# MATCHING THEORY FRAMEWORK FOR 5G WIRELESS COMMUNICATIONS

A Dissertation

Presented to

the Faculty of the Electrical and Computer Engineering

University of Houston

in Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical and Computer Engineering

by

Yunan Gu

December 2016

MATCHING THEORY FRAMEWORK FOR
5G WIRELESS COMMUNICATIONS

Yunan Gu

Approved:

Chair of the Committee
Dr. Zhu Han, Professor,
Electrical and Computer Engineering

Committee Members:

Dr. Walid Saad, Assistant Professor,
Electrical and Computer Engineering
Virginia Tech.

Dr. Saurabh Prasad, Assistant Professor,
Electrical and Computer Engineering

Dr. Lijun Qian, Professor,
Electrical and Computer Engineering
Prairie View A&M University

Dr. Miao Pan, Assistant Professor,
Electrical and Computer Engineering

Dr. Suresh K. Khator, Associate Dean,
Cullen College of Engineering

Dr. Badrinath Roysam, Professor and Chair,
Electrical and Computer Engineering

# Acknowledgements

I am so appreciative for everyone that has helped me during my Ph.D. study, which has made my short living in this beautiful country an unforgettable experience.

First, I would like to express my great appreciation to my advisor, Professor Zhu Han, for his esteemed guidance, constant encouragement, and continuous support during my graduate studies. His deep academic background and keen insights have helped me achieve significant improvement in my Ph.D. research and to be well prepared for future professional development, which will be invaluable assets for my future career.

Furthermore, I would like to express my sincere gratitude to Professor Miao Pan, who was my advisor during my master's degree in Texas Southern University. He has led me into the wireless communication research field and also helped me in getting such a precious opportunity to pursue a doctor's degree at the University of Houston. I would also like to thank the rest of my dissertation committee, Professor Walid Saad, Professor Saurabh Prasad and Professor Lijun Qian for their precious time and support on this dissertation.

My appreciation also goes to my dear colleagues Huaqing Zhang, Yanru Zhang, Hung Ngyuen, Yong Xiao, Ali Arab, Lanchao Liu, Radwa Sultan, Sai Mounika Muthyala, and Xunsheng Du, whom I always had a great time with in both on and off work times. I am also grateful to my husband Jianing Wu, who is always by my side and sharing both my happiness and sadness. I am grateful to all my friends who gave me strong support during my Ph.D. life, in Houston, back in my hometown China, and all around the world.

Last but by no means least, I want to thank my mom, dad, grandmothers, grandfathers, aunts, uncles, cousins, and nephew for their infinite love, encouragement, patience, and support during all those years, and bringing us together as a wonderful family.

# MATCHING THEORY FRAMEWORK FOR 5G WIRELESS COMMUNICATIONS

An Abstract

of a

Dissertation

Presented to

the Faculty of the Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical and Computer Engineering

by

Yunan Gu

December 2016

# Abstract

The prevalence of high-performance mobile devices such as smartphones and tablets has brought fundamental changes to the existing wireless networks. The growth of multimedia and location-based mobile services has exponentially increased the network congestion and the demands for more wireless resources. The extremely high computational complexity and communication overhead resulting from the conventional centralized resource management methods are no longer suitable to capture the scale of tomorrow's wireless networks. As a result, the resource management in next-generation networks is shifting from the centralized optimization to the self-organizing solutions. The goal of this thesis is to demonstrate the effectiveness of matching theory, a powerful operational research framework, for solving the wireless resource allocation problems in a distributed manner. Matching theory, as a Nobel-prize winning framework, has already been widely used in many economic fields. More recently, matching theory has been shown to have a promising potential for modeling and analyzing wireless resource allocation problems due to three reasons: (1) it offers suitable models that can inherently capture various wireless communication features; (2) the ability to use notions, such as preference relations, that can interpret complex system requirements; (3) it provides low-complexity and near-optimal matching algorithms while guaranteeing the system stability.

This dissertation provides a theoretical research of implementing the matching theory into the wireless communication fields. The main contributions of this dissertation are summarized as follows.

- An overview of the basic concepts, classifications, and models of the matching theory is provided. Furthermore, comparisons with existing mathematical solutions for the resource allocation problems in the wireless networks are conducted.

- Applications of matching theory in the wireless communications are studied. Especially, the stable marriage model, the student project allocation model and so on are introduced and

applied to solve the resource allocation problems, such as the device-to-device (D2D) communication, LTE-Unlicensed, and so on.

- Both theoretical and numerical analysis are provided to show that matching theory can model complex system requirements, and also provide semi-distributive matching algorithms to achieve stable and close-optimal results.

- The potential and challenges of the matching theory for designing resource allocation mechanisms in the future wireless networks are discussed.

# Table of Contents

# List of Figures

# List of Tables

## Chapter 1

# Introduction

The emergence of the novel wireless networking paradigms has forever transformed the way in which the wireless systems are operated. In particular, the need for the self-organizing solutions to manage the scarce spectral resources has become a prevalent theme in many wireless systems. To meet the exponentially increased traffic demand, a lot of new paradigms have proposed such as: a) cognitive radio networks, in which cognitive devices can adaptively opportunistically access the wireless spectrum, b) small cell networks, which boost the network capacity and coverage via the deployment of low-cost small cell base stations, and c) large-scale D2D communications, which can be deployed over both cellular and unlicensed bands to improve system throughput. This is gradually leading to the future multi-tiered heterogeneous wireless architecture, as shown in Fig. 1.1.

Effectively managing resource allocation in such a complex environment warrants a fundamental shift from the traditional centralized mechanisms towards the self-organizing optimizing approaches. The need for this shift is motivated by practical factors such as increased network density and the need for low-latency communications. In consequence, there is a need for self-organizing systems in which small cell base stations and even devices can have some intelligence to make resource management decisions rapidly. Indeed, there has been a recent surge in literature that proposes new mathematical tools for optimizing the wireless resource allocations. Examples include centralized optimization and game theory approach. The centralized optimization techniques can provide optimal solutions. However, they often require global network information and centralized control, thus yielding significant overhead and complexity. This complexity can rapidly increase when dealing with combinatorial, integer programming problems such as the channel allocation and user association. Moreover, centralized optimization may not be able to properly handle the challenges that emerge in the dense and heterogeneous wireless environments.

The aforementioned limitations of the centralized optimization have led to an interesting body

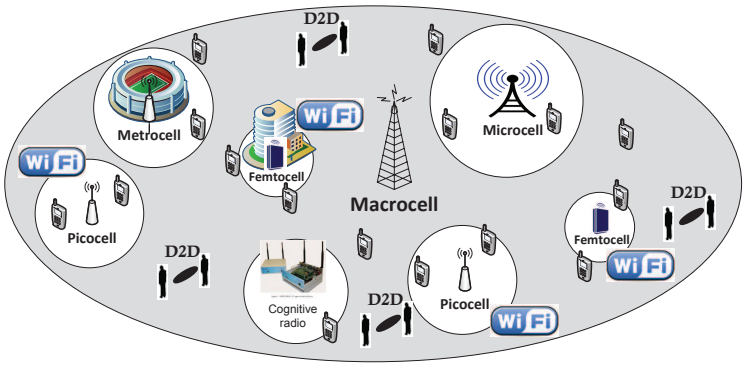



Figure 1.1: A future wireless network with a mixture of small cells, cognitive radio devices, and heterogeneous spectrum bands.

of literature that deals with the use of noncooperative games for the wireless resource allocations [1]. Despite the potential, such approaches present some shortcomings. First, classical game-theoretic algorithms, such as the best response mechanism, require the knowledge of other players' actions, thus limiting their distributive implementations. Second, most game-theoretic solutions, such as the Nash equilibrium, investigate the one-sided (or unilateral) stability notions in which the equilibrium deviations are evaluated unilaterally. Such unilateral deviations may not be practical when investigating the assignment problems between distinct sets of players. Last, but not least, the tractability of equilibrium in the game-theoretic methods requires having certain form of structure in the objective function, which cannot be satisfied in some practical wireless applications.

Recently, the matching theory framework has emerged as a promising technique for the wireless resource allocations which can overcome some limitations of the game theory and centralized optimization approaches [2]. The matching theory is a Nobel-prize winning framework that provides mathematically tractable solutions for the combinatorial problem of matching players from two distinct sets [3] [4], depending on the individual information and preference of each player. The advantages of matching theory for wireless resource management include: 1) suitable models for characterizing interactions between the heterogeneous nodes, each of which has its own type, objective, and information, 2) the ability to define the general "preferences" that can handle the heterogeneous and complex considerations related to the wireless quality-of-service (QoS) require-

ments, 3) suitable solutions, in terms of stability and optimality, that can accurately reflect different system objectives, and 4) efficient algorithmic implementations that are inherently self-organizing.

In this thesis, we aim to provide a matching-based framework that can solve various resource allocation problems in the 5G wireless networks. We show how to apply suitable matching models to solve specific wireless communication problems through the illustrations of our applications. For each of the matching applications, we provide the matching game modeling, solution discussion, and performance evaluations.

## 1.1 5G Wireless Communications

Various new paradigms have emerged during the development of the 5G networks. In the following sections, we introduce these new technologies regarding their potentials and challenges in boosting the 5G network capacities.

### 1.1.1 Device to Device Communication

To satisfy the increased traffic demand, the device-to-device (D2D) communication technology has been proposed for the Long Term Evaluation-Advanced (LTE-A) standard. In the D2D communications, the user equipment communicates with each other through a direct link by using the licensed resources instead of communicating with the BSs. It is considered to have some advantages such as: offloading the traffic, improving the system throughput, as well as extending the network coverage [5] [6].

Typically, in the D2D communications, the system throughput and reliability are considered as the optimization objectives in some existing works. For example, Kaufman and Aazhang in [7] intend to optimize the system throughput, while simultaneously guaranteeing the QoS requirements. However, being such a promising technology, the D2D communications also pose new challenges to the traditional cellular networks. One most critical issue is the interference brought by the channel reuse between the D2D and cellular users. Effective approaches to solve this problem include

transmission power management [8], interference avoiding multiple-input-multiple-output (MIMO) techniques [9] and advanced coding schemes [10].

### 1.1.2 Content Caching

Nowadays, with the emerging of mobile devices such as smartphones and tablets, more and more users are accessing the online social networks such as Facebook, Flickr and so on. As one of the largest online social networks, Facebook stores billions of photo contents. To deliver the contents to users efficiently, heterogeneous cache centers are used to support the Facebook Backend storage center. One important measurement to evaluate the user satisfaction is the response delay of the user request, which is highly depending on the data fetching paths, and thus relating to specific content caching allocation techniques. As a result, an appropriate content caching methodology can play a major role in improving the user satisfaction.

Before discussing any caching method, we first introduce the Facebook photo storage architecture. There are typically three layers of cache centers in front of the backend storage, also called the Haystack storage. These three cache layers are the Browser cache, the Edge cache, and the Origin cache. The Haystack storage stores all the data [11], part of which will be cached to the one of the three-layer cache centers to reduce the service latency. The first layer, which is the closest to the end users, is the Browser cache. The Browser cache centers are typically embedded in the user equipment such as desktops and mobile phones. The second layer cache is called the Facebook Edge cache [12], and the third layer is called the Origin cache. When a user requests data from Facebook, it first looks up the content in the user's local browser cache. If the fetch is a miss, the browser sends an HTTP request to the Internet, and the Facebook web server calculates a photo fetching path, which directs the search process to the higher layers of cache. Then, if the search in the Edge cache fails again, it will proceed to the Origin cache. If a miss happens again, the last try would be the Backend storage, which guarantees a $100\%$ hit since it stores all the data [13].

### 1.1.3 LTE-Unlicensed

To meet the mobile traffic demand, an intuitive idea is to exploit more licensed spectrum, which ensures the reliable and predictable performance. However, it is not quite possible that sufficient additional licensed spectrum can be available in the near future. A growing interest in exploiting the unlicensed spectrum to boost the network capacity has recently arisen. Some cellular network operators have deployed the Wi-Fi access points to offload the cellular traffic to the unlicensed spectrum. However, such efforts are limited by some disadvantages such as the extra cost due to the investment in backhaul and core networks, degradation of the Wi-Fi performance, and lack of good coordination between the cellular and Wi-Fi systems. Another way to augment the LTE capacity to meet the traffic demands is to integrate the unlicensed carriers into the LTE system to enhance the cellular transmission using the carrier aggregation (CA) technology. The CA technology provides the option of aggregating two or more component carriers into a combined virtual bandwidth for enhanced transmission [14]. By aggregating the unlicensed spectrum into cellular networks with CA, the capacity of the LTE network can be boosted, while maintaining the seamless mobility management and predictable performance. This technology is commonly referred to as the LTE-Unlicensed [15].

### 1.1.4 LTE Assisted V2V Communications

The technology of the connected vehicles has been envisioned as a paradigm capable of providing increased convenience to drivers, with applications ranging from road safety to traffic efficiency. In the traditional IEEE 802.11p based vehicle-to-vehicle (V2V) communications, reliable and efficient performance cannot be guaranteed since 802.11p is CSMA/CA based. Besides, the high cost of deploying roadside units (RSUs) cannot be ignored. Thus, the concept of integrating LTE into the V2X communications has been proposed, which is commonly referred to as the LTE-V [16] or the LTE-based V2X [17]. The objectives of the study on LTE V2X communications include the definition of an evaluation methodology and the possible scenarios for vehicular applications, and

the identification of necessary enhancements to the LTE physical layer, protocols, and interfaces. So far, 3GPP has defined 18 use cases in TR 22.885 for the LTE-based V2X services, such as Case 5.1: Forward collision warning, Case 5.8: Road safety services, Case 5.9: Automatic parking system, and so on [18].

### 1.1.5 Fog Computing

Cloud computing is an Internet-based computing platform that provides shared processing resources and data to computers and other devices on demand. The cloud computing and storage solutions can provide users and enterprises with various capabilities to store and process their data in the third-party data centers [19]. In particular, mobile cloud computing (MCC), as a combination of cloud computing, mobile computing and wireless networks, has made it possible for the mobile users to access the cloud resources to offload the computational tasks [20]. With the emerging of the new paradigm, namely, Internet of Things (IoT), a new range of services and applications have been enabled, such as the connected vehicles, smart grid, wireless sensor networks and so on. Not only facing the volume, velocity and variety increase in the communication contents, but also the new communication requirements, such as location awareness, real-time mobility management, and so on. Therefore, it requires a new designed MCC framework to meet these critical requirements.

CISCO first proposed the idea of Fog Computing in 2014, as a platform that exists between the end devices and the cloud data centers. It provides computation, storage and communication resources to the nearby mobile users [21]. Fog computing brings the cloud closer to the end users, and is characterized by features such as low latency, wide-spread distribution, support for mobility, heterogeneity, interoperability and federation [22]. Largely distributed at the network edge, the fog nodes (FNs) provide storage, computation and communication capabilities. An FN can be a cellular BS, a Wi-Fi AP or a femtocell router with upgraded CPU and memories in either fixed locations, such as a shopping mall, or being mobile [23].

### 1.1.6 Wireless Network Function Virtualization

Nowadays, virtualization has become a popular concept applied in many areas, such as the virtual memory, virtual machine and virtual data center. Virtualization refers to the abstraction and sharing of resources among different parties. It offers great network flexibility, maximizes network utilization, as well as inspires new services and products [24]. On the other hand, the mobile wireless traffic is expected to grow exponentially due to the massive user number and the rich communication contents. By extending the network virtualization into cellular networks, and more specifically, by abstracting, slicing and sharing the physical infrastructures and radio resources, the wireless traffic can be relieved. This paradigm is commonly referred to as the wireless network virtualization [25].

### 1.1.7 Mobile Crowd Sensing

The widespread of mobile devices, such as smartphones and vehicular systems, has provided us a new type of sensing infrastructure. These smart mobile devices are embedded with various types of sensors, such as camera, GPS, accelerator, digital compass, light sensor, and even health and pollution monitoring sensors in the future. These sensed data has enabled a broad range of applications such as the road transportation, health care, environmental surveillance, marketing and so on [26]. Compared with the traditional static sensor networks, utilizing the mobile people-centric measurement has many advantages, such as cost-efficient, scalable, better storage and computation capability, direct access to the Internet and so on [26]. This new paradigm is typically referred to as the mobile crowd sensing (MCS), where individual users with sensing and computing devices collectively share data and extract the information to measure and map the phenomena of common interest [27].

## 1.2 Thesis Organization

The rest of the thesis is organized as follow. In Chapter 2, we discuss the fundamental definitions of the matching theory, including the introduction of the stable (SM) marriage problem,

the conventional matching models, and the wireless-oriented matching models. Then in Chapter 3, we implement the SM model to solve the resource allocation problem in the D2D communications, and study the strategic issue in matching. Then, the Facebook content caching problem will be discussed in Chapter 3, modeled by the student admission (SA) matching game. In Chapter 4, the resource allocation problem in the LTE-Unlicensed with network dynamics is studied. Chapter 5 discusses the LTE assisted V2V communications, which is solved by modeling it as the stale fixture (SF) game. Then, we study the joint computational and radio resource allocations in the fog computing framework, and model it with the student project allocation (SPA) game in Chapter 6. The wireless network virtualization problem is discussed in Chapter 7, and is modeled as the three-sided stable marriage (3DSM) matching game. Besides, we study mobile crowd sensing application, and model the problem with the SF model in Chapter 8. Finally, two of my future works are proposed in Chapter 9.

# Chapter 2

# Fundamentals of Matching Theory

## 2.1 Priliminaries

Matching theory, in economics, is a mathematical framework attempting to describe the formation of mutually beneficial relationships over time. Before Gale and Shapley first studied the stable marriage and college admission problems in 1962, many matching problems were solved by the "free for all market[1]" [3]. Economists have identified several issues such as unraveling, congestion, and exploding offers in the "free for all market". Since then, with decades of efforts devoted to the matching algorithms (i.e., there arises a trusted third party, which collects information, runs the matching algorithm, and broadcasts the matching results), these problems could be avoided. As a result, these matching mechanisms are well developed and widely used in many areas, such as the national resident matching program in the United States, the college admission in Hungary, the incompatible kidney exchange market, the partnership formation in peer-to-peer (P2P) network, and so on.

To start with, we use the classical matching model *stable marriage* (SM) [28] as an illustration example. Assume a set of men and a set of women, and each of them is called a matching agent. A *preference list* for each agent is an ordered list based on the preferences over the other set of agents who he/she finds acceptable. A matching consists of (man, woman) pairs. As a fundamental requirement in any matching model, the *stability* concept refers to the case that, no *blocking pair* (BP) exists in a matching. A BP is defined as a (man, woman) pair, who both have the incentive to leave their current partners and form a new marriage relation with each other. A stable matching can be achieved by using the Gale-Shapley (GS) algorithm, which is widely deployed and has been customized to generate stable matchings in many other models.

The GS algorithm is an iterative procedure, where players in one set make proposals to the

[1] "The free for all market" term refers to the period before matching theory came into application, as well as the way that matching problems were dealt with during the period.

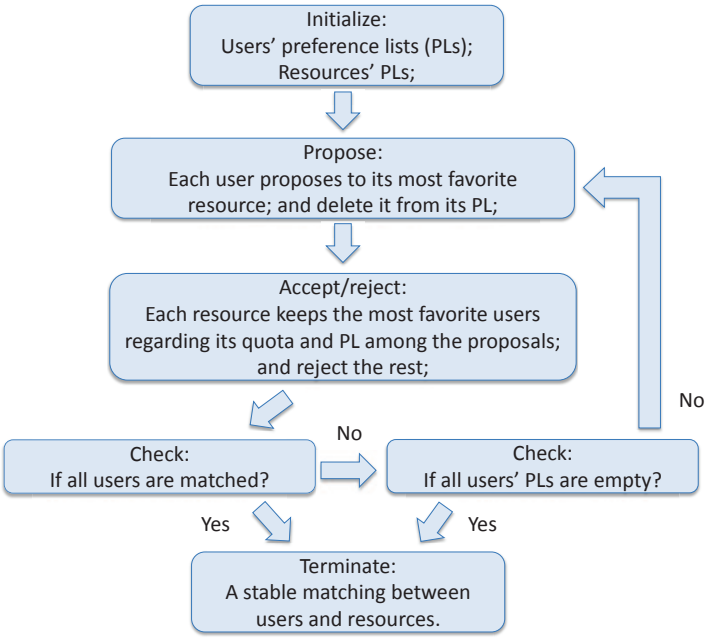



Figure 2.1: The Gale-Shapley algorithm.

other set, whose players, in turn, decide to accept or reject these proposals, w.r.t. their quota. Players make their decisions based on their individual preferences. This process admits many distributed implementations which do not require the players to know each other's preferences [28]. The GS algorithm terminates when no further proposals are made. A flow chart illustrating the execution of GS is shown in Fig. 2.1.

## 2.2 Conventional Matching Models

Matching problems can be classified in different ways. One typical classification is shown in Fig. 2.2. The detailed explanations are provided as follows:

- Bipartite matching problems with two-sided preferences. Here the participating agents can be partitioned into two disjoint sets, and each member of one set ranks a subset of the members in the other set in order of preference. Example applications include assigning junior doctors to hospitals, pupils to schools and school leavers to universities.

- Bipartite matching problems with one-sided preferences. Again the participating agents can

| Bipartite matching with two-sided preferences | Stable Marriage (SM); Hospital Resident (HR); Worker-Firm (WF); |
|---|---|
| Bipartite matching with one-sided preference | Housing allocation (HA); Assigning paper to reviewers; DVD rental markets; |
| Non-Bipartite matching with preferences | Stable roommate (SR); Forming chess tournament pairs; Creating P2P partnerships. |

Figure 2.2: Conventional classification of matching theory.

be partitioned into two disjoint sets, but this time only one set of players rank the subsets of the members in the other set in order of preferences. Example applications include campus housing allocation, DVD rental markets and assigning reviewers to conference papers.

- Non-bipartite matching problems with preferences. Here, all the participating agents form a single homogeneous set, and each agent ranks a subset of the others in order of preferences. Example applications include forming pairs of agents for chess tournaments, finding kidney exchanges involving incompatible (patient, donor) pairs and creating partnerships in P2P networks.

On the other hand, if we consider the capacity/quota allowed for each agent, we can have the following classification.

- One-to-one matching. It means each member of one set can be matched to at most one player from the opposite set. Examples include SM problem, forming roommate pairs, and so on. The one-to-one matching can be bipartite matchings with two-sided preferences (e.g., the SM problem), or bipartite matchings with one-sided preferences, or non-bipartite matchings (e.g., the stable roommate problem).

- Many-to-one matching. It means each agent of one set can be matched to more than one member from the opposite set up to the capacity, while agents from the opposite side can only be matched to one agent at most. Examples are like allocating residents to hospitals, assigning

school leavers to universities. Still, the many-to-one matching can be bipartite matchings with two-sided preferences (e.g., the hospital resident allocation problem), or bipartite matchings with one-sided preferences (e.g., the student housing allocation problem), or non-bipartite matchings.

- Many-to-many matching. It means agents from both matching sets can be matched to more than one agent up to their capacities. Examples include creating partnerships in P2P networks and assigning workers to firms problem. The many-to-many matching can be bipartite matchings with two-sided preferences (e.g., the finding kidney exchange problem), or bipartite matchings with one-sided preferences, or non-bipartite matchings.

## 2.3 Wireless-oriented Matching Models

To implement matching theory into the wireless resource allocations, we assume the wireless users and resources as the matching players. To capture the various wireless resource management features, we condense the rich matching literature into three new classes of problems, illustrated in Fig. 2.3, with the following properties:

1. *Class I: Canonical matching:* This constitutes the baseline class in which the preference of any resource (user) depends solely on the information available at this resource (user) and on the users (resources) to which it is seeking to match. This is useful to study the resource management within a single cell or for allocating orthogonal spectrum resources. Some of the applications discussed in the thesis belong to this category, for example, Chapter 3, 4, 6 and 9.

2. *Class II: Matching with externalities:* This class allows finding desirable matchings when the problem exhibits "externalities", which can be translated into the interdependencies between the players' preferences. For example, in a small cell network, whenever a user is associated with a resource, the preference of other users will change since this allocated resource can create interference to other users using the same frequency. Thus, the preference of any player

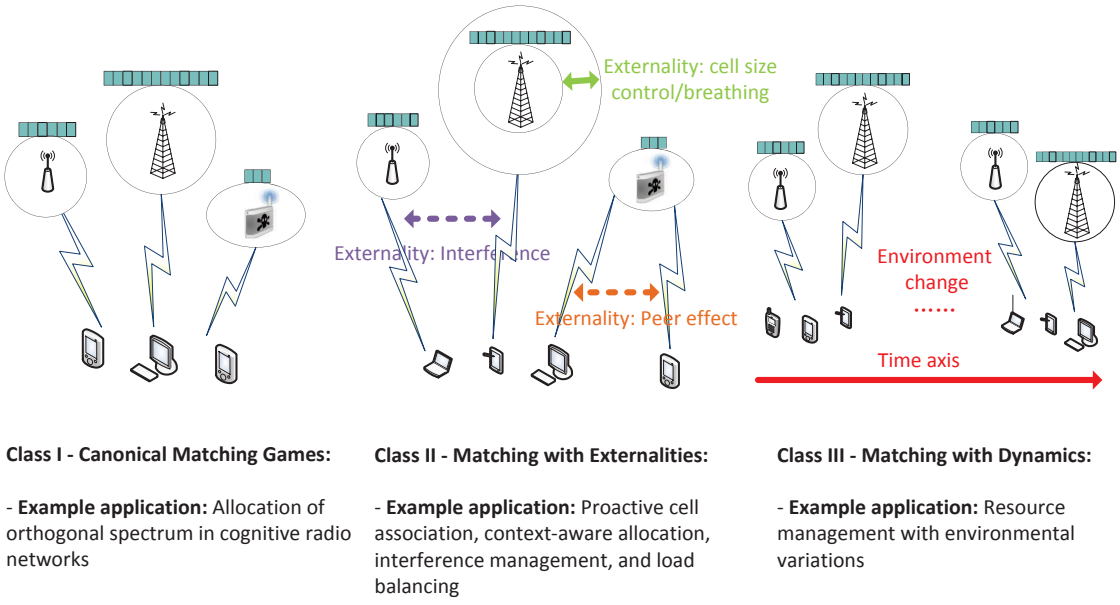



Figure 2.3: Wireless-oriented classification of matching theory.

depends not only on the information available at this player, but also on the entire matching. We distinguish between two types of externalities: conventional externalities and peer effects. In the former, the dependence of the preferences refers to the performance change due to the interference between the matched (user, resource) pair. In the latter, the preference of a user on a resource will depend on the identity and number of other users that are matched to the same resource. Such peer effects are abundant in the wireless environment. We have also discussed how to address the external effect in some of the chapters, such as Chapter 5 and 7.

3. *Class III: Matching with dynamics:* The third class, matching with dynamics, is suitable for scenarios in which one must adapt the matching processes to dynamics of the environment such as fast fading, mobility, or time-varying traffic. Here, at each time, the preferences of the players might change and, thus, the time dimension must be considered for in the matching solution. However, for a given time, the matching problem can either belong to class I or class II. The dynamic issue is also discussed in this thesis, such as in Chapter 5.

Mathematically, the formulation of problems in all three classes will follow the basics of

Section 2.1. For class I, the preferences of one player set simply depend on the other player set. However, for class II, the preferences will now depend not only on the matched player set, but also on the entire matching, due to externalities. For class III, one can introduce a time-dependent state variable in the matching to achieve dynamic stability. Subsequently, the preferences will be time and state dependent, if the problem has both dynamics and externalities. The transition between states depends on the application being studied. For example, if the state represents the activity pattern of a licensed user, the transition would follow a classical Markov model. In contrast, if the state represents a dynamically varying fast fading channel, one can use differential equations to represent the state transition.

## 2.4 Matching Theory in Wireless Communications

The basic wireless resource management problem can be modeled as a matching game between resources and users. Depending on the scenarios, the resources can be of different abstraction levels, representing base stations, time-frequency chunks, power, or others. Users can be devices, stations, or smartphone applications. Each user and resource has a quota that defines the maximum number of players with which it can be matched. The main goal of matching is to optimally match resources with users, given their individual and different objectives and also their self-learned information. Each user (resource) builds a ranking of the resources (users) using a preference relation. In its basic form, a preference can simply be defined in terms of a utility function that quantifies the quality of service (QoS) requirements achieved by a certain matching. However, a preference is more generic than a utility function in that it can incorporate additional qualitative measures extracted from the information available to users and resources. In the wireless resource management, the matching stability implies the robustness to deviations that can benefit both the resource owners and the users. In fact, an unstable matching can for example lead to undesirable cases in which a base station can swap its least preferred user with another base station since this swap is beneficial to both the resource and the user. Having such network-wide deviations ultimately leads to unstable network operations. This concept is very useful in matching problems and is broadly applicable to

all classes.

The information exchange during the matching is implemented in a semi-distributive way, meaning some of the operations are made based on the players' locally collected information, while some other decisions may require the global information from the centralized agent (e.g., eNBs). Before the execution of any matching algorithm, the first step is to set up the preference lists for all players. The preference list is set up through the local information collection by each player. The collected/exchanged information can be channel state information (CSI), player location, or any other information that interests the player. After the information collection, players will rank the other type of players, according to their preferences, into descending/ascending orders. Thus, the preference list set-up is implemented distributively, without requiring any centralized agent. To explain the information exchange during the actual matching algorithm implementations, we need to first clarify the major operations that involve message exchange during the matching. Take the GS algorithm for example, most operations taken by the players, are the proposing, accepting and rejecting operations. To realize these operations, players need to maintain their preference lists and the temporary matching matrix. The proposing information is sent from one type of player to the other type through the communication signal with specific overhead that indicates the proposing operation. Those users, who have received the proposal signals, will decide who to keep and who to reject according to the preference lists and capacity requirements, and also update their temporary matching matrixes. The rejecting/accepting operations are realized by sending the communication signals with rejecting/accepting overheads to the rejected players. The players, who received such signals will update their matching matrixes and then decide whether to start new proposals or not. Thus, GS algorithm can be implemented in a fully distributive way. However, there are still some operations that may require the assistance from the centralized agents in some other matching algorithms. For example, in the random path to stability (RPTS) algorithm, introduced in Chapter 5, each iteration involves the detection of a BP. Such detection requires the computation of the centralized agents (e.g., eNBs), who have access to all players' preference lists. Once a BP is detected, the centralized agent will inform the involved players in this BP of the BP status through commu-

nication signals. Then the informed BP players will take the divorce and remarry actions with the related users indicated in the communication signals. The divorce and remarry actions are realized by the rejecting and accepting operations. To summarize, most operations in a matching algorithm can be implemented distributively, while a small part of the operations require the assistance from the centralized agents, and thus making our matching framework a semi-distributive one.

# Chapter 3

# Matching and Cheating in Device to Device Communications Underlying Cellular Networks

In the device-to-device (D2D) communications, the mobile users communicate directly without going through the base station (BS). The D2D commutation has the advantage of improving spectrum efficiency. However, the interference introduced by the resource sharing in D2D has become a new challenge. In this chapter, we try to optimize the system throughput while simultaneously meeting the QoS requirements for both D2D users and cellular users (CUs). We implement the SM model to solve the resource allocation between D2D users and CUs. We introduce two stable matching algorithms to optimize the social welfare while ensuring the network stability. Moreover, we introduce the idea of cheating in matching to further improve D2D users' throughput. The cheating mechanism is proven to be able to benefit a subset of all the D2D users without reducing the rest of D2D users' performances. Through the simulation results, we demonstrate the effectiveness of our proposed algorithms in terms of improving both the D2D users' and the whole system's throughputs [29].

## 3.1 Introduction

To satisfy the increased traffic demand, the device-to-device (D2D) communication technology has been proposed for the Long Term Evaluation-Advanced (LTE-A) standard. In the D2D communications, the user equipment communicates with each other through a direct link by using the licensed resources instead of communicating with the BSs. It is considered to have some advantages such as: offloading the traffic, improving the system throughput and energy efficiency, as well as extending the network coverage [5] [6].

Typically, in the D2D communications, the system throughput and reliability are considered as the optimization objectives in some existing works. For example, Kaufman and Aazhang in [7]

intend to optimize the system throughput, while simultaneously guaranteeing the QoS requirements. However, being such a promising technology, the D2D communications also pose new challenges to the traditional cellular networks. One most critical issue is the interference brought by the channel reuse between D2D and cellular users. Effective approaches to solve this problem include transmission power management [8], interference avoiding multiple-input-multiple-output (MIMO) techniques [9] and advanced coding schemes [10].

Indeed, there has been a recent surge in the literature that proposes new mathematical frameworks, such as game theory [5] [30], auction theory [31], social networks [32] and graph theory [33], to solve the resource allocation problems in the D2D communications. For example, Wang et al. in [30] developed a Stackelberg game model between the CUs and D2D users, such that both the network throughput and user fairness are taken into account. Then in [31], Xu et al. introduce a reverse iterative combinatorial auction mechanism to deploy the D2D communication as an underlay for the downlink (DL) transmission in the cellular networks. Zhang et al. in [32] proposed an approach to improve the performance of D2D users by exploiting the social ties among individual users. Based on the users' social network profiles, they successfully offloaded the data traffic in the D2D networks by modeling the problem as the so-called Indian Buffet Process. In [33], a framework of the D2D resource allocation is presented to maximize the network throughput. A so-called Kuhn-Munkers algorithm is utilized to solve the bipartite matching between the D2D users and CUs, which achieves the optimal solution regarding the system throughput.

The Kuhn-Munkers algorithm adopted in [33] solves the resource allocation problem by modeling it as a weighted bipartite matching graph and achieves the optimal social welfare. However, beyond social welfare, another important concept for the resource management should be considered, which is called the network stability. This stability notion implies the robustness to deviations that can benefit both the resource owners and the users. In fact, an unstable matching may lead to undesirable and messy network operations. This motivates us to find a stable and efficient resource allocation method. Thus, in this chapter, we propose a one-to-one matching framework to solve the resource allocation problem in D2D communications. Our main contributions are summarized as

follows:

1) We propose a network consists of both D2D and cellular users, where D2D users seek to share CUs' spectrum to maximize the system utilization while satisfying the QoS requirements of both types of users. This resource sharing problem is formulated as a mixed integer non-linear programming (MINLP) problem.

2) We solve this resource allocation problem by modeling it as the SM game to find a stable matching between admissible D2D pairs and CUs. Two stable matching algorithms: the Gale-Shapley (GS) algorithm and minimum weight stable matching algorithm are proposed as two solution approaches.

3) We investigate how to further improve some D2D users' throughput by taking the cheating action. The cheating action is realized by implementing the so-called coalition strategy (CS). In addition, we discuss the impact of the cabal (defined in Definition 3.5) size on the cheating results.

The rest of the chapter is organized as follows. First in Section 3.2, we introduce the related works of matching theory. Then in Section 3.3, we model and formulate the resource allocation problem. The optimization problem is solved in Section 3.4 and then the cheating issue is discussed in Section 3.5. The proposed algorithms are evaluated in Section 3.6. Finally, conclusions are drawn in Section 3.7.

## 3.2 Related Works

Bayat et al. have studied the resource allocations in the cognitive radio networks, the physical layer security problem, as well as the femtocell negotiation problem using the matching theory. The spectrum allocation problem in the cognitive radio networks is studied in [34]. It's modeled as the one-to-one bipartite matching problem between the primary user and secondary user. The GS algorithm is utilized sequentially with dynamic preferences. The physical layer security issue is studied in [35] [36]. It's modeled as the one-to-one bipartite matching between the source-destination pairs and jammer nodes. In [37], a many-to-many bipartite matching between the wireless operators

(WOs) and femtocell access points (FAPs) is studied. The problem is solved with two rounds of matchings, where the second round matching is based on the negotiating results from the first round.

Saad et al. have applied matching theory into the channel assignment and the small cell association problems. They study the joint uplink(UL)/DL resource allocations in [38]. They model the interactions between the mobile users and UL subcarriers as the bipartite matching with two-sided preferences, and model the interactions between the users and DL subcarrier as the bipartite matching with one-sided preferences. The matching process is repeated until the system utility is maximized or no subcarrier needs to be reallocated. In [39], the context-aware user-cell association problem in small cell networks is discussed. They model it as the one-to-one bipartite matching between users and small base stations (SBSs). Starting from a random matching, users propose to the SBSs with better utilities and the SBSs decide whether to accept or reject based on their own interests. Then in [40], the many-to-one bipartite matching game is proposed to model the interactions between the UL users and SBSs. In this problem, the college admission game is performed at first, and then the transfer coalitional game is used repeatedly until the convergence to the Nash equilibrium. The many-to-many bipartite matching between the SBSs and the service providers' servers (SPS) is studied in [41]. The SPSs aim to cache the videos to SBSs to reduce end-user delay, while the SBSs agree to cache different videos based on their popularities in order to reduce the backhaul link load. Finally, a pair-wise stable matching is reached.

Leshem et al. have studied the channel assignment problems using matching theory, and also studied the fast matching algorithms. The one-to-one bipartite matching between the channel and users is discussed in [42]. The GS algorithm is used to solve the problem. Then in [43], a fast matching algorithm for asymptotically optimal channel assignment is proposed, which yields an asymptotically optimal matching. A similar idea is presented in [44], where a modified distributed auction algorithm is proposed, and then a matching algorithm is proposed to find the asymptotically optimal solution in a faster way.

Rami et al. have studied an adaptive cross-layer scheduling in the LTE DL in [45]. They apply

the many-to-one bipartite matching to model the problem and propose a Pareto-efficient allocation to find the trade-off between the system throughput and user fairness. In [46], a similar algorithm is applied in the channel allocation problem in the CR networks. In addition, a distributed English auction method is utilized to reach the Walrasian equilibrium in the network.

There are some other works that focus on the matching-based wireless resource allocations such as [47] [48] [49] [50] [51] [52], and so on. To the best of our knowledge, not many works in the D2D communications have implemented the matching theory approach. In this chapter, we try to optimize the performance of D2D networks while satisfying the QoS requirements utilizing two matching algorithms. Moreover, we investigate the cheating issue in matching, which has never been discussed in previous wireless resource allocation works.

## 3.3 System Model and Problem Formulation

We consider the spectrum sharing between CUs and D2D users, as shown in Fig. 3.1, where $L$ D2D user pairs coexist with $N$ CUs. Each D2D pair tries to find a suitable CU to share its allocated licensed channel. Fig. 3.2 will be explained in Section 3.4. In this chapter, we assume the UL transmission resources of CUs can be shared with D2D users since the UL spectrum is typically lighter loaded than the DL spectrum. Thus, the interference caused by the spectrum sharing only affects the BS side. Both CUs and D2D pairs need to satisfy certain signal to interference noise ratio (SINR) requirements before they can set up the spectrum sharing between them. We represent the CUs' set as $\mathcal{C} = \{c_1, ...c_i, ..., c_N\}, 1 \leq i \leq N$ and the D2D users' set as $\mathcal{D} = \{d_1, ..., d_j, ..., d_L\}, 1 \leq j \leq L$. In this chapter, we assume that $L = N$ [1].

We consider both fast fading and slow fading, caused by the multipath propagation effect and the shadowing effect, respectively. Thus, the channel gain between $c_i$ and the BS can be expressed as $g_{i,B} = K\beta_{i,B}\zeta_{i,B}L_{i,B}^{-\alpha}$, where $K$ is a constant value that determines system parameter, $\beta_{i,B}$ is the fast fading gain, $\zeta_{i,B}$ is the slowing fading gain, and $\alpha$ is the path loss exponent. Similarly, we

[1] This assumption is based on the concern that in order to realize stable matching and cheating algorithm as introduced in Section 3.4 and 3.5, we need to have the same number of matching agents in both sides (otherwise we can not guarantee the results). The necessity of this assumption is also validated through simulation.

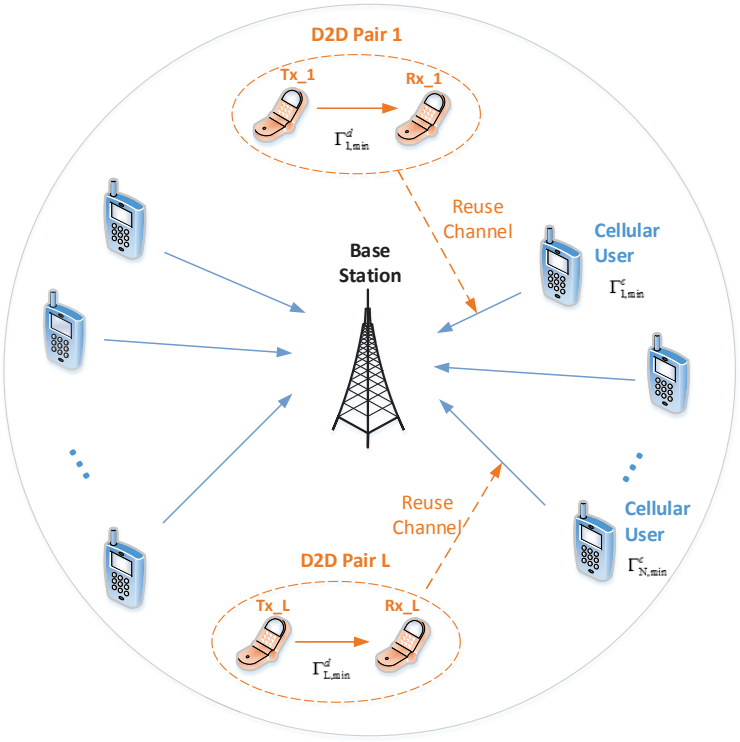



Figure 3.1: D2D communication underlying cellular network

define the channel gain between the D2D user pair $d_j$ as $g_j$, the channel gain for the interference link between $d_j$ and the BS as $h_{j,B}$, and the channel gain for the interference link between $c_i$ and $d_j$ as $h_{i,j}$. We use $L_{i,B}$ to represent the distance between $c_i$ and the BS.

To set up the spectrum sharing between any D2D pair and the CU, a minimum SINR requirement must be satisfied for either the D2D pair or the CU. Let $P_i^c$ and $P_j^d$ denote the transmission power of $c_i$ and $d_j$, and let $\Gamma_i^c$ and $\Gamma_j^d$ be the SINR of $c_i$ and $d_j$, respectively. The power of the

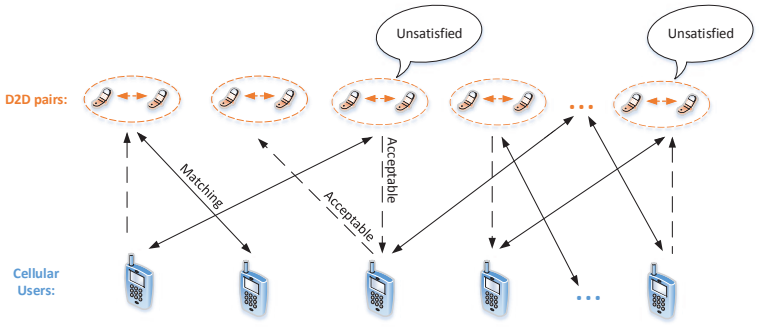



Figure 3.2: D2D matching system model

additive white Gaussian noise on each channel is assumed to be $\sigma^2$.

Then, we can formulate the maximum throughput problem in the D2D networks as follows:

$$\max_{\rho_{i,j},P_i^c,P_j^d} \sum_{c_i\in\mathcal{C}} \sum_{d_j\in\mathcal{D}} W_i[\log(1+\Gamma_i^c) + \rho_{i,j}\log(1+\Gamma_j^d)], \tag{3.1}$$

$$\text{subject to } \Gamma_i^c = \frac{P_i^c g_{i,B}}{\sigma^2 + \rho_{i,j} P_j^d h_{j,B}} \geq \Gamma_{i,\min}^c, \forall c_i \in \mathcal{C}, \tag{3.2}$$

$$\Gamma_j^d = \frac{P_j^d g_j}{\sigma^2 + \rho_{i,j} P_i^c h_{i,j}} \geq \Gamma_{j,\min}^d, \forall d_j \in \mathcal{D}, \tag{3.3}$$

$$\sum_{d_j\in\mathcal{S}} \rho_{i,j} \leq 1, \rho_{i,j} \in \{0,1\}, \forall c_i \in \mathcal{C}, \tag{3.4}$$

$$\sum_{c_i\in\mathcal{C}} \rho_{i,j} \leq 1, \rho_{i,j} \in \{0,1\}, \forall d_j \in \mathcal{D}, \tag{3.5}$$

$$P_i^c \leq P_{\text{max}}^c, \forall c_i \in \mathcal{C}, \text{and} \tag{3.6}$$

$$P_j^d \leq P_{\text{max}}^d, \forall d_j \in \mathcal{D}, \tag{3.7}$$

where $\rho_{i,j}$ is the binary resource indicator for $c_i$ and $d_j$. Here, $\rho_{i,j} = 1$ if $d_j$ reuses $c_i$'s channel $W_i$, and $\rho_{i,j} = 0$ otherwise. We assume that each CU $cu_i$ gets an equal share of spectrum $W_i$ from the BS. We use $\Gamma_{i,\text{min}}^c$ and $\Gamma_{j,\text{min}}^d$ to denote the minimum SINR requirements for $c_i$ and $d_j$, respectively. We use $P_{\text{max}}^c$ and $P_{\text{max}}^d$ to denote the maximum transmission power for $c_i$ and $d_j$, respectively. All the notation representations in this chapter are summarized in Table 3.1.

In order to optimize the system throughput while satisfying the QoS requirements, we try to find a proper CU for each D2D user and decide the optimal transmission powers for each sharing pair simultaneously. (3.1) represents the system objective, which aims to maximize the system throughput. (3.2) and (3.3) denote the SINR requirements for the CUs and D2D user, respectively. (3.4) and (3.5) indicate the capacity requriments for CUs and D2D users, respectively. (3.6) and (3.7) define the maximum transmission powers for the CUs and D2D users, respectively.

This optimization problem is an MINLP problem [53], which is generally NP-hard to solve.

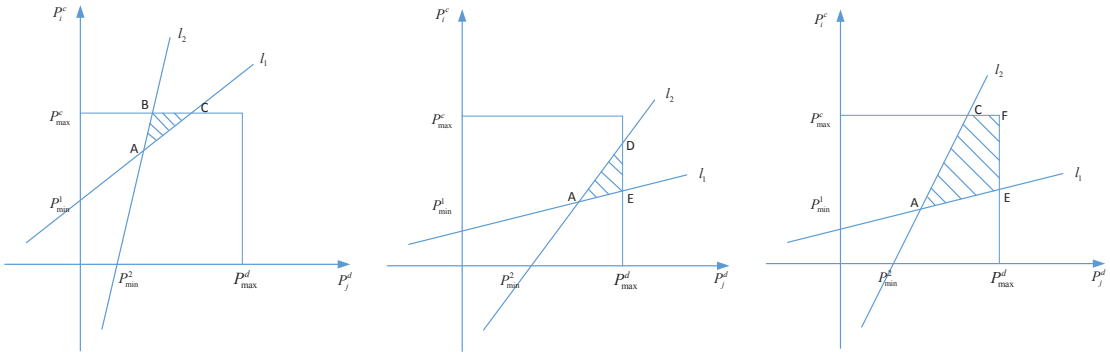
Table 3.1: Notation

| Symbol | Definition |
|---|---|
| $N$ | the number of CUs |
| $L$ | the number of D2D pairs |
| $\mathcal{C}$ | the set of CUs |
| $\mathcal{D}$ | the set of D2D pairs |
| $c_i$ | CU $c_i$ |
| $d_j$ | D2D pair $d_j$ |
| $\mathcal{S}_i^c$ | the set of D2D pairs that can be admitted by CU $c_i$ |
| $\mathcal{S}_j^d$ | the set of CUs that can be admitted by D2D pair $d_j$ |
| $\Gamma_i^c$ | SINR of $c_i$ |
| $\Gamma_j^d$ | SINR of $d_j$ |
| $\Gamma_{i,\text{min}}^c$ | minimum SINR requirement for $c_i$ |
| $\Gamma_{j,\text{min}}^d$ | minimum SINR requirement for $d_j$ |
| $P_i^c$ | transmission power of $c_i$ |
| $P_j^d$ | transmission power of $d_j$ |
| $P_{\text{max}}^c$ | maximum transmission power for $c_i$ |
| $P_{\text{max}}^d$ | maximum transmission power for $d_j$ |
| $g_{i,B}$ | channel gain between $c_i$ and BS |
| $g_j$ | channel gain between $d_j$ |
| $h_{j,B}$ | channel gain of interference link from $d_j$ to BS |
| $h_{i,j}$ | channel gain of interference link from $c_i$ to $d_j$ |
| $\rho_{i,j}$ | If $d_j$ share resource with $c_i$,then $\rho_{i,j} = 1$; otherwise $0$ |
| $\sigma^2$ | white Gaussian noise |

Thus, we try to solve the problem by dividing it into three subproblems in the next section.

## 3.4 Resource Allocation with True Preferences

We divide the resource allocation problem into three steps, i.e., the admission control, the optimal power allocation and the stable matching between admitted D2D users and CUs. The first two steps are presented in Section 3.4.1 and Section 3.4.2, respectively. In the third step, the GS algorithm is introduced to find a stable matching, and is discussed in Section 3.4.3. Moreover, we propose the minimum weight stable matching algorithm to pursue an optimal stable matching in Section 3.4.4.




(a) Admission area scenario 1 (b) Admission area scenario 2 (c) Admission area scenario 3

Figure 3.3: Admission area and power control illustration.

### 3.4.1 Admission Control

In this subsection, we will determine the acceptable pairs consisting of one CU and one D2D user pair. A set is called admissible only when both CU's and D2D user's transmission powers can be adjusted to satisfy the minimum SINR requirements. Thus, the admissible sets can be determined by the following equation,

$$\begin{cases} \Gamma_i^c = \frac{P_i^c g_{i,B}}{\sigma^2 + \rho_{i,j} P_j^d h_{j,B}} \geq \Gamma_{i,\min}^c, \\ \Gamma_j^d = \frac{P_j^d g_j}{\sigma^2 + \rho_{i,j} P_i^c h_{i,j}} \geq \Gamma_{j,\min}^d, \\ P_i^c \leq P_{\max}^c, \\ P_j^d \leq P_{\max}^d. \end{cases} \tag{3.8}$$

It's not hard to derive the above four linear relations between $c_i$'s transmission power $P_i^c$ and $d_j$'s transmission power $P_j^d$ from (3.8). A sharing pair is called admissible if and only if there exist $P_i^c$ and $P_j^d$ that can satisfy all the four linear relations. According to (3.8), we draw the three possible scenarios when a sharing pair consisting of $c_i$ and $d_j$ is admissible in Fig. 3.3, where $l_1 = \frac{\Gamma_{i,\min}^c h_{j,B}}{g_{i,B}}$, $l_2 = \frac{g_j}{\Gamma_{j,\min}^d h_{i,j}}$, $P_{\min}^1 = \frac{\sigma^2 \Gamma_{i,\min}^c}{g_{i,B}}$, and $P_{\min}^2 = \frac{\sigma^2 \Gamma_{j,\min}^d}{g_j}$. The shadow part $\mathcal{A}_{\text{admin}}$ in Fig. 3.3a, 3.3b and 3.3c are the transmission power pairs $(P_j^d, P_i^c)$ that can satisfy (3.8). If a sharing pair does not belong to any of the above scenarios (i.e., the shadow area is empty), then this pair is not admissible. We denote all the admissible D2D users for $c_i$ as $S_i^c$, and all the admitted CUs for $d_j$ as $S_j^d$.

### 3.4.2 Optimal Power Allocation

In the second step, the optimal transmission power for each admissible pair will be determined. The optimal power pair is represented as,

$$(P_i^{c*}, P_j^{d*}) = \underset{(P_i^c, P_j^d) \in \mathcal{A}_{\text{admin}}}{\arg\max} \; W_i[\log(1+\Gamma_i^c) + \log(1+\Gamma_j^d)], \tag{3.9}$$

where $\mathcal{A}_{\text{admin}}$ represents all the transmission power pairs that belong to the shadow area defined previously. We define $f(P_i^c, P_j^d) = W_i[\log(1+\Gamma_i^c) + \log(1+\Gamma_j^d)]$, and it's easy to derive that $f(\lambda P_i^c, \lambda P_j^d) > f(P_i^c, P_j^d)$ if $\lambda > 1$. Thus, we know that at least one transmission power in $(P_i^{c*} and P_j^{d*})$ is bounded by the peak value.

Now we are ready to find out optimal transmission powers for each adimissible sharing pair. In scenario one, as shown in Fig. 3.3a, in order to maximize $f(P_i^c, P_j^d)$, at least one user should transmit at its peak power. Thus, $c_i$ should transmit at $P_{\text{max}}^c$, while $d_j$ will reside on the segment $BC$. As proved in [54], $f(P_i^c, P_j^d)$ is a convex function over either $P_i^c$ or $P_j^d$, when the other value is fixed. Thus, $P_j^{d*}$ must be located on either point $B$ or $C$. Similar result holds for the scenario two, as shown in Fig. 3.3b, where $P_j^{d*}$ should be the peak transmission power, and $P_i^{c*}$ will locate on either point $D$ or $E$. Ad for the third scenario, as shown in Fig. 3.3c, the optimal power pair shall locate on segment $CF$ or segment $FE$.

### 3.4.3 Stable Matching by GS Algorithm

After we have identified all the admissible pairs and the optimal transmission powers, we need to find a proper CU for each D2D user pair. We apply the SM game to match the D2D users with the CUs [3], and the mapping is illustrated in Fig. 3.2.

The GS algorithm [28] is used to find a stable solution in the SM game. For an instance of the SM game, each man and each woman permute the opposite player set according to his/her preference. A stable matching is defined as a complete matching between men and women that admits

no BP. The basic story of the GS algorithm is that one gender makes a sequence of "proposals" to the other set. For example, each man proposes, in order, to the women in his preference list, and pauses when a woman agrees to consider his proposal, but continues if a proposal is rejected. When a woman receives a proposal, she rejects if she already holds a better proposal, and otherwise agrees to hold it for consideration. The progress ends until no man needs to propose. The computation complexity of the GS algorithm is $\mathcal{O}(m)$ [28], where $m$ is the number of acceptable man-woman pairs.

Similarly, D2D pairs and CUs can be regarded as men and women, respectively. The admissible pairs and the optimal transmission powers can be acquired after the admission control and power control. We use $c_i$'s throughput $W_i \log(1+\Gamma_i^c)$ when sharing spectrum with $d_j$ to denote $c_i$'s preference over $d_j$. Likewise, we use $d_j$'s throughout $W_i \log(1+\Gamma_j^d)$ to denote $d_j$'s preference over $c_i$. Thus, we define the "prefer" relation for $c_i$ between $d_j$ and $d_{j'}$ in Definition 3.1, and the "prefer" relation for $d_j$ between $c_i$ and $c_{i'}$ in Definition 3.2.

**Definition 3.1.** *$c_i$ prefers $d_j$ to $d_{j'}$, if $W_i \log(1+\Gamma_j^d) > W_i \log(1+\Gamma_{j'}^d)$, denoted by $d_j \succ_{c_i} d_{j'}$, for $c_i \in \mathcal{C}, d_j, d_{j'} \in \mathcal{S}_i^c, j \neq j'$.*

**Definition 3.2.** *$d_j$ prefers $c_i$ to $c_{i'}$, if $W_i \log(1+\Gamma_i^c) > W_i \log(1+\Gamma_{i'}^c)$, denoted by $c_i \succ_{d_j} c_{i'}$, for $d_j \in \mathcal{D}, c_i, c_{i'} \in \mathcal{S}_j^d, i \neq i'$.*

We denote the $rank(c_i, d_j)$ as the position of $d_j$ in $c_i$'s preference list $\mathcal{PL}_i^c$, and $rank(d_j, c_i)$ as the position of $c_i$ in $d_j$'s preference list $\mathcal{PL}_j^d$. For example, if $c_3$ is $d_6$'s second favourite choice, then $rank(d_6, c_3) = 2$. Now we define the stablity notation in Definition 3.3.

**Definition 3.3.** *A matching $M$ is stable, if there exists no blocking pair $(c_i, d_j)$, such that $d_j \succ_{c_i} M(c_i)$ and $c_i \succ_{d_j} M(d_j)$, where $M(c_i)$ represents CU $c_i$'s partner in $M$ and $M(d_j)$ represents $d_j$'s partner in $M$.*

Next, we present how to use the GS algorithm to find a stable matching between the D2D users and CUs in Algorithm 3.1.

**Algorithm 3.1 GS algorithm**

**Input:** D2D users' preference list $\mathcal{PL}^d$ and CUs' preference list $\mathcal{PL}^c$.
**Output:** Men-optimal stable matching $M$.
**Metode:**

Set up D2D pairs' preference lists as $\mathcal{PL}_j^d, \forall d_j \in \mathcal{S}$;
Set up CUs' preference lists as $\mathcal{PL}_i^c, \forall c_i \in \mathcal{C}$;
Set up a list of unmatched D2D users $\mathcal{UM} = \{d_j, \forall d_j \in \mathcal{S}\}$ ;
**while** $\mathcal{UM}$ is not empty **do**
    $d_j$ proposes to the CU that locates first in his list, $\forall d_j \in \mathcal{UM}$;
    **if** $c_i$ receives a proposal from $d_{j'}$, and $d_{j'}$ is more preferred than the current hold $d_j$ ($\forall d_j \in \mathcal{S}$ is considered more preferred by empty hold) **then**
        $c_i$ holds $d_{j'}$ and rejects $d_j$;
        $d_{j'}$ is removed from $\mathcal{UM}$ and $d_j$ is added into $\mathcal{UM}$;
    **else**
        CU rejects $d_{j'}$ and continues holding $d_j$;
    **end if**
**end while**
Output the matching $M$.

### 3.4.4 Stable Matching by Minimum Weight Algorithm

Gale and Shapley indicated in [28] that men and women whoever propose would be better off than being proposed to. Using Algorithm 3.1, we can have the men-optimal stable matching. On the other hand, if we let women propose, it would yield the women-optimal stable matching. Since different proposing methods can yield different stable matchings, which stable matching is "optimal" amongst all? There are typically several ways to define an "optimal" stable matching, such as the minimum regret stable matching [55], the egalitarian stable matching [56] [57], and minimum weight stable matching [56] [57]. The objective of this chapter is to maximize the social welfare. Thus, the minimum weight stable matching can be utilized to achieve our system objective. Notice here, it is possible that the minimum weight stable matching performance may be not so good as the Hungarian algorithm, w.r.t. the social welfare, since the Hungarian method doesn't ensure network stability. Next, we discuss how to deploy the minimum weight stable matching to achieve the optimal system throughput among all the possible stable matchings. The minimum weight stable matching is defined in Definition 3.4.

**Definition 3.4.** *We say that a stable matching $M$ is a minimum weight stable matching if it has the*

*minimum possible value of* $c(M)$*. The value* $c(M)$ *is given by*

$$c(M) = \sum_{c_i \in \mathcal{N}} wt(c_i, d_j) + \sum_{d_j \in \mathcal{L}} wt(d_j, c_i).$$

Here we define $wt(c_i, d_j) = rank(c_i, d_j)$, and $wt(d_j, c_i) = rank(d_j, c_i)$, representing the weights of $(c_i, d_j)$ and $(d_j, c_i)$, respectively. $rank(c_i, d_j)$ represents the ranking of CU $c_i$'s partner $d_j$ in $c_i$'s preference list, and $rank(d_j, c_i)$ represents the ranking of D2D user $d_j$'s partner $c_i$ in $d_j$'s preference list.

**Algorithm 3.2 Minimum Weight Stable Matching**

**Input:** D2D users' preference list $\mathcal{PL}^d$ and CUs' preference list $\mathcal{PL}^c$.
**Output:** Minimum Weight Stable Matching $M_{opt}$.
**Metode:**
1: Run the man-optimal GS algorithm with the true preference list, the output matching is $M_0$;
2: Find the men-oriented shortlist for the given problem;
3: According to the shortlist, find out all the rotations;
4: Construct a directed graph $P'$ presenting (in some way) the weighted rotation poset $P$;
5: Use the directed graph $P'$ to find the minimum weight closed subset $P$;
6: Eliminate the rotations in that closed subset to obtain the "optimal" stable matching $M_{opt}$;

In [56], Irving et al. derived a $\mathcal{O}(n^4)$ algorithm that outputs the minimum weight stable matching by exploiting the structure of the matching set that contains all the stable matchings. Leveraging this algorithm, we define the weight of a matching pair as the negative value of the throughput summation. Thus, the minimum weight stable matching can be used to achieve the maximum throughput. The basic idea of the minimum weight stable matching is to find all the rotations first, and then by eliminating these rotations, we are able to enumerate all the stable matchings. To reach the minimum weight stable matching, we search the closed subset of the rotation poset with the minimum weight, and eliminate this rotation poset from the existing men-optimal stable matching. A brief summary of the Irving's algorithm is presented in Algorithm 3.2. For further details please refer to [56][2].

In [33], Duan et al. adopted the so-called Hungarian algorithm, also known as the Kuhn Munkres algorithm. In an unweighted bipartite graph, the Hungarian algorithm can be used to find a maximum cardinality matching. While in a weighted situation, it can be used to find a maximum

[2]There is also another algorithm to find a minimum weight stable matching, which is described in [57].

weight matching in polynomial time ($O(n^3)$) [58]. The difference between the Hungarian method and the minimum weight stable matching is whether or not the network stability is guaranteed. In Section 3.6, the Hungarian algorithm will be used as a performance benchmark.

## 3.5 Cheating: Coalition Strategy

In this section, we discuss the strategic issue in matching, which means that some users can lie on their preference lists to get better partners. We introduce a cheating strategy to improve some D2D users' throughput. In addition, we endeavor to find a cabal (defined in Definition 3.5) that can benefit as many D2D users as possible, and thus further improving the throughput.

After we've reached a stable matching by the GS algorithm, some D2D pairs are not satisfied with their current partners (i.e., some D2D users are not matched to the first choices in their preferences). We allow these D2D users to pursue better partners by cheating. Here, the "cheating" action refers to the action of permutating some entries in the preference list or truncating the list [3]. In [59], Huang proposed a so-called coalition strategy (CS) in the SM game to allow some men to be better off by cheating. The general idea is presented as follows: 1) we construct a cabal consisting of men, within which each member prefers each other's partner (woman) to its own; 2) we find the accomplices for the cabal, who need to revise their preferences to assist the cabal members; 3) we run the GS algorithm with the falsified preferences. Thus, in the resulting matching, all men in cabal are strictly better off while the rest of men keep the same partners.

We denote $P_L(d)$ as the set of CUs, who are more preferred than $M(d)$ by the D2D pair $d$, and denote $P_R(d)$ as the set of CUs, who are less preferred than $M(d)$ by $d$. We let $M_0$ be the man-optimal stable matching.

**Definition 3.5.** *A cabal $\mathcal{K} = \{k_1, ..., k_m, ..., k_K\}$ is a subset of $\mathcal{D}$, such that for each $k_m, 1 \leq m \leq K$, we have $M(k_{m-1}) \succ_{k_m} M(k_m)$, $k_m \in \mathcal{D}$.*

**Definition 3.6.** *The accomplice set $\mathcal{H}(\mathcal{K})$ of the cabal $\mathcal{K}$ is a subset of $\mathcal{D}$, such that $h \in \mathcal{H}(\mathcal{K})$ if*

*1. $h \notin \mathcal{K}$, for any $k_m \in \mathcal{K}$, if $M(k_m) \succ_h M(h)$ and $h \succ_{M(k_m)} k_{m+1}$, or*

*2. $h \in \mathcal{K}$, and $h = k_l (k_l \in \mathcal{K})$, for any $k_m \in \mathcal{K}$, and $m \neq l$, if $M(k_m) \succ_{k_l} M(k_{l-1})$ and $k_l \succ_{M(k_m)} k_{m+1}$.*

Definition 3.5 defines the cabal consists of any D2D user $k_m \in \mathcal{K}$, who prefers $M(k_{m-1})$ to $M(k_m)$, where $M(k_m)$ is $k_m$'s current partner and $M(k_{m-1})$ is $k_m$'s desired partner. Definition 3.6 defines the subset of D2D users $\mathcal{H}(\mathcal{K})$ as the accomplices, who need to falsify their preferences to assist $\mathcal{K}$ to achieve their desired partners. Any D2D user $h$ outside the cabal $\mathcal{K}$, who would have prevented a cabal member $k_m$ from getting its desired partner, is defined as a accomplice of $\mathcal{K}$. We say $h$ prevents $k_m$ when $h$ prefers $M(k_m)$ to its own parter, while $M(k_m)$ prefers $h$ to $k_m$. Similarly, any D2D user $h$ within the cabal $\mathcal{K}$, represented by $k_l$, who would have prevented another cabal member $k_m$ from getting its desired partner, is also defined as an accomplice. We say $k_l$ prevents $k_m$ when $k_l$ prefers $M(k_m)$ to its desired partner $M(k_{l-1})$, while $M(k_m)$ prefers $k_l$ to $k_m$.

**Algorithm 3.3 Coalition Strategy**

**Input:** Men-optimal stable matching $M_0$.
**Output:** Men-optimal stable matching $M_s$ after cheating.
**Metode:**
1: Find the cabal $\mathcal{K}$ of $M_0$ as defined in Definition 3.5;
2: Find cabal $\mathcal{K}$'s accomplices $\mathcal{H}$ as defined in Definition 3.6;
3: **for all** D2D pair $d \in \mathcal{K}$ **do**
4: **if** $d \in \mathcal{H}(\mathcal{K}) - \mathcal{K}$ **then**
5: $d$ submits a preference list $(\pi_r(P_L(d) - X), M_0(d), \pi_r(P_R(d) + X))$, where $X = \{c | c = M_0(d_m) \in M_0(\mathcal{K}), d \succ_c d_{m+1}\}$;
6: **else**
7: $d = d_l$ submits a preference list $(\pi_r(P_L(d) - X), M_0(d_{l-1}), \pi_r(P_R(d) + X))$, where
8: $X = \{c | c = M_0(d_m) \in M_0(\mathcal{K}), c \succ_{d_l} d_{m-1}, d_l \succ_c d_{m+1}\}$;
9: **end if**
10: **end for**
11: Run the man-optimal GS algorithm the falsified preference list, and output matching is $M_s$.

Different from the Theorem 2 stated in [59], which doesn't specify how the unmatched users should perform, we have defined the actions for those unmatched users in Algorithm 3.3. We also find that, for the unmatched users within the cabal, their falsifying strategies should be different from those outside the cabal. Thus, we revised the CS algorithm proposed in [59] into our cheating strategy, which is stated in Algorithm 3.3.

In Algorithm 3.3, we use $\pi_r(P_L(d) - X)$ to denote a random permutation of $P_L(d) - X$, and

we use $\pi_r(P_R(d) + X)$ to denote a random permutation of $P_R(d) + X$.

In the resulting man-optimal matching $M_s$, we have $M_s(k_m) = M_0(k_{m-1})$ for $k_m \in \mathcal{K}$, and $M_s(k_m) = M_0(k_m)$ for $k_m \notin \mathcal{K}$. It means in the D2D-optimal stable matching after cheating, all the D2D users in the cabal have obtained their expected partners, and the rest of the D2D users have kept the same partners. The conclusion from [59] indicates that the CS algorithm is the only strategy that has the nice property of ensuring that some men are better off and the other men are at least as well off as before. This property is validated through our simulation.

Due to the NP-hardness of finding the largest cabal, which corresponds to finding the largest loop in the directed graph, we try to search a cabal as large as possible, so that more D2D users can be benefitted. In the random cabal search, we start from an arbitrary D2D user (whose current partner is not his first choice) and stop whenever a cycle is reached. In order to benefit more D2D users, we do the cabal search that starting from each possible D2D user so that we can find the cabal with a larger size than the random search. Later in Section 3.6, we will show whether a larger cabal can improve the D2D throughput or not through the simulation results.

## 3.6 Numerical Results

In this section, we consider a single cell network with the BS located at the cell center. We assume that the same number of CUs and D2D pairs (i.e., $N = L$) are uniformly distributed within the cell. The cell radius $R$ ranges from $350$ m to $650$ m. We assume the proximity $r$ between each D2D pair is randomly distributed within $(20, 40)$ m. The $5$ MHz UL bandwidth is equally shared within $N$ CUs. We assume Gaussian noise with power $-114$ dBm for all the licensed channels. The maximum transmission power is assumed to be identical for all the users as $24$ dBm. The SINR requirements for both D2D and cellular communications are uniformly distributed within $[20, 30]$ dB. As for the propagation gain, we set the pass loss constant $K$ as $10^{-2}$, the path loss exponent $\alpha$ as $4$, the multipath fading gain as the exponential distribution with unit mean, and the shadowing gain as the log-normal distribution with $8$ dB deviation [60].

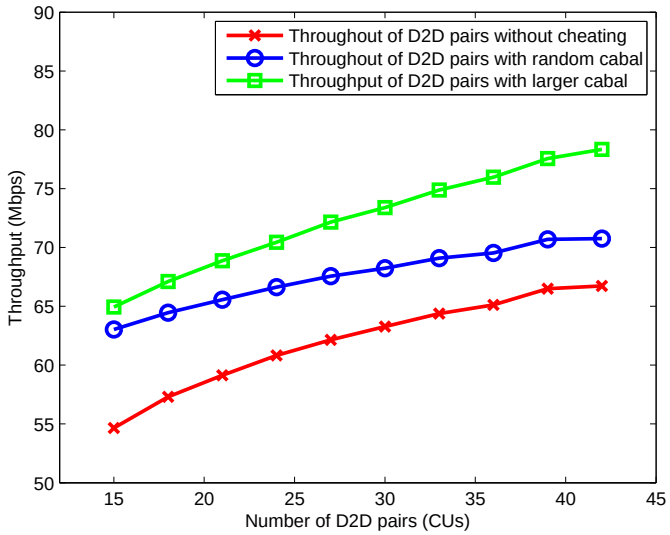



Figure 3.4: D2D users Throughput

In the evaluation, we first show how much the D2D users' and the system's throughput can be improved by cheating. Then by comparing the D2D users and the CUs' performances, we can show the advantage of proposing over being proposed to. Thirdly, we evaluate the D2D users' and CUs' performances, w.r.t. the SINR. Specifically, we show how each individual D2D user can be benefitted using the cheating algorithm. Finally, the probability of finding a cabal is provided.

In Fig. 3.4, we compare the throughput of D2D users under three methods, i.e., the GS algorithm, the CS algorithm with a random cabal and the CS algorithm with a bigger cabal. The D2D users' total throughput is improved after cheating as shown in the figure. The CS algorithm with a random cabal improves the performance for about $6.03\%$, while the CS algorithm with a bigger cabal can improve the performance for about $17.39\%$, compared the GS algorithm. It demonstrates that a cabal with larger size can yield better performance since more D2D users are benefitted.

In Fig. 3.5, we have evaluated the system throughput. The Hungarian algorithm [33] is used as a benchmark here. Although the Hungarian algorithm can achieve the highest system throughput, the resulted matching is not necessarily stable. In addition, we are not able to improve any user's satisfaction, since no preference is defined. The other three curves, i.e., the GS algorithm, the cheating method with a random cabal, and the cheating method with a larger cabal, achieve $92.51\%$,

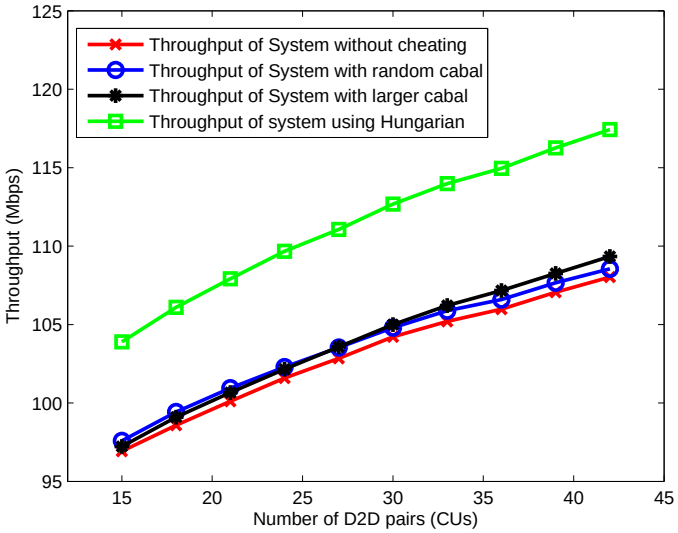



Figure 3.5: System Throughput

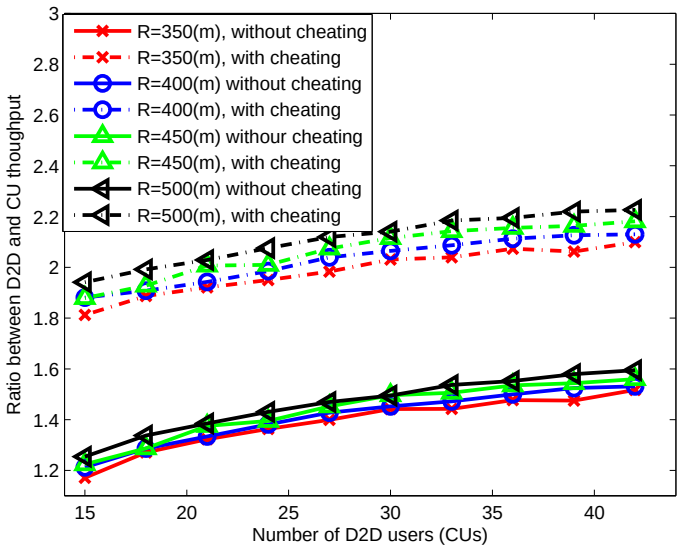



Figure 3.6: The ratio of D2D to CU throughput

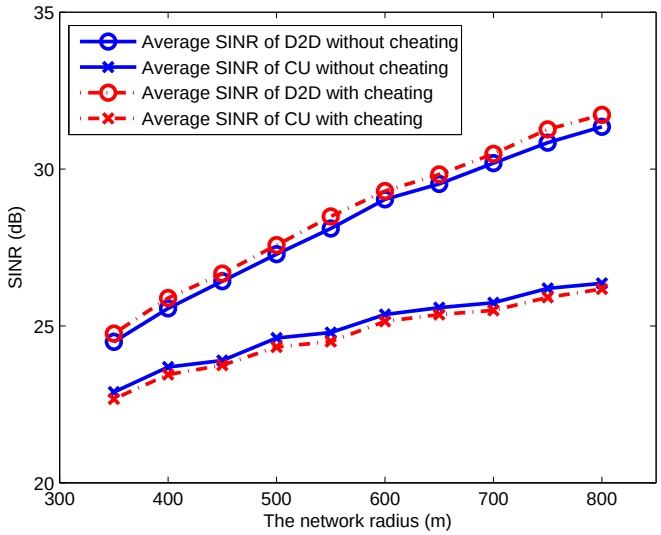
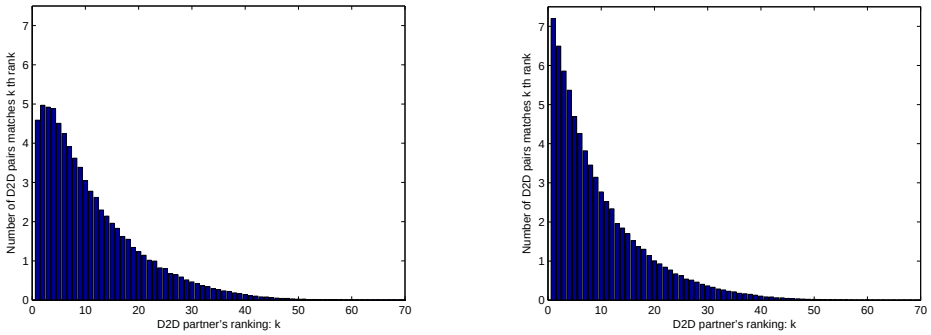



Figure 3.7: The average SINR of D2D and CU with and without cheating

$93.12\%$ and $93.24\%$ of the optimal solution, respectively. With such performance, we say that all the three algorithms are near-optimal while ensuring the system stability. In addition, the computation complexities of the GS and the CS algorithms are lower than the Hungarian method. For the two cheating strategies, we can also see from Fig. 3.5 that the CS algorithm not only benefits the D2D users but also improves the total system throughput. We can find from Fig. 3.5 that there's a cross point between the two cheating methods when the D2D user number reaches 27, and after this point, the cheating with a larger cabal outperforms the cheating with a random cabal. This is because when the D2D users get better partners, the CUs' partners get worse partners. It is difficult to tell who have more impact on the system throughput. Thus, when the cabal size gets larger, the system throughput not necessarily gets better, although the D2D performance is improved. However, when the system size becomes large enough (i.e., $N > 27$), finding a larger cabal can better improve the system throughput.

Fig. 3.6 illustrates the advantage of proposing than being proposed to by comparing the D2D users' and CUs' performances. For the case that everybody is telling the truth, which includes the bottom four curves, the ratio between D2D and CU throughput is between $1.1$ and $1.6$. It's reasonable since the men-optimal GS algorithm is performed. In the top four curves, which correspond to




(a) D2D partners' distribution without cheating. (b) D2D partners' distribution with cheating.

Figure 3.8: D2D partners' distribution with and without cheating.

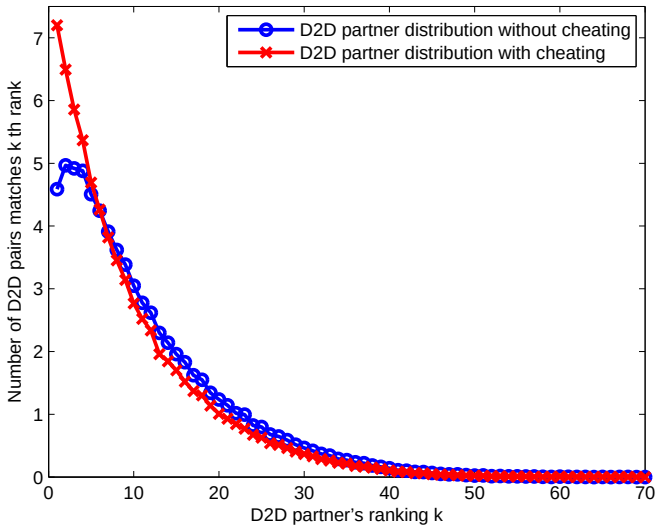



Figure 3.9: D2D partners' distribution comparison with and without cheating

the cheating case, this ratio is improved to almost 2. This results from the improvement of the D2D users' performance and the decrease of the CUs' performance due to cheating.

Fig. 3.7 compares the average SINR for D2D users and CUs with and without cheating. The network radius is increased from 350 m to 800 m by the step of 50 m. Both D2D users' and CUs' SINR values increase as the network radius increases. Apparently, D2D users' SINR values are improved, while CUs' SINRs are decreased after cheating. It's reasonable since the CS algorithm is designed to benefit the D2D users by sacrificing CUs' performance.

Fig. 3.8 shows how each D2D user's satisfaction (i.e., its partner's ranking) is improved by

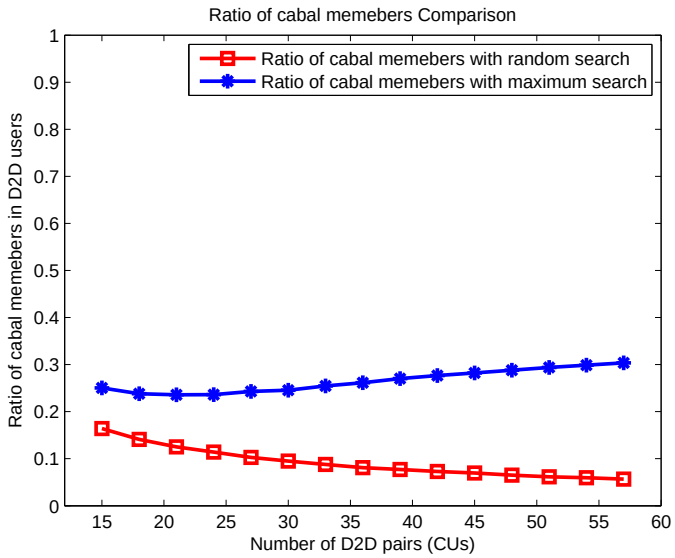



Figure 3.10: Ratio of D2D users that can improve performance

cheating. Fig. 3.8a and Fig. 3.8b provide the distributions of D2D users' partner rankings before and after cheating. If every user is honest, averagely $4.5$ users get their favorite partners, averagely $5$ users are matched to their second choices, and averagely $5$ users are matched to their third choices. Then in the case of cheating, we have more than 7 users matched to their first choices averagely, $6.5$ users matched to their second choices, and $6$ users matched to their third choices. In fact, more D2D users are matched to their top $5$ choices after cheating.

We evaluate the impact of different cabal sizes in Fig. 3.10. With the increase of user number (i.e., $L/N$), the ratio of the cabal members w.r.t. the total D2D users can achieve almost $30\%$ by using the cheating method with a larger cabal search. This value is about 3 times higher than the ratio achieved by the random search. The random cabal search stops as long as a cycle is detected no matter how large the cabal is. Thus, although the user number increases, the cabal size using the random search does not necessarily increase.

In Fig. 3.11, we approximate the probability of finding a cabal under four different radius values $R = 350$ m, $R = 450$ m, $R = 550$ m, and $R = 650$ m. Due to the high computational complexity ( $((L-1)!)^{(L-1)}$) of enumerating all the possible instances containing $L$ D2D users, we randomly generate $10000$ examples for each $L$, ranging from $15$ to $57$, to approximate the probabil-

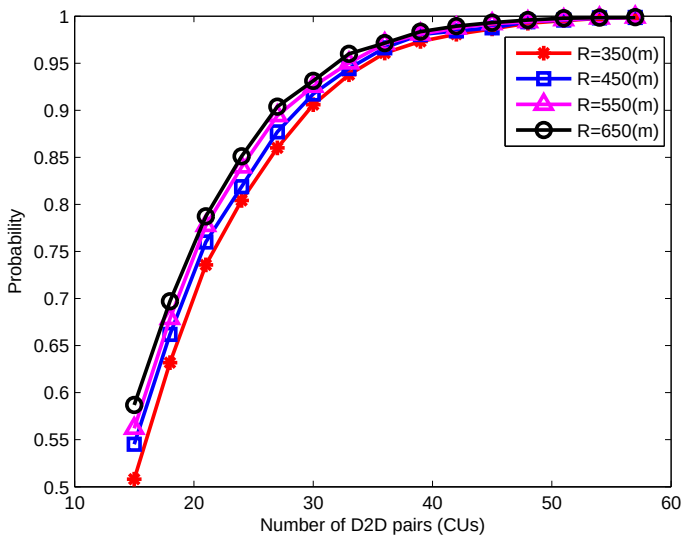



Figure 3.11: Probability of finding a cabal

ity. As can be seen from Fig. 3.11, with the increase of $L$, we have a higher probability of finding a cabal. For all the four network radiuses, when $L$ reaches $50$, the probabilities of finding a cabal can reach $100\%$.

## 3.7 Conclusion

In this chapter, we have implemented the SM game into the D2D communications to solve the resource allocation problem. The two stable matching algorithms, i.e., the GS and the minimum weight stable matching algorithms, are proposed and analyzed. The GS algorithm can achieve $92.51\%$ (under our simulation set-up) of the optimal system throughput by the Hungarian algorithm in polynomial time. In addition, the cheating mechanism, i.e., the CS algorithm, is utilized to further improve both the D2D users and the system performances. The simulation results also show that a larger cabal size can achieve even better performance.

# Chapter 4

# Student Admission Matching based Content-Cache Allocation

As a support to the backend storage, the content caching technique is of great importance to the online social networks (e.g., Facebook) in reducing the service latency and improving the user satisfaction. However, the limited caching capacity and booming user traffic have posed new challenges for the content caching allocation. In this chapter, we propose a three-layer caching model and investigate how to efficiently allocation the contents to different cache centers to minimize the overall service latency. We tackle this issue by utilizing both the centralized Mix Integer Linear Programming (MILP) optimization approach and the distributive Student Admission (SA) matching-based approach. In the SA model, we leverage the Resident-oriented Gale-Shapley (RGS) algorithm to yield a stable matching between the contents and the cache centers. We compare the performances of the centralized and distributed algorithms regarding both the system welfare and the computation complexity [61].

## 4.1 Introduction

Nowadays, with the emerging of mobile devices such as smartphones and tablets, more and more users are accessing the online social networks such as Facebook, Flickr and so on. As one of the largest online social networks, Facebook stores billions of photo contents. To deliver the contents to users efficiently, heterogeneous cache centers are used to support the Facebook Backend storage center. One important measurement to evaluate the user satisfaction is the response delay of the user request, which is highly depending on the data fetching paths, and thus relating to specific content caching allocation techniques. As a result, an appropriate content caching methodology can play a major role in improving the user satisfaction.

Before discussing any caching method, we first introduce the Facebook photo storage archi-

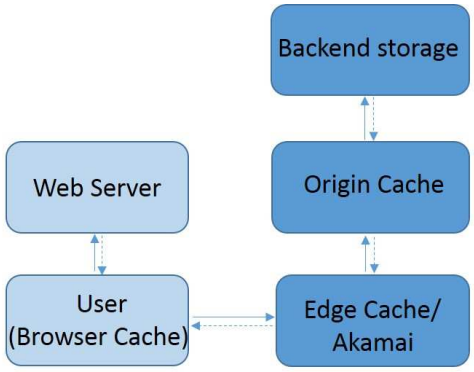



Figure 4.1: The caching system architecture (e.g., Facebook).

tecture, as shown in Fig. 4.1. There are typically three layers of cache centers in front of the backend storage, also called the Haystack storage. These three cache layers are the Browser cache, the Edge cache, and the Origin cache. The Haystack storage stores all the data [11], part of which will be cached to the one of the three-layer cache centers to reduce the service latency. The first layer, which is the closest to the end users, is the Browser cache. The Browser cache centers are typically embedded in the user equipment such as desktops and mobile phones. The second layer cache is called the Facebook Edge cache [12], and the third layer is called the Origin cache. When a user requests data from Facebook, it first looks up the content in the user's local browser cache. If the fetch is a miss, the browser sends an HTTP request to the Internet, and the Facebook web server calculates a photo fetching path, which directs the search process to the higher layers of cache. Then, if the search in the Edge cache fails again, it will proceed to the Origin cache. If a miss happens again, the last try would be the Backend storage, which guarantees a $100\%$ hit since it stores all the data [13].

Obviously, regarding the above data fetching procedure, the service latency increases as the fetch path goes to the higher layer cache centers. Besides, the service latency also varies for different cache centers in the same layer due to the geography diversity. Thus, as the preceding process of the photo fetching, the content caching mechanism must be well designed to improve the user satisfaction (i.e., to reduce the service latency). Some existing works have been proposed to efficiently store the contents to increase the hit ratio in different layers and minimize the service latency [62]. [63] introduces a domain name system that protects against the distributed denial of service attacks at-

tempting to overload the network to failure and the cache hacks.

In this chapter, we consder the Facebook photo storage system, and focus on the data caching mechanism to maximize the user satisfaction. Our main contributions are summarized as follows.

1) We propose a three-layer caching system, where photos can be cached to different cache centers to minimize the average service latency. Innovative metrics, such as the data's popularity and the cache centers' delay hierarchy, are considered during the optimization. We formulate this problem as an MILP problem, which can be solved by CPLEX [64].

2) We model the content caching problem as the SA matching game [3]. We treat the content and the cache center as two distinct matching parties. The preferences of both parties are built based on factors like popularity, locality and so on. We solve this matching problem by the proposed RGS algorithm, which is a distributive algorithm compared to the centralized MILP optimization.

3) Finally, we evaluate the proposed framework through simulation. We compare the performance of the centralized and distributed algorithms, as well as the random allocation mechanism. The computation complexity analysis is also provided.

The remainder of the chapter is organized as follows. In Section 4.2, we introduce the network model and the problem formulation. We discuss how to solve the problem in Section 4.3. Then, we conduct the performance evaluation in Section 4.4. Finally, conclusions are drawn in Section 4.5.

## 4.2 System Model and Problem Formulation

As indicated in [11], the Facebook web browser caches are co-located with the user devices. Currently, 9 Edge cache centers are located in San Jose, Palo Alto, LA, Dallas, Chicago, Atlanta, Miami, Washington D.C., and NYC. Besides, 4 Origin cache centers are located in Virginia, North Carolina, Oregon, and California. In this chapter, we consider a three-layer caching system, which includes the Edge cache, the Origin cache and the Backend storage. The reasons that we do not include the browser cache are stated as follows: 1) the Browser cache is dedicated for its co-located end user, so no matter whether a request hits or misses in its local browser cache, any other browser

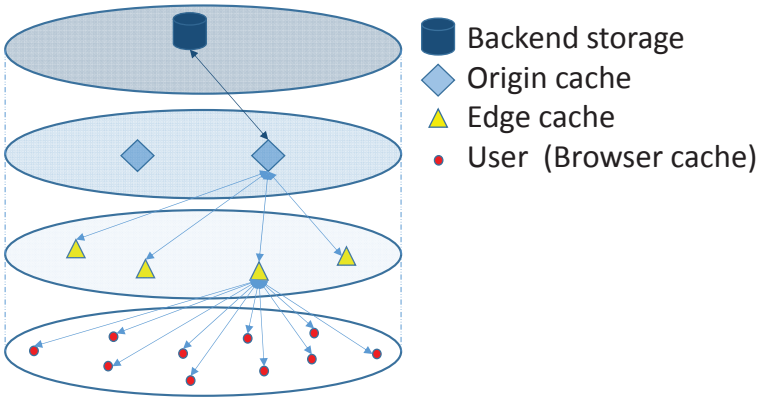



Figure 4.2: The system model.

cache could not be the candidate cache for this client; 2) the response delay of data fetching from the browser cache is almost ignorable compared to the other layer cache centers. Thus, we consider the content allocation within the Edge cache, the Origin cache and the Backend storage.

In this model, we assume our network as a circular area with the radius of $R$. We assume $K$ users $\mathcal{U} = \{u_1, ..., u_k, ..., u_K\}$ and $N$ cache centers $\mathcal{C} = \{c_1, ..., c_i, ..., c_N\}$ randomly located inside the circle, as shown in Fig. 4.2. The $N$ cache centers consist of $N_e$ Edge caches, $N_o$ Origin caches, and $N_s$ Backend storage, and thus, $N = N_e + N_o + N_s$. We denote the set of Edge caches by

$$\mathcal{C}_e = \{c_1^e, ..., c_i^e, ..., c_{N_e}^e\}, \quad 1 \leq i \leq N_e. \tag{4.1}$$

The set of Origin caches can be represented by

$$\mathcal{C}_o = \{c_1^o, ..., c_i^o, ..., c_{N_o}^o\}, \quad 1 \leq i \leq N_o. \tag{4.2}$$

The set of Backend storage can be represented by

$$\mathcal{C}_s = \{c_1^s, ..., c_i^s, ..., c_{N_s}^s\}, \quad 1 \leq i \leq N_s. \tag{4.3}$$

Without loss of generality, we assume that cache centers in the same layer have identical capacities, while the capacities of different layer cache centers are different, denoted as $q_e, q_o, q_s$ for the Edge cache, the Origin cache, and the Backend storage, respectively. The capacities of

the Edge cache, the Origin cache, and the Backend storage are defined as $q_e$, $q_o$ and $q_s$, respectively. For simplicity, we assume all the contents are of equal size $r$, and are denoted by $\mathcal{M} = \{m_1, ..., m_j, ..., m_M\}$, where $M$ is the number of contents.

**Definition 4.1.** Allocation Matrix $\mathbf{X}$*: An* $N \times M$ *matrix with the* $(i, j)$*th element* $x_{ij} \in \{0, 1\}$ *indicating the allocation of the content* $m_j$ *to the cache center* $c_i$*,* $\forall c_i \in \mathcal{N}$ *and* $\forall m_j \in \mathcal{M}$*. If* $x_{ij} = 1$*, the* $j$*th content is allocated to the* $i$*th cache center, and if* $x_{ij} = 0$*, otherwise.*

We assume that each content can be cached only once, thus we have

$$\sum_{i \in \mathcal{N}} x_{ij} \leq 1. \tag{4.4}$$

We assume the capacities of the cache centers as $Q$, $\quad Q(i) \in \{q_e, q_o, q_s\}$, $\forall c_i \in \mathcal{N}$. Each cache center should cache the amount of data no more than its capacity.

$$\sum_{m_j \in \mathcal{M}} r x_{ij} \leq Q(i), \forall c_i \in \mathcal{N}. \tag{4.5}$$

When determining the caching priority of different contents, we consider two factors: the content's popularity distribution and the cache center's service delay. We define them in the following two subsections.

### 4.2.1 Popularity

Intuitively, the popular contents are requested more frequently than those less popular contents. The work in [11] explores the geographical patterns in the data request flows and finds out that most of the users' traffic is served by the nearby cache centers. Thus, the popular contents should have higher priority to be cached to the centers that are in nearby locations to the users. On the other hand, as shown in [65, 66], the user online activity shows the homophily and locality effects, meaning people who are geographically close may have similar trends of accessing the contents. Thus, caching the content by its popularity regarding different districts can increase the hit probability.

A natural way to quantify the content popularity is by tracking the number of repeated requests for this content. Here we define the popularity matrix as follows.

**Definition 4.2.** Popularity Matrix **F***: A $K \times M$ matrix in which the $kj$th element $f_{kj}$ represents the number of requests for the content $m_j$ from the user $u_k$ during a certain period of time, $\forall u_k \in \mathcal{K}$ and $\forall m_j \in \mathcal{M}$.*

### 4.2.2 Delay

According to [11], when a user receives an HTML file from the Facebook web server, the fetching path is based on the URL information carried in the file. These URLs are generated by the web servers to control the traffic distribution across the serving stack. The routing policy is designed based on the joint consideration of the service latency, cache center capacity, Internet service providers (ISP) peering cost and so on. Unfortunately, to the best of our knowledge, there's no literature available online that shows how exactly the Facebook web server calculates the URLs. Thus, we make the following assumptions for our work: 1) the lower layer cache centers have higher caching priorities than higher layer, 2) for cache centers in the same layer, we assume the priority is geographically related. We denote the service delay for the Edge cache, the Origin cache, and the Backend storage as $t_e$, $t_o$, and $t_s$, respectively. Thus, we have the fetching delay inequality as follows

$$t_e < t_o < t_s. \tag{4.6}$$

Jointly considering the above two assumptions, we represent the general service delay as follows

$$t_{ki} = \begin{cases} t_e * \frac{d_{ki}}{2R}, & \text{if} \quad c_i \in \mathcal{C}_e, \\ t_e + t_o * \frac{d_{ki}}{2R}, & \text{if} \quad c_i \in \mathcal{C}_b, \\ t_e + t_o + t_s * \frac{d_{ki}}{2R}, & \text{if} \quad c_i \in \mathcal{C}_s, \end{cases} \tag{4.7}$$

where $d_{ki}$ is the distance between $u_k$ and $c_i$, and $R$ is the radius of our selected area. $\frac{d_{ki}}{2R}$ is a real number within $[0, 1]$. By utilizing $\frac{d_{ki}}{2R}$, we can add the geographical location into the delay

definition. On the other hand, by adding the lower-layer delay (i.e., $t_e$, $t_o$, $t_s$) to the current layer delay calculation, we guarantee that higher-layer cache centers have higher latencies. Thus, the delay matrix can be defined as follows.

**Definition 4.3.** Delay Matrix **T***: A $K \times N$ matrix in which the $kith$ element is the response delay $t_{ki}$ for client $u_k$ when fetching from cache center $c_i$, $\forall k \in \mathcal{K}$ and $\forall c_i \in \mathcal{N}$.*

### 4.2.3 Problem Formulation

Under the caching constraints discussed in Section 4.2, we try to minimize the average latency for the entire system. Since the allocation matrix **X** is the only variable matrix and is binary valued, we can formulate this content caching problem as an MILP optimization as follows.

$$\min_{X} : T \circ (F \times X'), \tag{4.8}$$

s.t.

$$\sum_{m_j \in \mathcal{M}} r x_{ij} \leq Q(i), \forall c_i \in \mathcal{C}, \tag{4.9}$$

$$\sum_{c_i \in \mathcal{N}} x_{ij} = 1, \forall m_j \in \mathcal{M}, \text{and} \tag{4.10}$$

$$x_{ij} = \{0, 1\}, \forall c_i \in \mathcal{C}, m_j \in \mathcal{M}, \tag{4.11}$$

where (4.8) is the objective function, presenting the overall response delay, defined as the Hadamard product [67] "$\circ$" of the corresponding response delay and the request times of the contents from users. The request times of the contents can be calculated by multiplying the popularity matrix $F$ with the transpose of the allocation matrix $X$. Constraint (4.9) defines the capacity of each cache center as $Q$. Constraint (4.10) indicates that each content can be only allocated to one cache center. Constraint (4.11) defines $x_{ij}$ as a binary variable, which represents the caching of a certain content to a certain cache center.

This MILP optimization can be solved by using the CPLEX [64] function for MATLAB. This function provides an extension to the IBM ILOG CPLEX Optimizers, and allows users to define

optimization problems and solve them with MATLAB. The centralized solution, will be used as the benchmark in Section 4.4.

## 4.3 SA Stable Matching Problem

The computational complexity of the optimization problem in (4.8) increases exponentially with the increase of the network size [53]. Thus, a low-complexity distributive solution is needed. In this section, we propose a matching-based distributive solution which can achieve the similar performance as the centralized optimization but with lower complexity. We introduce the SA game to model the many-to-one matching between the contents and cache centers.

In the SA game, the students apply to the colleges and the colleges decide whether to accept them or not. A student ranks all the colleges by order of his/her preferences over these colleges, which may depend on the college locations, or whether they offer a major that interests the student. On the other hand, after receiving applications from students, a college will rank the students who apply for it based on their scores or expertise in certain fields. Each college has a quota limiting the maximum number of students that it can recruit. Intuitively if a college receives applications more than its capacity, it chooses the most preferred ones up to the quota and rejects the rest of students. In this section, we introduce the SA model to formulate the content caching problem and leverage the RGS algorithm to solve it.

### 4.3.1 Preference list

As we have discussed previously, we can make use of the locality of the content's popularity to reduce the response delay. For each user, the preferences over different contents are different since people have various interests. Then taking the locality factor into consideration, the users are more likely to be served by the nearby cache centers. Thus, considering the user interests of $c_i$'s $K_{close}$ closest user set $\mathcal{U}_{close}(i)$, we calculate the average popularity of different contents among $\mathcal{U}_{close}(i)$, and define it as the preference of $c_i$ over these contents. It is represented as follows.

**Definition 4.4.** *For cache center $c_i$, $\forall c_i \in \mathcal{C}$, its preference over content $m_j$, $\forall m_j \in \mathcal{M}$ is*

$$PL_{cache}(i,j) = \frac{1}{K_{close}} \sum_{k \in \mathcal{U}_{close}(i)} f_{kj}. \tag{4.12}$$

By sorting each row of $N \times M$ matrix $PL_{cache}$ in a descending order, we can generate the preference lists for all the caches centers. On the other hand, to define contents' preferences, we use the average latencies of different cache centers. For the preference of content $m_j$ over cache center $c_i$, we consider the $K_{close}$ closest user set $\mathcal{U}_{close}(i)$ of $c_i$. By taking into consideration the popularity of $m_j$ over $\mathcal{U}_{close}(i)$, we can calculate the average service delay as $m_j$'s preference over $c_i$, which is represented as follows.

**Definition 4.5.** *For content $m_j$, $\forall m_j \in \mathcal{M}$, its preference over cache center $c_i$, $\forall c_i \in \mathcal{C}$ is*

$$PL_{content}(j,i) = \frac{1}{K_{close}} \sum_{k \in \mathcal{U}_{close}(i)} f_{kj} \times t_{ki}. \tag{4.13}$$

By sorting each row of $M \times N$ matrix $PL_{content}$ in an ascending order, we generate the preference lists for all the cache centers.

### 4.3.2 Proposed Matching Algorithm

In this subsection, we introduce the RGS algorithm to find the many-to-one stable matching solution [28]. In an SA instance consisting of $M$ students and $N$ colleges, the students keep proposing to the colleges, until all the students are accepted or all colleges have recruitted enough students. During the proposal, each student applies for his/her current favorite college, w.r.t. the student's preferences, and then removes this college from the preference list after proposing to it. Then for each iteration, after all the students have proposed, each college checks its received proposals, together with the students it has accepted in the previous iterations, and then keeps the most preferred students up to its quota and rejects the rest. The proposing and rejecting interaction continues until either all the student are accepted or all the colleges are full [28].

**Algorithm 4.1 RGS Algorithm for Cache-Content Allocation**

**Input:**$\mathcal{C}$, $\mathcal{M}$, $T$, $Q$, $F$, $r$
**Output:**$X$

```
Initialization;
Construct the preference list of cache centers PL_cache;
Construct the preference list of contents PL_content;
Construct the set of unmatched contents M_unmatch, set M_unmatch = M;
while M_unmatch ≠ ∅ do
    Contents propose to cache centers;
    for all m_j ∈ M_unmatch do
        Proposes to the first cache center c_i in its preference list PL_contents(j,:), set x_ij = 1;
        Remove c_i from PL_content(j,:);
    end for
    cache centers make decisions;
    for all c_i ∈ C do
        if Σ_{j∈M} r x_ij ≤ Q(i) then
            c_i keeps all of the proposed contents;
            Remove m_j from M_unmatch;
        else
            c_i keeps the most preferred Q(i) contents, and rejects the rest;
            Remove these Q(i) contents from the M_unmatch;
            Add the rejected contents into the M_unmatch, and set x_ij = 0;
        end if
    end for
end while
End of algorithm;
```

We model the content as the student and the cache center as the college. Firstly, the preference lists are set up using the preference values defined in (4.12) and (4.13). Secondly, the contents propose to their most favorite cache centers, and the cache centers, based on their preferences and capacities, decide whether to accept these applications or not. Finally, when all the contents are cached, the matching process terminates. The RGS algorithm is stated in Algorithm 4.1.

## 4.4 Numerical Results

Due to the lack of real data traces, we have made some assumptions to simplify the simulations. In our setting, we assume that there are $N_e = 10$ Edge caches, $N_o = 4$ Origin caches, and $N_s = 1$ Backend storage. Thus, the total number of cache centers is $N = N_1 + N_2 + N_3 = 15$. The capacities of the Edge cache, the Origin cache, and the Backend storage are assumed as $q_e = 0.1$ Gb, $q_o = 0.25$ Gb and $q_s = \infty$, respectively. The data size is assumed as $r = 50$ Mb. Thus, the quota for the Edge cache, Origin cache, and Backend storage are assumed as $\frac{q_e}{r} = 2, \frac{q_o}{r} = 5, \frac{q_s}{r} = \infty$, respectively. The delay parameters are set up as $t_e = 1$, $t_o = 10$, and $t_s = 20$. The total number of

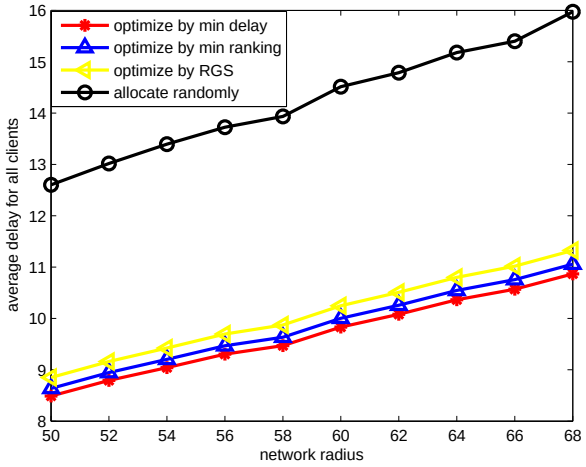



Figure 4.3: The response delay of the four mechanisms as the system scale varies.

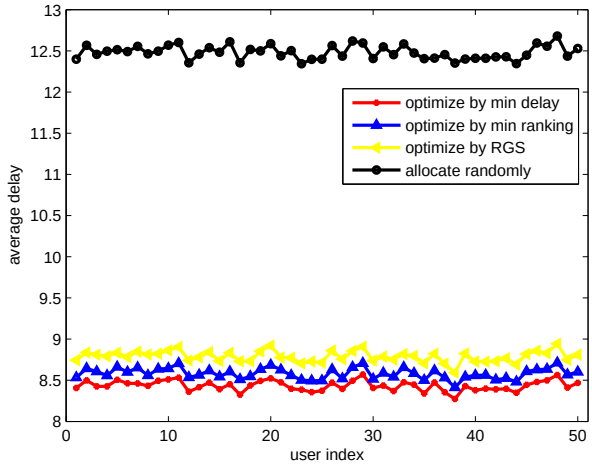
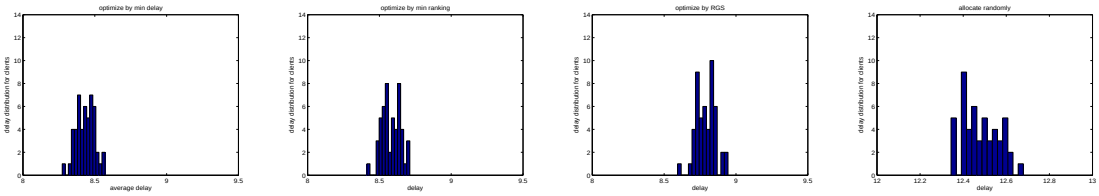



Figure 4.4: The response delay of the four mechanisms for all users.




(a) Min delay (b) Min ranking (c) RGS (d) Random

Figure 4.5: The average delay distribution at each user.

contents to be cached is $M = 70$, which slightly exceeds the total capacity of the Edge and Origin caches. We assume the content popularity distribution (the photo request frequency) is a random distribution within $[0, 10]$.

In the simulations, we propose another allocation mechanism, the min rank method, which is also solved by CPLEX. The optimization objective is to minimize the total popularity rankings. This mechanism is similar to a combination of the matching theory and the centralized optimization, which adopts the ranking definition in matching as the objective and solve the optimization using CPLEX. In addition, we introduce the random allocation as another benchmark, to compared with the three proposed mechanisms, i.e., the MILP with min delay, the MILP with min rank, and the RGS algorithm. In the random allocation, we assign the contents randomly to the cache centers in different layers while satisfying the capacity requirement.

Fig. 4.3 evaluates the average response delay for all users. Apparently, the proposed MILP with min delay method generates the smallest delay, followed by the MILP with min rank method. The difference between the two MILP curves shows that it's better to use the actual delay value instead of the ranking values when optimizing the response delay. For the other two curves, the RGS and the random allocation, the RGS curve achieves better performance than the random method. It is reasonable that the RGS method performs slightly worse than the two centralized mechanisms, since it is a distributive algorithm with much lower computation complexity. The complexity of the RGS algorithm is $\mathcal{O}(N \times M)$ [3], where $N \times M$ is the number of all possible (content, cache center) pairs. With the scaling of the system size, the distributive matching algorithm can be a good choice for reducing the computation complexity.

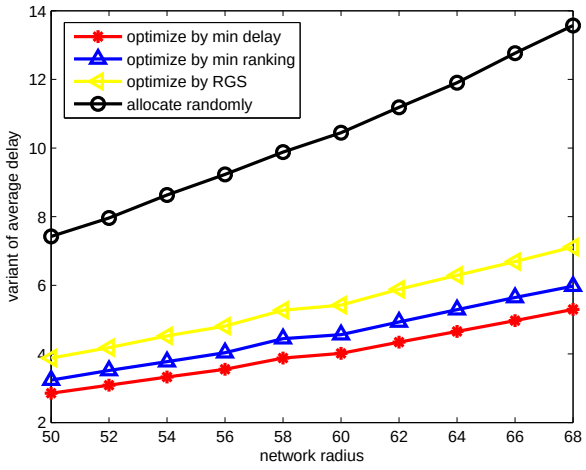



Figure 4.6: The response delay variants of the four mechanisms as the system scale varies.

In Fig. 4.4, we fix the network size and evaluate the delay distribution of all the users. We have run 1000 examples to obtain a relatively smooth and stable distribution. We can have similar conclusions in Fig. 4.4 as compared with Fig. 4.3. Fig. 4.5 is a another way to interpret Fig. 4.4, which is the histo graph for all clients' the delay distribution. Most users' latencies are distributed within $[8.5, 9.5]$ under the min delay, the min ranking, and the RGS methods, while the response delay using the random allocation is more than 12.

The variant of the delay distribution is evaluated in Fig. 4.6, which can represent the fairness between users. A small variant means a fair allocation. As can be seen from Fig. 4.6, the delay variants of the two centralized mechanisms are relatively smaller than the RGS and the random methods.

The hit ratio of each cache center is evaluated in Fig. 4.7, which represents the ratio of requests found in different cache layers. The way we cache the data determines the way it is fetched, and thus affecting the service latency. As shown in Fig. 4.7, the hit ratios in the lower layers using the three proposed mechanisms are higher than the random allocation, which indicates that more data are cached to the lower layers as we have expected. The RGS algorithm is achieving the same performance as two centralized mechanisms since we give priority to the lower layer cache centers

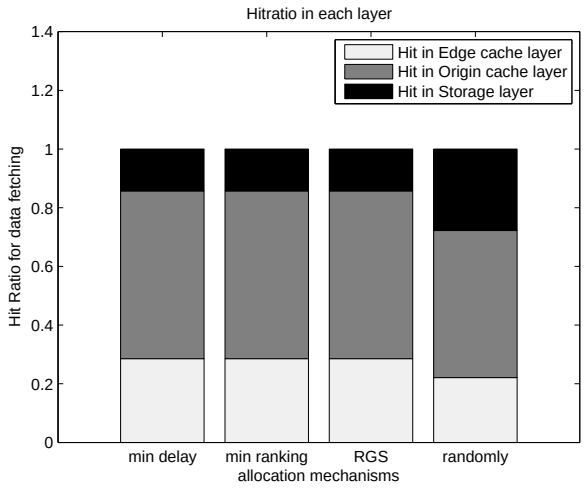



Figure 4.7: The hit ratio of the four mechanisms in three layer cache centers

when setting up the preference lists.

## 4.5 Conclusion

In this chapter, we have proposed an SA model to address the problem of caching online contents to different cache centers. We have considered the popularity and locality of the contents, together with the service delay of the cache centers, as the caching metrics. We have proposed two centralized optimization solutions as the benchmarks. Also, we have proposed a distributed algorithm, namely the RGS algorithm, to find the near-optimal performance while significantly reducing the computation complexity compared with the centralized mechanisms. Simulation results have demonstrated the effectiveness of the proposed mechanisms.

# Chapter 5

# Dynamic Path To Stability in LTE-Unlicensed with User Mobility: A Matching Framework

The LTE-Unlicensed, has recently captured intensive attention from both academic and industrial fields for its potential in boosting the LTE network capacity. By aggregating the unlicensed spectrum with the licensed spectrum, using the carrier aggregation technology, LTE-Unlicensed users can experience enhanced transmission, while maintaining the seamless mobility management and predictable performance. However, due to different transmission regulations, the coordination between the LTE and unlicensed systems (e.g., Wi-Fi) requires careful design. Especially, it's important to understand how to guarantee the transmission quality for the LTE users, and reduce Wi-Fi users' performance degradation, under the co-channel interference. In this chapter, we propose a matching theory framework to tackle this coexistence problem. Specifically, the interactions between the LTE and Wi-Fi users, are modeled using the SM game. The coexistence constraints are interpreted through the preference lists of both types of users. Two semi-distributed solutions, namely, the GS algorithm and the Random Path to Stability (RPTS) algorithm are proposed. In addition, the resource allocation problem is studied with network dynamics, and under which case, the RPTS method exhibits lower implementation cost compared with the GS algorithm. Moreover, to address the external effect, the Inter-Chanel Cooperation algorithm is proposed. Finally, the mechanisms are evaluated under two user mobility models: the Random Waypoint model and the HotSpot model.

## 5.1 Introduction

The ever-increasing mobile broadband traffic load has led to a pressing need for additional spectrum resources for the future 5G networks. To meet this demand, an intuitive idea is to exploit more licensed spectrum, which ensures the reliable and predictable performance. However, it is

not quite possible that sufficient additional licensed spectrum can be available in the near future. A growing interest in exploiting the unlicensed spectrum to boost the network capacity has recently arisen. Some cellular network operators have deployed the Wi-Fi access points to offload the cellular traffic to the unlicensed spectrum. However, such efforts are limited by some disadvantages such as the extra cost due to the investment in backhaul and core networks, degradation of the Wi-Fi performance, and lack of good coordination between the cellular and Wi-Fi systems. Another way to augment the LTE capacity to meet the traffic demands is to integrate the unlicensed carriers into the LTE system to enhance the cellular transmission using the carrier aggregation (CA) technology. The CA technology provides the option of aggregating two or more component carriers into a combined virtual bandwidth for enhanced transmission [14]. By aggregating the unlicensed spectrum into cellular networks with CA, the capacity of the LTE network can be boosted, while maintaining the seamless mobility management and predictable performance. This technology is commonly referred to as the LTE-Unlicensed [15].

### 5.1.1 LTE-Unlicensed Coexistence Issue

Recent studies have highlighted that the LTE technology has significant performance gains over the Wi-Fi when operating in the unlicensed band [68]. The main advantages of LTE-Unlicensed on the unlicensed spectrum include better link performance, seamless mobility management, and excellent coverage. These benefits have made LTE-Unlicensed a promising technology. Due to the low power transmission regulation imposed by the Federal Communications Commission (FCC) on the unlicensed spectrum, small cell (SC) deployment is an ideal implementation scenario for the LTE-Unlicensed. It is shown in [69] that the LTE-Unlicensed has a great potential in the ultra-dense cloud SC deployment, which combines the advantages of cloud radio access network and ultra-dense small cells. However, the LTE-Unlicensed technology is still in its infancy and requires much effort and careful design before it can fully meet the market requirements. More specifically, in this chapter, we want to study how to guarantee a fair coexistence of the newly joined cellular users (CUs) and the existing unlicensed users (UUs) on the unlicensed band. The traditional Wi-Fi transmission is colli-

sion avoidance based, so the UUs may back off when the interference caused by the co-channel CUs is higher than the energy detection threshold (e.g., -62dBm over 20MHz) [68]. On the other hand, the interference from the co-channel Wi-Fi users may also degrade the LTE-Unlicensed devices' performance, leading to the failure of meeting the QoS requirements of the cellular transmissions. In addition, with limited unlicensed bands, CUs need to compete with each other for the same unlicensed band. Thus, there may also exist interference between the co-channel CUs. Therefore, it is critical to design a coexistence mechanism to avoid such co-channel interferences and guarantee the harmonious coexistence of the Wi-Fi and LTE systems [70].

The existing projects on the LTE-Unlicensed come in multiple forms, the Licensed Assisted Access (LAA), the LTE-U and the MuLTEfire [71]. The LTE-U targets on the markets without the listen-before-talk (LBT) regulation on the unlicensed spectrum, while for markets with the LBT regulation, the LAA paradigm is specified. For both LTE-U and LAA, the signaling and control messages are sent through the reliable licensed anchor, and the unlicensed link is used only for data. The MuLTEfire broadens the LTE ecosystem to new deployment opportunities by operating solely in the unlicensed spectrum without a licensed anchor channel. For markets without the LBT regulations, the coexistence mechanism can be realized through careful software design and allows the fast-time-to-market launch. On the other hand, for markets with the LBT regulation, a number of modifications are needed to meet the channel occupancy requirements on the UL and DL transmissions [71].

A fair coexistence is always evaluated from both the CUs' and the existing Wi-Fi users' sides, and thus the coexisting interference can be summarized into the following three categories: (1) the interference that CUs bring to the existing UUs; (2) the interference that the existing UUs bring to CUs; and (3) the interference between multiple co-channel CUs. Therefore, to satisfy these coexisting constraints, certain transmission restrictions should be imposed on both the LTE and Wi-Fi systems. Some methods have been proposed to deal with the coexistence issues, for example, the Channel Selection mechanism, the Carrier-Sensing Adaptive Transmission (CSAT) and the Opportunistic SDL [71]. The Channel Selection method enables the SCs to select the cleanest channel

based on the channel measurements. When no clean channel is available, the CSAT algorithm can be used to apply adaptive TDM transmission based on the long-term carrier sensing of Wi-Fi activities. The SDL method allows to turn off the CA when the small cell is lightly loaded to avoid interference and transmission overheads. It is pointed out that, for most Wi-Fi and LTE-Unlicensed SC deployments, Channel Selection is usually sufficient to meet the coexistence requirements [71]. In a case that one unlicensed band is the cleanest choice for multiple CUs, instead of allocating them to the same unlicensed band, some of the CUs can be allocated to their second-best or third-best choices for improving the network utilization. Thus, it becomes a critical issue to find an efficient resource allocation method in the unlicensed bands so that we can not only achieve high network utilization but also guarantee both the CUs' and Wi-Fi users' performances.

### 5.1.2 Matching Theory for LTE-Unlicensed

The future 5G mobile networks are expected to be characterized by features such as higher data rates, reduced end-to-end latency, better network coverage and so on. The heterogeneous characteristics exhibited by the mobile users and the network density are the two major challenges in the 5G design. Current architectures for the cellular networks are highly centralized. The advantage of the centralized approach resides in its optimality, however with the huge amount of information to be collected by the centralized agent (e.g., eNBs) and the high computation complexity, the service latency at the user end can be very high. In addition, considering the network dynamics, such as the network topology change, channel condition change and so on, the distributive resource management approaches are considered to be more efficient.

Matching game, as a Nobel-prize winning framework, can overcome some limitations of game theory and the centralized optimization. It can model the competition and negotiation between the distinct user sets of LTE and Wi-Fi, and solve the resource allocation problem in a semi-distributive way. We claim it as a semi-distributive framework w.r.t. the fact that many operations in the matching algorithms are implemented distributively, including the information collection, preference list set-up, reject/accept decision making and so on, while some other operations may require global in-

formation from a centralized agent, such as the detection of a BP. Different from the static resource allocation studied in [72] [73], which is a one-time allocation, the dynamic case is not a simple repeat of the static allocation over time. In this chapter, we propose a matching-based framework to tackle the dynamic LTE-Unlicensed resource allocation problem. The major contributions are summarized as follows.

1) We have summarized the coexistence issues in the LTE-Unlicensed into three categories. To solve these issues we model the interactions between the CUs and UUs as the SM problem. The coexistence constraints are interpreted through the set-up of both CUs' and UUs' preference lists.

2) We have introduced two semi-distributed solutions: the GS algorithm and the RPTS algorithm to tackle the resource allocations dynamically. Both mechanisms guarantee network stability, while achieving relatively low computation complexity compared with the centralized optimization. Especially, the proposed RPTS algorithm further reduces the complexity compared with the GS algorithm, and is more suitable for dynamic networks.

3) The external effect, which refers to instability caused by the inter-dependence of the matching players' preference lists, is addressed by the proposed ICC mechanism. The ICC procedure not only re-stabilize the system but also further improves the network throughput.

4) We evaluate the adaptability and robustness of the GS-ICC and RPTS-ICC mechanisms under two user mobility models: the Random Waypoint model, and the HotSpot model. The computation complexity and system optimality analysis are performed both theoretically and through simulations.

The rest of the chapter is organized as follows. The related works of the LTE-Unlicensed are discussed in Section 5.2. The system model is provided in Section 5.3. Then, the problem formulation and the centralized solution are presented in Section 5.4. Due to the NP-hardness of the centralized solution, two semi-distributive matching approaches are introduced in Section 5.5. Both theoretical and simulation analysis are provided in Section 5.6 to evaluate the proposed mechanisms. Finally, conclusion remarks are drawn in Section 5.7.

## 5.2 Related Work

The performance evaluation of the LTE-Unlicensed has been conducted in recent studies. For example, [74] presents a system performance analysis, where the LTE and WLAN share the unlicensed resource using a simple fractional bandwidth sharing mechanism. The simulation results show that the coexistence has a negative impact on the WLAN system performance if without restrictions on LTE transmission. However, the severity of the impact can be controlled by restricting the LTE activities. The results also suggest the silent time of the WLAN, which is the period that the medium is idle when the WLAN users back off, can be exploited by the LTE users such that WLAN performance would not be degraded. Similar evaluations are conducted in [75], which again observes about $70\%$ to $100\%$ performance degradation of the Wi-Fi users if there is no inter-system coordination.

Efforts have been devoted to tackle the coexistence issues in the LTE-Unlicensed. To alleviate the coexistence interference, some techniques have been proposed, such as the channel selection, the transmission power control, the blank subframe and so on [76]. An intuitive way to prevent LTE/Wi-Fi users from accessing the channel at the same time is to use the blank subframe method, the idea of which is similar to the LTE almost blank subframe technique proposed in 3GPP Rel. 10. By silencing some of the subframes in the LTE UL/DL transmission, Wi-Fi users can access the channel during the blank subframe to increase the throughput [77]. A similar idea is proposed in [78]. Alternatively, LTE users can adopt the transmit power control to enable the LTE/Wi-Fi coexistence [79]. By measuring the interference at the LTE eNBs, LTE users estimate the presence and proximity of Wi-Fi users, and adjust their transmission powers to avoid strong interference to Wi-Fi. The idea of either blank subframe or power control can enable the coexistence of LTE/Wi-Fi, however, it more or less affects the transmission quality of the LTE users. Another enabler is the channel selection technique that can be used by both the Wi-Fi and LTE users [76]. For example, some Wi-Fi APs implement the least congested channel search (LCCS) to find the least congested channel. Meanwhile, except the fixed bandwidth allocation, the adaptive bandwidth channel allocation can

also be utilized in the LTE-Unlicensed environment.

There are some existing works on the resource allocation problem in the LTE-Unlicensed. For example, in [80], a joint user association and unlicensed resource allocation problem is proposed. The performance is measured by the average packet sojourn time. This work is solved by a centralized optimization approach. Some other works have been proposed by using the cooperative/noncooperative games. For example in [72], a coordinated hierarchical game is proposed for modeling the multi-operator spectrum sharing in the LTE-Unlicensed. The Kalai-Smorodinsky bargaining game is proposed to model the interactions among operators, and the Stackelberg game is proposed to model the interactions between operators and users. The final equilibrium is achieved by the price negotiation between operators and the transmission power control at the users. However, this work does not consider the interference caused by the unlicensed users. In addition, the assumption that each unlicensed subband is allocated to only one CU has limited its practical implementation. An interesting idea of leveraging the LTE-Unlicensed to transfer the Wi-Fi users to the LTE-Unlicensed system, while offering the unlicensed bands for compensation, is proposed in [81]. They developed a Nash bargaining solution (NBS) method to find the close-form expression for the unlicensed time slot allocation and the optimal number of transferred users. A matching based approach that addresses the LTE-Unlicensed coexistence issue has been discussed in [73]. The SPA matching game is utilized to model the interactions between the LTE and Wi-Fi users. The interference between LTE and Wi-Fi users can be avoided by generating the preference lists for both types of users, while the interference among co-channel LTE users can be avoided by utilizing the TDMA method. However, this work only considers the static resource allocation, and the dynamic resource management issue in the LTE-Unlicensed remains unexplored. A dynamic unlicensed resource sharing problem among multiple operators with time-varying traffic has been proposed in [82]. By modeling it as the repeated game, operators change the power spectral density to optimize the utilities in different time slots. Again, this dynamic game fails to consider the interference from the Wi-Fi system to the LTE system. Besides, only the DL transmission is discussed in this work.

The above-mentioned works and other existing works on the LTE-Unlicensed either address

only part of the coexistence issue, or fail to consider the network dynamic management. To the best of our knowledge, our work is the first that addresses the dynamic coexistence management problem in the LTE-Unlicensed, with joint consideration of different types of coexisting interference.

## 5.3 System Model

We consider a single carrier cellular network consisting of a set of CUs $\mathcal{CU} = \{cu_1, ..., cu_i, ..., cu_N\}$, as illustrated in Fig. 5.1. Each CU is served by its local eNB $\mathcal{BS} = \{bs_1, ..., bs_b, ..., bs_{B_1}\}$ with the allocated licensed spectrum. $B_1$ is the number of eNBs. Due to the time-varying traffic flow, some transmission requests can not be satisfied by the currently allocated licensed bands. We assume a set of such CUs, traveling around in the network with certain mobility patterns. Wherever the CUs are located, they search for nearby UUs, and seek to share their unlicensed spectrum using the CA technique for the supplemental downlink (SDL) transmission. The pre-assigned licensed bands of CUs will be the primary carrier and will be aggregated with the shared unlicensed bands to enhance transmission. To access a clean unlicensed channel, CUs need to have the channel sensing phase before joining any unlicensed channel, and this channel sensing shall be repeated each time they join the new unlicensed channels. During the channel sensing, CUs can detect the transmission energy on the target unlicensed channel and decide if this channel is clean or not. The CUs then communicate with the local eNBs, who assist the CUs in accessing the unlicensed bands, through control signal exchanges using the pre-assigned licensed bands. On the other hand, to model the interference incurred at the UUs from the co-channel CUs, the Wi-Fi medium utilization (MU) estimation should be performed. We denote the set of UUs as $\mathcal{UU} = \{uu_1, ..., uu_j, ..., uu_M\}$, and each UU is allocated with a specific unlicensed subband denoted as $\mathcal{F} = \{f_1, ..., f_j, ..., f_K\}$ for transmission. Typically, each unlicensed band is shared by multiple UUs according to the CSMA/CA regulation. Thus to simplify the representation, we assume that $uu_j, uu_j \in \mathcal{UU}$ is assigned with the unlicensed band $f_k, f_k \in \mathcal{F}$. Each UU is served by its local Wi-Fi AP, denoted as $\mathcal{AP} = \{ap_1, ..., ap_j, ..., ap_{B_2}\}$, for transmitting/receiving data, where $B_2$ is the number of Wi-Fi APs. The Wi-Fi MU estimation is conducted by the Wi-Fi APs through network listening, where all the CUs are required to turn off

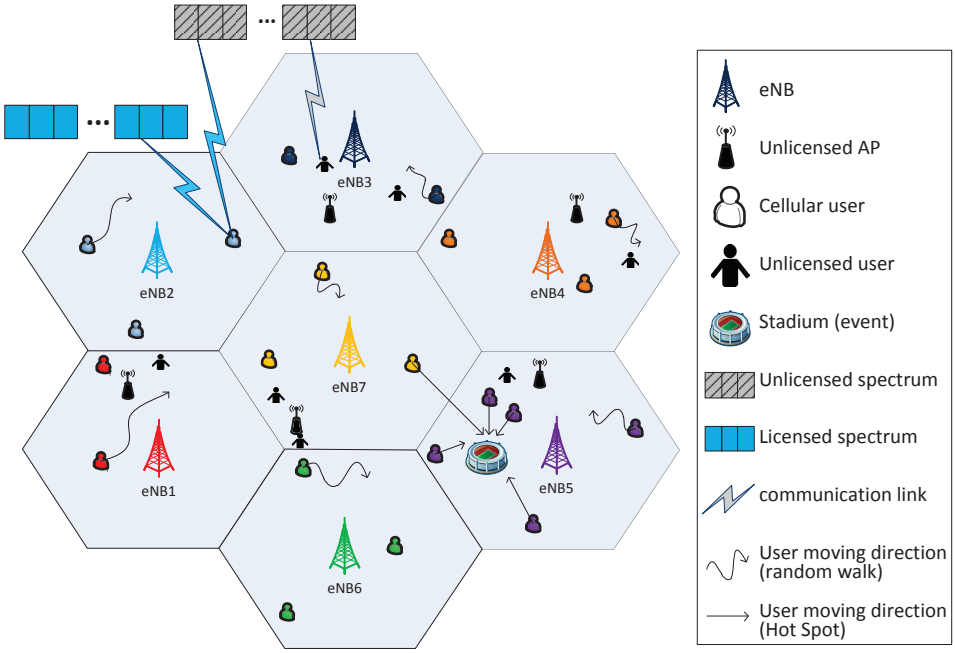



Figure 5.1: System Model

the unlicensed spectrum sharing in this period. The Wi-Fi network listening decodes the preamble of any WiFi packet detected during this time and records its corresponding received signal strength indicator (RSSI), duration, modulation, coding scheme and source/destination address [83]. With the above-estimated information, the Wi-Fi APs will share with the LTE-Unlicensed eNBs so that this information can be further shared with the CUs. To the best of our knowledge, there's no existing standard specifying how many unlicensed bands that each CU should use for aggregation in the LTE-Unlicensed, besides, SDL is only considered as an enhancement to the cellular transmission. Thus, without loss of generality, we assume that each CU will be matched to at most one UU, i.e., one unlicensed band. On the other hand, each unlicensed band can accommodate multiple CUs, depending on the number of existing UUs.

As discussed in Section 5.1, the coexistence issues are categorized as follows: (1) the interference that CUs bring to the existing UUs; (2) the interference that the existing UUs bring to CUs; (3) the interference between multiple co-channel CUs. We elaborate them one by one into the following constraints:

- It is well known that in Wi-Fi transmission, the UUs adopt the CSMA/CA mechanism for

coexistence, which is different from the way that LTE system operates. Thus, it is required that CUs should keep their interference to the UUs to be small enough, so that the channel is treated as "idle" by the UUs. To achieve this requirement, we set the threshold as the energy level of UU's normal channel noise, denoted as $\sigma_{\text{noise}}$.

- On the other hand, not all the unlicensed bands are clean enough for CUs to use. The existing UU can cause high interference that significantly reduces the transmission quality rather than enhancing the transmission. Thus, by restricting the SINR for $cu_i$ to be higher than the minimum requirement $\Gamma_i^{min}$, we can guarantee CUs' QoS requirements.

- The inter-CU interference can be avoided by the management at the eNBs. We assume that eNBs adopt the TDMA for the co-channel CUs, and each sharing CU is allocated an equal share of time. With more CUs assigned to the same unlicensed band, each CU gets a smaller share of the resource. Thus, it may happen that some CUs may prefer to switch to other channels which are allocated with fewer CUs. To avoid such situation, we propose the ICC strategy. The detailed mechanism discussion is provided in Section 5.5.2.2.

## 5.4 Problem Formulation

There are mainly two factors that may cause the network dynamics: the user mobility and the channel fading. To model the network dynamics, we divide the simulation period $[0, T]$ into identical time slots $\Delta T$. The slot duration $\Delta T$ can be set according to specific applications. To precisely model the dynamic network due to user mobility, we can set $\Delta T$ to be sufficiently small that during each time slot $(t, t+1), \forall t \in \{1, ..., t, ..., T\}$, the user distribution and channel conditions can be treated as static.

In order to pursue higher spectrum efficiency, we allow multiple CUs to share the same unlicensed channel as long as the coexisting interference is acceptable for all co-channel CUs and UUs. Each CU is allowed to be allocated to no more than one unlicensed channel. We use the binary matrix, denoted as $\rho(t) = \{\rho_{i,j} | cu_i \in \mathcal{CU}, uu_j \in \mathcal{UU}\}$, to model the spectrum sharing between CUs

and UUs. $\rho_{i,j}(t)$ is a binary value equal to 1 or 0 indicating if $cu_i$ is or is not assigned with $uu_j$ at time $t$. To dynamically maximize the social welfare, we endeavor to find the optimal allocation matrix $\rho(t)$ at each time slot that can achieve the highest overall performance of CUs and UUs.

### 5.4.1 CUs' Performance

We assume that CUs use the unlicensed spectrum for the SDL transmission. Thus, $cu_i$ is the receiver and its local eNB $bs_b$ is the transmitter. The interference from from the coexisting UU is also incurred at the receiver $cu_i$. The received SINR at $bs_b$ when sharing $f_j$ with $uu_j$ at time $t$, is represented as

$$\Gamma_{i,j}(t) = \frac{\rho_{i,j}(t)P_{b,i}(t)g_{b,i}(t)}{\sigma_N^l + P_{j,i}(t)h_{j,i}(t)}, \tag{5.1}$$

where $P_{b,i}(t)$ and $g_{b,i}(t)$ are the transmission power and the channel gain from $bs_b$ to $cu_i$ at time $t$, respectively. $P_{j,i}(t)$ and $h_{j,i}(t)$ represent the transmission power and the channel gain from $uu_j$ to $cu_i$, respectively. $\sigma_N^l$ is the licensed channel noise.

### 5.4.2 UUs' Performance

On the other hand, when $f_j$ is utilized by $cu_i$, both $uu_j$ and $ap_j$ can be interfered by the transmission power from $cu_i$ depending on whether $uu_j$ is transmitting to or receiving from $ap_j$. If $uu_j$ is the receiver, the interference from $cu_i$ at time $t$ is denoted as

$$\text{Intf}_{i,j}^{DL}(t) = P_{i,j}(t)h_{i,j}(t), \tag{5.2}$$

where $P_{i,j}(t)$ and $h_{i,j}(t)$ represent the transmission power and channel gain from $cu_i$ to $uu_j$, respectively.

While $uu_j$ is the transmitter, the interference is recieved at the Wi-Fi AP side, and is denoted as

$$\text{Intf}_{i,j}^{UL}(t) = P_{i,j}(t)h_{i,j}(t), \tag{5.3}$$

where $P_{i,j}(t)$ and $h_{i,j}(t)$ represent the transmission power and channel gain from $cu_i$ to $ap_j$, respectively.

Thus, $uu_j$'s interference $\text{Intf}_{i,j}$ equals to $\text{Intf}_{i,j}^{DL}(t)$ if $uu_j$ is the receiver, and equals $\text{Intf}_{i,j}^{UL}(t)$ if $uu_j$ is the transmitter. We represent $uu_j$'s SINR at time $t$ when sharing $f_j$ with $cu_i$ as

$$\Gamma_{j,i}^{UU}(t) = \frac{\rho_{i,j}(t)P_j(t)g_j(t)}{\sigma_N^u + \text{Intf}_{i,j}}, \tag{5.4}$$

where $P_j(t)$ and $g_j(t)$ is the transmission power and channel gain for $uu_j$, respectively. $\sigma_N^u$ is the unlicensed spectrum noise.

Now, we formulate the dynamic spectrum sharing problem in LTE-Unlicensed as a sequence of static resource allocation problems in each time slot. With the objective of dynamically maximizing the system throughput, the problem formulation is shown as follows.

$$\begin{aligned}
&\max_{\rho_{i,j}(t)} \sum_{i,j} \frac{f_k \rho_{i,j}(t)}{\sum_j \rho_{i,j}(t)} \log(1 + \Gamma_{i,j}^{CU}(t)) \\
&\quad + \sum_j \frac{\sum_i f_k \log(1 + \Gamma_{j,i}^{UU}(t))}{\sum_i \rho_{i,j}(t)}, \qquad (5.5)
\end{aligned}$$

**s.t. :**

$$\Gamma_{i,j}^{CU}(t) \geq \Gamma_i^{min}, \forall cu_i \in \mathcal{CU}, \tag{5.6}$$

$$\text{Intf}_{i,j}(t) \leq \sigma_{\text{noise}}, \forall uu_j \in \mathcal{UU}, \tag{5.7}$$

$$\sum_j \rho_{i,j}(t) \leq 1, \forall cu_i \in \mathcal{CU}, \text{and} \tag{5.8}$$

$$\sum_i \rho_{i,j}(t) \leq 1, \forall uu_j \in \mathcal{UU}. \tag{5.9}$$

Notice that for any $uu_j$, its associated unlicensed band is pre-assigned, and is denoted as $f_k, \forall f_k \in \mathcal{F}$. (5.6) is the SINR requirement that each CU should satisfy if to reuse a certain unlicensed band. (5.7) represents the maximum interference that each UU can allow from the coexisting CUs. (5.8) and (5.9) are the capacity requirements for CUs and UUs, respectively. Each CU can be allocated to only one UU (i.e., one unlicensed band), and each UU can be allocated to only one CU.

The formulated problem represents a sequential MINLP problems, which is in general NP-hard to solve [53]. In addition, distributive solutions usually act more quickly with lower computation complexities. Thus, we introduce the matching-based approach as the semi-distributive solution to cope with network dynamics.

## 5.5 Dynamic Matching Framework

The formulated optimization problem in Section 5.4 can be modeled as a one-to-one matching game between the CUs and UUs, which results in a many-to-one matching between the CUs and unlicensed bands. Typically, the two-sided one-to-one matching problem has been well studied using the SM model. Intuitively, we can tackle the sequential optimization problems by taking each individual time interval as a traditional SM game, and solving each of them independently over time. This idea will be elaborated in Section 5.5.2. However, in a dynamic network, both the network topology and channel conditions are not isolated in time, and thus there exist some relations between the resource allocations in two adjacent times. Thus, we may make use of such relations for the resource allocation. Under such belief, we propose another matching approach, called the RPTS algorithm, to address the network dynamics. This approach will be discussed in Section 5.5.3. An implementation flow chart for both approaches is shown in Fig. 5.2.

### 5.5.1 Basics of the SM Game

The SM problem is a bipartite matching problem with two-sided preferences. The final result of the SM matching consists of man-woman pairs. The stability definition for the SM instance is provided in Definition 5.1.

**Definition 5.1.** *Let $I$ be an instance of SM, and $\mathcal{M}$ be a matching in $I$. A pair $(m_i, w_j)$ blocks $M$, or is a blocking pair of $M$, if the following conditions are satisfied relative to $M$:*

*(1) $m_i$ is unassigned or prefers $w_j$ to $\mathcal{M}(m_i)$;*

*(2) $w_j$ is unassigned or prefers $m_i$ to $\mathcal{M}(w_j)$.*

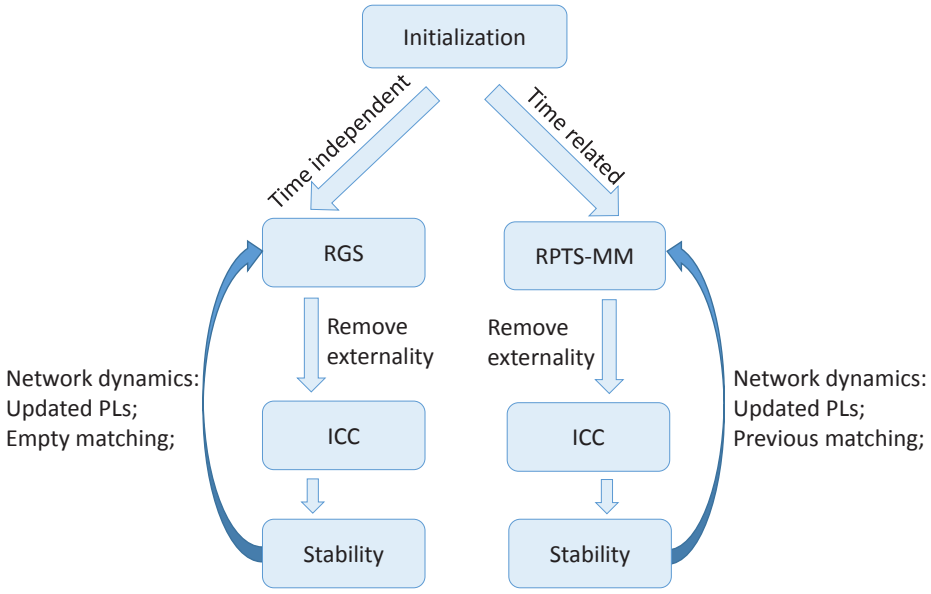



Figure 5.2: Matching Implementations

*$\mathcal{M}$ is said to be stable if it admits no blocking pair.*

$\mathcal{M}(x)$ refers to the partner of $x$ in $\mathcal{M}$, and $x$ can be either a man or a woman.

We assume the CUs to be men and the UUs to be women. Then, as the pre-procedure of any matching algorithm, we first establish each player's preference list. With the channel sensing results from both CUs and Wi-Fi APs, the CUs and UUs can set up their preference lists. The preference of a CU $cu_i, cu_i \in \mathcal{CU}$ over its neighboring UUs $uu_j, uu_j \in \mathcal{UU}$ is based on $cu_i$'s achievable transmission rate when sharing $uu_j$'s unlicensed spectrum $f_j$. Notice that each unlicensed band could be shared within multiple UUs as long as these UUs satisfy the unlicensed transmission regulation. Thus, each unlicensed band can also be shared within multiple CUs, which brings the co-channel interference between CUs. However, before the CUs join any unlicensed spectrum, they have no idea of the other coexisting CUs. Thus, the preference of $cu_i$ over $uu_j$ at time $t$ is simply assumed to be $cu_i$'s transmission rate when only itself is sharing $f_j$ with $uu_j$, and is represented as

$$\mathcal{PL}_{i,j}^{CU}(t) = f_j \log(1 + \Gamma_{i,j}^{CU}(t)). \tag{5.10}$$

On the other hand, the preferences of $uu_j$ over $cu_i$ at time $t$ is based on $uu_j$'s achievable

transmission rate when sharing spectrum with $cu_i$, which is shown as

$$\mathcal{PL}_{j,i}^{UU}(t) = f_j \log(1 + \Gamma_{j,i}^{UU}(t)). \tag{5.11}$$

### 5.5.2 Time-Independent Implementation

#### 5.5.2.1 The GS Algorithm

A stable matching is always guaranteed by using the GS algorithm for the SM game, which is stated in Theorem 5.2 [3].

**Theorem 5.2.** *Given an instance of SM, the GS algorithm constructs in $\mathcal{O}(m)$ time, the unique man-optimal stable matching, where $m$ is the number of acceptable man-woman pairs.*

The GS algorithm consists of sequential proposing and accepting/rejecting actions. Each iteration starts with the men proposing to the most favorite women on their current preference lists. After proposing, the women being proposed to will be removed from the men' preference lists. Then the women decide whether to accept or reject the proposals they've received so far. If the cumulative proposals exceed the capacity 1, each woman chooses to keep the man that she favors most, and rejects the rest. This proposing and accepting/rejecting iteration continues until all the men are matched or all men's preferences are empty. The convergence of the GS algorithm is provided in [28]. The implementation details of the modified GS algorithm for the LTE-Unlicensed can be found in Algorithm 5.1.

#### 5.5.2.2 Eliminating the External Effect

For the conventional SM game, a stable matching is guaranteed using the GS algorithm. However, this conclusion is only correct under the canonical matching assumption. In this problem, there exists the external effect, which refers to the inter-dependence of players' preferences. For example, if too many CUs are matched to the same unlicensed band, then each of them will be assigned a smaller share (by TDMA) than they have expected, in which case some CUs may have the incentive

**Algorithm 5.1 Man-oriented GS (GS) Algorithm**

**Input:**$\mathcal{CU}, \mathcal{UU}, \mathcal{PL}^{CU}(t), \mathcal{PL}^{UU}(t), q$
**Output:**Matching $\mathcal{M}(t)$

```
Construct the set of unmatched CU_un, set CU_un = CU;
while CU_un ≠ ∅ and PL^CU ≠ ∅ do
    CUs proposal to UUs;
    for all cu_i ∈ CU_un do
        Propose to the first UU it in its preference list uu_j, and remove uu_j from PL^UU;
    end for
    UUs make decisions;
    for all uu_j ∈ UU do
        if uu_j has received proposals no more than 1 then
            uu_j keeps the proposal, and remove this CU from CU_un;
        else
            uu_j keeps the most preferred proposal, and rejects the rest;
            Remove this favorite CU from the CU_un, and add the rejected CUs into the CU_un;
        end if
    end for
end while
```

to switch to other unlicensed bands with fewer CUs assigned. In addition, each CU is only admitted by its matched UU, but are not necessarily acceptable to the other existing UUs on this unlicensed band, thus making this matching pair no longer valid.

In order to eliminate such externalities, we propose the ICC strategy to validate and re-stabilize the matching. As the first step, the invalid sharing pairs should be removed. This operation is conducted by the eNBs by informing the related CU and UU to removing each other from their preference lists. The next step is to re-stabilize the matching. Pay attention that, since the UUs represented by the eNBs, and thus, the matching is designed for the benefit of the CUs. As long as the unlicensed transmission regulation is meet, the allocation strategy should focus on how to further improve CUs' performances. Therefore, at this time point, the external effect can be evaluated from the CUs' perspective. In other words, it becomes a one-sided "stability" problem. The new "stability", different from Definition 5.1, relies on the equilibrium among all CUs. We call this one-sided "stability" as the "Pareto Optimality" in matching theory [3]. The definition of Pareto optimal is provided as follows.

**Definition 5.3.** *Pareto Optimal: A matching is said to be Pareto Optimal if there is no other matching in which some player (i.e., CU) is better off, whilst no player is worse off.*

Accordingly, we provide the new definition of the BP for the one-sided matching problems in Definition 5.4.

**Definition 5.4.** *BP in the one-sided matching: A CU pair $(cu_i, cu_j)$ is defined as a BP, if both $cu_i$ and $cu_j$ are better off after exchanging their partners.*

The basic idea of ICC is described as follows. First, we remove all invalid (CU, UU) pairs. Secondly, we begin to search all the "unstable" CU-CU pairs regarding the current matching. Then, we try to check whether the partner switch between such a pair is allowed (beneficial to related CUs) or not. For all the BPs that are allowed to switch their partners, we find the pair that provides the greatest throughput improvement, and let them switch partners. We keep searching such "unstable" pairs until we reach a trade-in-free environment. The detailed ICC algorithm is stated in Algorithm 5.2.

**Algorithm 5.2 Inter-Channel Cooperation (ICC) Strategy**

**Input:** Existing matching $\mathcal{M}$, updated preference lists $\mathcal{PL}^{CU}(t)$ w.r.t. $\mathcal{M}$;
**Output:** Stable matching $\mathcal{M}'$.

$\mathcal{M}' = \mathcal{M}$;
Remove all invalid (CU, UU) pairs;
**while** $\mathcal{M}'$ is not Pareto optimal **do**
    Search the set of "unstable" CU-CU pairs $\mathcal{BP}(t)$ based on $\mathcal{PL}^{CU}(t)$;
    **for all** $(cu_{i1}, cu_{i2}) \in \mathcal{BP}(t)$ **do**
        **if** $\exists cu \in \mathcal{M}'(uu_{j1}^{k1}) \cup \mathcal{M}'(uu_{j2}^{k2})$, $\Delta U(cu) < 0$ **then**
            $(cu_{i1}, cu_{i2})$ are not allowed to exchange partners;
        **else**
            $(cu_{i1}, cu_{i2})$ are allowed to exchange partners;
        **end if**
    **end for**
    Find the optimal BP $(cu_{i1}^*, cu_{i2}^*)$;
    $cu_{i1}^*$ and $cu_{i2}^*$ switch partners;
    $\mathcal{M}' \leftarrow \mathcal{M}'/\{(cu_{i1}^*, \mathcal{M}'(cu_{i1}^*)), (cu_{i2}^*, \mathcal{M}'(cu_{i2}^*))\}$;
    $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{(cu_{i1}^*, \mathcal{M}'(cu_{i2}^*)), (cu_{i2}^*, \mathcal{M}'(cu_{i1}^*))\}$;
    Update $\mathcal{PL}^{CU}(t)$ based on $\mathcal{M}'$;
**end while**

In Algorithm 5.2, we transform the current matching $\mathcal{M}$ (i.e., $\mathcal{M}(t)$ generated by GS) into $\mathcal{M}'$. We define $\mathcal{M}(cu_{i1}) = uu_{j1}, \mathcal{M}(cu_{i2}) = vu_{j2}$. The utility of $cu_i$ is represented as $U(cu_i) = f_j \log(1 + \Gamma_{i,j}^{CU})$, and $\Delta U(cu_i) = U(cu_i)' - U(cu_i)$, where $U(cu_i)'$ is the utility after exchanging partner with another CU. The optimal BP is defined in (5.12).

$$(cu_{i1}^*, cu_{i2}^*) = \underset{(cu_{i1}, cu_{i2})}{\text{argmax}} \sum_{cu_{i1} \in \mathcal{M}_t(uu_{j1})} \Delta U(cu_{i1}) + \sum_{cu_{i2} \in \mathcal{M}_t(uu_{j2})} \Delta U(cu_{i2}), \tag{5.12}$$

where $(cu_{i1}, cu_{i2})$ is allowed to exchange partners. The convergence of ICC is guaranteed by the irreversibility of each switch. The dynamic stability, under the time-related implementation, is reached by adopting the GS-ICC algorithm repeatedly for each time slot.

### 5.5.3 Time-Dependent Implementation

Although we can use the GS-ICC method repeatedly in each time slot to find the dynamic stability, it is not computationally efficient to do so. Let's consider the case, where the network conditions vary very slightly for two adjacent time slots. In other words, only a small number of users' preferences are changed. Under such small network variation, the stable matching result also only varies very slightly and the change only involves a small number of players. Thus, instead of redoing the whole matching, we can utilize the relations between the two matchings, and try to transform the previous matching into stable again. Thus, we propose the RPTS algorithm, also called the Roth Vanda-Vate (RVV) Algorithm [84]. The basic idea of the RPTS algorithm is to use divorce and remarry operations to transform a random matching into stable again.

As shown in Algorithm 5.3, the RPTS algorithm starts from an initial matching $M_0$, which is the matching $\mathcal{M}(t-1)$ from the previous time slot $t-1$ [1]. Each loop of the RPTS algortihm comes with a matching $\mathcal{M}_i$, and it finally terminates with a stable matching. A set $A$ is utilized during the iterations of the RPTS algorithm, which is initially empty. $\mathcal{M}_i|_A$ denotes $\mathcal{M}_i \cap (A \times A)$, and $I|_A$ denotes the sub-instance of $I$ obtained by deleting every member of $(\mathcal{CU} \cup \mathcal{UU})/A$, including the preference lists. The RPTS iterates as long as $\mathcal{M}_i$ is not stable in $I$. During each iteration, if there's a BP $(a_i, b_j)$ such that $a_i \notin A$ and $b_j \in A$, the procedure *add* is called with parameter $a_i$. Otherwise, the *satisfy* procedure is called with parameters $a_i$ and $b_j$ ($a_i \notin A, b_j \notin A$). Notice that $a_i$ can be either a man or a woman. The two procedures *add* and *satisfy* are executed to ensure: 1) no member of $A$ is assigned in $\mathcal{M}_i$ to a member outside of $A$; 2) $\mathcal{M}_i|_A$ is stable in $I|_A$.

[1] We assume the initial matching $\mathcal{M}_0$ to be empty.

In the *add* procedure, $a_i$ is either a man or a woman, which doesn't belong to $A$. Our task is to ensure that upon the arrival of $a_i$, the matching can be restablized and $\mathcal{M}_i|_A$ is also stable in $I|_A$. We start by divorcing the pair $(a_i, \mathcal{M}_i(a_i))$ if $a_i$ is assigned in $\mathcal{M}_i$, and then add $a_i$ to the set $A$. If $a_i$, as the current proposer, is a blocking agent (i.e., involved in a blocking pair) in $(I|_A, \mathcal{M}_i|_A)$, we search $a_i$'s best blocking pair $(a_i, b_i)$ in $(I|_A, \mathcal{M}_i|_A)$. This $b_i$ must belong to $A$, and will be divorced from its current partner $\mathcal{M}_i(b_i)$ if it is currently assigned. Then this $\mathcal{M}_i(b_i)$ becomes the next proposer, and we can add the pair $(a_i, b_i)$ into $\mathcal{M}_i$. The while loop continues as long as the current proposer is a blocking agent in $(I|_A, \mathcal{M}_i|_A)$.

In the *satisfy* procedure, $a_i \notin A$ and $b_j \in A$. We assume $a_i$ and $b_j$ to be $m_i$ and $w_j$, respectively. Our task is to satisfy both $m_i$ and $w_j$. We start by adding $m_i$ and $w_j$ to $A$. If $m_i/w_j$ is assigned in $\mathcal{M}_i$, we divorce it from its partner $\mathcal{M}_i(m_i)/\mathcal{M}_i(w_j)$. Their partners (if any) will remain unassigned. Then we add this BP $(m_i, w_j)$ to $\mathcal{M}_i$.

**Algorithm 5.3 Random Path To Statbility (RPTS) Algorithm**

**Input:** Stable matching $\mathcal{M}(t-1)$ in the previous time $t-1$
**Output:** Stable matching $\mathcal{M}(t)$ at time $t$

**Initialization:**
$\mathcal{M}_i = \mathcal{M}(t-1), A = \emptyset$;
**while** $\mathcal{M}(t)$ is not stable in $\mathcal{I}$ **do**
    **if** There exists $(a_i, b_j) \in bp(I, \mathcal{M}_i)$ such that $a_i \notin A$, and $b_j \in A$ **then**
        ***add*** $a_i$;
    **else**
        choose $(m_i, w_j) \in bp(I, \mathcal{M}_i)$;
        ***satisfy*** $(m_i, w_j)$;
    **end if**
**end while**
$\mathcal{M}(t) = \mathcal{M}_i$

The dynamic stability, under the time-dependent implementation, is reached by adopting the RPTS-ICC algorithm iteratively. Regarding the convergence of the RPTS mechanism in the SM model, a conclusion is stated in Theorem 5.5 [84], and the proof is provided as follows.

**Theorem 5.5.** *Let $\mathcal{M}_0$ be an arbitrary matching for an SM instance $I$ with $N$ men and $M$ women. Then there exists a finite sequence of matchings $\mathcal{M}_0, ..., \mathcal{M}_s$, where $\mathcal{M}_s$ is stable, and for each $1 \leq i \leq s$, $\mathcal{M}_i$ is obtained from $\mathcal{M}_{i-1}$ by satisfying a blocking pair of $\mathcal{M}_{i-1}$. Moreover, $\mathcal{M}_s$ can be obtained in $\mathcal{O}((N+M)m)$ time, where $m$ is the number of acceptable man-woman pairs in $I$.*

**Algorithm 5.4** ***add*** **procedure for RPTS algorithm**

**Input:** $a_i$,$\mathcal{M}_i$
**Output:** $A$,$\mathcal{M}_i$

**if** $a_i$ is assigned in $\mathcal{M}_i$ **then**
    $\mathcal{M}_i = \mathcal{M}_i / \{(a_i, \mathcal{M}_i(a_i))\}$;
**end if**
$A = A \cup \{a_i\}$;
**while** $a_i$ is blocking agent in $(I|_A, \mathcal{M}_i|_A)$ **do**
    $a_i$ is the proposer;
    $(a_i, b_i) \doteq bestbp(I|_A, \mathcal{M}_i|_A, a_i)$;
    $a_z \doteq a_i$;
    **if** $b_i$ is assigned in $\mathcal{M}_i$ **then**
        $\mathcal{M}_i = \mathcal{M}_i / \{(\mathcal{M}_i(b_i), b_i)\}$;
        $a_i = \mathcal{M}_i(b_i)$;
    **end if**
    $\mathcal{M}_i = \mathcal{M}_i \cup \{(a_z, b_i)\}$;
**end while**

**Algorithm 5.5** ***satisfy*** **procedure for RPTS algorithm**

**Input:** $(m_i, w_j)$,$\mathcal{M}_i$
**Output:** $A$,$\mathcal{M}_i$

$A = A \cup \{(m_i, w_j)\}$;
**if** $m_i$ is assigned in $\mathcal{M}_i$ **then**
    $\mathcal{M}_i = \mathcal{M}_i / \{(m_i, \mathcal{M}_i(m_i))\}$;
**end if**
**if** $w_j$ is assigned in $\mathcal{M}_i$ **then**
    $\mathcal{M}_i = \mathcal{M}_i / \{(\mathcal{M}_i(w_j), w_j)\}$;
**end if**
$\mathcal{M}_i = \mathcal{M}_i \cup \{(m_i, w_j)\}$;

*Proof.* During each iteration of the RPTS algorithm, $A$ increases in size by either one (*add* procedure) or two elements (*satisfy* procedure). At the end of each such iteration, we have $\mathcal{M}_i|_A$ stable in $I|_A$. Hence we are bound to ultimately reach the outcome that $\mathcal{M}_s$ is stable (when $A$ reaches the size of $(N + M)$, in which case the RPTS algorithm terminates.

The complexity of the RPTS algorithm is obtained by observing that $A$ increases in size one or two element(s) at each iteration. Since $|A| \leq (N + M)$, it follows that the same upper bound that applies to the number of iterations of the RPTS algorithm. Each proposal-rejection sequence during an execution of the *add* procedure, at most $m$ pair of agents are involved. Thus, each iteration of *add* runs in $\mathcal{O}(m)$ time. While each call of the *satisfy* procedure takes $\mathcal{O}(1)$ time (no while loop inside). Thus, the overall computation complexity of finding a stable matching is $\mathcal{O}((N + M)m)$. □

## 5.6 Performance Evaluation

### 5.6.1 Complexity Analysis

The primary difference between the GS algorithm and the RPTS algorithm lies in their adaptabilities to network dynamics. Each time, the GS algorithm starts from an empty matching and by using the propose/reject operations to reach a stable matching. The RPTS algorithm begins with the matching from the previous time slot and takes the divorce/remarry operations to find its path to stability. The computation complexities or say iteration times for both algorithms depend on the number of users and how fast the network changes.

As provided in Section 5.5.2, the complexity of GS is $\mathcal{O}(m)$, where $m$ is the total length of all players' preference lists. It makes sense since the worst case of the GS is to traverse each player's preference lists and terminate. On the other hand, the computation complexity of the RPTS algorithm is $\mathcal{O}((N + M)m$, as indicated in Theorem 5.5. Again, it is not necessary for the RPTS algorithms to satisfy every possible BP to reach the stability. Regarding the ICC algorithm, it is realized by the iterative search of the currently best BP and the swapping of their partners. The complexity of finding all the BPs regarding the current matching, which requires the traverse of

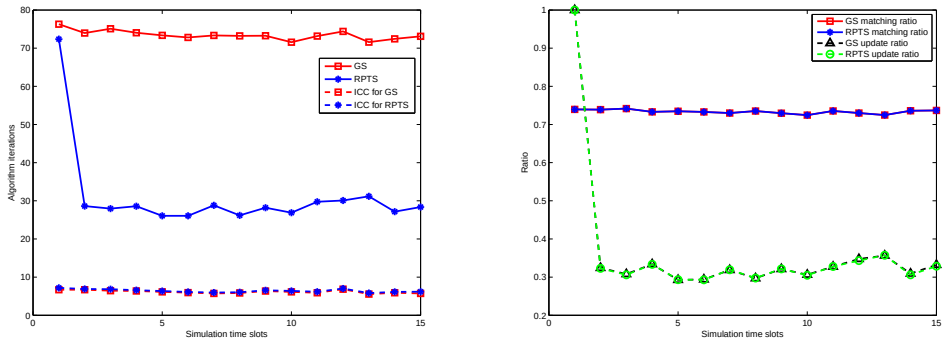
all users' preference lists, is bounded by $MN$ comparing operations. Since each swap in ICC is irreversible, meaning any two CUs can only swap partners with each other no more than once, the total iterations are bounded by $N^2$. Thus, the worst case complexity of the ICC algorithm is $\mathcal{O}(MN \times N^2)$ or denoted as $\mathcal{O}(N^3M)$. However, the actual computation cost is not necessarily as high as shown in the theoretical analysis.

Theoretically, the RPTS algorithm has a higher complexity than the GS algorithm. However, we should not ignore the piratical implementation. In practice, the actual complexity depends on many complicated network factors, such as the user velocity, network density and so on. To best evaluate the complexities of the matching algorithms in the LTE-unlicensed problem, we measure the complexity by counting the number of new (CU,UU) connections that are attempted to be set up during the matching. These new connections are not necessarily the final stable connections, since during the matching a partnership may break up later. However, building such a potential connection requires the exchange of information through the communication between the two user ends of the link. As we know, the communication overhead is a big concern in protocol/mechanism design regarding both cost and time efficiency. Thus, measuring the number of potential links that are set up during the matching is a reasonable assumption for the complexity cost in the practical implementations.

### 5.6.2 Experimental results

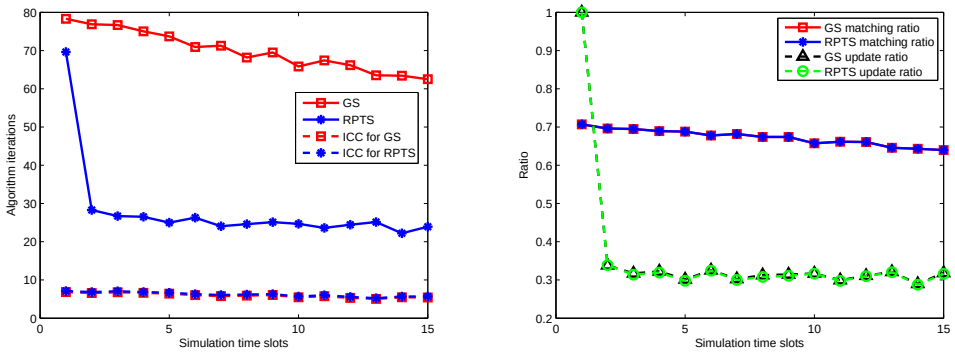
We first analyze the impact of the network dynamics on the resource allocations. Fig. 5.3, Fig. 5.4 and Fig. 5.5 evaluate the time dynamic performances of the proposed GS, RPTS and ICC algorithms, w.r.t. the computation complexity, matching update ratio, and system throughput.

The complexity of the three proposed algorithms are evaluated under the RWP and the HotSpot patterns in Fig. 5.3a and Fig. 5.4a, respectively. Apparently, the RPTS algorithm achieves a much lower complexity than the GS algorithm in both mobility models. Although the theoretical analysis indicates that the RPTS algorithm has higher complexity than the GS algorithm in the worst case,




(a) Computation complexity. (b) Connection updated ratio.

Figure 5.3: Time dynamics in the RWP mobility model.

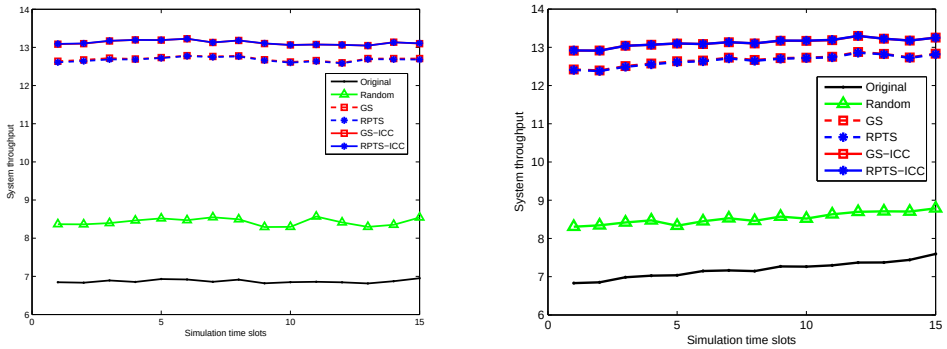



(a) Computation complexity. (b) Update ratio.

Figure 5.4: Time dynamics in the HotSpot mobility model.

the practical computation cost can be different. For the starting point, it's reasonable that the RPTS algorithm has a relatively high cost than the other time slots, but still lower than the GS algorithm, since the initial matching is empty. Comparing the two curves of the ICC algorithm in both Fig. 5.3a and Fig. 5.4a, we can see that they achieve similar results in both mobility models. In addition, we can find from both figures that it averagely takes 8 swaps to re-stabilize the matching using the ICC algorithm. In the HotSpot model, the complexities for all the three algorithms slowly decrease as time evolves. This is reasonable since in the HotSpot model CUs are gathering toward the event point (faster than UUs) and thus fewer CUs are matched as time evolves.

In Fig. 5.3b and 5.4b, we have evaluated both the user matching ratio and the matching update ratio. The user matching ratio represents the percentage of CUs who are allocated with the unlicensed bands. As indicated in both figures, the GS algorithm and the RPTS algorithm achieve




(a) RWP model. (b) HotSpot model.

Figure 5.5: Average system throughput comparison.

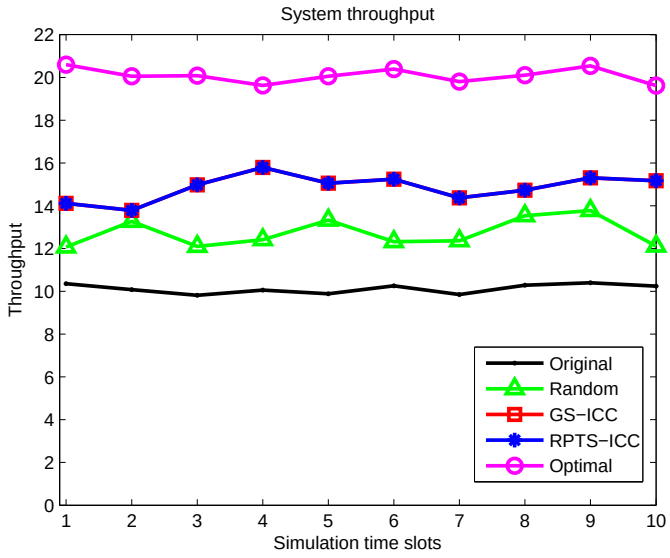
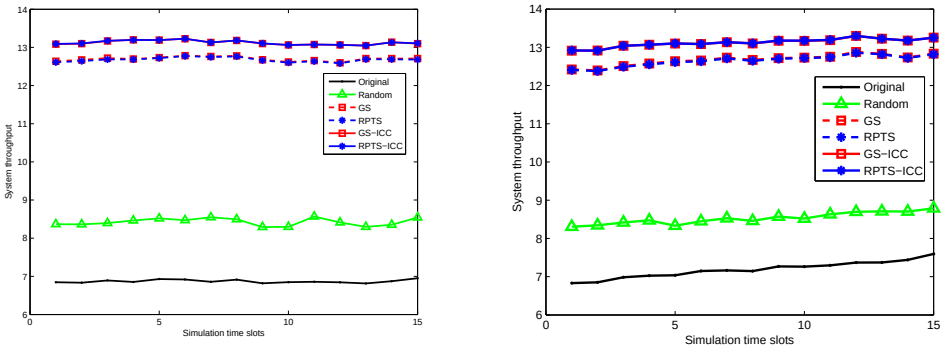



Figure 5.6: System throughput comparison with optimal solution.

similar matching ratios, which are both as high as $75\%$ in the RWP model and $70\%$ in the HotSpot model, averagely. The matching update ratio represents the percentage of partnerships changed in the next simulation time slot. Again, both algorithms achieve similar performances, which are both around $30\%$ averagely. The update ratios at the starting point for both algorithms reach $100\%$ since we assume to start with an empty matching.

For the throughput performance, we compare the GS and RPTS algorithms, with five other methods: the GS-ICC method, the RPTS-ICC method, the Random method, the Original method and the Optimal method. The GS-ICC and RPTS-ICC methods refer to the cases that the ICC



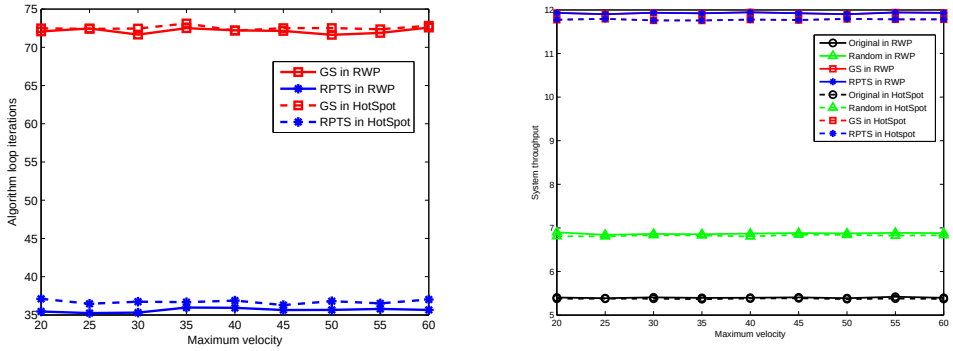

(a) Computation complexity.

(b) System Throughput

Figure 5.7: CU density dynamics in RWP&HotSpot mobility model.

algorithm is used after the GS and RPTS algorithms, respectively. The Random method refers to randomly allocating the unlicensed resources to the CUs, while the Original method refers to the case that no spectrum sharing happens. In the RWP model, as shown in Fig. 5.5a, the average system throughput is evaluated. Apparently, the four matching-based methods outperform the Random and Original methods a lot. The GS and RPTS methods achieve similar performances. Apparently, after using the ICC procedure, the system throughput is further improved with either the GS or the RPTS algorithm. More specifically, the average system throughput achieved by the GS-ICC method or the RPTS-ICC method is about $86\%$ higher than the Original method, and about $53\%$ higher than the Random method. In addition, we have also compared the performance of the proposed methods with the optimal solution in Fig. 5.6. The optimal result is achieved by using the brute force approach, which is very time-consuming. Thus, the number of CUs and UUs are set as $N = 4$ and $M = 4$, $B_1 = 2$, and $B_2 = 2$. As shown in Fig. 5.6, both the RPTS-ICC method and the GS-ICC method can achieve about $75\%$ of the optimal result regarding the system throughput.

Except the time dynamic analysis, we have also evaluated the impact of network density and user velocity to the resource allocations. As shown in Fig. , we change the network density by adding more users, including both CUs and UUs, into the network without adding any eNB, Wi-Fi AP, or unlicensed band. We add $5$ CUs and $5$ UUs to the network each time by starting from $N = M = 20$ and stop when $N = M = 65$. The number of unlicensed bands is set as $K = 30$. As shown in Fig. 5.7a, the complexity of the GS, RPTS and ICC algorithms all increase as more




(a) Computation complexity. (b) System Throughput

Figure 5.8: CU velocity dynamics in RWP&HotSpot mobility model.

users join the network. In addition, the complexity of the GS algorithm grows faster than the RPTS method, which demonstrates the good scalability of the RPTS algorithm. For the system throughput, as shown in Fig. 5.7b, the peak value is achieved when each unlicensed band accommodates exactly one CU. When more CUs join the network after this point, the unlicensed bands will be shared between multiple CUs by using TDMA.

We also change the maximum velocity value in both mobility models to test our proposed algorithms. As shown in Fig. 5.8, we increase the CU's maximum velocity (identical for all CUs) from 20 m/s to 60 m/s by 5 m/s. Apparently, the velocity changes do not necessarily have any impact on the computation complexity or the system throughput.

## 5.7 Conclusion

In this chapter, we have studied the dynamic resource allocation problem in the LTE-Unlicensed. The SM matching model has well interpreted the two-sided feature of the resource allocation problem. The proposed GS and RPTS algorithms provide near-optimal system throughput, and both methods can guarantee the QoS requirements and network stability. Especially, the RPTS algorithm achieves lower computation complexity than the GS algorithm. In other words, the RPTS algorithm is more adaptable than the GS algorithm under network dynamics in providing dynamic stability in the LTE-unlicensed.

# Chapter 6

# Exploiting the Stable Fixture Matching Game for Content Sharing in D2D-based LTE-V2X Communications

The study item: "Feasibility Study on LTE-based V2X Services", approved at 3GPP TSG RAN #68, has drawn much attention in the study of the LTE assisted vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications in the vehicular networks. By deploying the D2D technology adopted in the traditional cellular networks into the V2X communications (including both the V2V and V2I communications), performance improvements can be expected, such as better reliability, lower latency, and more efficient content sharing. This chapter investigates the content sharing problem in the D2D-based V2X communications. With both vehicles and eNBs carrying different types of data, this chapter studies how to optimize the information exchanged within the network. By jointly considering the data diversity and the communication link quality, the interactions between the vehicles/eNBs are modeled as the stable fixture (SF) matching game. Different from the traditional D2D communications, we allow vehicles.eNBs to set up multiple link connections for to further optimize the content sharing in the Vehicular ad hoc networks (VANETs). The SF game is solved by the proposed Irving's stable fixture (ISF) algorithm.

## 6.1 Introduction

The technology of the connected vehicles has been envisioned as a paradigm capable of providing increased convenience to drivers, with applications ranging from road safety to traffic efficiency. In the traditional IEEE 802.11p based vehicle-to-vehicle (V2V) communications, reliable and efficient performance cannot be guaranteed since 802.11p is CSMA/CA based. Besides, the high cost of deploying roadside units (RSUs) cannot be ignored. Thus, the concept of integrating LTE into the V2X communications has been proposed, which is commonly referred to as the LTE-V [16] or the LTE-based V2X [17]. The objectives of the study on LTE V2X communications include the

definition of an evaluation methodology and the possible scenarios for vehicular applications, and the identification of necessary enhancements to the LTE physical layer, protocols, and interfaces. So far, 3GPP has defined 18 use cases in TR 22.885 for the LTE-based V2X services, such as Case 5.1: Forward collision warning, Case 5.8: Road safety services, Case 5.9: Automatic parking system, and so on [18].

The LTE-based V2X technology allows low-cost and rapid deployment in the Intelligent Transportation System (ITS) compared with other solutions, since it can fully utilize the existing cellular base stations. LTE, as the currently most advanced wireless communication technology, features low-latency and high-reliability communications. Thus, the LTE-based V2X communications are perfect for some safety-critical VANET applications. Despite the above-mentioned advantages of LTE-V, this technology is also facing some challenges, for example, the cellular spectrum allocation, the network architecture design and so on.

The D2D communication technology has already paved the way for the LTE-V. Thus, if implementing the D2D-based LTE-V, the advantages of D2D communications, such as improving the spectrum efficiency, and extending the network coverage, can also be extended to the V2X communications. In fact, there have been some existing works on the D2D-based V2X communications. For example, in [85], the resource allocation for the D2D-based V2X communications is discussed. They addressed the latency and reliability issues by restricting the outage probability to be lower than a threshold, and solved the social welfare optimization problem using the heuristic RBSPA algorithm. The simulations proved the effectiveness of the proposed algorithm. To think one step further, if we allow each vehicle to be able to connect to multiple vehicles, then we may further increase the network capacity. In fact, the restriction of only one radio interface equipped for each traditional cellular user (e.g., cellphone), due to the size and energy issues, is no longer a problem in the VANETs, since vehicles are relatively abundant in space and energy for carrying more radio interfaces. In addition, in [85], only the transmission rate is evaluated regarding the network performance. However, the information value received by each vehicle through the data exchange is another important factor that should be considered in various V2X communication applications,

other than the pure transmission rate.

For the content sharing purpose, typically the cluster formation method is adopted in the VANETs. Clusters are formed within multiple vehicles, and a cluster head (CH) is selected to be responsible for all management and cooperation work of all the ordinary nodes (ONs) within the cluster. For example, in [86], the content sharing framework between the RSUs and the vehicles is formed by using the cooperative coalition formation game among the RSUs. In [86], the RSU acts as the CH, and estimates its utility based on the vehicles currently associated with it. Thus, any vehicle requiring information update has to go through the RSU, which may result in traffic congestion when communicating with the CH. In addition, ONs (i.e., vehicles) may lack the flexibility of setting up connections with any other node who may carry their interested information.

Motivated by the above reasons, we try to model the content sharing problem using a flexible many-to-many matching framework. The high complexity and the global information requirement have made the traditional centralized optimization less efficient in the high-density and high-mobility VANETs. Thus, in this chapter, we propose a matching-based approach to solve the content sharing problem in the V2X communications, and the major contributions are summarized as follows [87].

1) Different from the traditional D2D communications, our proposed V2X framework allows multiple connections for each vehicle. This further improves the network performance compared with the one-to-one communication case.

2) We introduce the weight factor for different data types to measure the data value. By jointly considering the data weight and communication link quality, we can improve the information diversity that circulated within the network, while achieving good throughput performance.

3) We model the content sharing problem as the SF game, where each node can set up multiple independent links with other nodes. Such V2X links are more flexible than the many-to-many relationships formed using the traditional clustering method. We solve the SF game by using the ISF algorithm, which can provide comparable performance with the centralized optimization approach.

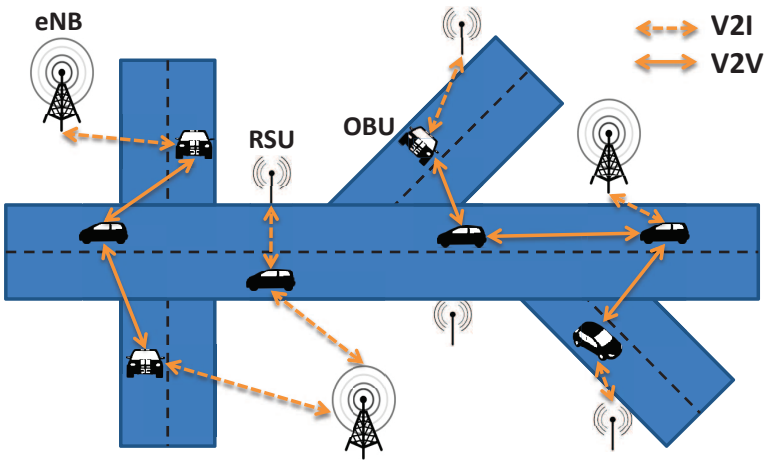



Figure 6.1: The V2X communication network model.

The rest of this chapter is organized as follows. The system model is described in Sec. 6.2, and we formulate the content sharing problem as a constrained optimization problem. The SF game is proposed to model the many-to-many relationships between vehicles, and the ISF algorithm is introduced to find a stable solution in Sec. 6.4. The performance of the proposed ISF algorithm is evaluated in Sec. 6.5. Finally, conclusions are drawn in Sec. 6.6.

## 6.2 System Model

Consider a VANET, as shown in Fig.6.1, where vehicles can communicate with either its neighboring vehicles, RSUs or eNBs through the cellular communications to receive the lasted information. We denote the set of vehicles as $\mathcal{V} = \{v_1, ..., v_i, ..., v_N\}$. For representation consistency, we use the set of eNBs $\mathcal{BS} = \{bs_1, ..., bs_i, ..., bs_M\}$ to denote both RSUs and eNBs, since they perform similar functionalities here. We assume that each vehicle is originally carrying a certain amount of data in various data types, for example, the entertainment information, the accident information, the road maintenance information and so on. We denote the data type set as $D = \{d_1, ..., d_j, ..., d_K\}$, where $K$ is the number of all the data types. Thus, the set of data carried by vehicle $v_i$ can be represented as $Q_i = \{q_1^i, ..., q_j^i, ...q_K^i\}$, where $q_j^i$ is the amount of data in type $d_j$ carried by $v_i$. A content sharing happens when any two vehicles (one vehicle and one eNB) set up a direct link connection,

referred to as the V2V link (V2I link). Since eNBs also exchange information with vehicles, from now on for simplicity, we denote the joint set of $\mathcal{V}$ and $\mathcal{BS}$ as $\mathcal{N} = \{n_1, ..., n_i, ...n_{N+M}\}$. We use the term node to represent either a vehicle or an eNB. Different data types are in different weights depending on the receiver nodes regarding their current interests. In other words, the data carried by any node means different values to different receivers. Thus, each node $n_i$ defines a positive weight value for each data type, denoted as $WT_i = \{wt_1, ..., wt_j, ..., wt_K\}$. Thus, the total information value that $n_i$ receives from $n_i'$ can be represented as $val_{i,i'} = \sum_{wt_j \in WT_i} wt_j q_j^{i'}$, where $n_i, n_i' \in \mathcal{N}, i \neq i'$. The eNBs typically carry more data than the vehicles.

By the direct communication set up between any two nodes, the exchanged information can improve the nodes' knowledge of the current network. To quantify the exchanged information value within the whole network, we first need to define the communication link matrix for all the nodes. We denote the matrix as $\rho = \{\rho_{i,j} | n_i, n_j \in \mathcal{N}\}$, where $\rho_{i,j}$ is a binary value in $\{0, 1\}$, representing if there is a communication link set up between $n_i$ and $n_j$. We allow each node to be connected to more than one nodes up to its capacity, which is defined by the number of radio interfaces equipped in each node, denoted as $c_i, n_i \in \mathcal{N}$.

### 6.2.1 System Requirements

#### 6.2.1.1 Latency requirement

For safety-critical V2X services, there is usually a latency requirement. In the cellular communication, we can use SINR as the latency metric. To satisfy the latency requirement for each node, we require its received SINR from any other potential node to be higher than a threshold $\gamma_{\text{min}}$. We assume that each node pair $(n_i, n_j)$ is assigned two subbands $f_{i,j}$ and $f_{j,i}$ for information exchange. In $f_{i,j}$ and $f_{j,i}$, the noise and interference powers are denoted as $\sigma_{i,j}$ and $\sigma_{i,j}$, respectively. Thus, the latency requirement for each link $f_{i,j}, n_i \neq n_j$ can be satisfied using the following inequality:

$$\gamma_{j,i} = \rho_{j,i} \frac{p_{j,i} g_{j,i}}{\sigma_{j,i}^2} \geq \gamma_{\text{min}}, \tag{6.1}$$

where $\gamma_{j,i}$ represents $n_i$'s received SINR from node $n_j$, for $n_i, n_j \in \mathcal{N}, i \neq j$.

#### 6.2.1.2 Capacity requirement

Except for the transmission quality requirement, each node should also satisfy the capacity requirement. The capacity requirement for each node $n_i, n_i \in \mathcal{N}$ is shown as follows:

$$\sum_{n_j \in \mathcal{N}} \rho_{i,j} \leq c_i. \tag{6.2}$$

### 6.2.2 User Uitlity

When deciding who to exchange information with, each node concerns about not only the information value carried by the potential partner, but also the communication link quality between them. In other words, on one hand, nodes hope to communicate with the nodes with higher information value. Besides, they also hope to set up good communication links so that more data can be received during a fixed communication period $T$. The received transmission rate of $n_i$ from $n_j$, is represented as

$$r_{j,i} = f_{j,i} \log \left( 1 + \frac{p_{j,i} g_{j,i}}{\sigma_{j,i}^2} \right), \tag{6.3}$$

where $p_{j,i}$ and $g_{j,i}$ denote the transmission power and channel propagation gain from $n_j$ to $n_i$, respectively. $g_{j,i} = k_p \beta_{j,i} \zeta_{j,i} d_{j,i}^{-\alpha}$, where $k_p$ is a constant system parameter, $\beta_{j,i}$ is the fast fading gain, $\zeta_{j,i}$ is the slowing fading gain, $\alpha$ is the path loss exponent, and $d_{j,i}$ is the distance between $n_i$ and $n_j$. $f_{j,i}$ is the subband assigned for transmission from $n_j$ to $n_i$. $\sigma_{j,i}$ is the channel noise of band $f_{j,i}$.

Each node can set up multiple V2X links with other nodes up to its capacity. Thus, to measure the overall content sharing, each user's utility $u_i$ is defined as the total information value it gains from all of its V2X links during the communication period $T$. Given the communication matrix $\rho$, the the utility for each vehicle $n_i, i \in \mathcal{N}$ is represented as

$$u_i = \sum_{n_j \in \mathcal{N}} \rho_{i,j} \cdot T \cdot r_{i,j} \cdot val_{i,j}, \tag{6.4}$$

where $T$ is the transmission period, and $val_{i,j}$ is the information value that $n_i$ can receive from $n_j$. The utilities of all the vehicles can be represented as the set $\mathcal{U} = \{u_i | v_i \in \mathcal{V}\}$.

## 6.3 Problem Formulation

In this section, we provide the problem formulation of the content sharing problem. The objective is to maximize the social welfare, which is the summation of all vehicles' received information value during the communication period $T$. The optimization problem is subject to the system QoS requirements. Based on the above discussions, the problem is formulated as follows.

$$\textbf{max:} \sum_{\rho_{i,j}} \sum_{v_i \in \mathcal{V}} \sum_{n_j \in \mathcal{N}} \rho_{i,j} \cdot T \cdot r_{i,j} \cdot val_{i,j}, \tag{6.5}$$

**s.t.:**

$$\gamma_{j,i} = \rho_{i,j} \frac{p_{j,i} g_{j,i}}{\sigma_{j,i}^2} \geq \gamma_{\text{min}}, \tag{6.6}$$

$$\sum_{n_j \in \mathcal{N}} \rho_{i,j} \leq c_i, \forall n_i \in \mathcal{N}, \text{and} \tag{6.7}$$

$$\rho_{i,j} = 0, \forall i = j. \tag{6.8}$$

(6.5) is the system objective that maximizes all vehicles' utilities. (6.6) represents SINR requirement for each node, while (6.7) indicates the capacity requirement of each node. (6.8) states that if any node cannot be assigned to itself as a V2X link.

The formulated problem is an MILP problem, which is NP-hard to solve in general [53]. The centralized solution typically requires the global information, and the complexity increases exponentially with the increase of the user number. In addition, in the V2X communications, nodes are in movement, and the channel conditions change as well. Thus it becomes hard to collect the global information from all the nodes to do the optimization in a real-time manner. Thus it motivates us to find a distributive approach where each vehicle can make local decisions rapidly. We introduce the matching-based approach in the following sections.

## 6.4 Matching-based Approach

In this section, we solve the V2X communication problem by modeling it as the SF game. We start by introducing some basic definitions of the SF game in Section 6.4.1 and then propose a distributive matching algorithm to solve the SF model in Section 6.4.2.

### 6.4.1 Stable Fixture Game

The SF problem stems from a practical situation, where players play against one another in a chess tournament. Each player ranks their potential opponents in order of preferences. The task is to construct a set of fixtures, consisting of distinct matched pairs (each involving two players, and each player can be involved in more than one match but cannot exceed its capacity), which is stable [88]. In the chess tournament, each competition happens independently. In other words, if player *a* plays with *b* first, then plays with *c*, then player *b* and *c* are not required to be involved in the same competition. Just like the many-to-many relationships in the chess tournament, in the V2X communications each node can set up multiple independent communications with other nodes up to its capacity.

Assume an SF instance consisting of a single set of players $A = \{a_1, ..., a_n\}$, where $n$ is the number of all players/agents. Each player first set up a list containing all the other acceptable agents. Then each agent ranks its acceptable list of the acceptable agents according to its preferences, and this list is called the preference list. The matching decisions are made based on the preference lists. A matching $M$ in a SF instance is defined as a subset of $E$, where $E = \{(a_i, a_j)|a_i, a_j \in A, i \neq j\}$. We denote $a_j = M(a_i)$ if pair $(a_i, a_j)$ is in the matching $M$. The formal definition of a stable matching in the SF model is provided in Definition 6.1:

**Definition 6.1.** *Stability: Let $I$ be an instance of SF and $M$ be a matching in $I$. A pair $(a_i, a_j) \in E/M$ blocks $M$, if the following conditions are satisfied relative to $M$:*

*(1) $a_i$ is under subscribed or prefers $a_j$ to its worst partner;*

*(2) $a_j$ is under subscribed or prefers $a_i$ to its worst partner.*

*A matching $M$ is said to be stable if it admits no BP.*

To model the V2X communications as the SF game, we assume the nodes (including both vehicle and eNBs) to be the chess players. Each node first search for its acceptable set of nodes, who can satisfy the SINR requirement. Then the preference value is calculated based on the received information value during the communication period from its acceptable partner. We represent the preferences of node $n_i$ over its acceptable set $\mathcal{A}(n_i)$ as

$$PL_i(j) = T \cdot r_{i,j} \cdot val_{i,j}, \forall n_j \in \mathcal{A}(n_i). \tag{6.9}$$

### 6.4.2 ISF Algorithm

In this section, we introduce the ISF algorithm to solve the SF game. The existence and convergence of the ISF algorithm are stated in Theorem 6.2, and the proof can be found in [88].

**Theorem 6.2.** *Given an instance of SF, the ISF algorithm constructs in $\mathcal{O}(m)$ time, a stable matching, or reports that no stable matching exists, where $m$ is the total length of all players' preference lists.*

The key idea of ISF is the reduction of players' preference lists $PL = \{PL_1, ..., PL_{N+M}\}$ and the construction of a player set $S$. This set $S$ consists of order player pairs, which is initially empty, and will be symmetric finally, in which case it reaches the stable matching. There are two cases in which a stable matching does not exist: (1) $\sum_i d_i$ is odd, (2) there's short list in $PL$ during the execution of the ISF algorithm. $d_i$ is the player's degree, and will be defined later. The step-to-step implementation of the ISF algorithm is provided in Algorithm 6.1.

The ISF algorithm consists of two phases. Phase 1 involves a sequence of bids from one node to another. These bids enable the construction of the player set $S$ and the reduction of the preference lists $PL$. To start with, each node $n_i$ bids for its most favorite node who is currently not in $A_i$, and denote it as $n_j$. We denote $n_i$'s target set and bidder set as $A_i$ and $B_i$, respectively. $A_i = \{n_j | (n_i, n_j) \in S\}, B_i = \{n_j | (n_j, n_i) \in S\}$, and $a_i = |A_i|, b_i = |B_i|$. Then, we construct

**Algorithm 6.1 ISF Algorithm**

**Input:**$\mathcal{N}$, $c$, $PL$, $S = \emptyset$
**Output:**Stable Matching $\mathcal{M}$

**Phase 1:**
**while** $a_i < \min(c_i, |PL_i|)$ **do**
    $n_j =$ the first player in $PL_i$ who is not in $A_i$;
    $S = S \cup \{(n_i, n_j)\}$;
    **if** $b_j \geq c_j$ **then**
        $n_k = c_j$th ranked bidder for $n_j$;
        **for all** successor $n_l$ of $n_k$ in $PL_j$ **do**
            **if** $(n_l, n_j) \in S$ **then**
                $S = S/\{(n_l, n_j)\}$
                delete $(n_l, n_j)$ from $PL$;
            **end if**
        **end for**
    **end if**
**end while**
**Phase 2:**
**if** $\sum_i d_i$ is odd **then**
    report instance unsolvable;
**else**
    **while** there's no short list in $PL$ **do**
        find a rotation $\rho$ in $PL$;
        $PL = PL/\rho$;
        **if** some list in $PL$ is short **then**
            report instance unsolvable;
        **else**
            $S = S(PL)$;
        **end if**
    **end while**
    $\mathcal{M} = S$;
**end if**
**End of algorithm**.

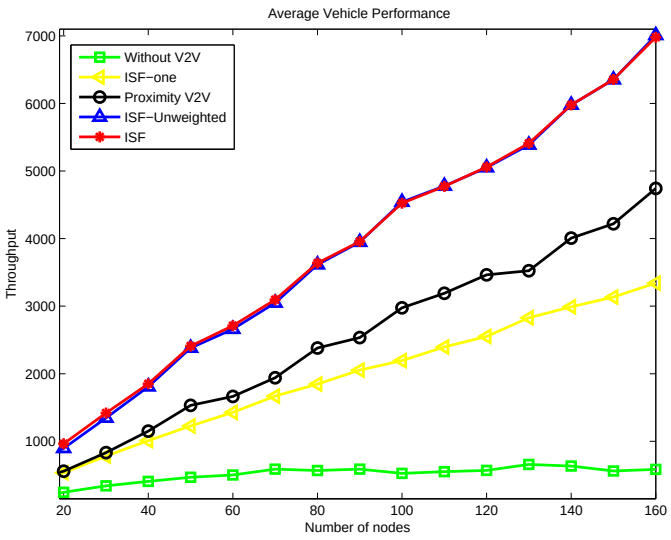



Figure 6.2: System social welfare.

$S$ by adding the pair $(\{(n_i, n_j)\})$ into it. Notice that all the pairs in $S$ are ordered, which means $(\{(n_i, n_j)\})$ and $(\{(n_j, n_i)\})$ are different in $S$. Then for the target node $n_j$, it checks whether its received bids has exceeded its capacity $c_j$. If yes, it deletes the bids who are worse than the $c_j$'s rank in $PL_j$. By the deletion of a pair $\{(n_i, j_j)\}$ from $PL$, we mean the removal of $n_i$ from $PL_j$ and the removal of $n_j$ from $PL_i$. The bid of $n_i$ continues as long as $a_i < \min(c_i, |PL_i|)$. Phase 1 terminates when each node's target set size has reached $\min(c_i, |PL_i|)$.

After the Phase 1, we have achieved a reduced preference list $PL$ and a constructed set $S$. We define $d_i$ as $min(c_i, |PL_i^1|)$, which is the degree of $n_i$. To start Phase 2, we first check if $\sum_i d_i$ is odd, if yes then we report this instance is unsolvable, otherwise we continue. The key idea of Phase 2 is the further construction of $S$, and the further reduction of $PL$. We classify all players' preference lists into the *short* lists and the *long* lists. We call $PL_i$ *short* if $|PL_i| < d_i$, and *long* if $|PL_i| > d_i$. During any time of the execution, if any node has a short list, then no stable matching exists. While no short list occurs, we try to find a rotation first, which is the key to the further reduce $PL$. A rotation is defined as a sequence of ordered pairs $\rho = ((n_{i_0}, n_{j_0}), (n_{i_1}, n_{j_1}), ..., (n_{i_{r-1}}, n_{j_{r-1}}))$, where for each $0 \leq k \leq r-1$, $n_{i_k} = n_{l(j_k)}$ and $n_{j_{k+1}} = n_{f(x_{i_k})}$. $x_{l(i)}$ is the last player in $PL_i$, and $x_{f(i)}$ is the first player in $PL_i$ who is not in $A_i$. To find a rotation, we begin by any node who

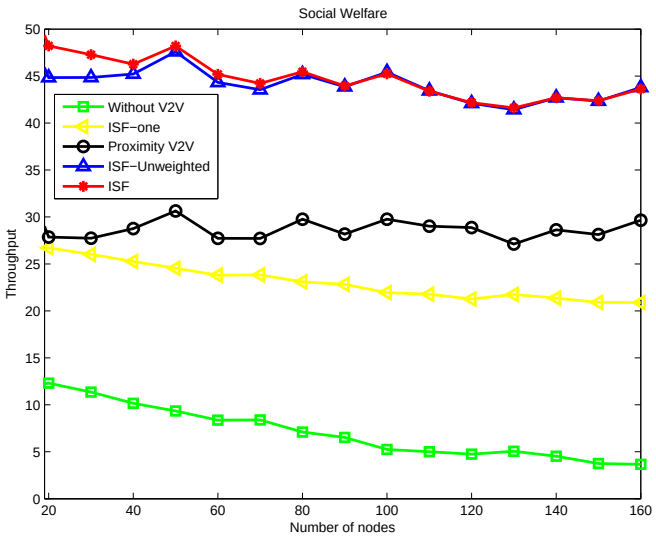



Figure 6.3: Average vehicle performance.

has a long list, and set it as $n_{j_0}$. Then by the relations between $n_{i_k}$, $n_{l(j_k)}$ and $n_{l(j_k+1)}$, we can start building the rotation $\rho$. The building process stops when any node is visited twice, and then we claim a rotation is found. To eliminate $\rho$ from $PL$, we delete all the pairs $(n_{j_k}, n_l)$, such that $n_{j_k}$ prefers $n_{g(j_k)}$ to $n_l$, where $n_g(j_k)$ is the least favored member of $n_{j_k}$ in $\{B_{j_k} \cup n_{i_{k-1}}\}/\{n_{i_k}\}$. The finding and eliminating rotation process terminates, whenever any short list occurs or when no rotation can be found, meaning that we have reached a stable matching.

## 6.5 Performance Evaluation

In this section, the performance of the proposed ISF algorithm will be evaluated by comparing with both the centralized optimization and four other heuristic mechanisms. Both the social welfare and the average user performance will be analyzed. In addition, the network connection ratio and the existence of stable results will also be evaluated.

### 6.5.1 Simulation Set Up

Within a circle VANET with radius of $R = 800$ m, we assume $N = [0, 200]$ vehicles and $M = 8$ eNBs. The channel bandwidth that allocated to each communication link is set as $1MHz$.

The SINR requirements for both vehicles and eNBs are uniformly distributed within $(20, 30)$ dB. For the propagation gain, we set the pass loss constant $k_p$ as $10^{-2}$, the path loss exponent $\alpha$ as $4$, the multipath fading gain as the exponential distribution with unit mean, and the shadowing gain as the log-normal distribution with $0$ mean and $4$ dB deviation. The capacity of each vehicle is set as $4$, and capacities of the eNBs are randomly distributed within $[10, 15]$. The total data type number is set as $K = 10$.

### 6.5.2 Numerical Results

To demonstrate the effectiveness of the ISF algorithm, we compare it with four heuristic mechanisms: (1) Without V2V, (2) ISF-one, (3) Proximity V2X, (4) ISF-Unweighted. In the Without V2V method, we simply assume each vehicle only communicates with its local eNBs. For the ISF-one method, each vehicle is only allowed to set up one connection. The one-to-one stable matching is also generated by the ISF algorithm with the capacity of each node set to 1. For the Proximity V2X method, each vehicle $n_i$ is connected to the $c(i)$ closest nodes which can meet its SINR requirement. While in the ISF-Unweighted method, the data weight is not considered in the ISF algorithm.

We evaluate the information exchanged in one communication period $T$. $N$ varies from 20 to 160 with the step of 20. As shown in Fig. 6.2, the total information values exchanged within the network, using all the five methods, increase as more users join the network. Among the five curves, the Without V2V curve achieves the worst performance, which demonstrates that the V2X communications can improve system performance. Then, by comparing the many-to-many V2X communications (i.e., the Proximity V2X, ISF-Unweighted and ISF methods) with the ISF-one method, we find that by setting up more connections, the network capacity can be improved. Besides, both the ISF-Unweighted and ISF methods outperform the Proximity V2X method. Last but not least, the ISF algorithm achieve slightly higher performance than the ISF-Unweighted because of the weight factor. Similar conclusions can be drawn from Fig. 6.3, which evaluates the average user performance. One thing to mention in the Fig. 6.3 is that the Without V2V curve decreases as $N$ increases. This is because there is no V2V communication in the network, while the eNBs'

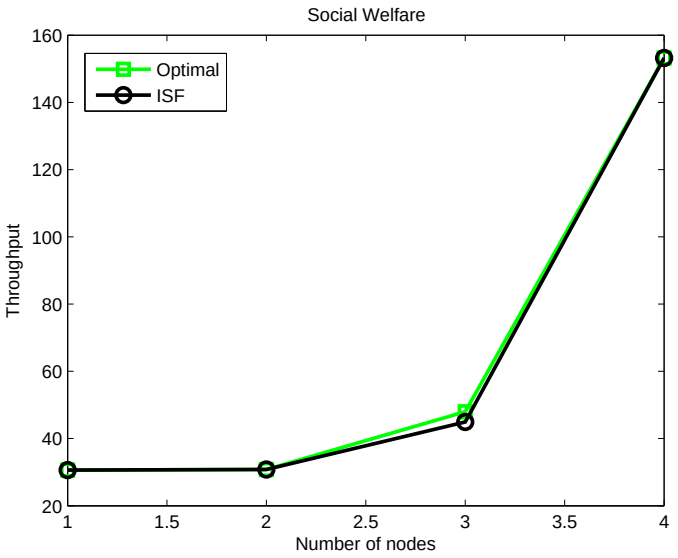



Figure 6.4: Optimal results comparison.

Table 6.1: Existence of stable matching ratio

| 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 |
|---|---|---|---|---|---|---|---|
| 93.9% | 81.2% | 67.5% | 58.6% | 53.3% | 45.2% | 43.5% | 39.8% |

capacities are limited. Thus, with more vehicles join the network, the average user performance decreases.

We also compare our distributive ISF algorithm with the centralized solution. We vary the number of vehicles $N$ from 1 to 4 and set the number of eNB as 1. As shown in Fig. 6.4, the ISF algorithm can achieve very close-optimal results, while the complexity of the ISF algorithm is much lower than the centralized optimization.

The connecting ratio is evaluated in Fig. 6.5. The ISF, ISF-Unweighted and ISF-one methods all achieve $100\%$ connectivity when $N > 20$. For the Without V2V method, since eNBs have limited service capacity, with more vehicles joining the network, only a certain number of vehicles can be served. As a result, the average vehicle performance decreases. While for the Proximity V2V method, its average connectivity is around $91\%$.

We have also provided the ratios of solvable SF instances in Table. 6.1 regarding different player numbers. With the increase of $N$ from 20 to 200, this ratio drops from $93.9\%$ to $39.8\%$. In

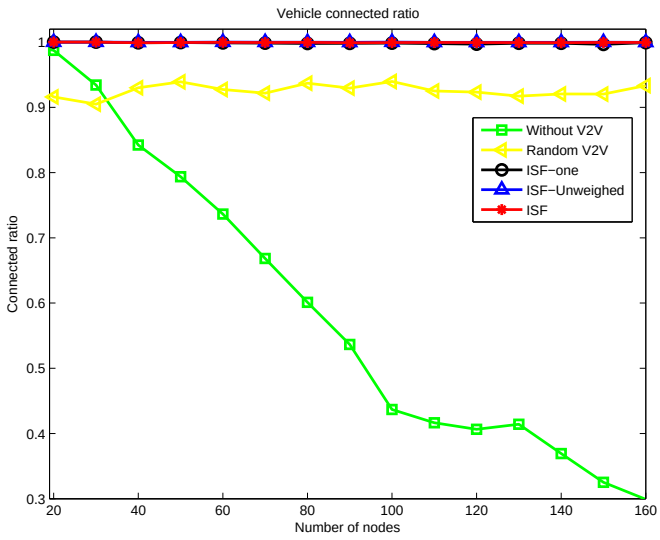



Figure 6.5: Ratio of connected vehicles.

addition, this decrease becomes slower as $N$ further increases.

## 6.6 Conclusion

This chapter has proposed a novel content sharing approach in the D2D based LTE-V networks. In particular, the communication latency and reliability issues that arise in the 802.11p-based V2X have been addressed, and the diversity of data classes is taken into consideration in the optimization problem. The proposed ISF algorithm can provide a stable result for the SF game in a distributive manner. The simulations have further demonstrated that the flexible many-to-many matching relations achieved by the ISF algorithm can greatly boost the network performance.

# Chapter 7

# Joint Radio and Computational Resource Allocation in Fog Computing: A Student Project Allocation Matching

The fog computing is considered as an emerging paradigm to complement the cloud computing platform for providing end-user with computational resources. With the fog computing, the service providers can better fulfill users' heterogeneous requirements by offloading the delay-sensitive tasks directly to the fog nodes (FNs). So far, most existing works are focused on either the radio or computational resource allocation in the fog computing. In this chapter, we investigate a joint radio and computational resource allocation problem to optimize the system performance and improve the user satisfaction. Some important factors, such as service delay, link quality, mandatory benefit and so on, are taken into considerations. Instead of the conventional centralized optimization, we propose the matching theory framework to offer a distributed solution. The formulated resource allocation problem is modeled as the student project allocation (SPA) game. The so-called SPA-(S,P) algorithm is implemented to find a stable matching for the SPA problem. In addition, the instability caused by the external effect is removed by the proposed the user-oriented cooperation (UOC) strategy.

## 7.1 Introduction

Cloud computing is an Internet-based computing platform that provides shared processing resources and data to computers and other devices on demand. The cloud computing and storage solutions can provide users and enterprises with various capabilities to store and process their data in the third-party data centers [19]. In particular, mobile cloud computing (MCC), as a combination of cloud computing, mobile computing and wireless networks, has made it possible for the mobile users to access the cloud resources to offload the computational tasks [20]. With the emerging of the new paradigm, namely, Internet of Things (IoT), a new range of services and applications have

been enabled, such as the connected vehicles, smart grid, wireless sensor networks and so on. Not only facing the volume, velocity and variety increase in the communication contents, but also the new communication requirements, such as location awareness, real-time mobility management, and so on. Therefore, it requires a new designed MCC framework to meet these critical requirements.

CISCO first proposed the idea of Fog Computing in 2014, as a platform that exists between the end devices and the cloud data centers. It provides computation, storage and communication resources to the nearby mobile users [21]. The fog computing brings the cloud closer to the end users, and is characterized by features such as low latency, large-scale distribution, support for mobility, heterogeneity, and federation [22]. Largely distributed at the network edge, the FNs provide storage, computation and communication services to the nearby users. An FN can be a cellular BS, a Wi-Fi AP or a femtocell router with upgraded CPU and memories in either fixed locations or being mobile [23]. FNs can communicate with nearby users for both control signal and actual data transmission. However, such direct communication may cause some security issues, such as eavesdropping and data hijack, without the surveillance from the cloud security system [89]. One way to avoid such security issues is to transfer them from the FNs to the cloud. In other words, the cloud, as the centralized controller of all the FNs and other resources, will be responsible for the security control, such as authentication, authorization and so on. Thus, the communication between FNs and users only involves the actual computation/storage data communication.

Currently, there are some obstacles that limit the performance of the fog computing. A lot of research has been done on studying how to efficiently allocate the cloud/fog computational resource to various users with heterogeneous requirements. Some frameworks, such as *MAUI*, *ThinkAir* and *Phone2Cloud* [90], have been proposed for offloading the computational tasks, which aim at reducing the energy cost by CPU and memories. A dynamic offloading framework for extending the lifetime of mobile users is discussed in [91]. The proposed algorithm, based on Lyapunov optimization, is able to extend the battery lifetime while satisfying the execution requirement. The energy efficiency issue of the mobile users is also discussed [92] [93]. For example, [92] studies the optimal offloading problem in the MCC under the stochastic wireless channel. By reconfiguring the

CPU frequency and varying the transmission rate, the objective is to conserve energy for the mobile devices. A closed-form solution for the optimal scheduling is derived. The above-mentioned works are all solved in a centralized way. However, the large scale and high mobility features of the IoT devices have made the centralized optimization less efficient considering the high computation complexity and heavy communication overhead. Game theory, as a popular distributive framework, has already been applied in the resource allocations of the MCC. For example, [94] discusses the resource management problem in the fog computing network, which is modeled as a 3-layer architecture: the FNs are in the upper layer, the data center operators are in the middle layer, and the users are in the bottom layer. A hierarchical Stackelberg game is proposed to find the network equilibrium. In [90], a joint radio and computation resource allocation in cloud computing is discussed, with users' energy and delay requirements considered. The optimization problem is solved in a distributive way. However, only one cloud provider is considered.

In this chapter, we want to study a jointly radio access and computational resources allocation when optimizing the system performance. To the best of our knowledge, this work is the first that investigates the joint radio and computational resource allocation problem with multiple cloud providers considered in the fog computing. The major contributions of this chapter are summarized as follows.

- We propose a joint radio and computational resource allocation framework for the fog computing. We allow users to express their requirements, regarding the delay and offloading data size, in the form of the mandatory offer to the cloud providers. On the hand, cloud providers try to find suitable FNs for offloading users' computation tasks to satisfy users' requirements.

- With the objective of optimizing the user satisfaction, we formulate this joint radio and computational resource allocation as an MINLP problem. In the formulation, system constraints, such as the service delay, transmission quality, power control and so on, are considered. We model the optimization problem using the SPA game, where cloud providers (modeled as the lecturers) own the radio/computation resources (modeled as the the projects), and are respon-

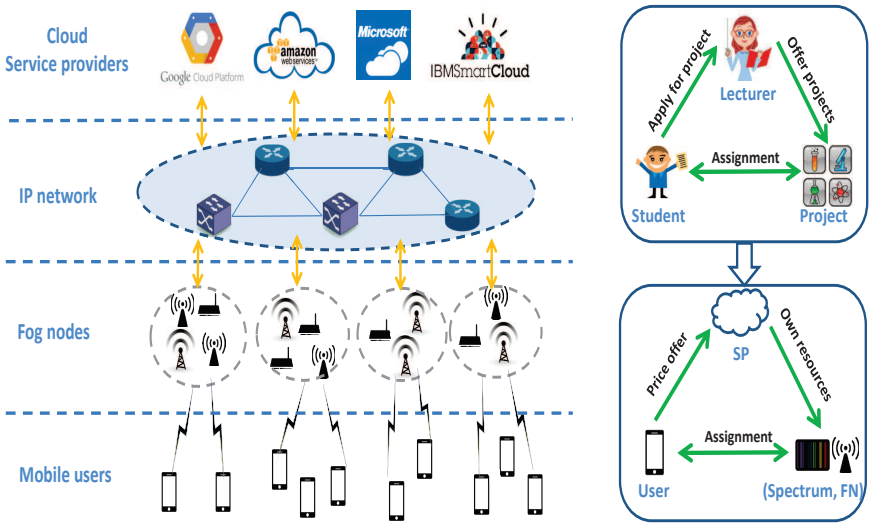
sible for the control information communications with the users (modeled as the students).

- We adopt the SPA-(S,P) algorithm to find a stable matching. In addition, the external effect, due to the inter-independence of players' preferences lists, is removed by the proposed UOC strategy. After the UOC procedure, the network stability is guaranteed and the system performance is further improved.

The rest of this chapter is organized as follows. In Section 7.2, we provide the system model. Then in Section 7.3, we formulate the optimization problem that aims at maximizing the system cost performance. Then, the SPA game is introduced to model the optimization problem, and the SPA-(S,P) algorithm is adopted as a distributed solution in Section 7.4. Simulations results are analyzed in Section 7.5 and conclusions are drawn in Section 7.6.

## 7.2 System Model

In this chapter, we assume a network comprised of a set of mobile users $\mathcal{U} = \{u_1, u_2, ..., u_M\}$, and a set of cloud service providers (SP) $\mathcal{SP} = \{sp_1, sp_2, ..., sp_N\}$, as shown in Fig. 7.1b. These SPs can meet different users with specific computing requirements regarding different data sizes and service latencies. For those users who are not delay sensitive, the computing tasks will be sent to the cloud, while for those users with certain delay requirements, the SPs will allocate the nearby fog nodes (FNs) to offload the computation tasks. However, the geography location is not the only factor that affects the service delay. In fact, the service latency consists of three time periods, which are the data transmitting time, the CPU processing time and the result receiving time. The transmitting and receiving periods are defined as the time used for sending data to FNs for processing and the time used for receiving the processed results, respectively. Such communication latency is not only related to the channel conditions but also affected by the data size of the computing task. On the other hand, the CPU processing time is related to the CPU rate of the FN. Thus, for any SP $sp_j$, when selecting an FN from the set $\mathcal{FN}^j = \{fn_1^j, fn_2^j, ..., fn_L^j\}$ for each user, it will jointly allocate its radio resource $\mathcal{W}^j = \{w_1^j, w_2^j, ..., w_K^j\}$ (the channel bandwidth) and computational resource




(a) The fog computing architecture. (b) The SPA model.

Figure 7.1: System model.

$\mathcal{C}^j = \{c_1^j, c_2^j, ..., c_L^j\}$ (the CPU cycle rate).

From the users' perspective, who have delay-sensitive tasks to process, will offer high prices to the SPs to compete for better resources (both the radio and computational resources). Intuitively, users who are requiring smaller latencies tend to offer higher prices. Besides, users also take their data sizes into consideration, since typically more data results in longer transmission period and longer CPU processing time. Notice here, the CPU cycles for the processing tasks are related to the data size but not exactly equal to it. Thus, we assume each user $u_i$ carries $D_i$ bits data, and the corresponding processing task requires $DC_i$ CPU cycles. Without loss of generality, we simply assume a linear relation between the $DC_i$ and $D_i$ [90].

The joint radio and computation resource allocation can be treated as the mapping between the user set $\mathcal{U}$ and the (radio,computation) resource pair set $\mathcal{RP}^j = \{(w_k^j, c_l^j) | \forall w_k^j \in \mathcal{W}^j, c_l^j \in \mathcal{C}^j\}$ owned by each SP $sp_j$, $sp_j \in \mathcal{SP}$. In the rest of this chapter, we may use $rp_{l,k}^j$ to denote the resource pair $(w_k^j, c_l^j)$ for simplicity. We represent such mapping relation with the binary value $\rho_{k,l}^{i,j}$, where $\rho_{k,l}^{i,j} = 1$ if $u_i$ is offloading a task to FN $fn_l^j$ using the channel $w_k^j$ owned by $sp_j$, and $\rho_{k,l}^{i,j} = 0$ otherwise. In order to optimize the joint resource allocation, we consider the profits of both users' and SPs', which will be discussed in the following sections.

### 7.2.1 User Satisfaction

One of the most important measurements that all SPs considers is the user experience or user satisfaction. As we mentioned previously, we are discussing a set of users with delay-sensitive tasks, so the service latency will be used as the user satisfaction measurement. However, before talking about the delay, we should first guarantee that the transmission quality between the users and FNs can meet the requirement. In other words, the SINR should be higher than a threshold $\Gamma_{\text{min}}$ in order to deliver the correct/complete data. We define the received SINR from $u_i$ at $fn_l^j$ using $w_k^j$ as

$$\Gamma_{k,l}^{i,j} = \frac{P_i g_{k,l}^{i,j}}{\sum_{u_{i'} \in \mathcal{U}, i' \neq i} \rho_{k,l}^{i',j} P_{i'} h_{k,l}^{i',j} + \sigma_N^2}, \tag{7.1}$$

where $P_i$ and $g_{k,l}^{i,j}$ are the transmission power and channel gain between $u_i$ and $fn_l^j$ using channel $w_k^j$, repectively. $h_{k,l}^{i',j}$ represents the interference channel gain from any other co-channel user $u_{i'}$ at $fn_j^j$. $\sigma_N^2$ represents the channel noise. We require $\Gamma_{k,l}^{i,j} \geq \Gamma_{\text{min}}$ for a successful transmission.

The data transmission rate from $u_i$ to $fn_l^j$ using $w_k^j$, if satisfying the SINR requirement, can be represented as

$$r_{k,l}^{i,j} = w_k^j \log(1 + \Gamma_{k,l}^{i,j}). \tag{7.2}$$

As we have discussed previously, the service delay consists of three time periods: the transmitting time $t_{\text{trans}}$, the CPU processing time $t_{\text{proc}}$, and the receiving time $t_{\text{recv}}$. Generally speaking, the received data size from the FN after processing is typically trivial compared to the original unprocessed data size. Besides, with no knowledge of the data process result, we cannot predict the exact size of the data to be returned. Thus, the receiving period is assumed to be very short, and we use a random variable $\delta t, \delta t \in [0, 1]$ to represent $t_{\text{recv}}$. When defining $t_{\text{trans}}$ and $t_{\text{proc}}$, we should consider the channel reuse and the CPU sharing among multiple users. We allow each channel to be shared among multiple users up to its capacity $q_R$, and also allow each FN to accommodate

multiple users to share its CPU up to its capacity $q_C$. Thus, the transmission rate for each user can be affected by the interference from the co-channel users, as represented in (7.1). In addition, the CPU processing rate for each user is also affected by the co-FN users. For simplicity, we assume each co-FN user will be allocated an equal share of the total CPU rate owned by the FN, denoted as $c_{k,l}^{i,j} = \frac{1}{\sum_{u_i \in \mathcal{U}} \rho_{k,l}^{i,j}} c_l^j$. Now, we can represent the service delay of $u_i$ when using the resource pair $(w_k^j, c_l^j)$ as follows:

$$t_{k,l}^{i,j} = t_{\text{trans}} + t_{\text{proc}} + t_{\text{recv}} = \frac{D_i}{r_{k,l}^{i,j}} + \frac{DC_i}{c_{k,l}^{i,j}} + \delta t. \tag{7.3}$$

### 7.2.2 SP Revenue

The monetary revenue is the incentive that SPs provide better services to the subscribed users. As another important factor to measure the system performance, the monetary offers from the users are considered, and are treated as SPs' benefits. As we have discussed, the price that each user offers is not only related to its delay requirement $T_{delay}$ but also its data size $D_i$. Without loss of generality, we assume a linear relation between the price and user's data size, and also a linear relation between the price and the inverse of user's delay requirement. Thus, the price offer from each user can be represented as

$$O_i = f(D_i, T_{delay}), \tag{7.4}$$

where $f(\cdot)$ is a monotonically increasing function for $D_i$ and monotonically decreasing function for $T_{delay}$. For Simplicity, we use the following function to define $f(D_i, T_{delay})$.

$$O_i = a\frac{D_i}{T_{delay}}, \tag{7.5}$$

where $a$ is a parameter with unit dollar/Mbps, and $O_i$ is the price that $u_i$ is willing to pay for any SP if matched.

Each SP serves more than one user, and thus receiving more than one offer. We define $sp_j$'s revenue as the summation of the mandatory offers collected from all the matched users, which is represented as

$$Rev_j = \sum_{u_i \in \mathcal{U}} \rho_{k,l}^{i,j} O_i. \tag{7.6}$$

## 7.3 Problem Formulation

In the previous section, we introduced two performance metrics, which are both essential for achieving a good resource allocation in the fog computing. The system objective is designed as a combination of the two metrics, and is named as the cost-performance (CP). The CP is defined as the ratio between the service delay and the price cost for each user, with the unit of sec/dollar. The total system CP $CP_{sys}$ is defined as the average value of all users' CP $CP(i)$, which can be represented as

$$CP_{sys} = \frac{\sum_{u_i \in \mathcal{U}} CP(i)}{M}, u_i \in \mathcal{U}, \tag{7.7}$$

where $CP(i)$ is the CP value for user $u_i$, and is defined as

$$CP(i) = \rho_{k,l}^{i,j} \frac{t_{k,l}^{i,j}}{O_i}. \tag{7.8}$$

Now we can formulate the optimization problem, which is shown as follows.

$$\max_{\rho^{i,j}_{k,l}} : \frac{\sum_{u_i \in \mathcal{U}} CP(i)}{M} \tag{7.9}$$

$$\text{s.t. : } \rho^{i,j}_{k,l} ti, j_{k,l} \leq T_{delay},$$

$$\forall u_i \in \mathcal{U}, rp^j_{l,k} \in \mathcal{RP}^j, sp_j \in \mathcal{SP}, \tag{7.10}$$

$$\rho^{i,j}_{k,l} \Gamma^{i,j}_{k,l} \geq \Gamma_{min},$$

$$\forall u_i \in \mathcal{U}, rp^j_{l,k} \in \mathcal{RP}^j, sp_j \in \mathcal{SP}, \tag{7.11}$$

$$\sum_{u_i \in \mathcal{U}, fn^j_l \in \mathcal{FN}^j} \rho^{i,j}_{k,l} \leq q_R, \forall w^j_k \in \mathcal{W}^j, sp_j \in \mathcal{SP}, \tag{7.12}$$

$$\sum_{u_i \in \mathcal{U}, w^j_k \in \mathcal{W}^j} \rho^{i,j}_{k,l} \leq q_C, \forall fn^j_l \in \mathcal{FN}^j, sp_j \in \mathcal{SP}, \tag{7.13}$$

$$\sum_{u_i \in \mathcal{U}, rp^j_{l,k} \in \mathcal{RP}^j} \rho^{i,j}_{k,l} \leq q_{SP}, and \forall sp_j \in \mathcal{SP}, \text{and} \tag{7.14}$$

$$\rho^{i,j}_{k,l} \in \{0, 1\}. \tag{7.15}$$

(7.9) is the system objective, representing the average cost performance for all the users. (7.10) represents the delay requirement for each user. (7.11) defines the minimum SINR requirement for each user. (7.12), (7.13) and (7.14) satisfy the capacity constraints for each channel, FN and SP, respectively.

Obviously, this optimization problem is an MINLP problem, which is NP-hard to solve in general [53]. Therefore, it motivates us to find a feasible solution. Thus, we introduce the SPA game as a distributive solution, which will be discussed in the next section.

## 7.4 A Student-Project Matching Game

The fact that the assignment of the radio and computational resources are coupled has motivated us to treat the (radio, computation) resource pair as one entity. We can enumerate all the possible combinations of the resource pairs, and then map the user set to the resource pair set. Ap-

parently, this process should be under the assistance of the SPs, who are responsible for the control signal communications with both the users and the resources.

A suitable matching model that offers such structure is the SPA problem [95], where various students will be assigned to various projects (owned by different lecturers) under the assistance of the lecturers. In this section, we first introduce how to model the proposed problem using the SPA model, and then implement the SPA-(S,P) algorithm to find a stable matching solution in Section 7.4.1. In addition, to deal with the externality that appears during the matching, we propose the UOC strategy to remove the external effect in Section 7.4.2.

### 7.4.1 Student-Project Allocation Modeling

In many university departments, students seek to undertake a project (e.g., senior design) from lecturers. Typically each lecturer will offer a variety of projects. Each student has preferences over the available projects, whilst a lecturer normally has some form of preferences over his/her projects and/or the students who find them acceptable. There are upper bounds on the number of students that can be assigned to a particular project and the number of students that can be assigned to a particular lecturer. In one variant of the SPA problem, the lecturers have preferences over the student-project pairs. This model is referred to as the SPA-(S,P) model, in which each lecturer has a preference list that depends on not only the students who find his/her projects acceptable but also the particular projects that these students would undertake [3].

We assume the SPs, the (radio, computation) resource pairs, and the users to be the lecturers, the projects and the students, respectively. The SPs offer available radio and CPU resource bundles, and the users propose to the SPs for the acceptable resource bundles. The SPs make decisions based on the monetary benefits that can be collected from the users regarding particular resource pairs. The stability notion here implies robustness to deviations that can benefit both the users and the resource pairs. The formal stability definition in this chapter is provided in Definition 7.1.

**Definition 7.1.** *Stability: A matching $\mathcal{M}$ is said be stable, if there's no blocking pair (BP). A pair*

*$(u_i, rp^j_{l,k})$ is defined as a BP that if all of the following conditions are satisfied:*

*(1) $u_i$ finds $rp^j_{l,k}$ acceptable;*

*(2) either $u_i$ is unmatched in $\mathcal{M}$, or $u_i$ prefers $rp^j_{l,k}$ to $\mathcal{M}(u_i)$;*

*(3) either*

*(3.1) $rp^j_{l,k}$ is under subscribed and either of the following three conditions is satisfied:*

*a)$\mathcal{M}(u_i) \in \mathcal{RP}^j$, and $sp_j$ prefers $(u_i, rp^j_{l,k})$ to $(u_i, \mathcal{M}(u_i))$; or*

*b)$\mathcal{M}(u_i) \notin \mathcal{RP}^j$ and $sp_j$ is under-subscribed; or*

*c)$\mathcal{M}(u_i) \notin \mathcal{RP}^j$ and $sp_j$ is full and $sp_j$ prefers $(u_i, rp^j_{l,k})$ to its current worst pair $(u_{wst}, rp^j_{wst})$;*

*(3.2) $rp^j_{l,k}$ is full and $sp_j$ prefers $(u_i, rp^j_{l,k})$ to the its current worst pair $(u_{wst}, rp^j_{wst})$, and either of the following two conditions is satisfied:*

*a)$\mathcal{M}(u_i) \notin \mathcal{RP}^j$;*

*b)$\mathcal{M}(u_i) \in \mathcal{RP}^j$ and $sp_j$ prefers $(u_i, rp^j_{l,k})$ to $(u_i, \mathcal{M}(u_i))$.*

In Definition 7.1, $\mathcal{M}(x)$ represents the partner/matching of the player $x$ in matching $\mathcal{M}$. More precisely, $\mathcal{M}(u_i) = rp^j_{l,k}, (w^j_k, c^j_l) \in \mathcal{RP}^j$.

To find a stable matching, both the users' and SPs' preference lists, denoted as $\mathcal{PL}^{user}$ and $\mathcal{PL}^{SP}$, need to be established first. During this procedure, the constraints (7.10) and (7.11) should be satisfied from the both users' and SPs' perspectives. After each player has found its acceptable set, it ranks the players in this acceptable set in a descending/ascending order according to the preferences. Intuitively, users prefer the resource pair most which can offer the smallest service latency. On the other hand, each resource pair will accommodate multiple users, and the coexistence will affect users' performances. For simplification, we assume these coexisting users will share the channel band and the CPU equally. Each SP, radio, and CPU have quotas, denoted as $q^{SP}$, $q^R$, and $q^C$, respectively. To calculate the potential service delay, each user assumes that it can get $\frac{1}{Q}$ share of the radio and CPU resources. The true performance for users may deviate from the evaluation, which causes the external effect during the matching (We'll address this issue in the Section 7.4.2). Thus, the preference of $u_i$ over the $rp^j_{l,k}$ is defined as the potential service delay $t'^{i,j}_{k,l}$, and is represented as

$$PL_i^{user}(j,k,l) = t'^{i,j}_{k,l} = t'_{\text{trans}} + t'_{\text{proc}} + t'_{\text{recv}}$$
$$= \frac{D_i}{\frac{1}{q^R} r'^{i,j}_{k,l}} + \frac{DC_i}{\frac{1}{q^R} c_l^j} + \delta t', \tag{7.16}$$

where $r'^{i,j}_{k,l}$ the data rate from $u_i$ to FN $fn_l^j$ when only $u_i$ is using the channel $w_k^j$, and is represented as $r'^{i,j}_{k,l} = w_k^j \log(1 + \frac{P_i g^{i,j}_{k,l}}{\sigma_N^2})$. $\delta t'$ is a real value within $[0, 1]$ that represents the possible period for sending back the processed result.

On the other hand, when selecting the users to match its resource pairs, the SPs not only consider the monetary benefit, but also the potential service delay. Users expect faster services, and SPs also pursue shorter service latencies in order to serve as many users as possible from the long-term consideration. Thus, the preference of the SP over any user is defined as the ratio between its price offer and the potential service delay, which is represented as

$$PL^{SP}_{j,k,l}(i) = \frac{O_i}{t'^{i,j}_{k,l}}. \tag{7.17}$$

With the preference lists set up, we can apply the SPA-(S,P) algorithm, as illustrated in Algorithm 7.1, to find a matching between the users and resources. The SPA-(S,P) algorithm consists of sequential propose and accept/reject operations. The convergence of the SPA-(S,P) algorithm is guaranteed and the proof is provided in [95].

### 7.4.2 User-Oriented Cooperation Strategy

Due to the inter-dependence of the preferences of users and resources, the matching yielded by the SPA-(S,P) algorithm is not necessarily stable. We call the matching framework with such inter-dependence between players' preferences as matching with externalities [4]. Our system objective is defined as the users' average cost performance, thus, we believe that it's reasonable to evaluate the stability notion solely from the user's side. In other words, we assume that only users have the

**Algorithm 7.1 SPA-(S,P) Algorithm**

**Input:** $\mathcal{U}, \mathcal{SP}, \mathcal{W}, \mathcal{FN}, \mathcal{PL}^{user}, \mathcal{PL}^{SP}$;
**Output:** Matching $\mathcal{M}$;
**Initialization:** set $\mathcal{M}$ empty, set all users free;

**while** some user $u_i$ is free and $u_i$ has a non-empty preference list **do**
  **for all** $u_i \in \mathcal{U}$ **do**
    $u_i$ proposes to the first entity $rp_{l,k}^j$ in $\mathcal{PL}_i^{user}$, and then remove $rp_{l,k}^j$ from $\mathcal{PL}_i^{user}$;
    $\mathcal{M} \leftarrow \mathcal{M} \cup (u_i, rp_{l,k}^j)$;
  **end for**
  **for all** $rp_{l,k}^j, rp_{l,k}^j \in \mathcal{RP}^j, sp_j \in \mathcal{SP}$ **do**
    **while** $rp_{l,k}^j$ is over-subscribed **do**
      Find the worst pair $(u_{\text{wst}}, rp_{\text{wst}})$ assigned to $rp_{l,k}^j$ in $sp_j$'s list;
      $\mathcal{M} \leftarrow \mathcal{M}/(u_{\text{wst}}, rp_{\text{wst}})$;
    **end while**
  **end for**
  **for all** $sp_j \in \mathcal{SP}$ **do**
    **while** $sp_j$ is over-subscribed **do**
      Find the worst pair $(u_{\text{wst}}, rp_{\text{wst}})$ in $sp_j$'s list;
      $\mathcal{M} \leftarrow \mathcal{M}/(u_{\text{wst}}, rp_{\text{wst}})$;
    **end while**
  **end for**
**end while**
Terminate with a matching $\mathcal{M}$.

incentive to make changes. Thus, a new "stability" notation should be defined among the users. Cooperations between the users are needed to transform the existing matching into stable again. We call such one-sided "stability" as the "Pareto Optimality" in matching theory [3]. The definition of Pareto optimal is provided as follows.

**Definition 7.2.** *Pareto Optimal: A matching is said to be Pareto Optimal if there is no other matching where some player is better off, whilst no player is worse off.*

Accordingly, the new BP definition in provided in Definition 7.3.

**Definition 7.3.** *BP in the one-sided matching: A user pair* $(u_i, u_j)$ *is defined as a BP, if both* $u_i$ *and* $u_j$ *are better off after exchanging their parters.*

To find such Pareto optimal matching, users again requires assistance from the SPs for the utility evaluation. The Pareto optimality is achieved through finite partner switch operations between the user pairs. As stated in Definition 7.2, the stability is reached when no player is better off without any other player being worse off. In other words, every swap operation should be beneficial to some

user(s) while doing no harm to the rest of users. Through finite swaps, we can finally reach a swap-free system, which means a stable system. We call such procedure as the UOC Strategy, and the details are illustrated in Algorithm 7.2.

**Algorithm 7.2 User-Oriented Cooperation (UOC) Strategy**

**Input:** Existing matching $\mathcal{M}_0$;
**Output:** Pareto optimal matching $\mathcal{M}_s$.

$\mathcal{M}_t = \mathcal{M}_0$;
**while** $\mathcal{M}_t$ is not Pareto optimal **do**
    Search the set of "unstable" (user,user) pairs $\mathcal{BP}$;
    **for all** $(u_{i1}, u_{i2}) \in \mathcal{BP}$ **do**
        **if** $\exists u \in \mathcal{M}_t(rp_{i1}) \cup \mathcal{M}_t(rp_{i2})$, $\Delta U(u) < 0$ **then**
            $(u_{i1}, u_{i2})$ are not allowed to switch partners;
        **else**
            $(u_{i1}, u_{i2})$ are allowed to switch partners;
        **end if**
    **end for**
    Find the optimal BP $(u^*_{i1}, u^*_{i2}) \in \mathcal{BP}$;
    $u^*_{i1}$ and $u^*_{i2}$ switch partners;
    $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t / \{(u^*_{i1}, \mathcal{M}_t(u^*_{i1})), (u^*_{i2}, \mathcal{M}_t(u^*_{i2}))\}$;
    $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t \cup \{(u^*_{i1}, \mathcal{M}_t(u^*_{i2})), (u^*_{i2}, \mathcal{M}_t(u^*_{i1}))\}$;
    Update $\mathcal{PL}^{user}$ based on $\mathcal{M}_t$;
**end while**
$\mathcal{M}_s = \mathcal{M}_t$.

In Algorithm 7.2, $rp_{i1} = \mathcal{M}_t(u_{i1}), rp_{i2} = \mathcal{M}_t(u_{i2})$. We define $U(x)$ as the utility function of user $x$, which is equal to its service delay. We define $\Delta U(x) = U(x)' - U(x)$, where $U(x)'$ is the utility after exchanging the partners. In other words, $\Delta U(x)$ represents $x$'s performance change, and is said to be improved if $\Delta U(x) > 0$ or decreased if $\Delta U(x) < 0$. A user pair is allowed to switch partners if and only if $\Delta U(x) \geq 0$ for any user $x$ that is affected in this switch. Then to find the optimal BP among all the BPs, we search for a BP which provides the highest the performance improvement. The performance here refers to the average time delay of all the users. We define the optimal BP as

$$(u^*_{i1}, u^*_{i2}) = \underset{(u_{i1}, u_{i2})}{\text{argmax}} \sum_{u \in \{u_{i1} \cup u_{i2} \cup \mathcal{M}_t(rp_{i1}) \cup \mathcal{M}_t(rp_{i2})\}} \Delta U(u), \tag{7.18}$$

We summarize the steps of the UOC strategy as follows. Firstly, we search all the "unstable" user-user pairs regarding the current matching. Secondly, we check whether the exchange/switch between such a pair is allowed (beneficial to related users). Thirdly, we find the pair that provides

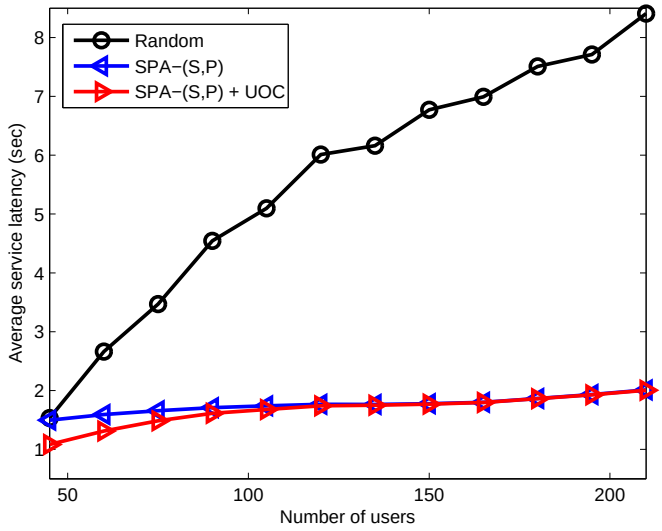



Figure 7.2: Users' average service latency.

the greatest throughput improvement, and allow this pair to switch partners. We keep searching for such BPs until we reach a trade-in-free network. The convergence of the UOC process is guaranteed by the irreversibility of each switch. Finally, UOC terminates with a Pareto optimal matching.

## 7.5 Performance Evaluation

In this Section, we evaluate the SPA-(S,P) algorithm and the UOC strategy regarding the users' service latency, SPs' profit and the system cost performance. In addition, the convergence of UOC will be analyzed.

We consider a network consisting of $N = 2$ SPs, and each equipped with $L = 5$ FNs. The network radius is $R = 1$ km. Assume a number of users $M, M \in [45, 210]$ are randomly distributed within the network. Each SP owns $K = 5$ channel bands, and each bandwidth is set to $w = 5$ MHz. The SINR requirements $\Gamma_{\text{min}}$ for users are uniformly distributed within $[20, 30]$ dB. We set equal capacity requirement for each channel and each FN, which is $q_R = q_C = 10$, and the SP's capacity is set as $q_{SP} = 80$. The service delay requirement $T_d$ for each user is uniformly distributed within $[6, 7]$ sec. The users' data sizes $D$ are uniformly distributed within $[2, 8]$ Mb, and the corresponding CPU cycles are set as $DC = D{*}10^4$. The CPU processing rates of the FNs are uniformly distributed

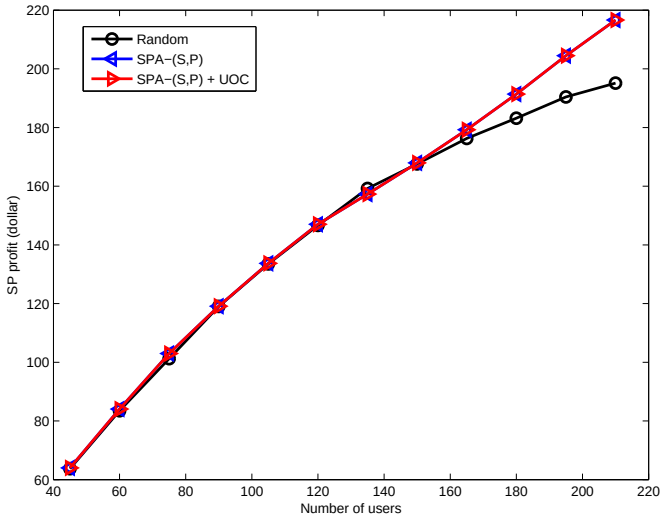



Figure 7.3: SPs' profit.

within $[5, 6] * 10^{10}$ cycles/sec. For the propagation gain $g$, we set the pass loss constant $C$ as $10^{-2}$, the path loss exponent $\alpha$ as $4$, the multipath fading gain as the exponential distribution with unit mean, and the shadowing gain as the log-normal distribution with $4$ dB deviation [60].

In Fig. 7.2 and Fig. 7.3, we evaluate the performance of the users and SPs, respectively. For comparison purposes, we use the Random method as the victim strategy, which refers to a random resource allocation between the users and the resource pairs. Fig. 7.2 evaluates the average service delay under the comparison of three methods: the Random method, the SPA-(S,P) method and the SPA-(S,P) with UOC method. We increase the number of users from $45$ to $210$ by the step of $15$. Apparently, the service latencies for all three strategies increase with the number of users. It is true since more users result in a smaller resource share for each, which thus leading to a higher average delay. Among the three methods, the Random curve gives the highest average latency, and is much higher than the other two methods. For the two matching curves, the SPA-(S,P) with UOC method is slightly better than the SPA-(S,P) method when $M < 150$, and is similar to the SPA-(S,P) method when $M > 150$. Thus, we can find the UOC strategy can further improve user performance while guaranteeing network stability. Besides, this improvement is less apparent when the user number is close to or has reached the network capacity $M = 160$. The network capacity refers to the maximum

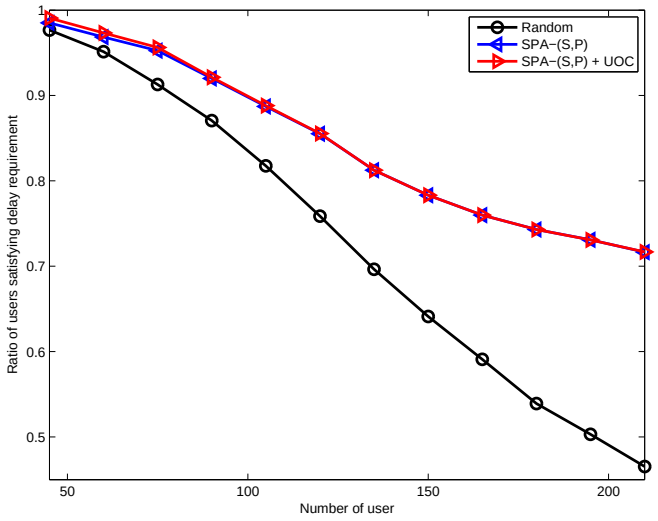



Figure 7.4: The ratio of users satisfying delay requirement.

number of users that the SPs can accommodate. As shown in Fig. 7.3, the average profits gained by the SPs are almost the same for all the three methods when $M < 150$, and after $M > 150$ both the SPA-(S,P) method and the SPA-(S,P) with UOC method outperform the Random method. Before the user number reaches the network capacity, almost all the users can be matched to a resource pair using different methods. Thus, SPs can still gain all the money. However, when the users are more than the network capacity, then users need to compete with other. Thus, which users will be kicked off and which ones will stay? As we discussed in Section 7.2, the users who have higher latency requirements offer higher prices, thus making them more likely to be selected by the SPs. In turn, the users with higher offers can make the SPs gain more profits. That is why the two matching curves can beat the Random method when $M > 150$.

The user satisfaction is evaluated in Fig. 7.4, w.r.t. the ratio of users whose actual service latencies can meet their requirements. Apparently, the ratio of satisfied users decreases as more users join the network. The starting points of all the three methods are almost $100\%$, and then the Random method drops faster than the two matching algorithms. The two matching curves drop with similar speeds, and drop slower after $M > 150$. At the end point when $M = 210$, both the SPA-(S,P) with UOC method and the SPA-(S,P) method reach around $75\%$ satisfaction ratio, while

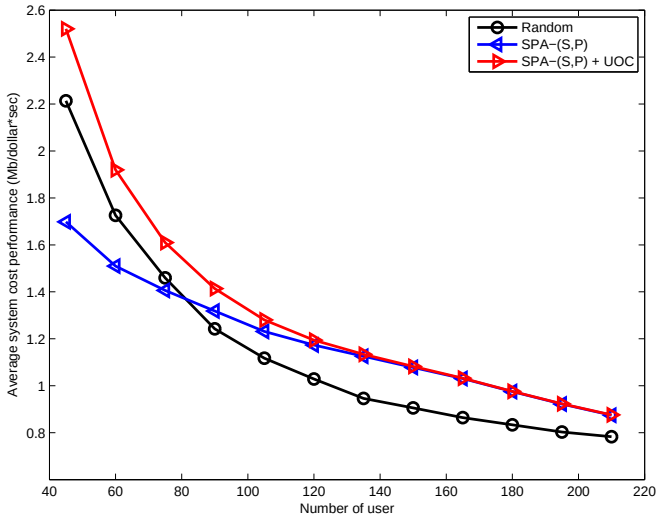



Figure 7.5: The system cost performance.

the Random method falls below $50\%$. Fig. 7.4, together with the average delay evaluation shown in Fig. 7.2, shows that our proposed matching algorithms not only consider the SPs' benefit, but also take each individual user's performance into consideration.

In Fig. 7.5, we evaluate the system cost performance under three methods. It's an joint consideration of both users' and SPs' benefits, with the objective to allocate the best resources to users who want them most (i.e., who offer the highest prices). As shown in Fig. 7.5, when $M < 75$, the SPA-(S,P) with UOC method outperforms the other two, while the Random allocation beats the SPA-(S,P) method. This happens because in the SPA-(S,P) method, users first propose to their favorite resources, and thus some good resources may receive many more proposals than the other resources. When the user number is relatively small and there are sufficient resources, the previously good resources, who are matched with more users, may be not so good compared with those resources who are not matched. Thus, when the user number is small, the SPA-(S,P) method is worse than the Random method. After $M > 80$, both matching algorithms are better than the Random one. The SPA-(S,P) with UOC method outperforms the SPA-(S,P) method when $M < 150$. The performances of all the three curves are decreasing with the increase of $M$. It is reasonable since the price offers and the resources are unchanged, and more users may result in less resource

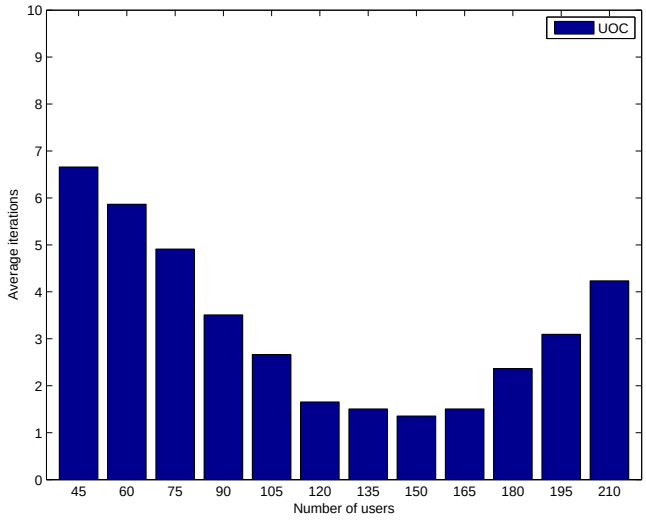



Figure 7.6: UOC Convergence Analysis.

for each, which results in the average cost performance reduce.

Lastly, the convergence of the proposed UOC strategy is analyzed in Fig. 7.6. The iteration of users switches during the UOC method is considered the measurement of its convergence. As we discussed in Section 7.4, the convergence of UOC to a Pareto optimal matching is guaranteed since each switch is inrevertible. By each switch, some users can switch to better resource pairs, which were previously under-evaluated. With so many switch options, our proposed UOC method selects the currently best pair to switch. It's not hard to understand, such pair selection procedure can greatly reduce the number of switches. We can see a decrease of the iteration number with the increase of the user number when $M < 150$, and then an increase when $M > 150$. Notice that the network capacity is $M = 160$, and the user increase step is 15, which means when $M > 150$ the user number exceeds the network capacity. So before the user number exceeds the capacity, SPs have some resource pairs who have spare rooms for more users. Thus with such rooms, users have more chances to improve their performance by switching. Thus, it explains that with the decrease when $M < 160$. Then, after the user number has reached the network capacity, there are more users who can not get any resource. Thus the competition between these unmatched users and the matched users will bring more switches. Thus, after $M > 160$, with more unmatched users, the

switch numbers start to increase. The average switch number is limited by 10 under this network setting, which is in fact a trivial number.

## 7.6 Conclusion

In this chapter, we have studied the joint radio and computational resource allocation problem in the fog computing. With the proposed semi-distributive SPA framework, we have modeled the interactions between the mobile users, the SPs and the FNs. System requirements, such as the transmission quality, service latency, and maximum power requirement, have been addressed through the representation of players' preference lists. The proposed SPA-(S,P) algorithm together with the UOC procedure can guarantee a stable matching. The simulation results have demonstrated that our proposed framework can provide close-optimal performance from both the users' and the SPs' perspectives.

# Chapter 8

# Cyclic Three-Sided Matching Game Inspired Wireless Network Virtualization

The network virtualization has drawn substantial attention from the wireless communication field recently. The key idea of the wireless network virtualization is about the abstraction, isolation and sharing of the wireless resources, including both the radio resources and the physical infrastructures. By such kind of isolation and sharing, the network virtualization can provide great flexibility, higher network efficiency and new services to the wireless networks. Traditionally, the virtual resource allocation job is controlled by the so-called virtual wireless network controller, who sell the service combos (consisting of the radio resources and physical infrastructures) to the service providers (SPs) in a wholesale way. The SPs then allocate the purchased services to their subscribed mobile users. However, such centralized allocation decouples the service generation from the user resource allocation. In this chapter, we propose a three-sided (3D) matching game to model the interactions between the radio resource, physical infrastructure and mobile users.

## 8.1 Introduction

Nowadays, virtualization has become a popular concept applied in many areas, such as the virtual memory, virtual machine and virtual data center. Virtualization refers to the abstraction and sharing of resources among different parties. It offers great network flexibility, maximizes network utilization, as well as inspires new services and products [24]. On the other hand, the mobile wireless traffic is expected to grow exponentially due to the massive user number and the rich communication contents. By extending the network virtualization into cellular networks, and more specifically, by abstracting, slicing and sharing the physical infrastructures and radio resources, the wireless traffic can be relieved. This paradigm is commonly referred to as the wireless network virtualization [25].

Wireless network virtualization decouples the services from the network resources, and en-

ables the sharing of both radio spectrum and physical infrastructures. Thus, the network utilization can be improved. In addition, with unified control and allocation of resources, new wireless services can be innovated in a more efficient way. Despite the visible potential of the wireless network virtualization, several design challenges remain to be addressed, which include the isolation, control signaling, resource discovery and allocation, mobility management, security and so on [25]. In particular, the resource management challenge calls for the comprehensive efforts, as it decides how the virtual networks are embedded on top of the physical networks, and directly affects the network utilization.

A popular way to define different roles in the wireless network virtualization is by classifying these roles into the service providers (SPs), the mobile virtual network operators (MVNOs), the infrastructure providers (InPs) and the end users [25]. The InPs own the infrastructure resources, while the MVNOs own the spectrum resources and are responsible for creating and managing the virtual resources. The SPs then allocate specific services, such as VoIP, video call and so on, to the end users, which are rented or purchased from the MVNOs in a wholesale way. Such purchasing and reselling jobs of the SPs are in fact the resource allocations between the MVNOs, SPS, and the end users [1]. The traditional resource allocation solution in the wireless network virtualization is to configure the virtual resource/services packages first and then offer the off-the-rack services to the users [96]. Such approaches decouple the virtual service generation procedure (accomplished by the MVNOs) from the user service assignment procedure (accomplished by the SPs), which lack the flexibility regarding some specific user requirements.

In this chapter, we jointly consider the service generation and service allocation procedures, and more specifically, we propose a matching-based resource allocation framework that matches the three network elements: spectrum, infrastructure and end users, simultaneously. The major contributions of this chapter are summarized as follows.

1) We propose a resource allocation framework in the wireless network virtualization. Un-

[1]In this chapter, we assume that the MVNOs create and manage the virtual resources, including both spectrum and infrastructure resources.

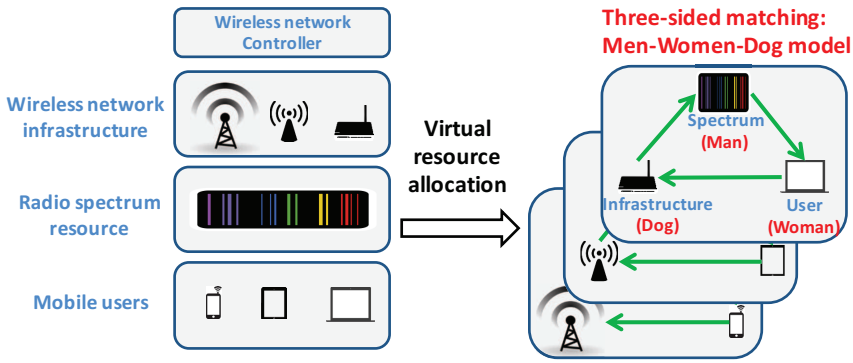



Figure 8.1: System Model

like the conventional decoupled service generation and service allocation, we tackle this problem by modeling it as a three-sided matching between the radio spectrum, physical infrastructures and mobile users.

2) With joint consideration of the user satisfaction, SPs' revenue and the system cost-performance, we formulate the three-sided matching as an optimization problem. We model the optimization problem by exploiting the three-dimensional stable marriage (3DSM) model.

3) The restricted three-sided matching with size and cyclic preference (R-TMSC) model can be solved by the proposed spectrum-oriented R-TMSC algorithm, 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. In Section 8.2, we provide the system model. Then in Section 8.3, we formulate the optimization problem that aiming at maximizing the system cost performance. Then, a three-sided matching approach is proposed to solve the optimization problem in Section 8.4. Simulations results are analyzed in Section 8.5 and conclusions are drawn in Section 8.6.

## 8.2 System Model

In this chapter, we assume a wireless network virtualization framework consisting of the SPs, MVNOs and end users, as shown in Fig. 8.1. We assume a set of spectrum bands $\mathcal{S} =$

$\{s_1, s_2, ..., s_K\}$ available for sharing, where $K$ is the number of bands. Each band is assumed to have identical bandwidth. A set of $N$ physical infrastructures $\mathcal{B} = \{b_1, b_2, ..., b_N\}$, which include the base stations, APs, core network elements and so on, can be rented. The group of subscribed mobile users are represented by $\mathcal{U} = \{u_1, u_2, ..., u_M\}$, where $M$ is the number of all users. 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, which was initially owned by the cellular network operators, can now be shared between multiple infrastructures, and is limited by its capacity $q^s$ regarding the number of users. On the other hand, each infrastructure is also shared between multiple bands, and is limited by its capacity $q^b$ regarding the number of users. In addition, any band assigned with some infrastructure can be shared between multiple users. In other words, the matching between $\mathcal{S}$ and $\mathcal{B}$ is a many-to-many matching, while the matching between the (band, infrastructure) resource pairs and the users $\mathcal{U}$ is one-to-many matching.

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

### 8.2.1 User Experience

One of the most important things that the SPs concern about is the user experience or the user satisfaction. We use the user's SINR to measure its user satisfaction in this chapter. Since the channel condition primarily relies on the transmitter and receiver, we define the user experience as the transmission SINR between the user and the infrastructure. It can be represented as

$$\Gamma_{i,j} = \frac{P_{i,j} g_{i,j}}{\sigma_I^2 + \sigma_N^2}, \tag{8.1}$$

where $\Gamma_{i,j}$ is the received SINR of infrastructure $b_j$ from user $u_i$. The UL transmission is discussed in this chapter. Thus the SINR will be the received SINR at the infrastructure side. $P_{i,j}$ and $g_{i,j}$ are the transmission power and channel gain between $u_i$ and $b_j$, respectively. $\sigma_N^2$ represents the channel noise, and $\sigma_I^2$ represents the channel interference from the other mobile users due to channel reuse.

### 8.2.2 SP Revenue

The monetary revenue is the incentive that SPs provide better services to the subscribed users. In this chapter, another factor that we use to measure the system performance is the revenue that SPs earn from the users. We assume each user offers a price based on its desired rate, thus the SPs would prefer to serve the ones with higher offers. We define SP's revenue as the summation of all the offers collected from the matched users, which is represented as

$$Revn = \sum_{u_i \in \mathcal{U}} O_i = \sum_{u_i \in \mathcal{U}} \alpha r_i, \tag{8.2}$$

where $O_i$ is the price that user $u_i$ offers to any spectrum, based on its desired transmission rate $r_i$. $\alpha$ is a parameter with the unit dollar/Mbps.

## 8.3 Problem Formulation

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

$$CP_{sys} = \frac{\sum_{u_i \in \mathcal{U}} CP(i)}{M}, u_i \in \mathcal{U}, \tag{8.3}$$

where $CP(i)$ is the CP value for user $u_i$, and the system $CP_{sys}$ is the averaged value of all the users, i.e.,

$$CP(i) = \frac{\sum_{b_j \in \mathcal{B}, s_k \in \mathcal{S}} \rho_{i,j,k} s_k \log(1 + \Gamma_{i,j}^k)}{O_i}, \tag{8.4}$$

where $\rho_{i,j,k}$ is a binary value, which is equal to 1 if $u_i$ is utilizing the channel band $s_k$ as its DL transmission through the infrastructure $b_j$, and 0 otherwise. $\Gamma_{i,j}^k$ is the actual SINR of user $u_i$ if

matched with infrastructure $b_i$ and spectrum $s_k$ (considering the interference from other users that share the same $s_k$ and $b_j$), which is represented as

$$\Gamma_{i,j}^{k} = \frac{P_{i,j} g_{i,j}}{\sigma_N^2 + \sigma_I^2} = \frac{P_{i,j} g_{i,j}}{\sigma_N^2 + \sum_{i' \neq i} \rho_{i',j,k} P_{i',j} g_{i',j}}. \tag{8.5}$$

Now we can formulate the optimization problem, which is shown as follows.

$$\textbf{max :} \underset{\rho_{i,j,k}}{} CP_{sys} \tag{8.6}$$

$$\textbf{s.t. :} \sum_{u_i \in \mathcal{U}, b_j \in \mathcal{B}} \rho_{i,j,k} \leq q^s, \forall s_k \in \mathcal{S}, \tag{8.7}$$

$$\sum_{u_i \in \mathcal{U}, s_k \in \mathcal{S}} \rho_{i,j,k} \leq q^b, \forall b_j \in \mathcal{B}, \tag{8.8}$$

$$\Gamma_{i,j} \geq \Gamma_{min}, \forall u_i \in \mathcal{U}, b_j \in \mathcal{B}, \text{and} \tag{8.9}$$

$$\rho_{i,j,k} \in \{0, 1\}, \tag{8.10}$$

where (8.6) is the system objective, representing the overall cost performance. (8.7) and (8.8) are the capacity constraints for the spectrum and infrastructure, respectively. (8.9) defines the minimum SINR requirement for each user.

Obviously, this optimization problem is an MINLP problem [2], which is NP-hard to solve [53]. Therefore, it motivates us to find a feasible solution. In the next section, we introduce a matching-based distributive solution: the man-woman-dog model to solve the problem.

## 8.4 Cyclic Three-sided Matching Game

Three-sided relationship is very common in the social and economic fields, e.g., the supplier-firm-buyer relationship, the kidney exchange problem, and so on. Generally, the three-sided matching can be treated as the 3D generalization of the SM model [3], where the three types of matching

[2] The nonlinearity is caused by $\Gamma_{i,j}^{k}$ in the objective function.

agents can be referred to as the men, women and dogs. This three-dimensional variant of the SM game is usually referred to as the 3DSM problem. Primarily, there are two types of models studying the 3DSM problem, depending on the nature of agents' preferences. For the first model, each agent may rank the pairs of other agents that they are prepared to form triples with in order of preferences. In the other model, the agents' preference lists involve only one type of agents (e.g., men only rank women in order of preferences, women only rank dogs, and dogs only rank men). This type of preference lists are commonly referred to as the cyclic preferences. In this section, we are discussing the 3DSM model with cyclic preferences.

### 8.4.1 3DSM Model

As an intriguing variant of the 3DSM, the three-dimensional stable marriage problem with cyclic preferences (3DSM-CYC) model refers to case that the matching agents' preference lists involve only one type of agents (instead of pairs of agents). The problem of deciding whether a given instance of the 3DSM-CYC model admits a weakly or strongly stable matching is NP-complete as studied by [97]. However, in [98], Cui and Jia studied an interesting variant of the 3DSM-CYC model, where a stable matching can always be found with certain restrictions in the preference lists.

In the man-woman-dog instance, we assume that men only rank women in order of preferences, women only rank dogs, and dogs only rank men. We use $m$, $w$ and $d$ to represent man, woman and dog, respectively. Each agent can be matched to the other type of agents in its preference list up to its capacity. We define the three-sided matching with size and cyclic preference problem (TMSC) problem as follows:

**Definition 8.1.** ***Three-sided matching with size and cyclic preference problem (TMSC):*** *the three-sided matching problem of TMSC is to find a matching* $M = \{(m_i, w_j, d_k)\}$ *with the maximum*

*cardinality:*

$$\textbf{\textit{max :}}\ |M| \tag{8.11}$$

$$\textbf{\textit{s.t. :}}\ \mathcal{N}(M, m_i) \leq c_m, \tag{8.12}$$

$$\mathcal{N}(M, w_j) \leq c_w, and \tag{8.13}$$

$$\mathcal{N}(M, d_k) \leq c_d, \tag{8.14}$$

where (8.11) represents the cardinality of the matching $M$ (i.e., the number of $(m, w, d)$ triples in the matching). $\mathcal{N}(M, x)$ represents the number of partners that $x$ has in the matching $M$[3] (8.12), (8.13) and (8.14) represent the capacity of each man, woman, and dog, respectively. As claimed in [98], the TMSC problem is NP-hard.

In order to solve the TMSC problem, we add some extra restrictions to transform it into the R-TMSC model as follows: (1) The preference lists of men are derived from a master preference list. This master list is the set of all the women with strict order (e.g., according to the age), and then all men's preference lists are derived from this master list. (2) The dogs are indifferent with the men, thus for each dog, the men in its preference list forms a tie. We refer to this restricted TMSC model, satisfying both (1) and (2), as the R-TMSC model. Before we discuss the property of the R-TMSC model, we provide some basic matching definitions. As a fundamental requirement for a matching result, the concept of stability is widely used in algorithm design. A matching is said to be stable is if it admits no BP in case of the two-sided matching, and thus no blocking triple in case of the three-sided matching. Under restriction (1) and (2), the definition of the blocking triple can be updated as in Definition 8.2. Intuitively, a blocking triple consists of one man, one woman and one dog, each of which has the desire to be matched to each other as a triple instead of staying with their current partners.

**Definition 8.2.** ***Blocking triple in 3DSM:***

$\{\mathcal{N}(M, d_k) \leq c_d\} \wedge \{M(m_i) = \emptyset \vee w_j \succ_{m_i} M(m_i)\} \wedge$

$\{M(w_j) = \emptyset \vee d_k \succ_{w_j} M(w_j)\}$.

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

Finding the maximum cardinality matching of the R-TMSC problem is still NP-hard as proved in [98]. However, regarding the stability perspective of the matching, we have a nice conclusion, which is stated in Theorem 8.3. Due to page limitation, the proof of Theorem 8.3 is not provided in this chapter. Please refer to [98] for the detailed proof.

**Theorem 8.3.** *The user-oriented R-TMSC algorithm will stop and output a stable matching after a finite number of steps.*

### 8.4.2 Stable Matching solution for R-TMSC

We assume the spectrum to be man, user to be woman, and infrastructure to be dog. Following the R-TMSC model, we define the preference lists for the spectrum, user and infrastructure. The preference lists of the spectrum over users are derived from the master list that ranks all users' offers in a descending order [4]. In this case, all the spectrum resources create the same preference lists, which are represented as follows

$$PL_s(k,i) = O_i, \forall s_k \in \mathcal{S}. \tag{8.15}$$

On the other hand, the users rank the acceptable infrastructures according to the service quality (the acceptable set is generated by applying (8.9)), which is measured by the SINR $\Gamma_{i,j}$ [5]. We represent the preference lists of users as follows

$$PL_u(i,j) = \Gamma_{i,j}, \forall u_i \in \mathcal{U}, b_j \in \mathcal{B}. \tag{8.16}$$

According the R-TMSC model, the infrastructures are indifferent with all the spectrum resources. In other words, the preference list of any infrastructure consists of identical rankings, which can be represented as

$$PL_b(j,k) = 1, \forall b_j \in \mathcal{B}, s_k \in \mathcal{S}. \tag{8.17}$$

[4] Here, the spectrum is on behalf of the SP's interest/revenue.

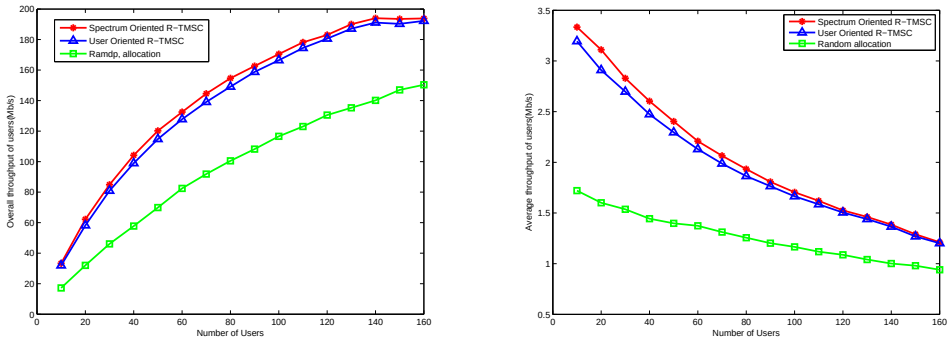
[5] We assume the $\sigma_I^2 = 0$ in building the preference lists, since the matching actions of other users are not known in advance to any user.

After generating all agents' preference lists, we propose the spectrum-oriented R-TMSC algorithm. Slightly different from the R-TMSC algorithm discussed in [98], we tailor it to fit our problem setting. The basic idea of this algorithm is to iteratively search for the "best" triple and add this triple to the matching, which starts from empty. Each "best" triple (in the form of $(u, b, s)$) is generated by first selecting a spectrum satisfying the requirements, and then this selected spectrum decides the best user that can meet the requirements, and finally this selected user decides the best qualified infrastructure. The detailed procedure is provided in Algorithm 8.1.

In Algorithm 8.1, we use $A^{+1}(M, s_k) = \{u_i | u_i \succ_{s_k} M(s_k)\}$to represent all the users that spectrum $s_k$ prefers to its own partner $M(s_k)$. We use $A^{+1}(M, u_i) = \{b_j | b_j \succ_{u_i} M(u_i)\}$ to represent all the infrastructures that user $u_i$ prefers to its own partner $M(u_i)$. Then we define $A^{-1}(M, u_i) = \{b_j, |\mathcal{N}(M, b_j) \leq q^b\}$, which contains all the infrastructures that still have rooms to accept $u_i$. We also define $A^{-2}(M, s_k) = \{u_i | A^{+1}(M, u_i) \cap A^{-1}(M, u_i) \neq \emptyset\}$, which represents any user $u_i$, such that there exists a $b_j$, who still has room to accept $u_i$, and $u_i$ prefers it to its current partner.

**Algorithm 8.1 Spectrum-oriented R-TMSC Matching**

**Input:** $\mathcal{U}, \mathcal{B}, \mathcal{S}$
**Output:** $M$
**Initialization**;
Construct the preference lists $PL_u$, $PL_b$, and $PL_s$;
$M = \emptyset$, $flag = 1$;
**while** $flag = 1$ **do**
  Set $flag = 0$;
  **for all** each $s_k \in \mathcal{S}$ **do**
    $\mathcal{U}' = A^{+1}(M, s_k) \cap A^{-2}(M, s_k)$;
    **if** $\mathcal{U}' \neq \emptyset$ **then**
      $u_i = Head(\mathcal{U}', s_k)$;
      $\mathcal{B}' = A^{+1}(M, u_i) \cap A^{-1}(M, u_i)$;
      $b_j = Head(\mathcal{B}', u_i)$;
    **end if**
    **if** $\mathcal{N}(M, s_k) == 1$ **then**
      $M = M \cup \{M(s_k), M(M(s_k)), s_k\}$;
      $flag = 1$;
    **end if**
    **if** $\mathcal{N}(M, u_i) == 1$ **then**
      $M = M \cup \{u_i, M(u_i), *\}$;
      $flag = 1$;
    **end if**
    $M = M \cup \{u_i, b_j, s_k\}$;
  **end for**
**end while**
**End of algorithm**;




(a) Overall user throughput.

(b) Average user throughput.

Figure 8.2: User throughput Analysis.

## 8.5 Performance Evaluation

In this section, we evaluate the proposed the spectrum-oriented R-TMSC algorithm by comparing it with the user-oriented R-TMSC algorithm, the decoupled allocation, and the random allocation, regarding the user throughput, spectrum revenue and system cost performance. 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 finding the best qualified user for this spectrum, and finally finding the best qualified infrastructure for this user. The user-oriented R-TMSC algorithm operates similarly by adding one triple per time, but the triple is generated from finding a qualified user first. The decoupled allocation is the scheme that decouples the service generation from the service assignment. For simplicity, we assume a random combination of the spectrum $\mathcal{S}$ and infrastructure $\mathcal{B}$ resources. Then, with the generated services (i.e., the resource pairs), it will be assigned to the users $\mathcal{U}$ using the GS matching algorithm, which can be used to find a stable solution for the two-sided matching problems [3].

We assume a circle network with the radius of $R = 800$ m, consisting of $M \in [0, 160]$ mobile users, $N = 7$ infrastructures, and $K = 20$ spectrum bands. The bandwidth of each band is set to be $5$ MHz. The capacity of each infrastructure is $20$, while the capacity of each frequency band is $7$. The minimum SINR requirements for all mobile users are set as $25$ dB. For the propagation gain $g = C\beta\zeta d^{-\alpha}$, we set the pass loss constant $C$ as $10^{-2}$, the path loss exponent $\alpha$ as $4$, the

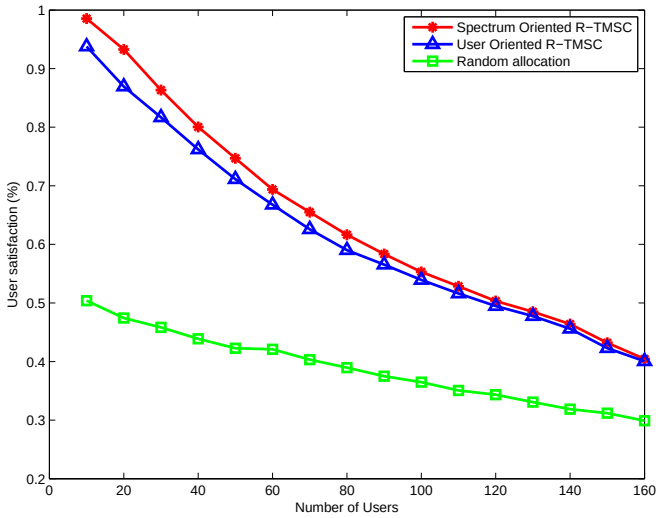



Figure 8.3: User satisfaction.

multipath fading gain $\beta$ as the exponential distribution with unit mean, and the shadowing gain $\zeta$ as the log-normal distribution with $4$ dB deviation.

In Fig. 8.2a and Fig. 8.2b, the overall and average throughput of users are evaluated. We increase the user number from $30$ to $170$ by $20$. As shown in Fig. 8.2a, the network throughput increases as more users join the network. It's reasonable since spectrum is reused between users and more users can bring higher spectrum utilization. On the other hand, Fig. 8.2b shows that the average user performance decreases slightly as more users joint. This is due to the interference caused by users who are sharing the same infrastructure and spectrum. Similar conclusions can be drawn from Fig. 8.2a and Fig. 8.2b, that the spectrum-oriented R-TMSC algorithm outperforms the user-oriented one slightly, and both outperform the decouple and random allocation methods a lot, w.r.t. both the overall and average throughput.

Fig. 8.3 gives another insight on the performance from the perspective of the user satisfaction. The user satisfaction is defined as the ratio between the actual transmission rate and the expected data rate. As discussed in Section 8.1, users make offers to the SP according to the expected rate. As a result, any user who has a higher rate requirement will offer a higher price. In turn, users who provider higher offers are more preferred by the spectrum, and thus better served. It is obvious that

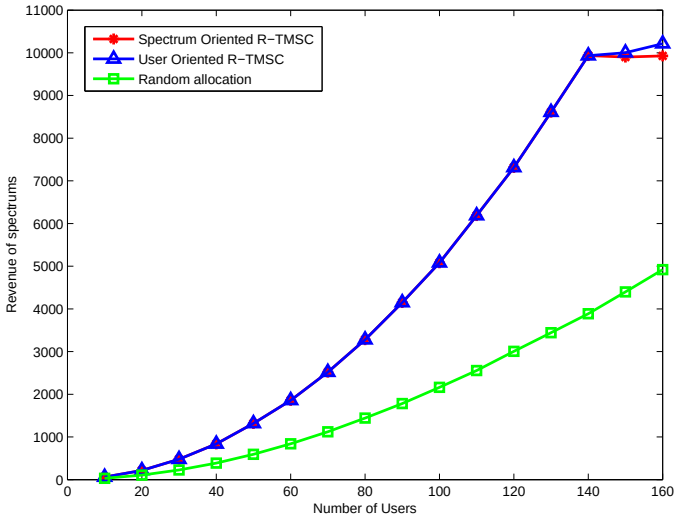



Figure 8.4: Spectrum Revenue.

as more users join the network, the user satisfaction decreases. With more users sharing the same radio and infrastructure resources, the interference grows and leads to the performance degradation. However, the spectrum-oriented algorithm still outperforms the user-oriented one, and both achieve better performance than the decoupled and random allocation.

Fig. 8.4 evaluates the revenue of the SPs. The revenue is defined as the summation of the all the income collected from the matched users. Apparently, more users will bring more revenue. A turning point can be found for both algorithms when the user number $M$ becomes $140$. This turning point is related to the network capacity, since both the infrastructure and the spectrum have the same capacity of $140$. Thus, after the user number reaches $140$, the revenue no loner increases linear with the increase of user number. Both matching algorithms achieve the same system revenue since all the users are accepted and served when $M < 140$.

In Fig. 8.5, we analyze the cost performance of the system. As defined in Section 8.3, the system objective is to optimize the system cost performance, which is calculated by averaging all the users' transmission rates over their price offers. This cost performance metric not only evaluates how good the users perform, but also evaluates how much benefit the SPs make. As can be found in Fig. 8.5, the average CP value decreases as more users join. This is caused by the average user

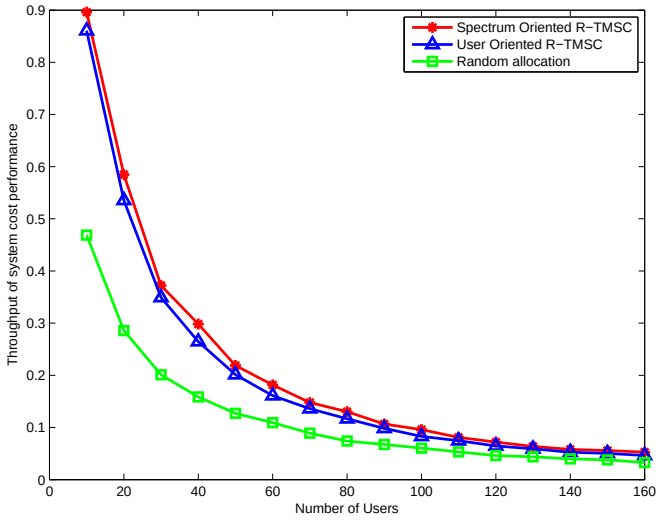



Figure 8.5: System cost performance

throughput reduce as indicated in Fig. 8.2b and Fig. 8.3. Again, the spectrum-oriented algorithm outperforms the user-oriented one. In addition, both matching algorithms beat the random allocation.

## 8.6 Conclusion

In this chapter, we have developed a framework to solve the resource allocation problem in the wireless network virtualization. Utilizing the three-sided matching: man-woman-dog model, we are able to formulate the interactions between the radio resource, the physical infrastructures, and the mobile users. The proposed spectrum-oriented R-TMSC algorithm can always generate a stable three-sided matching result. The simulation results have proved the effectiveness of the proposed matching approach in improving the user satisfaction, SP revenue, as well as the system cost-performance.

# Chapter 9

# Exploiting the Stable Fixture Matching Game for Mobile Crowd Sensing: A Local Event Sharing Framework

The surging of the smartphone sensing and social media have enabled a great variety of applications, such as the environment surveillance, marketing, health monitoring and so on. This new paradigm is typically referred to as the mobile crowd sensing (MCS). Existing solutions on the MCS are majorly based on text, image and video analysis using the distributively sensed/collected data. One shortcoming of such approaches is the data processing delay. Considered as an enhanced function to the temporary social media services, e.g., Twitter, we propose a real-time event sharing framework that gives mobile users the freedom of expressing their various interests. We introduce a novel Hashtag, which is the combination of five types of information: time, location, keywords, data type and data size. Users, by comparing their interests with the uploaded Hashtags in the Twitter server, can search for suitable partners to share their information with in real time. The formulated user pairing problem is modeled as the SF matching game, and can be solved by the ISF algorithm in a semi-distributive manner.

## 9.1 Introduction

The widespread of mobile devices, such as smartphones and vehicular networks, has provided us a new type of sensing infrastructure. These smart mobile devices are embedded with various types of sensors, such as camera, GPS, accelerator, digital compass, light sensor, and even health and pollution monitoring sensors in the future. These sensed data has enabled a broad range of applications such as the road transportation, health care, environmental surveillance, and so on [26]. Compared with the traditional static sensor networks, the mobile people-centric measurement has many advantages, such as scalable, better computation capability, direct access to the Internet and so on [26]. This new paradigm is typically referred to as the MCS, where individual users with sensing

and computing devices collectively share data and extract the information to measure and map the phenomena of common interest [27].

MCS applications can be majorly classified into three categories depending on the type of phenomenon being measured, which are environmental, infrastructure, and social applications [27]. Among these three types, the social-based type is intrinsically more complicated due to the interactive nature between social users. There have been several examples of applying the MCS platform into the mobile social networks. For example, the mCrowd is a platform implemented in iPhone that relies on the general public to capture the geo-tagged images, audio snippets, and so on [99]. Twitter is a popular online social media that provides time-critic world trends and events based on the analysis of the tweet contents. Microblogs, like Twitter, are typically more efficient than the traditional blogs in terms of timeliness, which can provide the ongoing information collected from users all over the world [100]. With the popularity of the microblogs, there have been a growing number of works on the event detection/discovery using the MCS approach.

Different user tweets are collected and analyzed by the Twitter server to study the trends and events. However, most of them are based on text analysis, where the outputs are only the event keywords with locations and times. The advantage of such text-based MCS analysis is its time efficiency, but it does not provide enough event details. Thus, some works have proposed to analyze the image or video tweets for better event visualization. This type of MCS analysis is called the photo mining [100] or visual sensing [101], and is usually accomplished by feature extraction and clustering methods. Although image/video tweets are commonly seen, the disadvantages of such approaches are the huge data size, high processing complexity and delay. Suppose in a local event, such as a concert or a sports game, people who attend the event are already familiar with the general information, i.e., the event location and time. Thus, they expect to see more real-time and diverse event details about the event when using the social network, e.g., Twitter. From this point, using the image/video MCS analysis to summarize the event may result in not only the data uploading delay but also the processing and analyzing delay. In addition, users who want to review specific event moments or view from a different angle may have to download and go through some irrelevant data

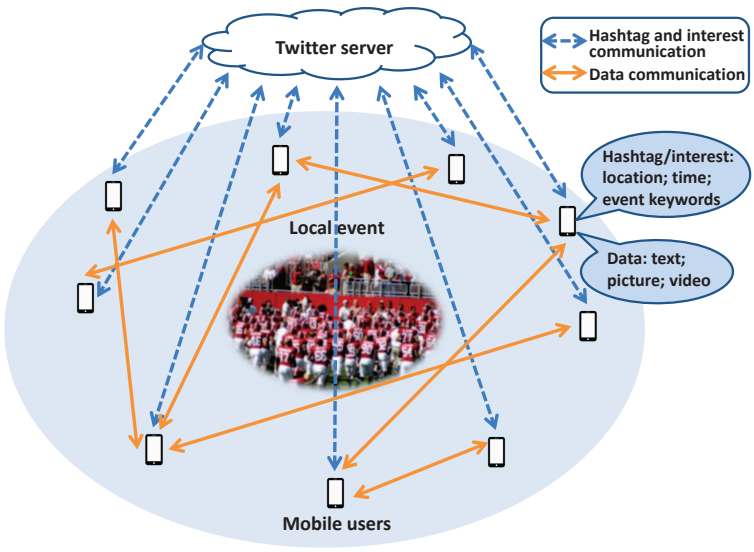



Figure 9.1: The mobile crowd sensing communication model.

before they find the right content, which is a waste of phone energy, storage and time.

Motivated by the above reasons, in this chapter, we propose an MCS-based platform for the real-time information sharing in a local event. It can be considered as an enhancement to the existing text/image/video analysis. Our novelly defined Hashtags, generated by distributed users, can provide detailed location and time information, data type and size information, as well as the keyword description of the data content. Users can search for the Hashtags according to their interests. Then we allow suitable users pairs to communicate directly with each other to exchange their data (text/image/video). To guarantee users' sharing incentives, only two users who both have interests in each other's Hashtags are allowed to exchange. Now the question becomes how to map these users into sharing pairs so that the information circulating within the network can be optimized. In this chapter, we propose the SF game to model the many-to-many matching relations between the users.

The rest of this chapter is organized as follows. The system model is described in Sec. 9.2, and then we formulate the crowd sensing problem as an optimization problem. The SF game is utilized to model the problem, and we tackle it with the ISF algorithm in Sec. 9.4. Performance of the ISF algorithm is evaluated in Sec. 9.5. Finally, conclusions are drawn in Sec. 9.6.

## 9.2 System Model

Consider a circular event area with the radius of $R$, as shown in Fig. 9.1, with mobile users are uniformly distributed within it. We denote the set of mobile users (i.e., audiences) as $\mathcal{U} = \{u_1, ..., u_i, ..., u_N\}$, where $N$ is the total number of users. For the convenience of quantifying the relative positions of the mobile users, we abstract the actual area of the event as one point. We assume the geometry center of the event as its event location. For example, in an NBA game, the geometry center of the rectangle basketball court is taken as the event location. It's very common that people can miss some exciting moments when they are distracted, and sometimes in a large stadium people are not guaranteed with clear or complete views of the event. Thus, it is incentive and necessary that people share their self-generated event information with each other, which may include text, picture and video information, for a better understanding of the event. To quantify the data exchange/sharing among the users, we should first define the communication link matrix. We denote the matrix as $\rho = \{\rho_{i,j} | u_i, u_j \in \mathcal{U}\}$, where $\rho_{i,j}$ is a binary value in $\{0, 1\}$, representing if there is or not a communication link set up between $u_i$ and $u_j$. Each user can be connected to multiple users simultaneously. The capacity of each user is defined as the number of radio interfaces equipped in the mobile devices, and is denoted as $c_i, u_i \in \mathcal{U}$.

Now we assume this set of users have agreed to share their information within themselves. To achieve the information sharing purpose, a social media platform is required, and without loss of generality, we adopt Twitter as the communication platform in this chapter. The information content generated by different users vary with generating time, location, data type and data size. In order to efficiently organize and analyze the generated data, a "Hashtag" in Twitter is often utilized to highlight and organize the data content. A traditional Hashtag is represented with a # symbol, and is used to index keywords or topics on Twitter. These Hashtags usually come at the beginning or the end of the actual tweets. However, using such traditional tags will make it hard to distinguish the data generated for the same event, and thus failing to help the users to identify the specific information that interests them. Thus, we propose a novel Hashtag definition, which allows users to

better distinguish between different tweets in the event.

### 9.2.1 A Novel Hashtag

We define the Hashtag generated by user $u_i$ at time $t$ as

$$\mathcal{H}_i = \{tim_i, loc_i, key_i, dtp_i, dsz_i\}. \tag{9.1}$$

$tim_i$ is the data generation time $t$. In this chapter, we assume the period for each time slot as $1$ sec. $loc_i$ represents the user $u_i$'s location. Since the event area is defined as a circle, we can divide it into sub-areas by direction and distance to the event center point. Suppose we divide the whole $360$ degree area into $8$ sections evenly according the angle range. Thus, each $45$ degree represents a view direction, staring from $0$ to $360$ degree. Then, we further divide each such section into $2$ parts according to the distance range to the event center point. The closer $R/2$ area of each sector represents the closer user group and the farther $R/2$ represents the farther user group. In other words, the whole area is divided into $16$ sections and is numbered in order of direction and distance. Thus, $loc_i$ must belong to one of the sub-areas in $\mathcal{L} = \{l_1, l_2, ..., l_{16}\}$. Except the location and time, the data content also varies other factors. For example in the sports game, the text/picture/video data may cover different targets, such as players, coaches or judges, and different topics, such as audiences' reactions or players' performances. These content differences can be realized by the combination of the user-defined keywords. We denote the keywords of $u_i$ as $key_i$, where $key_i = \{k_1, k_2, ..., k_K\}$. The number of keywords for each user is a fixed number $K$. Finally, an unignorable factor that may affect the user's interest is data type. Some users may prefer image data, while some others may prefer text or video data. We denote the data type of $u_i$ as $dtp_i$, which is an integer value within $\{1, 2, 3\}$ with $1$ representing text, $2$ representing image, and $3$ representing video. With the knowledge of data type, users also care about the data size, denoted as $dsz_i$, to fulfill different levels of requirements. $dsz_i$ is a positive real number, with the unit of MB.

With different tweets generated, users upload their Hashtags to the Twitter server. Notice that in this chapter, we assume that the Hashtag and the actual data are generated together but are

uploaded/transmitted separately[1]. The uploaded Hashtags can be viewed by all the event audiences in real time and will be used for making the paring decisions.

### 9.2.2 User Interest

As we have stated in Section 9.1, one of the advantages of our framework is allowing users to select the information according to their interests without downloading redundant/irrelevant data. To achieve the information filtering, we need to rely on the Hashtag defined in the previous section. Users can review the uploaded Hashtags and then compare them to their interests to decide if they want to set up communication links with other users who uploaded those Hashtags. Following the structure of the Hashtag $\mathcal{H}_i = \{tim_i, loc_i, key_i, dtp_i, dsz_i\}$, users can express their interests in the same way so that the comparison can be done easily. More specifically, the user interest for $u_i$ is defined as $\mathcal{I}_i = \{tim_i, loc_i, key_i, dtp_i, dsz_i\}$. The definition of each element in $\mathcal{I}_i$ is the same as the Hashtag, except that the elements in $\mathcal{I}_i$ are the value that $u_i$ is interested in. In practice, to generate the user interest, the Twitter system can be revised to provide five types of options (i.e., time, location, keywords, data type and data size) for users to select. Especially, in the keyword selection, a keyword library is generated from the existing uploaded Hashtags. Users can select $K$ keywords from the library to describe their interests.

To find the best Hashtag that matches the user interest, we measure the difference between these two vectors. The difference between $u_i$'s interest and a Hashtag uploaded by $u_j$ is denoted as $\delta_{i,j} = \mathcal{I}_i - \mathcal{H}_j$. The difference is still a vector, and thus we define its value as

$$\mathcal{V}(\delta_{i,j}) = a_1|tim_i - tim_j| + a_2|loc_i - loc_j| + \\ a_3(K - |key_i \cap key_j|) + a_4|dtp_i \cap dtp_j| + a_5|dsz_i - dsz_j|. \tag{9.2}$$

In the above equation, $|tim_i - tim_j|$ represents the non-negative time difference, $|loc_i - loc_j|$ represents the non-negative location section difference, $K - |key_i \cap key_j|$ represents the keyword

[1]The event summary function of traditional crowd sensing is beyond the scope of this chapter. Thus, for information sharing purpose only, it's not necessary to upload the actual data to the Twitter server.

difference, $|dtp_i \cap dtp_j|$ represents the non-negative data type difference, and $|dsz_i - dsz_j|$ represents the non-negative data size difference between $u_i$'s interest and $u_j$'s Hashtag. $a_1, a_2, a_3, a_4, a_5$ are the weight parameters for measuring the five types of differences. These five parameters are defined to round $\mathcal{V}(\delta_{i,j})$ within $[0, 1]$. $\mathcal{V}(\delta_{i,j})$ can used to measure the similarity/difference of the user interest and an existing Hashtag. The smaller it is, the more matched $\mathcal{I}_i$ and $\mathcal{H}_j$ are.

### 9.2.3 Link Quality

However, when selecting an user to exchange data with, not only the user interest, but also the link quality between two users should be considered. First of all, each potential link should satisfy a minimum SINR requirement, denoted as $\gamma_{\text{min}}$. We assume that each sharing pair $(u_i, u_j)$ is assigned with two bands $f_{i,j}$ and $f_{j,i}$ for information exchange. For both $f_{i,j}$ and $f_{j,i}$, the noise levels vary with $u_i$ and $u_j$, and are denoted as $\sigma_{i,j}$ and $\sigma_{i,j}$, respectively. Thus, the QoS requirement for each data link can be satisfied using the following inequality:

$$\gamma_{j,i} = \frac{p_{j,i} g_{j,i}}{\sigma_{j,i}^2} \geq \gamma_{\text{min}}, \tag{9.3}$$

where $\gamma_{j,i}$ represents $u_i$'s received SINR from $u_j$, for $u_i, u_j \in \mathcal{U}, i \neq j$. $p_{j,i}$ and $g_{j,i}$ denote the transmission power and channel propagation gain from $u_j$ to $u_i$, respectively. $g_{j,i} = C\beta_{j,i}\zeta_{j,i}D_{j,i}^{-\alpha}$, where $C$ is a constant system parameter, $\beta_{j,i}$ is the fast fading gain, $\zeta_{j,i}$ is the slowing fading gain, $\alpha$ is the path loss exponent, and $D_{j,i}$ is the distance between user $u_i$ and $u_j$.

## 9.3 Problem Formulation

With joint consideration of the user interest and the link quality, we define the utility for user $u_i$ as the weighted transmission rate when exchanging data with $u_j$. The user interest is expressed with $\mathcal{V}(\delta_{i,j})$ and is used as the data weight of the transmission. Thus, we can represent the utility of $u_i$ when exchanging information with $u_i$ as follows

$$\mathcal{U}_{i,j} = \mathcal{V}(\delta_{i,j}) \times r_{j,i} = \mathcal{V}(\delta_{i,j}) f_{j,i} \log\left(1 + \gamma_{j,i}\right), \tag{9.4}$$

where $r_{j,i}$ is the received transmission rate of $u_i$ from $u_j$. $f_{j,i}$ is the subband assigned for transmission from $u_j$ to $u_i$.

Intuitively, by allowing users to select the their sharing partners, we aim to maximize the information exchanged within the whole network. This objective can be achieved by optimizing the binary link matrix $\rho$. Thus, we define the summation of all users' utility as our system objective. Based on the above discussions, the problem can be formulated as follows.

$$\textbf{max:} \sum_{u_i \in \mathcal{U}} \sum_{u_j \in \mathcal{U}} \rho_{i,j} \cdot \mathcal{U}_{i,j}, \tag{9.5}$$

**s.t.:**

$$\gamma_{j,i} = \rho_{i,j} \frac{p_{j,i} g_{j,i}}{\sigma^2_{j,i}} \geq \gamma_{\text{min}}, \tag{9.6}$$

$$\sum_{u_j \in \mathcal{N}} \rho_{i,j} \leq c_i, \forall u_j \in \mathcal{U}, \text{and} \tag{9.7}$$

$$\rho_{i,j} = 0, \forall i = j. \tag{9.8}$$

(9.5) is the system objective that maximizes all users' utilities. (9.6) represents SINR requirement for each potential communication link, while (9.7) indicates the user capacity requirement. (9.8) states that if no user can be assigned to itself. The formulated problem is an MILP problem, which is NP-hard in general [53]. Thus, it motivates us to find a distributive approach where each user can make the decision directly. We introduce the matching-based approach in the following sections.

## 9.4 Matching-based Approach

In this section, the optimization problem is modeled as the SF game. We start by introducing some definitions of the SF game in Section 9.4.1 and then propose a distributive solution for the SF

game in Section 9.4.2.

### 9.4.1 Stable Fixture Game

The SF problem stems from a practical situation, where players play against each another in a chess tournament. Each player ranks their potential opponents in order of preference, and the task is to construct a set of fixtures, consisting of distinct matches (each involving two players), which is stable [88]. In the chess tournament, each competition happens independently. In other words, if player *a* plays with player *b* and player *c*, then players *b* and *c* are not necessarily required to compete with each other. Just like the many-to-many relationships in the chess tournament, each mobile user can set up multiple independent communication links with other users, and thus forming a mesh-style network.

An SF instance consists of a single set of agents $A = \{a_1, ..., a_n\}$, where $n$ is the number of all players/agents. Each player maintains a list of all the acceptable agents. Then each agent ranks its acceptable list of agents according to its preferences, and this list is called the preference list. The matching decisions are made based on the preference lists. A matching $M$ in an SF instance is defined as a subset of $E$, where $E = \{(a_i, a_j)|a_i, a_j \in A, i \neq j\}$. We denote $a_j = M(a_i)$ if pair $(a_i, a_j)$ is within the matching $M$. A fundamental requirement for a good matching is to be stable. Intuitively, a stable matching refers to the matching in which no player has the incentive to make any changes (switch partners). The formal definition of a stable matching in the SF model is provided in Definition 9.1:

**Definition 9.1.** *Stability: Let $I$ be an instance of the SF game and $M$ be a matching in $I$. A pair $(a_i, a_j) \in E/M$ blocks $M$, or is a blocking pair (BP) of $M$, if the following conditions are satisfied relative to $M$:*

*(1) $a_i$ is under subscribed or prefers $a_j$ to its worst partner;*

*(2) $a_j$ is under subscribed or prefers $a_i$ to its worst partner.*

*A matching $M$ is said to be stable if it admits no BP.*

Modeled as the SF game, we assume mobile users as the agents. Each user first searches for its acceptable set $\mathcal{A}(u_i)$ of partner users, who satisfy the SINR requirement, as shown in (9.6). Then each user's preference is ranked based on its utility function defined in (9.4) for all its acceptable partners. We denote $u_i$'s preference list as follows

$$PL_i(j) = \mathcal{U}_{i,j} = \mathcal{V}(\delta_{i,j}) \times r_{j,i}, \forall u_j \in \mathcal{A}(u_i). \tag{9.9}$$

### 9.4.2 ISF Algorithm

Now we modify the SF stable matching algorithm proposed by Irving et al. in [88] to fit our crowd sensing problem. We name the algorithm as the ISF algorithm in this chapter. The existence and complexity property of the ISF algorithm is stated in Theorem 9.2, and the proof can be found in [88].

**Theorem 9.2.** *Given an instance of the SF game, the ISF algorithm constructs in $\mathcal{O}(m)$ time, a stable matching, or reports that non exists, where $m$ is the sum of the preference list lengths.*

The general idea of the ISF algorithm is the reduction of users' preference lists and the construction of a player set $S$. $S$ is initially empty, and will finally be symmetric, in which case it becomes the stable matching if one exists. There are two cases that a stable matching does not exist: (1) $\sum_i d_i$ is odd, (2) there exists short list in $PL$ during the execution of the ISF algorithm. The step-to-step implementations are provided in Algorithm 9.1.

The ISF algorithm consists of two phases. Phase 1 involves a sequence of bids from one user to another. These bids enable the construction of the set $S$ and the reduction of the preference lists $PL$. We denote $u_i$'s target set and bidder set as $\mathcal{TS}_i$ and $\mathcal{BS}_i$, respectively. $\mathcal{TS}_i = \{u_j | (u_i, u_j) \in S\}, \mathcal{BS}_i = \{u_j | (u_j, u_i) \in S\}$, and we define $ts_i = |\mathcal{TS}_i|, bs_i = |\mathcal{BS}_i|$. To start with, each user $u_i$ bids for its most favorite user in $PL_i$ who is not in $\mathcal{TS}_i$, and denote it as $u_j$. Then, we construct $S$ by adding the pair $(u_i, u_j)$ into it. Notice here that all the pairs in $S$ are ordered, which means $(u_i, u_j)$ and $(u_j, u_i)$ are different. Then for the target user $u_j$, it check whether the received bids has

exceeded its capacity $c_j$. If yes, it deletes the bidders who are worse than the $c_j$'s rank in $PL_j$. The bidding action of $u_i$ continues as long as $ts_i < \min(c_i, |PL_i|)$. Phase 1 terminates when each user $u_i$'s target set size has reached $\min(c_i, |PL_i|)$.

**Algorithm 9.1 ISF Algorithm**

**Input:** $\mathcal{U}$, $PL$, $S = \emptyset$
**Output:** Stable Matching $\mathcal{M}$

**Phase 1:**
**while** $ts_i < \min(c_i, |PL_i|)$ **do**
    $u_j =$ the first player in $PL_i$ who is not in $\mathcal{TS}_i$;
    $S = S \cup \{(u_i, u_j)\}$;
    **if** $bs_j \geq c_j$ **then**
        $u_k = c_j$th ranked bidder for $u_j$;
        **for all** successor $u_l$ of $u_k$ in $PL_j$ **do**
            **if** $(u_l, u_j) \in S$ **then**
                $S = S/\{(u_l, u_j)\}$
                delete $(u_l, u_j)$ from $PL$;
            **end if**
        **end for**
    **end if**
**end while**
**Phase 2:**
**if** $\sum_i d_i$ is odd **then**
    report instance unsolvable;
**else**
    **while** there's no short list in $PL$ **do**
        find a rotation $rot$ in $PL$;
        $PL = PL/rot$;
        **if** some list in $PL$ is short **then**
            report instance unsolvable;
        **else**
            $S = S(PL)$;
        **end if**
    **end while**
    $\mathcal{M} = S$;
**end if**
**End of algorithm**.

After Phase 1, we have achieved a reduced preference list $PL$ and an increased players set $S$. We represent $min(c_i, |PL_i^1|)$ as $d_i$, which is the degree of $u_i$. To start Phase 2, we first check if $\sum_i d_i$ is odd, if yes then we report this instance is unsolvable, otherwise we continue. The key ideas of Phase 2 are the further construction of $S$ and the further reduction of $PL$. We first classify $PL$ into *short* lists and *long* lists. We call $PL_i$ *short* if $|PL_i| < d_i$, and *long* if $|PL_i| > d_i$. During the execution, if any user has a short list, then no stable matching exists. If no short list occurs, we search for a rotation in $PL$, which is the key to the further reduction of $PL$. A rotation is a sequence of ordered pairs $rot = ((u_{i_0}, u_{j_0}), (u_{i_1}, u_{j_1}), ..., (u_{i_{r-1}}, u_{j_{r-1}}))$, where for each $0 \leq k \leq r-1$,

$u_{i_k} = u_{l(j_k)}$ and $u_{j_{k+1}} = u_{f(x_{i_k})}$. $x_{l(i)}$ the last player in $PL_i$, and $x_{f(i)}$ is the first player in $PL_i$ who is not in $\mathcal{TS}_i$. To find a rotation, we begin by any user who has a long list, and set it as $u_{j_0}$. Then by the relations between $u_{i_k}$, $u_{l(j_k)}$ and $u_{l(j_k+1)}$, we can start building the rotation $rot$. The building process stops when any user is visited twice, and the rotation is found. To eliminate $rot$ from $PL$, we delete all the pairs $(u_{j_k}, n_l)$, such that $u_{j_k}$ prefers $u_{g(j_k)}$ to $n_l$, where $n_g(j_k)$ is the least favored member of $u_{j_k}$ in $\{\mathcal{BS}_{j_k} \cup u_{i_{k-1}}\}/\{u_{i_k}\}$. The rotation searching process stops whenever a short list occurs (i.e., there is no stable result) or when no rotation can be found (i.e., we have reached a stable matching).

## 9.5 Performance Evaluation

In this section, the performance of the proposed ISF algorithm will be evaluated by comparing it with the centralized optimization, the greedy and one heuristic method. Both the social welfare and the average user performance will be analyzed.

### 9.5.1 Simulation Set Up

Within a circle area with radius of $R = 800$ m, we assume a local event with $N = [20, 120]$ uniformly distributed audiences. The bandwidth that allocated to each potential communication link is set within $[1, 3]$ MHz. The SINR requirement for each link is a uniform random distribution, within $(20, 30)$ dB. For the propagation gain, we set the pass loss constant $C$ as $10^{-2}$, the path loss exponent $\alpha$ as $4$, the multipath fading gain as the exponential distribution with unit mean, and the shadowing gain as the log-normal distribution with $0$ mean and $4$ dB deviation. The user capacity $c_i$ is set as $4$, identical for each one. For the Hashtag parameters, the time element $tim_i$ is set between $[1, 10]$, where the unit is set as min. The location section $loc_i$ belongs to one of the $16$ sections $\mathcal{L}$ as defined in Section 9.1. The keyword number $K$ is set as $5$, meaning that the overlap of uploaded and interested keywords is within $\{0, 1, 2, 3, 4, 5\}$. The data type $dtp_i$ is set as $\{1, 2, 3\}$ representing text, image, video, respectively, and the data size $dsz_i$ is set within $[1, 100]$ MB. $a_1, ..., a_5$ are fine-turned

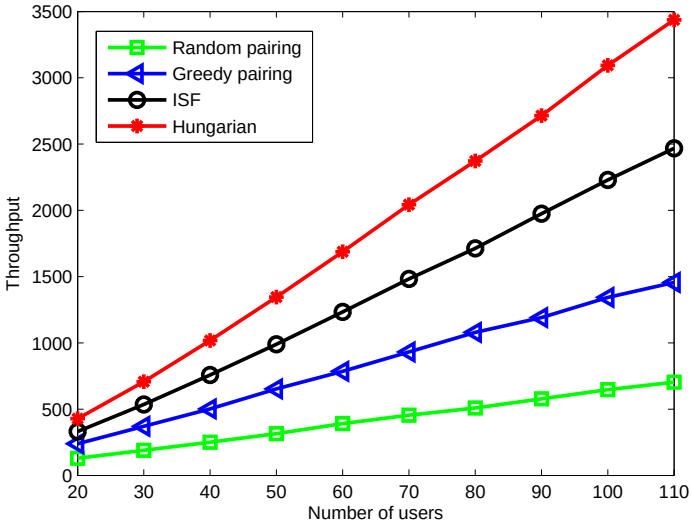



Figure 9.2: System welfare evaluation.

in the simulations to unitize $\mathcal{V}(\delta_{i,j})$.

### 9.5.2 Numerical Results

To demonstrate the effectiveness of the ISF algorithm, we compare it with three other methods: (1) Hungarian, (2) Greedy, and (3) Random. The Hungarian method represents the centralized optimal solution in finding the maximum weight one-to-one matching [102]. The edge weight is set as the weighted data rate, as defined in (9.9). The Greedy method is a method proposed by [102], that offers a non-iterative algorithm for finding a one-to-one matching in the D2D communications. The Random method is a method that randomly matches the users into pairs while satisfying the capacity and SINR requirements.

We evaluate the information sharing by measuring the weighted data rate from the perspectives of the social welfare and the average user performance. Notice here, since both the Hungarian method and the Greedy method are designed for the one-to-one matching, we set the user capacity as 1 when implementing the four algorithms in Fig. 9.2, Fig. 9.3 and Fig. 9.5. In Fig. 9.2, we increase the user number from 20 to 120 by 10. Apparently, the centralized Hungarian method provides the best system throughout, followed by our proposed ISF method. However, the ISF algorithm is a dis-

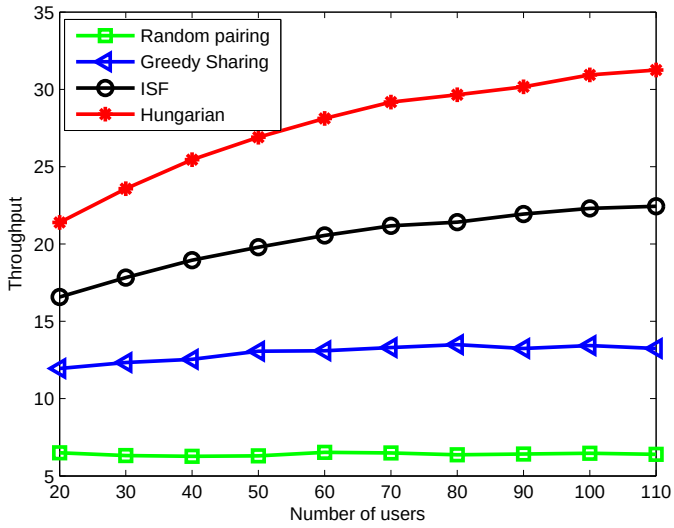



Figure 9.3: Average user performance evaluation.

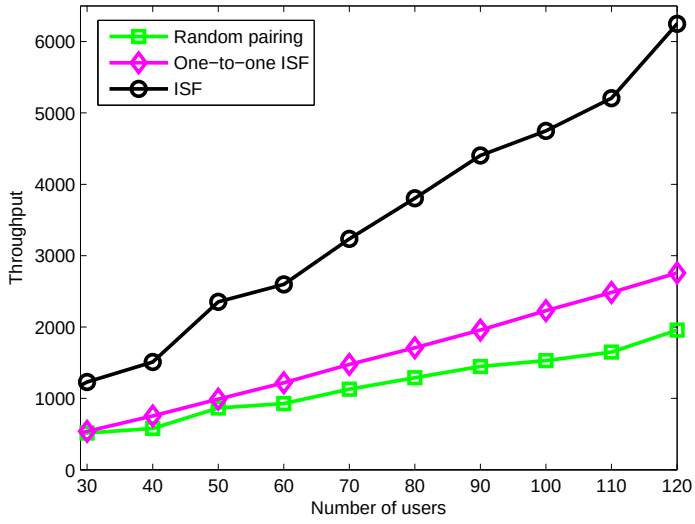



Figure 9.4: System social welfare.

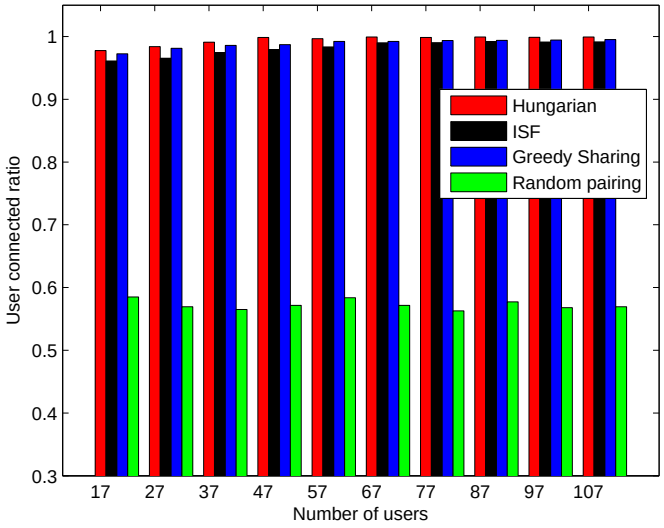



Figure 9.5: Ratio of connected users.

tributive algorithm that has a much lower complexity ($\mathcal{O}(m)$) compared to the Hungarian method ($\mathcal{O}(n^4)$). What's more, the ISF algorithm beats both the Greedy and the Random methods, which are both non-iterative. A similar conclusion can be drawn from Fig.9.3, which evaluates the average user performance. In fact, we can find a performance increase with the increase of the user number in both Fig. 9.2 and Fig. 9.3. It's easy to understand the increase in Fig. 9.2, since more users can bring more system throughput. For average user performance in Fig. 9.3, the increase is credited to the more partner options brought by new users. Apparently, the Random method doesn't benefit from having more choices since it's a random selection.

Except the special case of the one-to-one matching, we also provide the many-to-many matching evaluation in Fig. 9.4. We compared the ISF method with the one-to-one ISF method and the Random method regarding the social welfare. The one-to-one ISF method represents the matching using the ISF algorithm with each user's capacity set as 1. Apparently, the ISF method outperforms the other two methods. What's more, the one-to-one ISF method also beats the Random method, which shows the superiority of the ISF algorithm even under specific capacity assumptions.

Finally, we evaluate the percentage of users that successfully shared data with others in Fig. 9.5. The Hungarian method still achieves the best performance, with the Greedy and ISF methods

following closely. With the increase of the user number, this ratio almost reaches $100\%$ when using the Hungarian or the ISF method. For the Random method, this ratio keeps an average value of $57\%$.

## 9.6 Conclusion

This chapter has proposed a novel event sharing framework using the mobile crowd sensing technique. It offers mobile users with real-time information update, and provide users with the freedom of selecting their interested data. Using the SF matching game, the flexible many-to-many sharing relations between users can be accurately modeled. The proposed ISF algorithm provides a stable solution for the SF game in a distributive manner. The simulation results have further proved that superiority of the ISF algorithm in achieving close-optimal network performance.

# Chapter 10

# Future Works

Beyond the previous works, we are also working on some new topics/ideas. In the following sections, two future works will be briefly discussed.

## 10.1 Dynamic Stability in LTE assisted V2V commmunications

The technology of connected vehicles has been envisioned as a paradigm capable of providing increased convenience to drivers, enabling applications ranging from road safety to traffic efficiency. In traditional IEEE 802.11p based V2V communications, reliable and efficient performance cannot be guaranteed since 802.11p is CSMA/CA based. Besides, the high cost of deploying roadside units (RSUs) cannot be ignored. Thus, the concept of integrating LTE into the V2X communications has been proposed, which is commonly referred to as LTE-V [16] or LTE assisted V2X [17]. The objectives of the study on LTE-based V2X services include the definition of an evaluation methodology and possible application scenarios, and the identification and evaluation of necessary enhancements to LTE physical layer, protocols, and interfaces. So far, 3GPP has defined 18 use cases for LTE-based V2X services, such as Case 5.1: Forward collision warning, Case 5.8: Road safety services, Case 5.9: Automatic parking system, 5.18: vulnerable road user safety and so on [18]. These use cases cover major road safety and traffic efficiency applications.

On the other hand, dynamic management is one of the expected characteristics of the 5G wireless communications. Typically, network dynamics can be caused by user mobility, channel fading, or some other factors. Many popular mobility models have been used to model different movement patterns. For example, the RWP and HotSpot models can be used to model unpredictable and predictable user motions, respectively. User mobility, in fact, can further cause channel fading. Therefore, in the vehicular communications networks, where vehicles can be in fast movement and the surroundings can change very rapidly, it is necessary to find an efficient solution for the LTE

V2X content sharing applications to guarantee user satisfactions.

In Chapter 6, we have studied the content sharing in the LTE V2X networks, where we have considered a static (i.e., one-time) resource allocation. In order to cope with the network dynamics, new solutions approaches should be proposed. Instead of solving the static resource allocations of the different time slots in a repeated manner, i.e., using the ISF algorithm introduced in Chapter 6 repeatedly for each time slot, we can try to transform the previously unstable matching into stable again. Just like the RTPS algorithm adopted in Chapter 5 to find the dynamic stabilities in the two-sided matching problem, there are also algorithms that can be used to re-stabilize the matching in the one-sided case. For example, Inarra et al. in [103] proposed an algorithm for finding a p-stable matching by continuously satisfying BPs. Biro and Norman in [104] extended the Tan-Hsueh algorithm to construct a sequence of matchings that ultimately yield a p-stable matching in the one-sided matching problem. Thus, by following the idea of the above-mentioned approaches, we can either directly implement or slightly revise one of these algorithms to find the many-to-many dynamic stability in the LTE V2X scenario.

## 10.2 Stable Allocation Modeling in the HetNets

With the exponential increase of smart mobiles devices, there has been an explosion in not only the communication content diversity but also the device diversity. On the other hand, mobile users are characterized by more accurate user information and specific QoS requirements. All these factors can result in user heterogeneity and will bring more challenges for the resource management. Previously implemented matching models, such as the SF model studied in Chapter 6, only consider one type of connection relationship between players. However, in the heterogeneous networks (HetNets), it is very likely that the communication links between different users are diverse, and even for the same player pair, the communication contents, delay requirements, and channel conditions have various specifications. Thus, the single-type matching relationship may no longer be adequate to fulfill the resource management requirements for the HetNets.

On the other hand, there are some existing models that study the multi-type partnerships between players in the matching theory. For example, the Stable Multiple Activities (SMA) game is a generalization of the SF model in which the underlying graph may have parallel edges. The SMA model represents a practical situation, where players form multiple partnerships according to different sports activities: for example, player $a_i$ might play tennis, chess, and badminton games with player $a_j$. A stable solution to such an instance can be achieved in $\mathcal{O}(m^2)$, where $m$ is the edge number in the underlying graph, as proved by Cechlarova and Fleiner in [105]. Thus, we may utilize this SMA model to characterize the various user specifications in the HetNets. For example, two mobile users may communicate with each other through multiple types of connections, such as voice call, text messaging, video sharing and so on. Different types of communications are characterized by different requirements, and thus should be allocated with different resources. By modeling the resource allocation in the HetNets as the SMA model, we can provide higher user satisfaction while improving the spectrum utilization.

# References


[1] 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.

[2] 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, May 2015.

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

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

[5] C. Xu, L. Song, and Z. Han, *Resource Management for Device-to-Device Underlay Communication*. Springer, US, 2013.

[6] L. Song, D. Niyato, Z. Han, and E. Hossain, *Wireless Device-to-Device Communications and Networks*. Cambridge University Press, UK, 2014.

[7] B. Kaufman and B. Aazhang, "Cellular networks with an overlaid device to device network," in *42nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 2008.

[8] M. Belleschi, G. Fodor, and A. Abrardo, "Performance analysis of a distributed resource allocation scheme for D2D communications," in *IEEE Global Communications Conference (GLOBECOM) Workshops*, Houston, TX, Dec. 2011.

[9] P. Janis, V. Koivunen, C. Ribeiro, K. Doppler, and K. Hugl, "Interference-avoiding mimo schemes for device-to-device radio underlaying cellular networks," in *IEEE 20th International Symposium on Indoor and Mobile Radio Communications*, Tokyo, Japan, Sep. 2009.

[10] K. Doppler, J. Manssour, A. Osseiran, and M. Xiao, “Innovative concepts in peer-to-peer and network coding,” Project: Wireless World Initiative New Radio, WINNER+, 2008.

[11] Q. Huang, K. Birman, R. V. Renesse, W. Lloyd, S. Kumar, and L. H. C, “An analysis of facebook photo caching,” in *the 24th ACM Symposium on Operating Systems Principles*, 2013.

[12] “Akamai technologies.” [Online]. Available: http://www.akamai.com/

[13] S. Press, “Web caching,” technical report. [Online]. Available: http://www.silicon-press.com/briefs/brief.webcaching/brief.pdf

[14] J. Wannstrom, “Carrier Aggregation,” Jun. 2013. [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/101-carrier-aggregation-explained

[15] 3GPP, “LTE in Unlicensed Spectrum,” Jun. 2014. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1603-lte-in-unlicensed

[16] Huawei, “LTE-V.” [Online]. Available: http://www.huawei.com/minisite/hwmbbf15/en/lte-v.html

[17] G. T. RP-151109, “New SI proposal: Feasibility Study on LTE-based V2X Services.” [Online]. Available: http://www.3gpp.org/DynaReport/TDocExMtg--RP-68--31197.htm

[18] G. TR22.885, “Study on LTE Support for V2X sevices.” [Online]. Available: http://www.3gpp.org/DynaReport/22885.htm

[19] “Cloud Computing.” [Online]. Available: https://en.wikipedia.org/wiki/Cloud-computing

[20] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013.

[21] CISCO, “Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are,” 2015, white paper. [Online]. Available: https://www.cisco.com/c/dam/en-us/solutions/trends/iot/docs/computing-overview.pdf

[22] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. ACM, 2012, pp. 13–16.

[23] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. We, and L. Sun, "Fog computing: Focusing on mobile users at the edge." *arXiv*, vol. arXiv:1502.01815, 2015.

[24] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862 – 876, Apr. 2010.

[25] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 358–380, Mar. 2015.

[26] M. Talasila, R. Curtmola, and C. Borcea, "Mobile crowd sensing," in *Handbook of Sensor Networking: Advanced Technologies and Applications*, J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds. CRC Press, 2015.

[27] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[28] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.

[29] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Matching and cheating in device to device communications underlying cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2156–2166, Oct. 2015.

[30] F. Wang, L. Song, Z. Han, Q. Zhao, and X. Wang, "Joint scheduling and resource allocation for device-to-device underlay communication," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, Apr. 2013.

[31] C. Xu, L. Song, Z. Han, D. Li, and B. Jiao, "Resource allocation using a reverse iterative combinatorial auction for device-to-device underlay cellular networks," in *IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, Dec. 2012.

[32] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Exploring social ties for enhanced device-to-device communications in wireless networks," in *IEEE Globe Communication Conference (Globecom)*, Atlanta, GA, Dec. 2013.

[33] D. Feng, L. Lu, Y. Yuan-Wu, G. Li, G. Feng, and S. Li, "Device-to-device communications underlaying cellular networks," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3541–3551, Aug. 2013.

[34] S. Bayat, R. H. Y. Louie, Y. Li, and B. Vucetic, "Cognitive radio relay networks with multiple primary and secondary users: Distributed stable matching algorithms for spectrum access," in *2011 IEEE International Conference on Communications (ICC)*, Tokyo, Japan, Jun. 2011.

[35] S. Bayat, R. H. Y. Louie, Z. Han, Y. Li, and B. Vucetic, "Distributed stable matching algorithm for physical layer security with multiple source-destination pairs and jammer nodes," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, Paris, France, Apr. 2012.

[36] S. Bayat, R. H. Y. Louie, Z. Han, B. Vucetic, and Y. Li, "Physical-layer security in distributed wireless networks using matching theory," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 717–732, May 2013.

[37] S. Bayat, R. H. Y. Louie, Z. Han, Y. Li, and B. Vucetic, "Multiple operator and multiple femtocell networks: Distributed stable matching," in *2012 IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012.

[38] A. El-Hajj, Z. Dawy, and W. Saad, "A stable matching game for joint uplink/downlink resource allocation In OFDMA wireless networks," in *2012 IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012.

[39] F. Pantisano, M. Bennis, W. Saad, S. Valentin, and M. Debbah, "Matching with externalities for context-aware user-cell association in small cell networks," in *IEEE Global Communications Conference*, Atlanta, GA, Dec. 2013.

[40] W. Saad, Z. Han, R. Zheng, M. Debbah, and H. V. Poor, "A college admissions game for uplink user association in wireless small cell networks," in *The 33rd Annual IEEE International Conference on Computer Communications*, Toronto, Canada, Apr.-May 2014.

[41] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *Proc. 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Hammamet, Tunisia, May 2014.

[42] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 1, pp. 82–95, Jan. 2012.

[43] O. Naparstek and A. Leshem, "A fast matching algorithm for asymptotically optimal distributed channel assignment," in *18th International Conference on Digital Signal Processing (DSP)*, Santorini, Greece, Jul. 2013.

[44] O. Naparstek, A. Leshem, and E. A. Jorswieck, "Distributed medium access control for energy efficient transmission in cognitive radios," *Computing Research Repository (CoRR)*, vol. abs/1401.1671, 2014.

[45] B. Holfeld, R. Mochaourab, and T. Wirth, "Stable Matching for Adaptive Cross-Layer Scheduling in the LTE Downlink," in *IEEE 77th Vehicular Technology Conference (VTC Spring)*, Dresden, Germany, Jun. 2013.

[46] R. Mochaourab, B. Holfeld, and T. Wirth, "Distributed channel assignment in cognitive radio networks: Stable matching and walrasian equilibrium," *Computing Research Repository (CoRR)*, 2014. [Online]. Available: http://arxiv.org/abs/1404.5859

[47] Y. Zhang, Y. Gu, M. Pan, and Z. Han, "Distributed matching based spectrum allocation in cognitive radio networks," in *IEEE Globe Communication Conference*, Austin, TX, Dec. 2014.

[48] E. A. Jorswieck, "Stable matchings for resource allocation in wireless networks," in *2011 17th International Conference on Digital Signal Processing (DSP)*, Corfu, Greece, Jul. 2011.

[49] E. A. Jorswieck and P. Cao, "Matching and exchange market based resource allocation in mimo cognitive radio networks," in *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*, Marrakech, Morroco, Sep. 2013.

[50] L. Huang, G. Zhu, X. Du, and K. Bian, "Stable multiuser channel allocations in opportunistic spectrum access," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013.

[51] L. Guo, Q. Cui, Y. Liu, X. Li, T. Fu, and Z. Chen, "Graph theory based channel reallocation technique in channel borrowing in mobile satellite communication," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013.

[52] J. Han, Q. Cui, C. Yang, and X. Tao, "Bipartite matching approach to optimal resource allocation in device to device underlaying cellular network," *Electronics Letters*, vol. 50, no. 3, Jan. 2014.

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

[54] A. Gjendemsjo, D. Gesbert, G. E. Oien, and S. G. Kiani, "Optimal power allocation and scheduling for two-cell capacity maximization," in *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Boston, MA, Apr. 2006.

[55] D. Gusfield, "Three fast algorithms for four problems in stable marriage," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 111–128, Feb. 1987.

[56] R. W. Irving, P. Leather, and D. Gusfield, "An efficient algorithm for the optimal stable marriage," *Journal of the ACM*, vol. 34, no. 3, pp. 532–543, Jul. 1987.

[57] T. Feder, "Network flow and 2-satisfiability," *Algorithmica*, vol. 11, no. 3, pp. 291–319, Mar. 1994.

[58] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[59] C. Huang, "Cheating by men in the gale-shapley stable matching algorithm," in *Algorithms-ESA*, 2006, vol. 4168, pp. 418–431.

[60] A. Goldsmith, *Wireless Communications*. UK: Cambridge University Press, 2004.

[61] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Student admission matching based content-cache allocation," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, Louisiana, Mar. 2015.

[62] T. Geoff Huston, "Web caching," *The Internet Protocol Journal*, vol. 2, no. 3, pp. 2–20, Sep. 1999.

[63] C. OHanlon, "Infoblox dns caching appliance helps speed web experiences," technical report, 2012, Available: http://thevarguy.com/network-security-and-data-protection-software-solutions/infoblox-dns-caching-appliance-helps-speed-w.

[64] IBM, "Cplex optimizer." [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[65] M. P. Wittie, V. Pejovic, L. Deek, K. C. Almeroth, and B. Y. Zhao, "Exploiting locality of interest in online social networks," in *ACM Proc. of the 6th International Conference on emerging Networking Experiments and Technologies (CoNEXT)*, Philadelphia, PA, Nov. - Dec. 2010.

[66] E. Jaho and I. Stavrakakis, "Joint interest and locality-aware content dissemination in social networks," in *IEEE The 6th International Conference on Wireless On-demand Network Systems and Services (WONS)*, Snowbird, UT, Feb. 2009.

[67] E. Million, "The hadamard product," 2007. [Online]. Available: http://buzzard.ups.edu/courses/2007spring/projects/million-paper.pdf

[68] Alcatel-Lucent, Ericsson, Q. T. Inc., S. Electronics, and Verizon, "LTE-U Technical Report: Coexistence Study for LTE-U SDL V1.0," Tech. Rep., Feb. 2015.

[69] H. Zhang, Y. Dong, J. Cheng, M. J. Hossain, and V. C. M. Leung, "Fronthauling for 5g lte-u ultra dense cloud small cell networks," *accepted by IEEE Trans. on Wireless Communs.*, 2016.

[70] Y. Gu, Y. Zhang, L. Cai, M. Pan, L. Song, and Z. Han, "LTE-Unlicensed Co-existence Mechanism: A Matching Game Framework," *IEEE Wireless Communications Magazine*, 2016, to appear.

[71] QUALCOMM, "LTE in Unlicensed Spectrum: Harmonious Coexistence with Wi-Fi," Jun. 2014, White Paper.

[72] H. Zhang, Y. Xiao, L. X. Cai, D. Niyato, L. Song, and Z. Han, "A hierarchical game approach for multi-operator spectrum sharing in lte unlicensed," in *IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, Dec. 2015, pp. 1–6.

[73] Y. Gu, Y. Zhang, L. X. Cai, M. Pan, L. Song, and Z. Han, "Exploiting student-project allocation matching for spectrum sharing in lte-unlicensed," in *IEEE Global Communications Conference*, San Diego, CA, Dec. 2015.

[74] T. Nihtilä, V. Tykhomyrov, O. Alanen, M. A. Uusitalo, A. Sorri, M. Moisio, S. Iraji, R. Ratasuk, and N. Mangalvedhe, "System Performance of LTE and IEEE 802.11 Coexisting on A Shared Frequency Band," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Shanghai, China, Apr. 2013.

[75] A. M. Cavalcante, E. Almeida, R. D. Vieira, F. Chaves, R. Paiva, F. Abinader, S. Choudhury, E. Tuomaala, and K. Doppler, "Performance Evaluation of LTE and Wi-Fi Coexistence in Unlicensed Bands," in *IEEE Vehicular Technology Conference*, Dresden, Germany, Jun. 2013.

[76] F. M. Abinader, E. Almeida, F. Chaves, A. Cavalcante, R. Vieira, R. Paiva, A. Sobrinho, S. Choudhury, E. Tuomaala, K. Doppler, and V. Sousa, "Enabling the Coexistence of LTE and Wi-Fi In Unlicensed Bands," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 54–61, Nov. 2014.

[77] E. Almeida, A. M. Cavalcante, R. C. D. Paiva, F. S. Chaves, F. M. Abinader, R. D. Vieira, S. Choudhury, E. Tuomaala, and K. Doppler, "Enabling LTE/WiFi Coexistence by LTE Blank Subframe Allocation," in *IEEE International Conference on Communications. (ICC)*, Budapest, Hungary, Jun. 2013.

[78] H. Zhang, X. Chu, W. Guo, and S. Wang, "Coexistence of Wi-Fi and Heterogeneous Small Cell Networks Sharing Unlicensed Spectrum," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 158–164, Mar. 2015.

[79] F. S. Chaves, E. P. L. Almeida, R. D. Vieira, A. M. Cavalcante, F. M. Abinader, S. Choudhury, and K. Doppler, "LTE UL Power Control for the Improvement of LTE/Wi-Fi Coexistence," in *IEEE 78th Vehicular Technology Conference (VTC Fall)*, Las Vegas, NV, Sep. 2013.

[80] Z. Zhou, D. Guo, and M. L. Honig, "Allocation of Licensed and Unlicensed Spectrum in Heterogeneous Networks," in *Australian Communications Theory Workshop (AusCTW)*, Melbourne, Australia, Jan. 2016.

[81] Q. Chen, G. Yu, A. Maaref, G. Y. Li, and A. Huang, "Rethinking mobile data offloading for lte in unlicensed spectrum," *IEEE Trans. on Wireless Communs.*, vol. 15, no. 7, pp. 4987–5000, Jul. 2016.

[82] F. Teng, D. Guo, and M. L. Honig, "Sharing of Unlicensed Spectrum by Strategic Operators," in *GlobalSIP Symposium on Game Theory in Communications and Signal Processing*, Atlanta, GA, Dec. 2014.

[83] J. Zhou and M. Fan, "LTE-U Forum and Coexistence Overview," Tech. Rep., Jul. 2015. [Online]. Available: https://mentor.ieee.org/802.19/dcn/15/19-15-0057-00-0000-lte-u-forum-and-coexistence-overview.pdf

[84] A. E. Roth and J. H. V. Vate, "Random paths to stability in two-sided matching," *Econometrica*, vol. 58, no. 6, pp. 1475–1480, Nov. 1990.

[85] W. Sun, D. Yuan, E. G. Ström, and F. Brännström, "Resource Sharing and Power Allocation for D2D-based safety-critical V2X communications," in *IEEE International Conference on Communication Workshop*, London, UK, June 2015.

[86] W. Saad, Z. Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 48–60, January 2011.

[87] Y. Gu, L. Wang, M. Pan, L. Song, and Z. Han, "Exploiting the stable fixture matching game for content sharing in d2d-based lte-v2x communications," in *IEEE Global Communications Conference*, Washington, DC, Dec. 2016.

[88] R. W. Irving and S. Scott, "The stable fixtures problem: A many-to-many extension of stable roommates," *Discrete Applied Mathematics*, vol. 155, no. 16, pp. 2118 – 2129, 2007.

[89] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 10, pp. 2991–3005, 2016.

[90] G. S. S. Sardellitti and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[91] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[92] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[93] Y. Cui, X. Ma, H. Wang, I. Stojmenovic, and J. Liu, "A survey of energy efficient wireless transmission and modeling in mobile cloud computing," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 148–155, 2013.

[94] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, "Fog computing in multi-tier data center networks: A hierarchical game approach," in *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[95] A. H. A. El-Atta and M. I. Moussa, "Student project allocation with preference lists over (student, project) pairs," in *Second International Conference on Computer and Electrical Engineering*, Dubai, Dec. 2009.

[96] 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.

[97] P. Biro and E. McDermid, "Three-sided stable matchings with cyclic preferences," *Algorithmica*, vol. 58, no. 1, pp. 5–18, Sep. 2010.

[98] L. Cui and W. Jia, "Cyclic stable matching for three-sided networking services," *Computer Networks*, vol. 57, no. 1, pp. 351 – 363, Jan. 2013.

[99] U. Amherst., "mCrowd." [Online]. Available: https://crowd.cs.umass.edu/

[100] T. Kaneko and K. Yanai, "Event photo mining from twitter using keyword bursts and image clustering," *Neurocomputing*, vol. 172, pp. 143 – 158, Jan. 2016.

[101] H. Chen, B. Guo, Z. Yu, and Q. Han, "Toward real-time and cooperative mobile visual sensing and sharing," in *The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, Apr. 2016.

[102] L. Wang and H. Wu, "Fast pairing of device-to-device link underlay for spectrum sharing with cellular users," *IEEE Communications Letters*, vol. 18, no. 10, pp. 1803–1806, Oct. 2014.

[103] E. Inarra, C. Larrea, and E. Molis, "Random paths to p-stability in the roommate problem," *International Journal of Game Theory*, vol. 36, p. 461–471, 2008.

[104] P. Biro and G. Norman, "Analysis of stochastic matching markets," *International Journal of Game Theory*, vol. 43, Oct. 2012.

[105] K. Cechlárová and T. Fleiner, "On a generalization of the stable roommates problem," *ACM Trans. Algorithms*, vol. 1, no. 1, pp. 143–156, Jul. 2005.