for actual data input, and for model data input, respectively. The average testing loss values can be seen to be converging within the total number of testing steps, even



(a) Generator loss.

(b) Discriminator loss.

(c) Discriminator loss with data input.

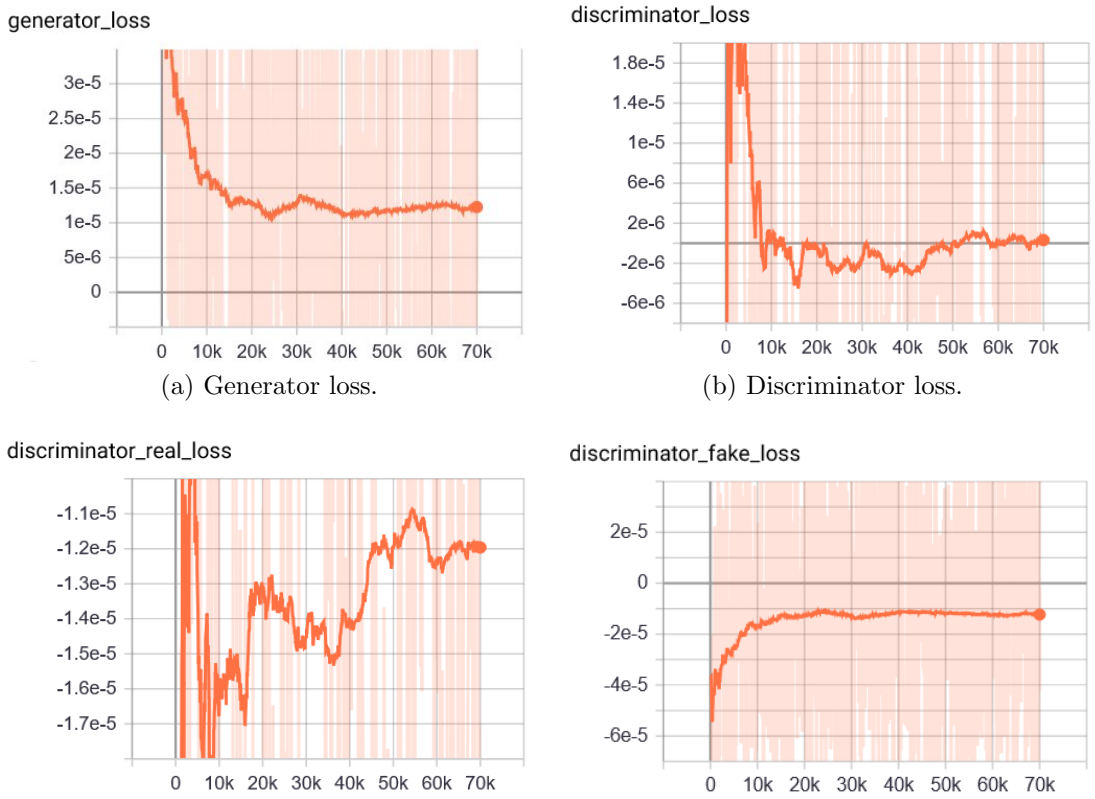
(d) Discriminator loss with model input.

Figure 6.3: Generator and discriminator losses vs. number of steps for the training dataset.

though Fig. 6.4b and Fig. 6.4c show some oscillations. This demonstrates that the proposed WGAN model for PUE attack classification is effective.

6.6 Conclusion

In this chapter, we propose a novel WGAN based framework for PUE attack detection in CRNs. We obtain the CDP values of the signals sensed by the SUs, and utilize these values as inputs for classification by the GAN. The generator and discriminator losses of the proposed model are observed to be converging, which denotes that our GAN framework can be employed for efficient PUE attack classification in CRNs.



(a) Generator loss.

(b) Discriminator loss.

(c) Discriminator loss with data input.

(d) Discriminator loss with model input.

Figure 6.4: Generator and discriminator losses vs. number of steps for the testing dataset.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

This dissertation proposes efficient resource allocation solutions for some of the key facets of next generation mobile networks: network virtualization, fog computing, heterogeneous networks, and spectrum sharing. Distributed frameworks based on game theoretic and ML techniques are proposed by considering the objectives and behaviors of various autonomous entities involved, which are summarized below:

- *The three-sided matching based model from the perspective of the spectrum improves user throughput and satisfaction in wireless network virtualization.* The interrelationships between the spectrum and infrastructure slices, and the mobile users can be modeled using the R-TMSC problem, and it always generates stable matching results in a finite number of steps. The spectrum-oriented R-TMSC algorithm can be employed in virtualization scenarios in order to enhance the user throughput and satisfaction, as well as to serve more number of users compared to the centralized decoupled allocation by the MVNO. The proposed three-sided matching algorithm can be utilized to achieve distributed allocation of network resources in wireless network virtualization, which is flexible to user requirements.

- *The three-sided matching framework can jointly address both the user requirements for different VNF instances as well as their placements in different CNs in NFV.* The proposed R-TMSC algorithm not only improves the average data rate provided by the CNs and the user satisfaction, but also leads to more number of VNF instances being hosted in CNs. Therefore, the proposed model can

be effectively employed in 5G vEPC to consider the VNF demands along with allocating the required resources from CNs.

- *Pricing optimization for DSOs by simultaneously considering the ADSS requirements, as well as resource allocation optimization for FNs is possible for NFV integrated fog computing.* The interactions between the DSOs and the ADSSs in fog computing can be effectively modeled by EPEC, and a large-scale optimization can be facilitated by ADMM. The optimization of resource pricing for the DSOs and the amount of resources to be purchased by the ADSSs can be achieved in real-time using the proposed ADMM for EPEC algorithm. Moreover, the resources from the FNs can be allocated according to the VNF resource requirements of the ADSSs using the proposed many-to-many matching model. The many-to-many matching approach can reduce the costs of FN resource allocation compared to the centralized allocation by the NFVO.

- *RL can be employed for routing optimization in dynamic heterogeneous network scenarios, along with ADMM for EPEC for large scale revenue optimization.* The proposed RL based technique can determine optimal data transmission routes for a VLC-D2D heterogeneous network in a distributed fashion, while simultaneously performing revenue optimization for the D2D users by utilizing ADMM for EPEC. Data transmission routes with low delays and high capacities can be achieved through the RL based algorithm. The proposed RL and EPEC based framework can be employed in similar heterogeneous networks to determine optimal data transmission routes.

- *Deep learning networks like GAN can be used for attack classification in CRNs.* The proposed WGAN based model with CNNs as both the generator and the discriminator, demonstrates efficient PUE attack classification performance. The convergence of the generator and discriminator loss values of the model

shows that cyclostationarity based PUE attack detection frameworks utilizing GANs are effective. The proposed model can be employed in other cyclostationarity based classification scenarios in CRNs.

7.2 Future Works

Finally, there is always room for improvement. On that account, some of the possible directions for future research are listed here:

- **Incorporate user mobility into wireless network virtualization resource allocation**: Considering the widespread integration of V2X in next generation networks, the proposed matching based algorithm can be extended to incorporate user mobility in the virtualization scenario. User mobility can lead to continuous changes in the preference lists of the different sets of entities. A possible workaround for this situation is that the three-sided matching algorithm can be executed repeatedly to reflect the continuous changes in preference lists. Another possible workaround is that models such as the RVV algorithm can be adopted as mentioned in Chapter 2, which can transform a random matching into a stable matching through minimal updates.

- **Extend the proposed framework for 5G vEPC to virtual 5G Core resource allocation**: The proposed three-sided matching framework for 5G vEPC involving TAs, VNF instances and CNs, can be easily extended to the 5G Core network. In that case, the set of VNF functions will change from MME, HSS, PGW, or/and SGW to AMF, SMF, AUSF, or/and UPF. The deployment costs of these VNF instances on to the CNs will also change according to the flavor requirements of the instances.

- **Extend fog computing optimization to meet the diverse requirements**

of IoT: The proposed EPEC and matching based framework can be extended to address the heterogeneous resource requirements of different kinds of IoT devices (ADSSs). Considering the diverse sources of computation tasks in the emerging IoT ecosystem like *smart grids*, wearable devices, VR etc., the demand profiles for various VNF instances can change as per the applications, and accordingly, their deployments on to the FNs can be further optimized.

- **Routing optimization using RL and EPEC can be employed in other heterogenous networks**: Heterogeneous networks are a promising solution for traffic offloading and extending coverage in next generation networks. The proposed RL and EPEC based algorithm can be employed in heterogeneous networks involving *small cells*, *femto cells*, etc. in 5G mobile networks for routing and D2D revenue optimization.

- **GAN based frameworks can be utilized for intelligent PUE attack detection**: Taking into account the role of DSA in improving the spectrum efficiency of next generation networks, the proposed GAN based framework can be extended to detect more intelligent PUE attacks. The huge potential of deep learning models like GANs can be exploited to detect radio-aware intelligent adversaries of CRNs by learning certain features which set them apart from the legitimate users.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper," available at `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html`.

[2] Gemalto, "Introducing 5G technology and networks (definition, features, 5G vs 4G and use cases)," available at `https://www.gemalto.com/mobile/inspired/5G`.

[3] Nokia, "5G for Mission Critical Communication," available at `http://www.hit.bme.hu/~jakab/edu/litr/5G/Nokia_5G_for_Mission_Critical_Communication_White_Paper.pdf`.

[4] M. J. Osborne and A. Rubinstein, *A Course in Game Theory,* MIT Press, Cambridge, MA, 1994.

[5] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjorungnes, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications,* Cambridge University Press, UK, 2011.

[6] E. Alpaydin, *Introduction to Machine Learning,* MIT Press, Cambridge, MA, 2004.

[7] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications,* vol. 24, no. 2, pp. 98–105, Apr. 2017.

[8] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 358–380, Mar. 2015.

[9] W. Xie, J. Zhu, C. Huang, M. Luo, and W. Chou, "Network Virtualization with Dynamic Resource Pooling and Trading Mechanism," *IEEE Global Communications Conference (Globecom)*, Austin, TX, Dec. 2014.

[10] N. M. M. K. Chowdhury and R. Boutaba, "A Survey of Network Virtualization," *Computer Networks,* vol. 54, no. 5, pp. 862–876, Apr. 2010.

[11] X. Wang, P. Krishnamurthy, and D. Tipper, "Wireless Network Virtualization," *International Conference on Computing, Networking and Communications (ICNC),* San Diego, CA, Jan. 2013.

[12] F. Fu and U. C. Kozat, "Stochastic Game for Wireless Network Virtualization," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 84–97, Feb. 2013.

[13] C. Liang and F. R. Yu, "Mobile Virtual Network Admission Control and Resource Allocation for Wireless Network Virtualization: A Robust Optimization Approach," *IEEE Global Communications Conference (Globecom)*, San Diego, CA, Dec. 2015.

[14] E. A. Jorswieck, "Stable Matchings for Resource Allocation in Wireless Networks," *Proceedings of the 17th International Conference on Digital Signal Processing*, Corfu, Greece, Jul. 2011.

[15] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel Opportunistic Carrier Sensing for Stable Channel Access Control in Cognitive Radio Systems," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 82–95, Jan. 2012.

[16] F. Pantisano, M. Bennis, W. Saad, S. Valentin, and M. Debbah, "Matching with Externalities for Context-Aware User-Cell Association in Small Cell Networks," *IEEE Global Communications Conference (Globecom)*, Atlanta, GA, Dec. 2013.

[17] O. Semiari, W. Saad, S. Valentin, M. Bennis, and H. V. Poor, "Context-Aware Small Cell Networks: How Social Metrics Improve Wireless Resource Allocation," *IEEE Transaction on Wireless Communications*, vol. 14, no. 11, pp. 5927–5940, Nov. 2015.

[18] D. Gale and L. S. Shapley, "College Admissions and the Stability of Marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.

[19] A. Roth and M. A. O. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis,* Cambridge University Press, UK, 1992.

[20] D. F. Manlove, *Algorithmics of Matching Under Preferences,* World Scientific, 2013.

[21] R. W. Irving, P. Leather, and D. Gusfield, "An Efficient Algorithm for the Optimal Stable Marriage," *Journal of the ACM (JACM),* vol. 34, no. 3, pp. 532–543, Jul. 1987.

[22] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching Theory for Future Wireless Networks: Fundamentals and Applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, Apr. 2015.

[23] R. A. Banez, H. Xu, N. H. Tran, J. B. Song, C. S. Hong, and Z. Han, "Network Virtualization Resource Allocation and Economics Based on Prey-Predator Food Chain Model," *IEEE Transactions on Communications*, vol. 66, no. 10, pp. 4738–4752, Oct. 2018.

[24] E. Hossain and M. Hasan, "5g Cellular: Key Enabling Technologies and Research Challenges," *IEEE Instrumentation and Measurement Magazine*, vol. 18, no. 3, pp. 11–21, Jun. 2015.

[25] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource Discovery and Allocation in Network Virtualization," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 4, pp. 1114–1128, Feb. 2012.

[26] M. I. Kamel, L. B. Le, and A. Girard, "LTE Wireless Network Virtualization: Dynamic Slicing via Flexible Scheduling," *80th IEEE Vehicular Technology Conference (VTC)*, Vancouver, Canada, Sep. 2014.

[27] Z. B. Zhu, P. Gupta, Q. Wang, S. Kalyanaraman, Y. Lin, H. Franke, and S. Sarangi, "Virtual Base Station Pool: Towards a Wireless Network Cloud for Radio Access Networks," *Proceedings of the 8th ACM International Conference on Computing Frontiers*, Ischia, Italy, May 2011.

[28] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[29] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor, and A. Monje, "On the Optimal Allocation of Virtual Resources in Cloud Computing Networks," *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1060–1071, Jun. 2013.

[30] J. van de Belt, H. Ahmadi, and L. E. Doyle, "Defining and Surveying Wireless Link Virtualization and Wireless Network Virtualization," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1603–1627, May 2017.

[31] J. Feng, Q. Zhang, G. Dong, P. Cao, and Z. Feng, "An Approach to 5G Wireless Network Virtualization: Architecture and Trial Environment," *IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, Mar. 2017.

[32] G. Zhang, K. Yang, H. Jiang, X. Lu, K. Xu, and L. Zhang, "Equilibrium Price and Dynamic Virtual Resource Allocation for Wireless Network Virtualization," *Mobile Networks and Applications,* vol. 22, no. 3, pp. 564–576, Jun. 2017.

[33] M. Kalil, A. Al-Dweik, M. F. A. Sharkh, A. Shami, and A. Refaey, "A Framework for Joint Wireless Network Virtualization and Cloud Radio Access Networks for Next Generation Wireless Networks," *IEEE Access*, vol. 5, pp. 20814–20827, Aug. 2017.

[34] N. Zhang, P. Yang, S. Zhang, D. Chen, W. Zhuang, B. Liang, and X. S. Shen, "Software Defined Networking Enabled Wireless Network Virtualization: Challenges and Solutions," *IEEE Network*, vol. 31, no. 5, pp. 42–49, May 2017.

[35] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "Nvs: A Substrate for Virtualizing Wireless Resources in Cellular Networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.

[36] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G Infrastructure Markets: The Business of Network Slicing," *36th IEEE International Conference on Computer Communications (INFOCOM)*, Atlanta, GA, May 2017.

[37] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A. A. Gebremariam, T. Taleb, and M. Bagaa, "End-to-End Network Slicing for 5G Mobile Networks," *Journal of Information Processing*, vol. 25, pp. 153–163, Feb. 2017.

[38] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network Slicing in 5G: An Auction-Based Model," *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[39] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network Slicing Management and Prioritization in 5G Mobile Systems," *Proceedings of the 22nd European Wireless Conference*, Oulu, Finland, May 2016.

[40] S. Retal and M. Bagaa and T. Taleb and H. Flinck, "Content Delivery Network Slicing: QoE and Cost Awareness," *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[41] S. M. A. Kazmi, N. H. Tran, T. M. Ho, and C. S. Hong, "Hierarchical Matching Game for Service Selection and Resource Purchasing in Wireless Network Virtualization," *IEEE Communications Letters*, vol. 22, no. 1, pp. 121–124, Jan. 2018.

[42] T. H. T. Le, N. H. Tran, T. LeAnh, and C. S. Hong, "User Matching Game in Virtualized 5G Cellular Networks," *18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Kanazawa, Japan, Oct. 2016.

[43] S. M. A. Kazmi and C. S. Hong, "A Matching Game Approach for Resource Allocation in Wireless Network Virtualization," *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Beppu, Japan, Jan. 2017.

[44] D. H. Ho and S. Valaee, "Information Raining and Optimal Link-Layer Design for Mobile Hotspots," *IEEE Transactions On Mobile Computing*, vol. 4, no. 3, pp. 271–284, May 2005.

[45] S. Arabi and E. Sabir and T. Taleb and M. Sadik, "The Right Content for the Right Relay in Self-Organizing Delay Tolerant Networks: A Matching Game Perspective," *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[46] P. Yang, G. Qin, H. Wang, L. Zhang, and G. Chen, "MOTOROLA: Mobility Tolerable Route Selection Algorithm in Wireless Networks," *IET Communications*, vol. 4, no. 7, pp. 837–839, Apr. 2010.

[47] Y. Zhang, Y. Gu, M. Pan, and Z. Han, "Distributed Matching Based Spectrum Allocation in Cognitive Radio Networks," *IEEE Global Communications Conference (Globecom)*, Austin, TX, Dec. 2014.

[48] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Student Admission Matching Based Content-Cache Allocation," *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, Mar. 2015.

[49] Y. Gu, Y. Zhang, L. X. Cai, M. Pan, L. Song, and Z. Han, "LTE-Unlicensed Coexistence Mechanism: A Matching Game Framework," *IEEE Wireless Communications*, vol. 23, no. 6, pp. 54–60, Dec. 2016.

[50] H. W. Stuart Jr, "The Supplier–Firm–Buyer Game and Its M-Sided Generalization," *Mathematical Social Sciences*, vol. 34, no. 1, pp. 21–27, Aug. 1997.

[51] L. Cui and W. Jia, "Cyclic Stable Matching for Three-Sided Networking Services," *Computer Networks*, vol. 57, no. 1, pp. 351–363, Jan. 2013.

[52] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization,* Athena Scientific, 1997.

[53] D. E. Knuth, "Mariages Stables," *Les Presses de l'Universit de Montral*, 1976.

[54] C. Ng and D. S. Hirschberg, "Three-Dimensional Stable Matching Problems," *SIAM Journal on Discrete Mathematics*, vol. 4, no. 2, pp. 245–252, May 1991.

[55] P. Biro and E. McDermid, "Three-Sided Stable Matchings with Cyclic Preferences," *Algorithmica*, vol. 58, no. 1, pp. 5–18, Sep. 2010.

[56] R. Mijumbi, J. Serrat, J-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 236–262, Sep. 2015.

[57] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional Game for the Creation of Efficient Virtual Core Network Slices in 5G Mobile Systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 469–484, Mar. 2018.

[58] Y. Li, and M. Chen, "Software-Defined Network Function Virtualization: A Survey," *IEEE Access*, vol. 3, pp. 2542–2553, Dec. 2015.

[59] C. Liang, F. R. Yu, and X. Zhang, "Information-Centric Network Function Virtualization Over 5G Mobile Wireless Networks," *IEEE Network*, vol. 29, no. 3, pp. 68–74, May 2015.

[60] M. Condoluci, F. Sardis, and T. Mahmoodi, "Softwarization and Virtualization in 5G Networks for Smart Cities," *Proceedings of the International Conference on Cyber Physical Systems, IoT and Sensors Networks (CYCLONE)*, Rome, Italy, Oct. 2015.

[61] T. Taleb, M. Bagaa, and A. Ksentini, "User Mobility-Aware Virtual Network Function Placement for Virtual 5G Network Infrastructure," *IEEE International Conference on Communications (ICC)*, London, UK, Jun. 2015.

[62] C. Hyser, B. McKee, R. Gardner, and B. J. Watson, "Autonomic Virtual Machine Placement in the Data Center," *Tech. Rep. HPL-2007-189, HP Labs*, Palo Alto, CA, Feb. 2008.

[63] L. Mashayekhy and D. Grosu, "A Coalitional Game-Based Mechanism for Forming Cloud Federations," *Proceedings of the 5th IEEE/ACM International Con-*

ference on Utility and Cloud Computing (UCC), Chicago, IL, pp. 223–227, Nov. 2012.

[64] B. K. Ray, S. Khatua, and S. Roy, "Cloud Federation Formation Using Coalitional Game Theory," *Proceedings of the 11th International Conference on Distributed Computing and Internet Technology (ICDCIT)*, Bhubaneswar, India, pp. 345–350, Feb. 2015.

[65] T. Taleb and A. Ksentini, "Gateway Relocation Avoidance-Aware Network Function Placement in Carrier Cloud," *Proceedings of the 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, Barcelona, Spain, pp. 341–346, Nov. 2013.

[66] M. Bagaa, T. Taleb, and A. Ksentini, "Service-Aware Network Function Placement for Efficient Traffic Handling in Carrier Cloud," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Istanbul, Turkey, pp. 2402–2407, Apr. 2014.

[67] F. Z. Yousaf, J. Lessmann, P. Loureiro, and S. Schmid, "SoftEPC—Dynamic Instantiation of Mobile Core Network Entities for Efficient Resource Utilization," *Proceedings of the IEEE International Conference on Communications (ICC)*, Budapest, Hungary, pp. 3602–3606, Jun. 2013.

[68] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems,* vol. 29, nol. 7, pp. 1645–1660, Sep. 2013.

[69] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile Edge Computing: Survey and Research Outlook," arXiv preprint arXiv:1701.01090, 2017.

[70] R. Vilalta, A. Mayoral, R. Casellas, R. Martínez, and R. Muñoz, "Experimental Demonstration of Distributed Multi-Tenant Cloud/Fog and Heterogeneous

SDN/NFV Orchestration for 5G Services," *European Conference on Networks and Communications (EuCNC),* Athens, Greece, Jun. 2016.

[71] R. Vilalta, A. Mayoral, D. Pubill, R. Casellas, R. Martínez, J. Serra, C. Verikoukis, and R. Muñoz, "End-to-End SDN Orchestration of IoT Services Using an SDN/NFV-Enabled Edge Node," *Optical Fiber Communications Conference and Exhibition (OFC),* Anaheim, CA, Mar. 2016.

[72] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," *Proceedings of the Workshop on Mobile Big Data (Mobidata),* pp. 37–42, Hangzhou, China, Jun. 2015.

[73] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-Efficient NFV-Enabled Mobile Edge-Cloud for Low Latency Mobile Applications," *IEEE Transactions on Network and Service Management,* vol. 15, no. 1, pp. 475–488, Mar. 2018.

[74] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing Resource Allocation in Three-Tier IoT Fog Networks: a Joint Optimization Approach Combining Stackelberg Game and Matching," *IEEE Internet Of Things Journal, Special Issue on Fog Computing in the Internet of Things,* vol. 4, no. 5, p.p. 1204–1215, Oct. 2017.

[75] B. S. Mordukhovich, "Equilibrium Problems with Equilibrium Constraints via Multiobjective Optimization," *Research Report 17,* p. 14, 2003, *Optimization Methods and Software 19,* pp. 479–492, 2004.

[76] Z. Zheng, L. Song and Z. Han, "Bridge the Gap Between ADMM and Stackelberg Game: Incentive Mechanism Design for Big Data Networks," *IEEE Signal Processing Letters,* vol. 24, no. 2, pp. 191–195, Feb. 2017.

[77] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning,* vol. 3, no. 1, pp. 1–122, Jul. 2011.

[78] N. Raveendran, H. Zhang, Z. Zheng, L. Song, and Z. Han, "Large-Scale Fog Computing Optimization using Equilibrium Problem with Equilibrium Constraints," *IEEE Global Communications Conference (Globecom),* Singapore, Dec. 2017.

[79] H. Zhang and Z. Han, "Distributed Resource Allocation for Network Virtualization," *Handbook of Cognitive Radio,* Springer, editors: L. Song and W. Zhang, 2019.

[80] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management,* vol. 13, no. 3, pp. 518–532, Sep. 2016.

[81] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC)," *IEEE Network,* vol. 28, no. 6, pp. 18–26, Nov. 2014.

[82] N. Raveendran, Y. Gu, C. Jiang, N. H. Tran, M. Pan, L. Song, and Z. Han, "Cyclic Three-Sided Matching Game Inspired Wireless Network Virtualization," *IEEE Transactions on Mobile Computing*, DOI 10.1109/TMC.2019.2947522.

[83] N. Raveendran, Y. Zha, Y. Zhang, X. Liu, and Z. Han, "Virtual Core Network Resource Allocation in 5G Systems Using Three-Sided Matching," *IEEE International Conference on Communications (ICC),* Shanghai, China, May 2019.

[84] Y. Zhang, N. H. Tran, D. Niyato, and Z. Han, "Multi-Dimensional Payment Plan in Fog Computing with Moral Hazard," *15th IEEE International Conference on Communication Systems (ICCS),* Shenzhen, China, Dec. 2016.

[85] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-Oriented Heterogeneous Resource Sharing for Optimizing Service Latency in Mobile Cloud," *Proceedings of the 1st International Workshop on Mobile Cloud Computing and Networking (MobileCloud),* pp. 19–26, Bangalore, India, Jul. 2013.

[86] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing IoT Applications in the Fog: A Distributed Dataflow Approach," *5th International Conference on the Internet of Things (IOT),* Seoul, South Korea, Oct. 2015.

[87] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint Radio and Computational Resource Allocation in IoT Fog Computing," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 8, p.p. 7475–7484, Aug. 2018.

[88] C. Anglano, M. Canonico, P. Castagno, M. Guazzone, and M. Sereno, "A Game-Theoretic Approach to Coalition Formation in Fog Provider Federations," *Proceedings of the 3rd IEEE International Conference on Fog and Mobile Edge Computing (FMEC),* pp. 123–130, Barcelona, Spain, Apr. 2018.

[89] J. Li, J. Jin, D. Yuan, and H. Zhang, "Virtual Fog: A Virtualization Enabled Fog Computing Framework for Internet of Things," *IEEE Internet of Things Journal,* vol. 5, no. 1, pp. 121–131, Feb. 2018.

[90] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence Analysis of Alternating Direction Method of Multipliers for a Family of Nonconvex Problems," *SIAM Journal on Optimization,* vol. 26, no. 1, pp. 337–364, Jan. 2016.

[91] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A General Analysis of the Convergence of ADMM," *Proceedings of the 32nd International Conference on Machine Learning (ICML),* pp. 343–352, Lille, France, Jul. 2015.

[92] Z. Han, D. Niyato, W. Saad, and T. Basar, *Game Theory for Next Generation Wireless and Communication Networks: Modeling, Analysis, and Design,* Cambridge University Press, UK, 2019.

[93] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-Many Matching Games for Proactive Social-Caching in Wireless Small Cell Networks," *12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt),* Hammamet, Tunisia, May 2014.

[94] C. Pohlmann, "Visible Light Communication," available at `https://pdfs.semanticscholar.org/cd51/c741f013885834a5d05df6abf4213399f3e7.pdf`.

[95] T. Komine and M. Nakagawa, "Fundamental Analysis for Visible-Light Communication System Using LED Lights," *IEEE Transactions on Consumer Electronics,* vol. 50, no. 1, pp. 100–107, Feb. 2004.

[96] E. T. Won, D. Shin, D. K. Jung, Y. J. Oh, T. Bae, H.-C. Kwon, C. Cho, J. Son, D. O'Brien, T.-G. Kang, and T. Matsumura, "Visible Light Communication: Tutorial," available at `http://www.ieee802.org/802_tutorials/2008-03/15-08-0114-02-0000-VLC_Tutorial_MCO_Samsung-VLCC-Oxford_2008-03-17.pdf`.

[97] A. Vavoulas, H. G. Sandalidis, T. A. Tsiftsis, and N. Vaiopoulos, "Coverage Aspects of Indoor VLC Networks," *Journal of Lightwave Technology,* vol. 33, no. 23, pp. 4915–4921, Dec. 2015.

[98] H. Zhang, W. Ding, J. Song, and Z. Han, "A Hierarchical Game Approach for Visible Light Communication and D2D Heterogeneous Network," *IEEE Global Communications Conference (Globecom),* Washington, D.C., MD, Dec. 2016.

[99] P. Gandotra, R. K. Jha, and S. Jain, "A Survey on Device-to-Device (D2D) Communication: Architecture and Security Issues," *Journal of Network and Computer Applications,* vol. 78, pp. 9–29, Jan. 2017.

[100] D. Silver, "Introduction to reinforcement learning," available at `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf`.

[101] D. Silver, "Deep reinforcement learning," available at `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Resources_files/deep_rl.pdf`.

[102] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of AI Research,* vol. 4, pp. 237–285, Jan. 1996.

[103] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction,* MIT Press, Cambridge, MA, 1998.

[104] M. L. Littman and C. Szepesvari, "A Generalized Reinforcement-Learning Model: Convergence and Applications," *Proceedings of the 13th International Conference on Machine Learning (ICML),* pp. 310–318, Bari, Italy, Feb. 1996.

[105] T. Z. Oo, N. H. Tran, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Offloading in HetNet: A Coordination of Interference Mitigation, User Association, and Resource Allocation," *IEEE Transactions on Mobile Computing,* vol. 16, no. 8, pp. 2276–2291, Aug. 2017.

[106] M. Ali and S. Mumtaz and S. Qaisar and M. Naeem, "Smart Heterogeneous Networks: A 5G Paradigm," *Telecommunication Systems,* vol. 66, no. 2, pp. 311–330, Oct. 2017.

[107] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic Resource Allocation Methods for Device-to-Device Communication," *IEEE Wireless Communications,* vol. 21, no. 3, pp. 136–144, Jun. 2014.

[108] A. Abrardo, G. Fodor, and B. Tola, "Network Coding Schemes for Device-to-Device Communications Based Relaying for Cellular Coverage Extension," *16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC),* Stockholm, Sweden, Jun. 2015.

[109] M. Ali, S. Qaisar, M. Naeem, and S. Mumtaz, "Energy Efficient Resource Allocation in D2D-Assisted Heterogeneous Networks with Relays," *IEEE Access: Green Communications and Networking for 5G Wireless,* vol. 4, pp. 4902–4911, Aug. 2016.

[110] R. Liu and C. Zhang, "Dynamic Dwell Timer for Vertical Handover in VLC-WLAN Heterogeneous Networks," *13th International Wireless Communications and Mobile Computing Conference (IWCMC),* Valencia, Spain, Jun. 2017.

[111] M. S. Saud, H. Chowdhury, and M. Katz, "Heterogeneous Software-Defined Networks: Implementation of a Hybrid Radio-Optical Wireless Network," *IEEE Wireless Communications and Networking Conference (WCNC),* San Francisco, CA, Mar. 2017.

[112] M. S. Saud and M. Katz, "Implementation of a Hybrid Optical-RF Wireless Network with Fast Network Handover," *European Wireless,* Dresden, Germany, May 2017.

[113] W. O. Popoola, E. Pikasis, and I. Osahon, "Hybrid Polymer Optical Fibre and Visible Light Communication Link for In-home Network," *26th Wireless and Optical Communication Conference (WOCC),* Newark, NJ, Apr. 2017.

[114] C.-X. Wang, F. Haider, X. Gao, X.-H. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular Architecture and Key Technologies for 5G Wireless Communication Networks," *IEEE Communications Magazine,* vol. 52, no. 2, pp. 122–130, Feb. 2014.

[115] R. Zhang, J. Wang, Z. Wang, Z. Xu, C. Zhao, and L. Hanzo, "Visible Light Communications in Heterogeneous Networks: Paving the Way for User-centric Design," *IEEE Wireless Communications,* vol. 22, no. 2, pp. 8–16, Apr. 2015.

[116] X. Bao, J. Dai, and X. Zhu, "Visible Light Communications Heterogeneous Network (VLC-HetNet): New Model and Protocols for Mobile Scenario," *Wireless Networks,* vol. 23, no. 1, pp. 299–309, Jan. 2017.

[117] M. B. Rahaim, A. M. Vegni, and T. D. C. Little, "A Hybrid Radio Frequency and Broadcast Visible Light Communication System," *IEEE Global Communications Conference (GLOBECOM) Workshops,* Houston, TX, Dec. 2011.

[118] M. Kashef, M. Ismail, M. Abdallah, K. A. Qaraqe, and E. Serpedin, "Energy Efficient Resource Allocation for Mixed RF/VLC Heterogeneous Wireless Networks," *IEEE Journal on Selected Areas in Communications,* vol. 34, no. 4, pp. 883–893, Apr. 2016.

[119] F. Jin, R. Zhang, and L. Hanzo, "Resource Allocation Under Delay-Guarantee Constraints for Heterogeneous Visible-Light and RF Femtocell," *IEEE Transactions on Wireless Communications,* vol. 14, no. 2, pp. 1020–1034, Feb. 2015.

[120] J. Hu and M. P. Wellman, "Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm," *Proceedings of the 15th International Conference on Machine Learning (ICML),* pp. 242–250, Madison, WI, Jul. 1998.

[121] X. Xu, L. Zuo, and Z. Huang, "Reinforcement Learning Algorithms with Function Approximation: Recent Advances and Applications," *Information Sciences,* vol. 261, pp. 1–31, Sep. 2014.

[122] DeepMind, "AlphaGo," available at `https://deepmind.com/research/alphago/`.

[123] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement Learning in Robotics: A Survey," *The International Journal of Robotics Research,* vol. 32, no. 11, pp. 1238–1274, Aug. 2013.

[124] M. L. Littman, "Markov Games as a Framework for Multi-agent Reinforcement Learning," *11th International Conference on Machine Learning (ICML),* New Brunswick, NJ, Jul. 1994.

[125] M. Zolfpour-Arokhlo, A. Selamat, S. M. Hashim, and H. Afkhami, "Modeling of Route Planning System Based on Q Value-based Dynamic Programming with Multi-agent Reinforcement Learning Algorithms," *Engineering Applications of Artificial Intelligence,* vol. 29, pp. 163–177, Mar. 2014.

[126] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-agent Reinforcement Learning Based Cognitive Anti-jamming," *IEEE Wireless Communications and Networking Conference (WCNC),* San Francisco, CA, Mar. 2017.

[127] M. A. Aref, S. Machuzak, S. K. Jayaweera, and S. Lane, "Replicated Q-learning Based Sub-band Selection for Wideband Spectrum Sensing in Cognitive Radio," *IEEE/CIC International Conference on Communications in China (ICCC),* Chengdu, China, Jul. 2016.

[128] R. Ghasemaghaei, M. A. Rahman, W. Gueaieb, and A. E. Saddik, "Ant Colony-Based Reinforcement Learning Algorithm for Routing in Wireless Sensor Net-

works," *IEEE Instrumentation and Measurement Technology Conference,* Warsaw, Poland, May 2007.

[129] M. Hasan, E. Hossain, and D. Niyato, "Random Access for Machine-to-Machine Communication in LTE-Advanced Networks: Issues and Approaches," *IEEE Communications Magazine,* vol. 51, no. 6, pp. 86–93, Jun. 2013.

[130] M. Lee, D. Marconett, X. Ye, and S. J. B. Yoo, "Cognitive Network Management with Reinforcement Learning for Wireless Mesh Networks," *7th IEEE International Conference on IP Operations and Management (IPOM'07),* San Jose, CA, Oct. 2007.

[131] Z. Liu and I. Elhanany, "RL-MAC: A QoS-aware Reinforcement Learning Based MAC Protocol for Wireless Sensor Networks," *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC),* pp. 768–773, Ft. Lauderdale, FL, Jun. 2006.

[132] A. R. Syed, K.-L. A. Yau, J. Qadir, H. Mohamad, N. Ramli, and S. L. Keoh, "Route Selection for Multi-hop Cognitive Radio Networks Using Reinforcement Learning: An Experimental Study," *IEEE Access,* vol. 4, pp. 6304–6324, Sep. 2016.

[133] L. Giupponi, R. Agustí, J. Pérez-Romero, and O. Sallent, "A Novel Joint Radio Resource Management Approach with Reinforcement Learning Mechanisms," *24th IEEE International Performance, Computing, and Communications Conference (IPCCC),* Phoenix, AZ, Apr. 2005.

[134] R. S. H. Istepanian, N. Y. Philip, and M. G. Martini, "Medical QoS Provision Based on Reinforcement Learning in Ultrasound Streaming Over 3.5G Wireless Systems," *IEEE Journal on Selected Areas in Communications,* vol. 27, no. 4, pp. 566–574, May 2009.

[135] N. Mastronarde and M. van der Schaar, "Fast Reinforcement Learning for Energy-Efficient Wireless Communication," *IEEE Transactions in Signal Processing,* vol. 59, no. 12, pp. 6262–6267, Dec. 2011.

[136] C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Machine Learning,* vol. 8, no. 3, pp. 279–292, May. 1992.

[137] S. Leyffer and T. Munson, "Solving Multi-Leader-Common-Follower Games," *Optimization Methods and Software,* vol. 25, no. 4, pp. 601–623, Aug. 2010.

[138] R. Yu, Y. Zhang, Y. Liu, S. Gjessing, and M. Guizani, "Securing Cognitive Radio Networks Against Primary User Emulation Attacks," *IEEE Network,* vol. 30, no. 6, pp. 62–69, Dec. 2016.

[139] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive Radio Networking and Communications: An Overview," *IEEE Transactions on Vehicular Technology,* vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[140] Q. Zhao and A. Swami, "A Survey of Dynamic Spectrum Access: Signal Processing and Networking Perspectives," *in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Honolulu, HI, Apr. 2007.

[141] Z. Yuan, "A Framework of Belief Propagation and Game Theory for Cognitive Radio Security and Routing," 2012. Available at `https://uh-ir.tdl.org/uh-ir/handle/10657/737`.

[142] T. Yucek and H. Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications," *IEEE Communications Surveys & Tutorials,* vol. 11, no. 1, pp. 116–130, Mar. 2009.

[143] S. Geirhofer, L. Tong, and B. M. Sadler, "Cognitive Radios for Dynamic Spectrum Access - Dynamic Spectrum Access in the Time Domain: Modeling and

Exploiting White Space," *IEEE Communications Magazine,* vol. 45, no. 5, pp. 66–72, May 2007.

[144] D. Pu, Y. Shi, A. V. Ilyashenko, and A. M. Wyglinski, "Detecting Primary User Emulation Attack in Cognitive Radio Networks," *in IEEE Global Telecommunications Conference (GLOBECOM),* Houston, TX, Dec. 2011.

[145] Z. Shu, Y. Qian, and S. Ci, "On Physical Layer Security for Cognitive Radio Networks," *IEEE Network,* vol. 27, no. 3, pp. 28–33, May 2013

[146] R. Chen, J.-M. Park, and J. H. Reed, "Defense Against Primary User Emulation Attacks in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications,* vol. 26, no. 1, pp. 25–37, Jan. 2008.

[147] C. S. Xin and M. Song, "Detection of PUE Attacks in Cognitive Radio Networks Based on Signal Activity Pattern," *IEEE Transactions on Mobile Computing,* vol. 13, no. 5, pp. 1022–1034, May 2014.

[148] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications,* vol. 31, no. 11, pp. 2209–2221, Nov. 2013.

[149] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep Learning for Wireless Physical Layer: Opportunities and Challenges," *China Communications,* vol. 14, no. 11, pp. 92–111, Nov. 2017.

[150] N. E. West and T. O'Shea, "Deep Architectures for Modulation Recognition," *in IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN),* Piscataway, NJ, Mar. 2017.

[151] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," *Computer,* vol. 29, no. 3, pp. 31–44, Mar. 1996.

[152] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning,* MIT Press, Cambridge, MA, 2016.

[153] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature,* vol. 521, pp. 436–444, May 2015.

[154] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks,* vol. 61, pp. 85–117, Jan. 2015.

[155] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *in 27th International Conference on Neural Information Processing Systems,* Montreal, Canada, Dec. 2014.

[156] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *in Proceedings of the IEEE,* vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[157] K. Kim, I. A. Akbar, K. K. Bae, J.-S. Um, C. M. Spooner, and J. H. Reed, "Cyclostationary Approaches to Signal Detection and Classification in Cognitive Radio," *in IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks,* Dublin, Ireland, Apr. 2007.

[158] P. D. Sutton, K. E. Nolan, and L. E. Doyle, "Cyclostationary Signatures in Practical Cognitive Radio Applications," *IEEE Journal on Selected Areas in Communications,* vol. 26, no. 1, pp. 13–24, Jan. 2008.

[159] D. Cabric, S. Mishra, and R. Brodersen, "Implementation Issues in Spectrum Sensing for Cognitive Radios," *in Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, Nov. 2004.

[160] V. K. Tumuluru, P. Wang, and D. Niyato, "A Neural Network Based Spectrum Prediction Scheme for Cognitive Radio," *in IEEE International Conference on Communications (ICC),* Cape Town, South Africa, May 2010.

[161] A. Fehske, J. Gaeddert, and J. H. Reed, "A New Approach to Signal Classification Using Spectral Correlation and Neural Networks," *in First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* Baltimore, MD, Nov. 2005.

[162] K. Tsagkaris, A. Katidiotis, and P. Demestichas, "Neural Network-Based Learning Schemes for Cognitive Radio Systems," *Computer Communications,* vol. 31, no. 14, pp. 3394–3404, Sep. 2008.

[163] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks," *IEEE Journal of Selected Topics in Signal Processing,* vol. 12, no. 1, pp. 160–167, Feb. 2018.

[164] G. J. Mendis, J. Wei, and A. Madanayake, "Deep Learning-Based Automated Modulation Classification for Cognitive Radio," *in IEEE International Conference on Communication Systems (ICCS),* Shenzhen, China, Dec. 2016.

[165] M. Zhang, M. Diao, and L. Guo, "Convolutional Neural Networks for Automatic Cognitive Radio Waveform Recognition," *IEEE Access,* vol. 5, pp. 11074–11082, Jun. 2017.

[166] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial Deep Learning for Cognitive Radio Security: Jamming Attack and Defense

Strategies," *in IEEE International Conference on Communications Workshops (ICC Workshops),* Kansas City, MO, May 2018.

[167] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep Learning for Launching and Mitigating Wireless Jamming Attacks," 2018. Available at arXiv:1807.02567.

[168] K. Davaslioglu and Y. E. Sagduyu, "Generative Adversarial Learning for Spectrum Sensing," *in IEEE International Conference on Communications (ICC),* Kansas City, MO, May 2018.

[169] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital Signal Modulation Classification with Data Augmentation Using Generative Adversarial Nets in Cognitive Radio Networks," *IEEE Access,* vol. 6, pp. 15713–15722, Mar. 2018.

[170] S. B. A. Khaliq, M. F. Amjad, H. Abbas, N. Shafqat, and H. Afzal, "Defence Against PUE Attacks in Ad Hoc Cognitive Radio Networks: A Mean Field Game Approach," *Telecommunication Systems,* pp. 1–18, May 2018.

[171] P. Patil, P. Patil, and K. P. Paradeshi, "Detection of Primary User Emulation (PUE) Attack in Cognitive Radio Network," *International Journal of Engineering Science and Computing,* vol. 7, no. 3, pp. 5845–5847, Mar. 2017.

[172] R. Sultana and M. Hussain, "Mitigating Primary User Emulation Attack in Cognitive Radio Network Using Localization and Variance Detection," *Proceedings of the 1st International Conference on Smart System, Innovations and Computing,* pp. 433–444, Jan. 2018.

[173] Q. Dong, Y. Chen, X. Li, and K. Zeng, "An Adaptive Primary User Emulation Attack Detection Mechanism for Cognitive Radio Networks," 2018. Available at arXiv:1804.09266.

[174] Q. Dong, Y. Chen, X. Li, and K. Zeng, "Explore Recurrent Neural Network for PUE Attack Detection in Practical CRN Models," *in IEEE International Smart Cities Conference (ISC2),* Kansas City, MO, Sept. 2018.

[175] Q. Dong, Y. Chen, X. Li, and K. Zeng, "An Adaptive Primary User Emulation Attack Detection Mechanism for Cognitive Radio Networks," *in 14th International Conference on Security and Privacy in Communication Networks (SecureComm),* Singapore, Singapore, Aug. 2018.

[176] J. Unnikrishnan and V. V. Veeravalli, "Cooperative Sensing for Primary Detection in Cognitive Radio," *IEEE Journal of Selected Topics in Signal Processing,* vol. 2, no. 1, pp. 18–27, Feb. 2008.

[177] J. Nash, "Equilibrium Points in N-Person Games," *in Proceedings of the National Academy of Sciences of the United States of America,* vol. 36, no. 1, pp. 48–49, Jan. 1950.

[178] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior,* Princeton University Press, Princeton, NJ, 1944.

[179] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a Century of Research," *Signal Processing,* vol. 86, no. 4, pp. 639–697, Apr. 2006.

[180] B. Ramkumar, "Automatic Modulation Classification for Cognitive Radios Using Cyclic Feature Detection," *IEEE Circuits and Systems Magazine,* vol. 9, no. 2, pp. 27–45, Jun. 2009.

[181] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv preprint arXiv:1701.07875, 2017.

[182] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," arXiv preprint arXiv:1704.00028, 2017.

[183] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

[184] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM,* vol. 60, no. 6, pp. 84–90, Jun. 2017.

[185] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," *in IEEE Conference on Computer Vision and Pattern Recognition,* Columbus, OH, Jun. 2014.