Computation Offloading in Mobile Edge Networks: A Minority Game Model

by

Shermila Ranadheera

A Thesis submitted to The Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

December 2017

Abstract

Fifth generation (5G) dense small cell networks are expected to meet the thousand-fold mobile traffic challenge within the next few years. Due to the ever-increasing popularity of resource-hungry and delay-constrained mobile applications, the computation and storage capabilities of remote cloud has partially migrated towards the mobile edge, giving rise to the concept known as Mobile Edge Computing (MEC). One application of MEC is computation offloading, where users offload computationally expensive tasks to the edge nodes. Two main challenges of MEC are offloading decision making problem and MEC server resource allocation problem. While MEC servers enjoy the close proximity to the end-users to provide services at reduced latency and lower energy costs, they suffer from limitations in computational and radio resources. This calls for efficient resource management in the MEC servers. This problem is challenging due to the ultra-high density, distributed nature, and intrinsic randomness of next generation wireless networks. Thus, when developing solution schemes, conventional centralized control may no longer be viable. Instead, distributed decision making mechanisms with low complexity would be desirable to make the network self-organizing and autonomous. Hence, it is imperative to develop distributed mechanisms for computation offloading, such that the users' latency constraints are fulfilled, while the computational servers are utilized at their best capacity. Game theory is well-established as a classic tool to mathematically model the wireless resource allocation problems and to develop distributed decision making schemes. Since game theory focuses on strategic interactions among players, it eliminates the need for a central controller which is a major advantage. In this thesis, I investigate two main challenges in computation offloading mentioned above: (i) computation offloading decision making and (ii) energy efficient activation of MEC servers. For both cases, I focus on the objective of achieving efficient resource allocation of MEC servers while meeting users' latency requirements. To this end, I develop distributed decision making schemes to solve these problems using the theory of minority games. I demonstrate the performance of the proposed methods using simulations.

Acknowledgements

First and foremost, I sincerely thank my academic advisor Dr. Ekram Hossain for giving me the opportunity to work with him. I was able to push myself beyond my expectations with his excellent supervising, dedication and encouragement. Secondly, I would like to thank Dr. Pradeepa Yahampath and Dr. Noman Mohammed for their invaluable feedback and advice. I especially thank Dr. Setareh Maghsudi for her guidance and constant support provided for my research work.

I acknowledge University of Manitoba and the government of Manitoba for providing me with financial support.

I would like to convey my immense gratitude to my mother Chandrika Dias and my father Sunil Ranadheera for their unconditional love, enormous sacrifices and most of all, for always believing in me. My sincere appreciation goes to my husband Madhuranga Bandara for his love, caring and constant support. I would also like to thank my dear friends for the continuous encouragement given. I finally thank my lab mates and the staff of the Department of Electrical and Computer Engineering, University of Manitoba for providing a peaceful and productive working environment.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

5G	Fifth Generation
cdf	Cumulative distribution function
CG	Congestion game
CPU	Central Processing Unit
CSI	Channel state information
D2D	Device-to-Device
MBS	Macro base station
MCC	Mobile Cloud Computing
MEC	Mobile Edge Computing
MG	Minority game
NE	Nash Equilibrium
pdf	Probability density function
QoE	Quality of Experience
QoS	Quality of Service
SBS	Small cell base station
SCN	Small cell network
TDMA	Time division multiple access

Publications

- Magazine Publications:

    1. **Shermila Ranadheera**, Setareh Maghsudi, and Ekram Hossain, "Minority Games With Applications to Distributed Decision Making and Control in Wireless Networks," *IEEE Wireless Communications*, vol. PP, no. 99, pp. 2–10, 2017

    2. **Shermila Ranadheera**, Setareh Maghsudi, and Ekram Hossain, "Mobile Edge Computation Offloading Using Game Theory and Reinforcement Learning," submitted to the *IEEE Communications Magazine*.

- Letter Publications:

    1. **Shermila Ranadheera**, Setareh Maghsudi, and Ekram Hossain, "Computation Offloading and Activation of Mobile Edge Computing Servers: A Minority Game," submitted to the *IEEE Wireless Communications Letters*.

Chapter 1

Introduction

1.1 Computation Offloading

With the ever-increasing popularity of computationally intensive applications such as augmented reality, online gaming and image processing (e.g., facial recognition), small user devices often face with a particular challenge. This is because, such applications are typically resource-hungry and delay-sensitive whereas the user devices have very limited computation, battery and storage capacities. Therefore, user devices are not always capable of performing the desired task locally, or doing so might become inefficient. This gives rise to the idea of transferring such tasks to powerful remote computing servers (e.g. the cloud), which typically have much higher computational capability than local devices. This idea is referred to as *computation offloading*. Computation offloading is expected to save the energy cost spent for local execution, thereby saving the battery life of end user devices. Therefore, computation offloading capability has become a prerequisite for next generation wireless networks.

1.1.1 Mobile Cloud Computing

The technology known as mobile cloud computing (MCC) can be considered as a means of computation offloading. In MCC, the user devices are able to utilize the resources of dedicated servers for executing their tasks. These servers are equipped with high power, CPU and storage capabilities. Moreover, these servers are typically located outsider the local network so that the users access these remote servers via a mobile network [1]. Therefore, despite its higher resource capacity, remote cloud may not be the ideal option for computation offloading, as the long distance between the cloud server and the user device yields substantial communication costs in terms of latency and energy. This is a major disadvantage of MCC since low latency is a stringent requirement for many popular applications such as online gaming. An illustration of computation offloading in MCC is given in Fig. 1.1. In addition, in dense networks, excessive back-haul traffic might arise as a result of the large number of offloading requests that are being sent to the cloud. Moreover, MCC has a centralized architecture where the cloud server farms are located in relatively few number of locations within the central core network. Therefore, solutions that eliminate the above limitations of MCC are required.

1.1.2 Mobile Edge Computing

In contrast to the cloud servers, the small scale computing servers located in the network edge are more likely to provide services at reduced latency and energy cost, compared to the remote cloud. Therefore, in next generation networks, the computation and storage capabilities of the core network have partially migrated towards the mobile edge, giving rise to the concept known as Mobile Edge Computing (MEC). Some initial concepts related to edge computing are fog computing [2], cloudlets [3]

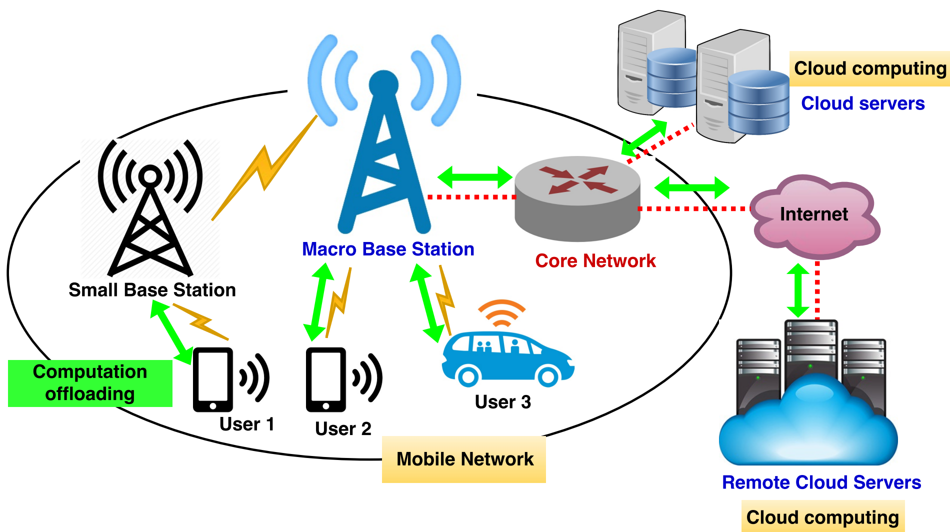


Figure 1.1: Mobile cloud computing.

and ad-hoc clouds formed by user devices [1]. However, all these concepts have shown some drawbacks in integration and meeting users' desired quality of service (QoS). In order to alleviate these issues, MEC is proposed as a standardized solution by the European Telecommunications Standards Institute (ETSI) and the Industry Specification Group (ISG). MEC is also recognized as a key enabler for 5G networks by the European 5G PPP (5G Infrastructure Public Private Partnership). MEC is expected to integrate the computation and storage capabilities of the mobile edge such that users' QoS levels can be guaranteed [1, 4–6].

In short, MEC is implemented possibly by a dense deployment of computational servers or by strengthening the already-deployed edge entities such as small cell base stations (SBS) with computation and storage resources. In a MEC network, mobile devices are able to offload their computationally expensive tasks to the edge servers while requesting some specific QoS. This process is feasible due to the fact that edge servers are deployed in close proximity of mobile users, specifically in comparison to the remote cloud servers. Therefore, MEC is foreseen as a remedy to address the challenges of reducing the latency, energy cost and bandwidth consumption of mobile users. Moreover, for dense networks, the amount of back-haul traffic generated due to computation offloading is negligible in MEC compared to MCC. This is because, the offloading requests are processed at each edge node itself rather than forwarding to a central remote cloud. Hence MEC can enhance the user quality of experience (QoE) and deliver higher user satisfaction in future wireless networks. Moreover, unlike MCC, MEC follows a decentralized architecture since the edge servers are deployed in a distributed manner. In MEC, there can be a variety of edge servers such as SBS, macro base stations (MBS), wireless access points, peer user devices (e.g., in device-to-device communication), etc. Moreover, the user devices can be either mobile phones and/or IoT devices. An illustration of computation offloading in MEC is provided in Fig. 1.2.

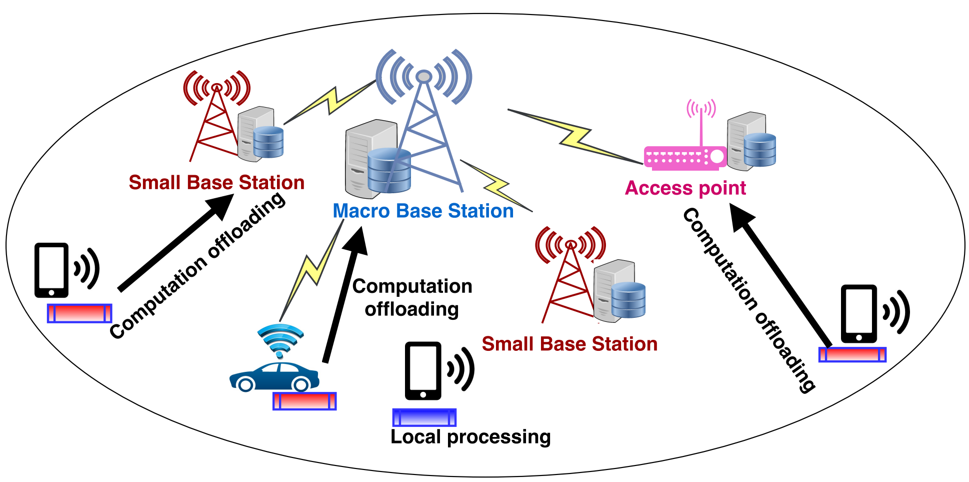


Figure 1.2: Mobile edge computing.

1.2 Challenges in MEC

Despite its great potential in improving the latency and energy consumption, realizing the concept of MEC is associated with a variety of challenges. As previously mentioned, from the mobile devices' perspective, the energy consumption and the latency experienced should be minimized. Apart from that, resource scarcity and distributed nature of MEC requires efficient resource management in the absence of centralized control. In other words, the limited radio, power and computational resources of mobile edge servers shall need to be efficiently utilized so that the users' quality of service requirements are met with the minimal effort. These problems become aggravated when the randomness and dynamics of wireless networks are taken into account. The factors that contribute to this include, but are not limited to, users' mobility, random channel quality, time-varying and random task-arrival and non-deterministic energy resources (for instance in case of energy harvesting). In what follows, we briefly discuss some important problems, including computation offloading and few other closely-related issues.

- **Computational Resource Allocation:** As a result of being deployed at the edge, MEC suffers from restrictions of computational resources, in particular when compared to the central mobile cloud computing. As a result, it becomes imperative to allocate the limited resources in an efficient manner. This includes, but is not limited to, efficient MEC server utilization, MEC server activation and scheduling, load balancing, request management, task allocation, and the like. At the same time, users' latency requirements should be taken into account. Therefore, addressing this trade-off is a major issue in developing efficient MEC systems.

- **Radio Resource Allocation:** Enhancing the wireless network with MEC complicates the radio resource management. For instance, the necessary uploading and downloading of task-related data results in radio bandwidth consumption and interference. Consequently, smart bandwidth allocation shall need to be performed for mobile devices/servers. Moreover, the energy consumption at the servers should be kept at the minimum. To increase energy efficiency, servers might share the energy resources and/or harvest ambient energy. Such remedies however introduce uncertainty in the system, in contrast to using deterministic power resources such as a grid.

- **Computation Offloading Decision:** While the allocation of computational resources is performed on the MEC servers' side, mobile devices decide about computation offloading. In essence, each device decides which and what part of every task shall be offloaded to an edge server. In some cases, the specific server to which the task is uploaded can be determined by the device as well. Moreover, mobile devices might be able to demand a specific QoS guarantee. Naturally, mobile devices might compete with each other for limited computational services, whereas each server would compete with others to increase its number of offloaded tasks. Moreover, a conflict arises between the set of servers and the set of devices, since the latter requests low prices for services, whereas the former benefits from high service prices.

1.3 Motivation

Main motivation of this thesis is to address the challenge of efficient utilization of MEC servers in next generation networks. The majority of the existing literature in this regard focuses on the user-centric objectives such as meeting users' delay constraints and minimizing users' energy consumption. On the contrary, this thesis presents a hybrid view where both servers' and users' standpoints for a computation offloading environment are considered. To elaborate, I focus on achieving a desired latency threshold for the users. Meanwhile, I focus on achieving efficient utilization of edge servers (chapter 3) and selective activation of edge computing servers that ensure energy efficiency (chapter 4).

It is clear that the above cannot be addressed by conventional centralized resource allocation schemes, since such mechanisms necessitate the availability of global information at a central node. This is infeasible to acquire in ultra-dense distributed networks with large number of users and/or servers. Moreover, for such networks, the computational complexity becomes overwhelming as well. As a result, it is vital to develop distributed and autonomous approaches, where the individual mobile devices and mobile edge servers make decisions for the system to settle at efficient and stable operating points. Therefore, to develop such distributed schemes, tools that eliminate the need of centralized controller are required. Therefore, I use game theory to model the distributed resource allocation problem since game theory is a well-established tool that focuses on strategic interactions among players without the use of a central controller.

Game theory offers a variety of game models in which, each game has its own distinct set of properties, that make them suitable for different types of decision making problems based on the context. In this thesis, I use the theory of minority games to model the resource management problem in MEC. Motivation for using minority game is mainly because of its straight-forward applicability as a tool for modeling resource allocation problems. Minority game is a type of congestion game and thus essentially models a number of players competing for a limited resource. In a MEC system, the offloading users attempt to utilize the limited amount of available computational resources of the edge servers. Hence it is clear that, this scenario essentially maps to a congestion problem and thus can be easily modeled using minority game. Moreover, in a minority game, players make independent decisions with the availability of very small amount of external information. Other advantages of minority game such as simple implementation, low overhead, and scalability to large set of players also make it ideal to model dense wireless MEC systems.

1.4 Related Work

In this section, we briefly explore the cutting edge research in the area of computation offloading and resource management for MEC. In doing so, we focus on computation offloading and resource management methods that are developed based on game theory and/or reinforcement learning. A summary is given in Table 1.1. A significant number of detailed surveys on MEC that discuss the MEC architecture and standardization, enabling technologies, MEC use cases/applications in future wireless networks, challenges, state-of-the-art and future research directions, etc. are available in [1, 4–6].

Computation Offloading

A large body of existing literature focuses on minimizing offloading users' computation overhead in terms of energy and latency. To this end, researchers have developed distributed decision making methodologies. Authors in [7, 8] presented a multi-user computation offloading decision making mechanism using potential games. They proposed distributed algorithms for the users to converge into a Nash equilibrium (NE) solution. In [9], authors proposed a multi-user computation offloading scheme for cloudlet based MCC network. The offloading decision making problem is modeled as a potential game where existence of NE is proved. A distributed offloading algorithm is proposed to achieve the equilibrium. Similarly, authors in [10] proposed a game theoretic model for computation offloading in MCC. Therein, the mobile users choose one of many wireless access points to offload their tasks such that the global offloading cost defined in terms of energy and delay is minimized. Existence of NE is proven and algorithms are provided to converge to the equilibrium. In [11], authors investigated the same energy and latency cost minimization problem for offloading users. However, they considered a drone (UAV) based network as the mobile edge and modeled the offloading problem as a potential game to develop a decentralized offloading strategy for the users. In [12], the authors considered a multi-cell, quasi-static environment, and formulated the computation offloading problem as a dynamic sequential game. They further established the existence of NE and develop a distributed convergent offloading scheme. In [13], the authors considered the offloading problem with the set of mobile devices varying randomly during the offloading period. The problem is modeled using a stochastic game framework, which is afterward shown to be equivalent to a potential game. The existence of NE is proved and a stochastic learning algorithm is developed. For cloud-enhanced vehicular networks with edge computing capability, an offloading mechanism based on a Stackelberg game is proposed in [14]. The servers and the offloading vehicles are modeled as the leaders and the followers, respectively. Similar to the aforementioned references, the existence of NE is proved and a distributed algorithm is designed that maximizes the edge server's utility while meeting the tasks' latency constraints. In [15], the authors investigated the multi-user offloading decision making problem in a dynamic environment, where users' states and offloading requests are time-variant. The number of tasks offloaded to each server (machine) is modeled as an a priori unknown time-varying Markov process. The authors then formulated the offloading problem as a Markov decision process. Online learning algorithms are developed to solve for the optimal offloading policy for both centralized and decentralized scenarios.

Radio and Computational Resource Management

In [16], the authors applied coalitional game theory to solve a resource allocation problem in MEC-enabled IoT networks with software-defined network (SDN) capability. In such a network, delay sensitive tasks are offloaded to the edge servers by the IoT applications. The developed game-theoretical framework is guaranteed to adaptively provision the available computational resources in the MEC servers in order to satisfy the quality of service requirements of IoT applications. Moreover, a deterministic algorithm is proposed to minimize the task processing cost and the latency. Reference [17] investigated joint offloading decision making and dynamic edge server provisioning in an offloading mobile edge network with energy harvesting capability. They modeled the problem as a Markov decision process. A reinforcement learning algorithm is developed for offloading computation jobs and activating edge servers while minimizing the overall cost and delay. The authors of [18] proposed a resource allocation mechanism using auction theory. Therein, service providers in the mobile edge network design contracts with the edge node infrastructure providers. The contracts enable the edge servers to efficiently provision their assigned computational resources and to schedule the offloaded tasks in a way that the latency is minimized. In [19], the focus is on a dynamically-changing vehicular networks with MEC capabilities including computation and caching. A network operator allocates computation, caching and network resources to the vehicles for different vehicular ap-

Table 1.1: A comparison of the state-of-the-art.

Reference	**Objective**	**Model**	**MEC type**
[7], [8], [9], [10], [11]	Minimize users' energy and latency cost	Potential game	Quasi-static
[12]	Minimize users' energy and latency cost	Dynamic sequential game	Quasi-static
[13]	Minimize users' energy and latency cost	Stochastic game	Dynamic
[14]	Maximize utilities of users and servers	Stackelberg game	Vehicular
[15]	Minimize unprocessed offloading requests	Markov decision process	Dynamic
[16]	Optimize resource usage and QoS guarantee	Coalitional game	Edge IoT
[17]	Minimize overall cost and latency	Markov decision process	Energy harvesting MEC
[18]	Minimize latency	Auction theory	Dynamic workload arrival
[19]	Efficient resource allocation	Deep Q-learning	Vehicular
[20]	Minimize servers' energy and QoS guarantee	Minority game	Random

plications. To address high complexity, the authors developed a deep reinforcement learning algorithm based on deep Q-learning.

1.5 Contributions and Scope of the Thesis

This thesis introduces minority games as a tool to mathematically model the distributed resource allocation problems in dense wireless networks. Furthermore, as its main contribution, this thesis investigates the problem of efficient edge server utilization problem in MEC networks. To this end, distributed algorithms are developed using the minority game for two different scenarios: (i) multiple user offloading decision making problem and (ii) efficient edge server activation problem.

In what follows, I briefly discuss the main contributions of this thesis.

1. I provide a brief tutorial on minority games and how the theory of minority

games is applicable to solve distributed decision making problems in wireless networks, mainly focusing on dense small cell networks (SCN). I discuss the state-of-the-art minority game applications and provide the future research directions and open problems.

2. I investigate the distributed mobile computation offloading problem in an SCN with MEC capability. In such networks, SBSs are provided with computational capability and thus are capable of serving as MEC servers. I formulate the offloading decision making problem of users accessing a given SBS. Therein, I focus on how the computation resources of SBS can be utilized to its maximum capacity while the users' latency would remain below a specific threshold. I consider how the total number of offloading users affect on users' experienced latency and investigate how the users would make the offloading decision based on the base station being crowded vs uncrowded. Thus, in this work, both users' and server's perspectives are considered. I cast the formulated problem using a minority game theoretic framework and provide an offloading decision making algorithm. The developed mechanism is distributed and hence does not require communication among users. I numerically investigate the performance of the proposed method.

3. Considering a pool of edge servers in a MEC network, I formulate the efficient edge server activation problem. Therein, I address the uncertainty caused by the randomness in channel quality and users' requests. I first analyze the statistical characteristics of the offloading delay. Based on this, I model the computation offloading problem as a planned market, where the price of computational services is determined by a central governor. Afterward, by using the theory of minority games, I develop a novel approach for efficient mode selec-

tion at the servers' side. The proposed pricing framework in combination with the designed mode selection mechanism guarantee a minimal server activation to ensure energy efficiency, while meeting the users' delay constraints with adjustable certainty. Thus, both users' and servers' standpoints are taken into account. The proposed mode selection scheme is distributed, and does not require any prior information at the servers' side. Moreover I apply different reinforcement learning and stochastic learning methods to solve the formulated minority game and numerically investigate their performances to examine which methods provide the best performance in terms of both users' and servers' utility. Numerical results are presented to illustrate the trade-off between the efficient server activation and meeting users' expected latency threshold.

1.6 Organization of the Thesis

The remainder of the thesis is organized into four chapters as follows:

- **Chapter 2** discusses the basic concepts, some variants, and equilibrium notions of the minority game. Furthermore, the applicability of minority games for modeling distributed decision making in wireless networks are discussed. To this end, a brief review of the state-of-the-art applications of minority games in communication networks is provided. Potential future applications and open problems are also presented.

- **Chapter 3** presents a minority game based distributed computation offloading decision making scheme for offloading users in an SCN. Detailed system model, assumptions and simulation results are given.

- **Chapter 4** presents a minority game based distributed decision making scheme for energy efficient MEC server activation problem in mobile edge networks. Moreover, numerous reinforcement learning and stochastic learning techniques that are applicable to solve the formulated minority game model are discussed. System model, assumptions and simulation results are given.

- **Chapter 5** provides a summary of the research presented in this thesis along with the future research directions and open problems.

Symbols and notations used throughout the chapters are given in a table at the beginning of each chapter.

Chapter 2

Minority Games With Applications to Distributed Decision Making in Wireless Networks

In this chapter, I provide an introductory tutorial of the minority game and discuss its applicability as a mathematical tool to model wireless resource allocation problems. Therein, I specifically focus on applying minority game (MG) to model the distributed decision making/control problems that arise in dense small cell networks (SCN). To this end, I provide a brief summary of the state-of-the-art, where MG is applied to solve the problems that arise in communication networks. Finally, I provide the future research directions and open problems.

2.1 Distributed Decision Making in Dense Wireless Networks

The next generation of wireless networks, also known as 5G, is expected to face a thousand-fold growth in mobile data traffic due to the increased smart device usage, proliferation of data hungry applications and pervasive connectivity requirement. Since the existing traditional macro cellular networks are not designed to cope with such large data traffic, network densification using small cell base stations (SBS) and implementation of SCNs are proposed. In particular, SCNs are expected to improve the efficiency of the utilization of radio resources, including energy and spectrum.

Although SCNs might become the key enablers of 5G, they impose some challenges that need to be addressed. For instance, the typical wireless resource allocation problems become more complicated in a dense network. Since the SCNs are expected to be hyper-dense and multi-tier, they must be self-organizing and self-healing, to avoid high complexity and fault-intolerance of central management. In other words, network management tasks such as resource allocation are preferred to be performed in a distributed manner. Also, the unavailability of global and precise channel state information in dense networks needs to be addressed. Moreover, feedback and signaling overhead should be minimized. In order to address these challenges, in this chapter, I focus on applying minority game (MG) to model the distributed decision making/control problems that arise in 5G SCNs.

2.2 Overview of Game Theory

Game theory is well-established as a classic tool to mathematically model the wireless resource allocation problems. Based on the fact that many of the wireless resource allocation problems can be reduced to distributed decision making problems, game theory becomes an ideal fit. Game theory focuses on strategic interactions among players and thus eliminates the need for a central controller which is a major advantage. As it is well-known, game theory has two main branches: non-cooperative and cooperative. Non-cooperative game theory studies the interactions of rational and self-interested players that compete against each other, and the goal is to achieve an efficient equilibrium point. Important notions of equilibrium include Nash, correlated and Walrasian equilibrium. Cooperative game theory promotes a cooperative behavior that is supposedly beneficial to all agents. In summary, game theory offers a variety of game models in which, each game has its own distinct set of properties, that make them suitable for different types of decision making problems based on the context.

2.3 Minority Game for Dense SCNs

Minority game (MG) has recently gained attention of the research community as a tool to model congestion problems encountered in wireless networks. MG is a type of non-cooperative games that can be used to model distributed resource allocation problems. In simple terms, in an MG, an odd number of players select between two alternatives in the hope of being in the minority, because only the minority group receives a pay-off. In other words, MG is a simple congestion game (CG) with a standard CG theoretic framework. As in a typical CG, in an MG, the objective of each player is to maximize her own pay-off, which depends on the number of players choosing the same option. However, MG carries its own unique properties and learning model and thus differs from other variants of CG such as route-choice games. For instance, a main difference between route-choice games and MG is that the former has strict pure strategy Nash equilibria (NE) while none of the pure strategy NEs of the latter is strict [21]. An MG is able to model a congested system with a large number of agents competing for shared resources, where pair-wise communication between agents does not take place. This finds application in 5G SCNs that accommodate a large number of users, where congestion can occur due to the scarcity of the (radio and/or computational) resources. In such scenarios, users would naturally prefer to select the less-crowded option. Moreover, MG involves self-organized decision making with minimal external information available to the agents as desired in dense SCNs.

MG proclaims its own unique characteristics that make it a more suitable model to apply in wireless networking problems compared to traditional non-cooperative games. For instance, unlike many game models, MG involves players with bounded rationality using *inductive reasoning* to make decisions [22]. Moreover, the players are anonymous in MG, making it ideal to model wireless problems where the user identity should not be compromised, for instance, in crowdsourcing and recommender systems. In addition, MG benefits a variety of settings that can be useful to model many different types of resource management problems.

When developing distributed solution schemes for wireless resource allocation problems, conventional distributed approaches (e.g. those based on traditional game theory) are not always applicable, since such models become increasingly complex for systems with large number of agents. In essence, many such game models require pairwise interactions among the agents. In contrast, the agents' interaction in MG exhibit mean field like behavior [22], i.e., an individual agent interacts with the aggregate behavior of all other agents. This makes MG a promising technique, especially since mean field-based models are widely used as fitting tools to model large systems that are often studied in distributed resource allocation problems. With such characteristics, MG is an emergent tool that can be used to model distributed decision making problems in wireless networks. The main motivations of this chapter are to study MG and to investigate the applicability of MG in wireless networks.

I will first describe the basic concepts, some variants, and equilibrium notions of MG. Then, the state-of-the-art applications of minority games in communication networks will be discussed.

2.4 Minority Games: Basics, Equilibrium, Solution Approaches, and Variants

2.4.1 Basics of a Minority Game

The concept of MG stems from *El Farol bar problem* [23], and was initially formulated and presented in [24]. In the most basic setting of such a game, an odd number of players choose between two actions while competing to be in the minority group through selecting the less popular action, since only the minority receives a reward. After each round of play, all players are informed of the winning action, which is then used as history data by the players to improve the decision making in the upcoming rounds. Let us denote the two actions by 0 and 1. Moreover, the action of player i at time t is shown by $a_i(t)$. The number of players (N) is required to be an odd number to avoid ties. Each player has a given set of decision making strategies that help her select future actions. A *strategy* predicts the winning action of the next round based on the previous m number of winning actions, with m being the size of memory, also known as the *brain size*. In other words, a *strategy* is essentially a mapping of the m-bit length history string ($\mu(t)$) to an action. An example strategy table for an agent is given in Table 2.1, where the agent has two strategies S_1 and

S_2. Since there are two actions to select from, it is clear that the strategy space consists of 2^{2^m} total number of strategies, which become very large even for small m. Thus, *reduced strategy space* (RSS) is introduced to make the strategy space remarkably smaller without any significant impact on the dynamics of the MG. RSS is formulated by choosing 2^m strategy pairs so that in each pair, one strategy is *anti-correlated* to the other. In other words, the predictions given by one strategy are the exact opposites of the predictions given by the other strategy [25]. An example for two anti-correlated strategies is shown in Table 2.1. Thus RSS constitutes of 2^{m+1} total number of strategies, which is much smaller than the size of universal strategy space 2^{2^m}.

Table 2.1: An example strategy table for an agent.

History string	Predicted winning action	
	S_1	S_2
00	1	0
01	1	0
10	0	1
11	0	1

At the outset of the game, each agent randomly draws S strategies from the strategy space which remain fixed for each player throughout the game. There is no a priori best strategy. Intuitively, if such strategy exists, all agents would use it and therefore lose due to the minority rule, which contradicts the initial assumption. As the game is played iteratively, each player evaluates her own strategies as follows: The strategies that make accurate predictions about the winning action are given a point and the poorly performing strategies are penalized. In other words, strategies are reinforced as they predict the winning action over a number of plays. Note that all strategies are scored after each round regardless of being used by the agent or not. Thus the score of each strategy is updated after each round of play according to its performance and the players use the strategy with the largest accumulated score at each round. Each player's objective is to maximize her utility over the time as she plays the game repeatedly. In MG, often the players compete for a limited resource without communicating with each other. Consequently, since players do not have any knowledge about other players' decisions, the decision making becomes almost autonomous [22].

2.4.2 Properties of a Minority Game

The properties of a minority game are described by the following parameters and behaviors:

- *Attendance*: One of the most important parameters of an MG is the collective sum of the actions of all players at a given time t, known as the *attendance*, $A(t)$.

- *Volatility*: Basically, the attendance value never settles but fluctuates around the mean attendance (i.e. cut-off value) [22]. The fluctuation around the mean attendance is known as *volatility*, σ. Volatility is an inverse measure of the system's performance and hence, the term σ^2/N corresponds to an *inverse global efficiency*. When the fluctuations are smaller, that implies that the size of the minority, thus the number of winners, is larger. Hence, smaller volatility corresponds to higher users' satisfaction levels along with better resource utilization. It is known that volatility depends on the ratio $\alpha = 2^m/N$, which is commonly referred to as the *training parameter* or *control parameter* [22] [25] [26].

- *Phase transition*: Using the variation of the global efficiency w.r.t. α, the game can be divided into two *phases* by the minimum value of α (denoted by α^*), namely *crowded phase* and *uncrowded phase*. MG is said to be in the crowded phase when $\alpha < \alpha^*$. This is because, for smaller m, the number of strategies, 2^{2^m}, is quite smaller compared to the number of agents N, thus many agents could be using the same strategy, leading them to make the same decision. This then creates a *herding effect*, causing the MG to enter the crowded phase. Once $\alpha > \alpha^*$, the m values are large enough to make the strategy space larger than the number of agents N, so that the probability of any two agents using identical strategies diminishes, thus making MG enter the uncrowded phase. Note that, α^* corresponds to the minimum volatility indicating the system's ability to self-organize into a state where the number of satisfied agents and the resource utilization are maximized. Moreover, it is shown that the performance of MG surpasses that of the random choice game (where all agents choose each action with a probability = 0.5) for a certain range of α values. This is referred to as the *better than random regime* [22] [25] [27] [28]. Variation of σ/N w.r.t α is illustrated in Fig. 2.1.

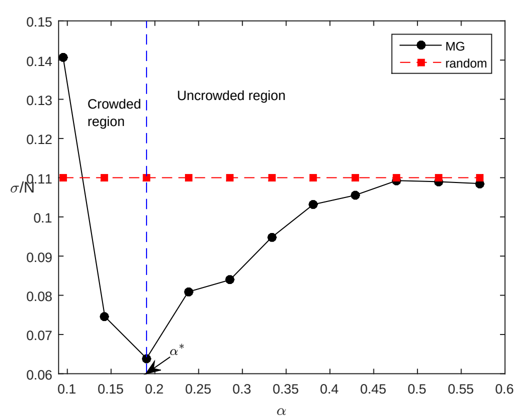


Figure 2.1: Variation of inverse global efficiency w.r.t. α.

- *Predictability*: This is an important physical property of MG. It measures the information content in the previous set of *attendance* values, that is available to agents. Predictability is denoted by H, where $H = 0$ corresponds to the situation in which the game outcome is unpredictable. Moreover, the predictability is the parameter that characterizes the two phases in the MG. In MG, $H = 0$ for $\alpha < \alpha^*$ and $H \neq 0$ for $\alpha > \alpha^*$. This implies that during the crowded phase, the game outcome is unpredictable and when MG enters the uncrowded phase, the game outcome becomes more predictable [25].

2.4.3 Equilibrium Notions for an MG

In this section, I provide a brief introduction to the notion of equilibria of MG. More details can be found in [27], [28].

Assuming the number of agents is an odd number equal to N, an MG is in an equilibrium if each of the two alternatives is selected by $(N-1)/2$ and $(N+1)/2$ agents. Then, no agent would gain by unilaterally deviating from its state since, if any of the agents in majority group does so, the groups would switch thus the state of the deviated agent would not improve. In an MG stage game, three types of Nash equilibria (NE) are applicable. Note that the NE corresponds to the local minima of volatility values [28].

- *Pure strategy Nash equilibria*: If there are N agents playing the MG and $(N-1)/2$ of them choose to select one alternative with probability $= 1$ while the other $(N+1)/2$ agents select the other alternative with probability $= 1$, system is said to be in a pure strategy NE. There are $\binom{N}{\frac{N-1}{2}}$ number of such NEs that exist. These NEs are considered the *globally optimal* states [27].

- *Symmetric mixed strategy Nash equilibria*: There exists only a single symmetric mixed strategy NE to the MG. It corresponds to the so called *random choice* *game* where, all agents choose between each of the two alternatives with a probability of 0.5 [27].

- *Asymmetric mixed strategy Nash equilibria*: If $(N-1)/2$ agents select one alternative with probability $= 1$, another $(N-1)/2$ agents select the other alternative with probability $= 1$ and the remaining agent selects an alternative with an arbitrary mixed probability, the MG stage game is said to be in an asymmetric mixed strategy NE. There can be an infinite number of such NEs [27].

2.4.4 Solution Approaches

Both qualitative and analytical approaches have been studied in the literature to solve an MG. The qualitative approach investigates how the volatility of the system varies with respect to the brain size and the population size. Moreover, it interprets the phase transitions of MG from the crowded phase to the uncrowded phase along with the volatility variation. Hence, this approach is also referred to as *crowd-anticrowd theory* in the literature [25]. A brief overview of the phase transition in MG in relation to the variation of volatility is given in Section 2.4.2. Rigorous explanations can be found in [28].

The analytical solution of MG obtains its statistical characterizations of the stationary states and the NE. In MG, the stationary states correspond to the minima of *predictability* whereas the NE of the MG correspond to the minima of *volatility*. In order to derive the analytical solutions, numerous mathematical techniques are used, including reinforcement learning, replicator dynamics, and tools from statistical physics of disordered systems (e.g., Hamiltonian, replica method, spin glass model, Ising model). Reference [27] includes a rigorous analysis on the solution of MG where aforementioned techniques are employed to derive the solution. Therein, complete statistical characterizations of the stationary state of MG are realized. First, the authors use *multi-population replicator dynamics* technique to obtain some evolutionarily stable NEs of the stage game. Then, a generalized version of the repeated MG model is analyzed where *exponential learning* is used by the agents to adapt their strategies.

For this scenario, analysis is done considering two different types of agents, namely *naive*[1] and *sophisticated*.[2] In their analysis the authors show that for the repeated MG with naive agents, stationary state is not a NE. Moreover, authors show that for the systems with sophisticated agents and exponential learning, the system converges to a NE.

2.4.5 Variants of Minority Game

In this section, I briefly introduce few different variations of MG beyond the basic form described before. Comprehensive discussions can be found in [25], [26] and [29], among many others.

1. *MG with arbitrary cut-off*: A generalized version of the basic MG, referred to as *MG with arbitrary cut-offs* is introduced in [26]. In such games, the minority rule is defined at an arbitrary cut-off value (ϕ) rather than the 50% cut-off used in the seminal MG. In [26], authors show the behavioral change of MG when the cut-off value is varied. In brief, it is shown that the attendance values fluctuate around the new cut-off, exhibiting the adaptation of the population. Furthermore, the analysis shows that when the cut-off ϕ is decreased below $N/2$,

[1] In seminal MG, the agents are naive, since they only know the pay-off received by the played strategies.

[2] Unlike naive agents, sophisticated agents are assumed to have the knowledge of the pay-off they would receive for any strategy that they play (including the strategies that are not played). More details on naive and sophisticated agents can be found in [27].

the brain size yielding the minimum volatility is also decreased. This variant is particularly useful to model some resource allocation problems where the cut-off (also known as the *comfort value*) of the capacity of a particular resource is a value other than 50%.

2. *Multiple-choice MG (Simplex game)*: This variant is introduced in [29] as a direct generalization of the basic MG where every agent might select among K different choices ($K > 2$). Thus, a simplex game is defined by the set of N players, the set of K choices and the history winning actions (m-bit long). Similar to the seminal MG, a strategy is a mapping of the history data to one of the K choices, and each player is given a set of S strategies.

   Moreover, the strategy space of the simplex game is associated with probability values p_{is}, which indicates the satisfaction of the i^{th} agent with her s^{th} strategy. Each player's choice is referred to as a *bid* and a quantity called *aggregate bid* is defined as the sum of the choices of all agents. Similar to the *attendance* property in the basic MG, the aggregate bid contains the information about the number of agents that select a given choice and determines the pay-off that each user receives. As the game is played iteratively, after scoring the strategies, the probability values (p_{is}) are updated using the *exponential learning*[3] method. Thus, although the players start out naive, they become sophisticated as the game evolves. Therefore, unlike basic MG, the simplex game exhibits evolutionary behavior. In [29], it is shown that compared to playing an MG with few options, in a game with a large action set, the overall system performance improves, resulting in higher resource utilization.

[3] In exponential learning, the probability values are modified by following a *logit model*-like formula. This formula contains exponential functions of strategy score and the agent's *learning rate*, which is a numerical constant that might differ for each agent [27].

3. *Evolutionary MG (Genetic model)*: In this version of MG, unlike the basic case, all users apply a single strategy. Each agent i chooses the action predicted by the strategy with some probability p_i referred to as the agent's *gene value*. Each agent selects the opposite action with probability $1 - p_i$. At each play, +1 (or −1) point is assigned to each agent in the minority (or majority). As the game evolves, if the accumulated score falls below a certain threshold, a new gene value is drawn (known as mutation of gene value) [25].

4. *Grand canonical MG (GCMG)*: In this type of MG, the number of players who participate in the game can vary since the players have the freedom of being *active* or *inactive* at any round of the game. More precisely, in a GCMG, agents would score their strategies as usual and if the highest strategy score falls below a certain threshold, agents would abstain from playing the game for that round of play. Any inactive agent re-enters the game when participation becomes profitable. Consequently, the attendance is calculated based on active players only [25].

2.5 MG Models in Communication Networks: State-of-the-Art

In this section, I provide a brief summary of the state-of-the-art, where MG is applied to solve the problems that arise in communication networks.

- *Backhaul management in cache-enabled SCNs*: In [30], authors provide a MG based framework for distributed backhaul management in cache-enabled SCNs. They model the SBSs as the players where the players are required to determine the number of predicted files to be cached from the core network. The total number of predicted files should be kept under a given threshold since the number of predicted files affects the allocated backhaul rate for the SBSs.

- *Interference management*: In [31], the authors investigated distributed interference management in Cognitive Radio (CR) networks using a novel MG-driven approach. They proposed a decentralized transmission control policy for secondary users, who share the spectrum with primary users thus causing interference. In this work, secondary users play an MG, selecting between two options, namely *transmit* or *not transmit*. The winning group is determined based on the interference experienced by the primary user. For instance, if the majority transmits, the interference power measured at the primary receiver exceeds the threshold so that the minority who does not transmit become the winners and vice versa, ensuring that the minority always wins. At each round of play, the primary receiver announces the winning group through sending a control bit to secondary transmitters.

- *Wireless resource allocation and opportunistic spectrum access*: An example of wireless channel allocation using MG can be found in [32] and [33] where an MG-based mechanism for energy-efficient spectrum sensing in cognitive radio networks (CRNs) is presented. The authors emphasized how the MG, due to its self-organizing nature, befits to model such problems to achieve cooperation and coordination gain without causing a large signaling cost in a CRN. In the applied MG model, the agents are the secondary users, who choose between *sensing* or *not sensing*, in the process of detecting an idle channel. Two different distributed learning algorithms are then developed that are applied by the agents to converge into equilibrium states characterized by pure and mixed strategy NE. The authors in [34] developed an MG-based slotted ALOHA channel allocation mechanism along with a traffic rate adaptation scheme for CRNs. In [35], multiple-choice MG was used for dynamic channel allocation in self-organizing networks. Therein, the set of transmitter and receiver pairs in the wireless network behave as players who attempt to access a channel from a limited number of available channels. In this model, each player is assigned a weight to account for different types of players. In [36], a multiple-choice MG (simplex game) was used to model the resource allocation problem in heterogeneous networks, where a large number of non-cooperative users compete for limited radio resources. The existence of correlated equilibrium was proved. Moreover, the authors compared the equilibria with the optimal states using the concept of the price of anarchy.

- *Coordination in delay tolerant networks*: In [37], an MG-based model was applied to coordinate the relay activation in delay tolerant networks in order to guarantee an efficient resource consumption. In the MG model, relays act as the players who decide to *transmit* (participate in relaying) or *not to transmit* (not to participate in relaying). In their work, the authors developed a stochastic learning algorithm that converges to a desired equilibrium solution.

2.6 MG in Next Generation Networks: Potential Applications and Open Problems

Although the applications of MG for 5G SCNs remain unexplored in the existing literature, such models can be well-justified. For instance, a variety of resource allocation problems in 5G SCNs can be considered as congestion problems. There, wireless users often attempt to either maximize a certain utility or minimize a certain cost rather than achieving a certain quality of service (QoS). Moreover, such systems typically involve a large number of users and hence, when modeled as large games, it is difficult to figure out an option that would necessarily yield a particular QoS. Thus, the users' best alternative is to opt for a less ambitious goal, for instance to be in the minority. This strategy increases the probability of achieving the required QoS. Consequently, the MG appears to be an appropriate tool to model such problems. In what follows, I discuss some possible applications of MG as well as open theoretical issues.

- Transmission mode selection: Device-to-Device (D2D) communication is considered as a building block in 5G networks. In D2D network underlaying an SCN, users have two modes of transmission: (i) direct communication without using the core network infrastructure, (ii) communication with the aid of the base station as in regular cellular networks. Clearly, the transmission mode selection problem can be modeled as an MG, where users are modeled as agents and the two options correspond to transmission via *D2D mode* or *cellular mode*. Given limited resource, the reward of each mode (for instance the throughput) then depends on the number of users selecting that mode. Thus, the cut-off value that defines the minority depends on factors such as number of interferers, number of available direct channels, etc.

- Multiple-choice games: It is clear that the current state-of-the-art mostly use the basic MG, which limits the agents to select between only two alternatives. Nonetheless, in many practical resource allocation problems, the decision is made among multiple choices. Examples include the channel selection or computation offloading problems where a channel/computational server is selected from many potential options.

- Evolutionary variations of MG: In a basic MG, the agents' selected set of strategies do not evolve. In other words, they stay fixed through out the iterations and are only scored in each play so that the agents can learn the best strategy for them. Hence, the basic iterated MG cannot be classified as an evolutionary game, making it inapplicable to model the problems where users should have the capability of altering their given strategies. On the other hand, complex dynamic systems require the agents to not only learn the best strategy but also to *adjust* the strategies as the game advances in a dynamic manner. To accomplish this, the modified versions of the seminal MG such as evolutionary MG (EMG) can be used. Other options include MG models that use learning methods such as *exponential learning* to adjust the strategies. The current state-of-the-art consists of very few applications of such evolutionary variations of the MG, thus it can be noted as a potential research direction.

- MGs for players with heterogeneity: From a practical point of view, it is very likely for the SCN users to be heterogeneous and to have diverse QoS requirements (e.g. in terms of delay, rate and energy efficiency). However, in the basic MG, every player is assumed to have similar capabilities and uses identical information and learning methods. Thus, generalization of the basic MG model to include such heterogeneities among users can be considered as a potential line of research.

The theory of MG introduced in this chapter will be used throughout this thesis. The following two chapters present the main contributions of the thesis.

Chapter 3

Computation Offloading in Small Cell Networks: When Minority Wins

3.1 Introduction

3.1.1 Overview

In this chapter, I provide a minority game (MG) based distributed computation offloading decision making mechanism for users in a small cell network (SCN). I consider multiple users accessing a single small base station (SBS) with edge computation capabilities. In other words, the small base station is equipped with computation resources, so that users attached to it compete for its computational resources by offloading their computationally expensive tasks. The users have the two options of either (i) *offloading their task to the SBS* or (ii) *to execute it locally.* Obviously, each user wants to make the more beneficial option out of the two, in terms of latency.

However, the more beneficial option varies with the total number of offloading users in each offloading period. For instance, if more users than the allowed capacity of SBS decide to offload, the SBS will become crowded and thus all offloading users will experience higher latency. Using the proposed scheme, the users are able to make offloading decisions in such a way that they can eventually coordinate to a state where the SBS is utilized with its maximum capacity, while the users' also achieve desired latency. It should be noted that using the MG-based method, users achieve this with no information exchange among them while using a minimal amount of external information. In other words, users learn from the environment and each user interacts with the aggregate behavior of all other users. This allows the decision making process to become almost autonomous. It is also shown that when users are coordinating, the SBS utilization is near maximum and users' utility is also comparatively higher than that of a random offloading decision making scenario.

3.1.2 Contribution

The main contributions of this chapter can be summarized as follows:

1. I modeled the computation offloading decision making problem considering a multi-user, single edge server (SBS) system as a minority game.

2. I developed a distributed offloading decision making algorithm for mobile users that enables users to make decisions almost autonomously by learning from the environment.

3. I provided simulation results for the given system model to show that the users reach a coordinated state where the the SBS utilization reaches near maximum, while meeting their required latency constraint.

Table 3.1: Symbols used in this chapter.

Symbol	Description
ϕ	System threshold for number of offloading users
$a_i(t)$	Action of user i at time t
$b(t)$	Control information (Winning choice)
C_b	Computational capability of SBS
C_u	Computational capability of local user device
$L(t)$	Latency experienced by an offloading user at offloading period t
L_{th}	Latency experienced by a locally computing user (Latency threshold)
M	Number of CPU cycles required to complete the task
$n(t)$	Number of offloading users at time t
S	Number of strategies per user
T	Number of offloading periods
$U_l(t)$	Utility received by a locally computing user
$U_o(t)$	Utility received by an offloading user
$V_{i,s}(t)$	Score of the strategy s of ith user at time t

3.2 System Model and Assumptions

In order to increase the readability, all the symbols that are used throughout this chapter are gathered in Table 3.1.

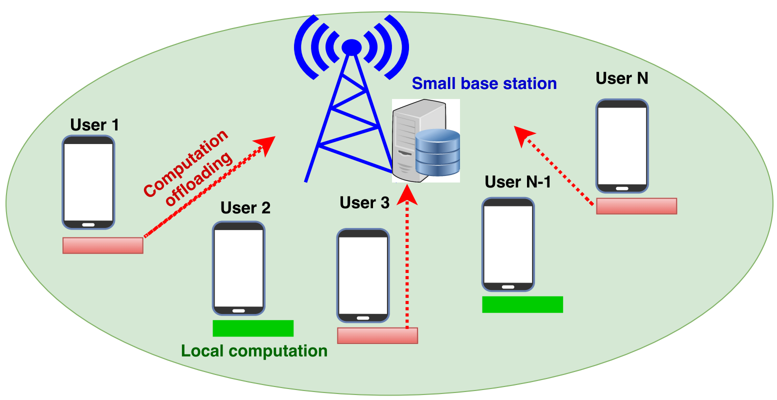


Figure 3.1: System model.

I use a minority game with an arbitrary cut-off. Consider an SBS serving N number of homogeneous (with respect to both computational capability and the task potentially to be offloaded) users. Each computational offloading period t is considered to be a round of play of the MG. All users participate as the players of the game and individually decide whether they offload the task to the local SBS or they execute it locally using their own resources. Users select one of these two options simultaneously within each offloading period and they have no information about other users' actions. Note that users naturally prefer to offload to the local SBS rather than executing the tasks locally, provided that the local SBS does not become crowded with offloading requests. The reason is as follows. Analogous to the original bar problem [23] where the customers naturally like to go to the bar than staying home if the bar is uncrowded, I assume that by offloading to an uncrowded SBS, users can experience lower latency. The SBS supports all of the computation offloading requests it receives, by completing all tasks in a TDMA manner. (This can be done using virtual parallel processing, where the time slot given to each task is small enough to assume that all tasks are performed simultaneously.) Therefore, if the number of offloading requests exceeds a certain threshold, the latency experienced by users would increase, making local computation to become the preferred option. Note that, for the sake of simplicity, here I only focus on the users' objective of meeting the latency requirements. Hence, for this case, I do not consider the other objective of users, which is minimizing the energy cost. Therefore, to make the analysis simple, I assume the amount of energy required for the local computation is approximately equal to the amount of transmission energy required for the offloading. Thus I omit the energy parameter in this model. An illustration of the system model is given in Fig. 3.1.

The problem described above is a distributed resource allocation problem where I analyze how to optimally utilize the computational resources of the local SBS while the latency remains below a specific threshold. As conventional, I assume that the local SBS has a fixed computational capability. In each round, users have the two options of either offloading or locally computing. The number of offloading requests that the SBS can handle is an arbitrary cut-off value denoted by ϕ. Clearly, this offloading threshold (ϕ) counts as the minority rule of the game. Note that ϕ remains unknown to the users throughout the game. For every user, L_{th} (in seconds) is the maximum tolerable latency, which is experienced if the computation is performed locally. Then the cut-off value ϕ is defined such that, when the number of offloading users approaches ϕ, the latency for offloading users reaches the threshold, L_{th}. Thus, for a user to benefit from offloading, the number of offloading users should not exceed the limit of ϕ. As a result, being in the population minority (defined by ϕ) is always desired. After each round of play, one of the outcomes mentioned below would occur.

- *If minority chooses to offload and majority chooses to compute locally*: In this case, the minority is rewarded since offloading yields lower latency than local computation since the local SBS is uncrowded.

- *If minority chooses to locally compute and majority chooses to offload*: In this case, the number of offloading requests exceeds the threshold ϕ so that the SBS becomes too crowded. Consequently, the latency for the offloading users would exceed the allowable latency threshold. Thus minority who did not offload, wins and is rewarded.

3.3 Minority Game Model

3.3.1 Attendance

Conventionally in MG, the winning choice is announced to all users after each round of play, so that users take advantage of this information to score their strategies. Accordingly, in our model, after each round of play, the SBS broadcasts the winning group by sending an one-bit control information $b(t)$ defined as

$$b(t) = \begin{cases} 1, & \text{if } n(t) < \phi \\ 0, & \text{if } n(t) \geq \phi \end{cases} \tag{3.1}$$

where $n(t)$ = number of offloading users at offloading period t.

In accordance with the MG terminology, I refer to this $n(t)$ as the *attendance*. Given the control information, users evaluate their strategies to improve decision making in the next round of play.

3.3.2 Reward

The reward of each winning user is defined based on the computation latency experienced by the user. Note that the transmission delays and propagation delays are considered negligible compared to computation latency. Thus the reward depends on the number of other users who select the same option, be it offloading or local computation.

- $L(t)$ = Latency experienced by an offloading user at offloading period t (in seconds),
- C_b = Computation capability of SBS (in number of CPU cycles per unit time),

- C_u = Computation capability of local user device (in number of CPU cycles per unit time),
- M = Number of CPU cycles required to complete the task.

Thus the latency experienced by an offloading user yields $L(t) = n(t) \cdot M/C_b$, and the latency experienced by a locally computing user is given as $L_{th} = M/C_u$. For $L(t) = L_{th}$, $n(t) = \phi$, hence $\phi = C_b/C_u$. If $L(t) \geq L_{th}$ (equivalent to $n(t) \geq \phi$), offloading users (majority) lose and locally computing users (minority) win. In contrast, when $L < L_{th}$ (i.e. $n(t) < \phi$), offloading users (minority) win and locally computing users (majority) lose. Let $U_o(t)$ and $U_l(t)$ denote the utility that each user respectively receives, in case of offloading and local computing. Thus

$$U_o(t) = b(t) = \begin{cases} 1, & \text{if } n(t) < \phi \\ 0, & \text{if } n(t) \geq \phi \end{cases} \tag{3.2}$$

and

$$U_l(t) = \begin{cases} 0, & \text{if } n(t) < \phi \\ 1, & \text{if } n(t) \geq \phi. \end{cases} \tag{3.3}$$

3.3.3 Distributed Learning Algorithm

To solve the designed MG, I use the seminal reinforcement learning technique [24, 25, 38] proposed in the original MG. For comparison purpose, MG-based method is compared with a random selection scenario.

Algorithm 1 Distributed learning algorithm to solve offloading MG [24]

Initialization: Each user i randomly draws S strategies.
for $t = 2 : T$ **do**
  Each user i selects action $a_i(t)$ predicted by the best strategy.
  SBS broadcasts the control information $b(t)$.
  for $s = 1 : S$ **do**
    Each user i updates the score of the strategy $s, V_{i,s}$.
    if prediction of $s = b(t)$ **then** $V_{i,s}(t+1) = V_{i,s}(t) + 1$,
    else$V_{i,s}(t+1) = V_{i,s}(t)$.
    end if
  end for
  Each user i selects best strategy $s_i(t)$, defined as $\arg\max_{s \in \mathcal{S}} V_{i,s}(t)$
  $t \leftarrow t + 1$
end for

Table 3.2: Simulation parameters.

Parameter	Value
ϕ	20
C_u	0.5 GHz
C_b	10 GHz
M	10
N	31
S	2
T	10000

3.4 Simulation Results and Discussion

For numerical analysis, I consider an SBS serving $N = 31$ users. The task that the users have to perform is assumed to require $M = 10$ Megacycles of CPU cycles. The CPU capacity of each user device is $C_u = 0.5$ GHz. Moreover, the SBS allocates $C_b = 10$ GHz of CPU capacity to serve users' offloading requests. Hence the system's cut-off value becomes $\phi = C_b/C_u = 20$. Thus, if the attendance is less than 20, the offloading users win, and vice versa. I simulate the system for different brain sizes (m) to observe the system behavior. For each m value, 32 runs are carried out, where each time, users randomly draw a new set of strategies $(S = 2)$. In each of the 32 runs, $T = 10000$ offloading periods (rounds of MG) are executed. For comparison, I also implement the random choice game where users select one of the two actions (to offload or not) with equal probabilities at each round of the game.

The variation of *attendance* over time for different brain sizes (m) is shown in Fig. 3.2. From the figure it is clear that the number of offloading users always fluctuates near the cut-off value, when users play an MG. It implies that, the system self-organizes into a state where the number of users, who experience a latency less than the threshold, is near its maximum value, thereby maintaining the optimal SBS utilization. This is interesting since the users are not given any prior information about the exact cut-off value of the system. As mentioned earlier, fluctuation of the attendance (i.e., the standard deviation) is known as the volatility in MG literature. It is clear that for different m values, the amount of fluctuation differs as explained in Fig. 3.3 (see below). Note that the fluctuations correspond to the amount of cooperation in the MG. They provide a measure for the number of users who could have offloaded if the attendance is lower than the cut-off or for the number of agents who could have not selected offloading option, if the attendance exceeds the cut-off. From the attendance figures, it can be seen that, even though the cut-off value is not advertised to the agents, the population adapts to the cut-off value of the system. The reason for this behavior is the agents' adaptation to the environment they collectively create.

Fig. 3.3 shows that the variation of the standard deviation (σ/N) of the number of offloading users over different m values follows the expected MG behavior [22] [25] [26], described in the following. As discussed, the volatility corresponds to the fluctuations of the attendance and serves as an established measure for the system performance. Lower volatility values mean that the fluctuations around the cut-off decrease. This

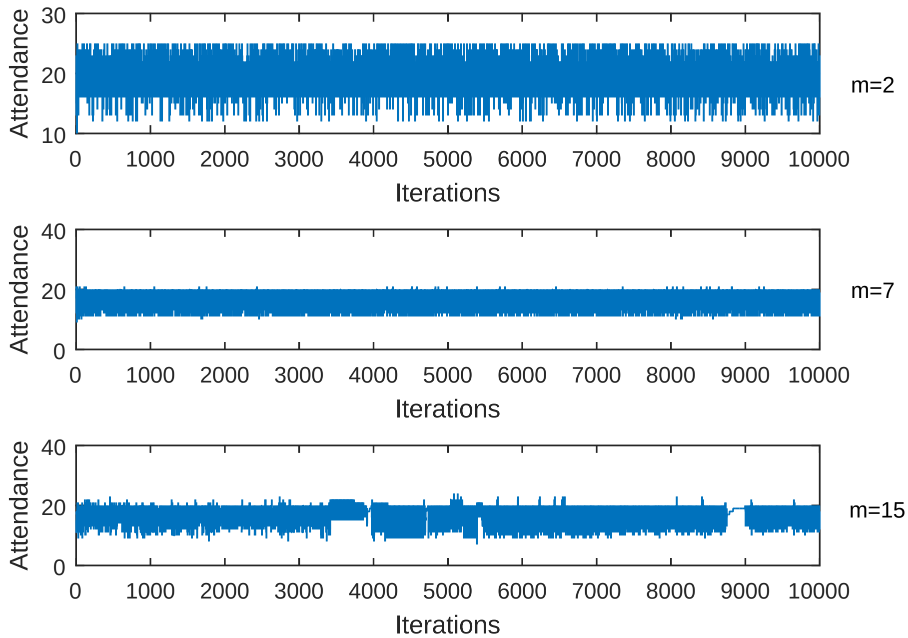


Figure 3.2: Time evolution of attendance.

corresponds to the size of the minority being larger, resulting in a larger number of winners, thus a better performance. Accordingly, lower volatility corresponds to better resource utilization and higher user satisfaction. From Fig. 3.3, for almost all values of m, volatility is lower than that of the random choice game. Hence one can conclude that the resource utilization is improved when MG-based offloading method is used. This shows the self-organizing nature of the MG, where agents coordinate to reduce the fluctuations in the absence of any communication or information other than the history data. It can also be seen that a minimum of the average volatility occurs at $m = 3$, where the phase transition from the *crowded phase* to the *uncrowded phase* occurs.

To investigate the improvement in the latency experienced by the users, average utility is shown in Fig. 3.4. It is clear that for MG, the utility achieved by individual

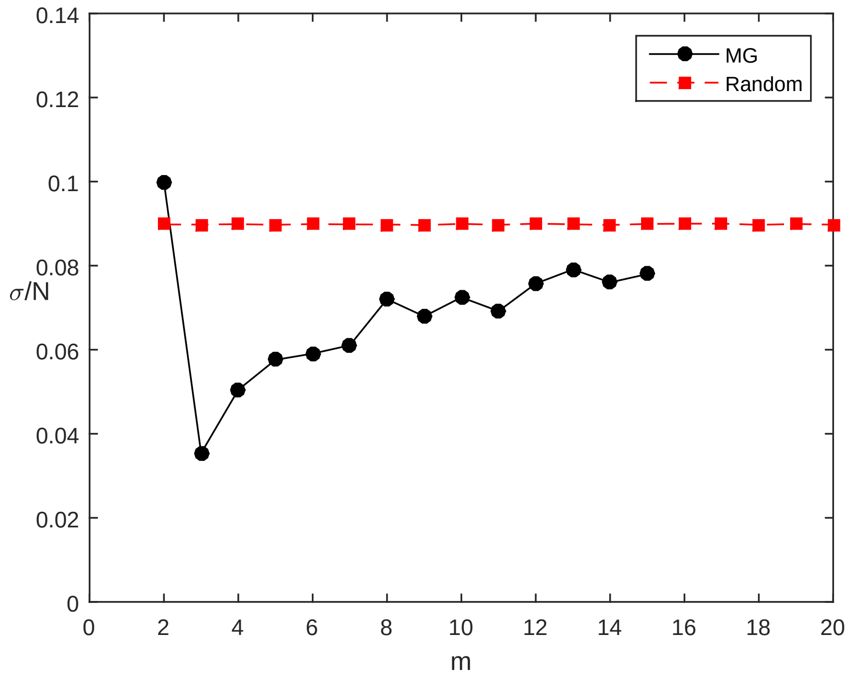


Figure 3.3: Variation of volatility with m.

users is better than that of the random choice game. However, it can be seen that the average utility received by a user who applies MG-based offloading is still lower than that of an optimal situation, where the number of offloading users is always equal to $\phi - 1$ (here 19). This is the price of the lack of coordination between agents and the use of minimal external information. Fig. 3.5 depicts the influence of the brain size on the average utility achieved per user. As expected, for larger volatility values, the achieved utility is substantially smaller. Roughly speaking, Fig. 3.5 is approximately an inverse of the volatility figure (Fig. 3.3). Therefore, the volatility is indeed an inverse performance measure for the MG-based system.

In Fig. 3.6, I illustrate the benefit of MG-based offloading. When the MG-based offloading mechanism is used, the number of users who experience latency below the threshold fluctuates near its maximum value ($\phi - 1 = 19$, in this example), compared

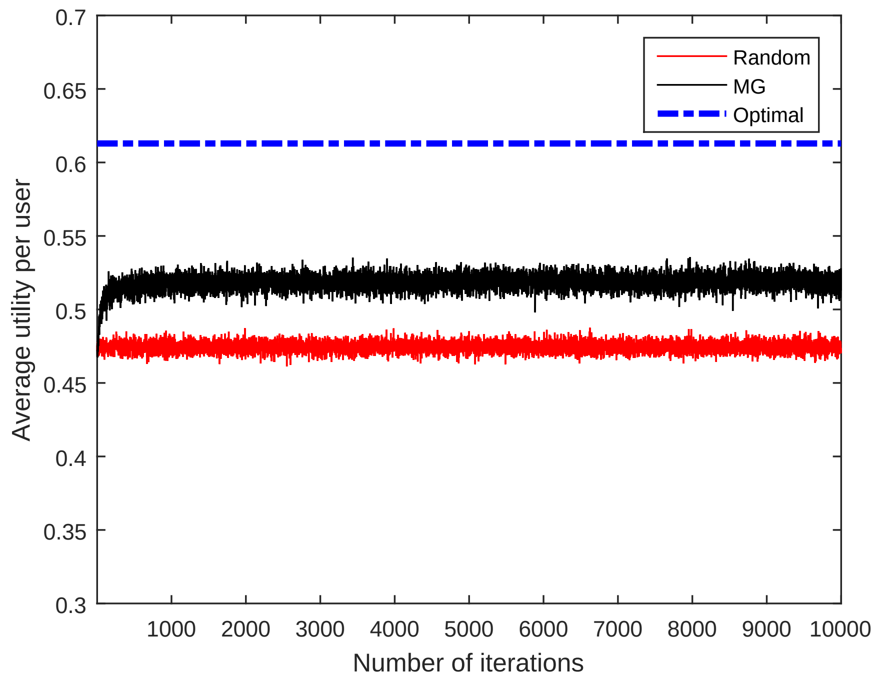


Figure 3.4: Average utility received by users.

to the two general cases where all users simply offload or compute the task locally. In the last two cases, none of the users is able to achieve a latency below the threshold. Using our defined model, if all users offload, the latency experienced by each user yields 31 milliseconds. Similarly, if all users choose to compute locally, the latency is 20 milliseconds. Since the threshold latency is 20 milliseconds, it is clear that none of the above two methods would allow the users to experience latency values *below* the threshold. Consequently, using an MG-based approach results in achieving some latency below the threshold for a larger number of users, thereby utilizing the available SBS resources in a productive manner.

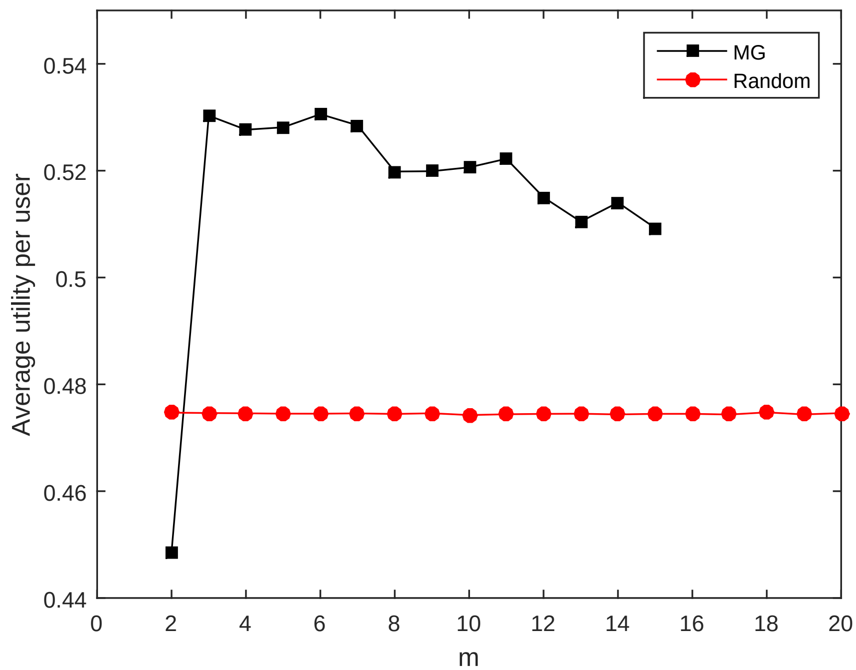


Figure 3.5: Average utility vs. m.

3.5 Summary

In this chapter, I have investigated the distributed mobile computation offloading problem as a potential application of MG in 5G SCNs. I considered a multi-user, single SBS system where the users attached to the considered SBS competing for its CPU resources. Therein, I developed a distributed offloading decision making algorithm using MG. This scheme achieves better SBS resource utilization while keeping the users' experienced latency below a desired threshold. Moreover, simulation results show that the MG-based learning method provides better performance compared to random offloading decision making method.

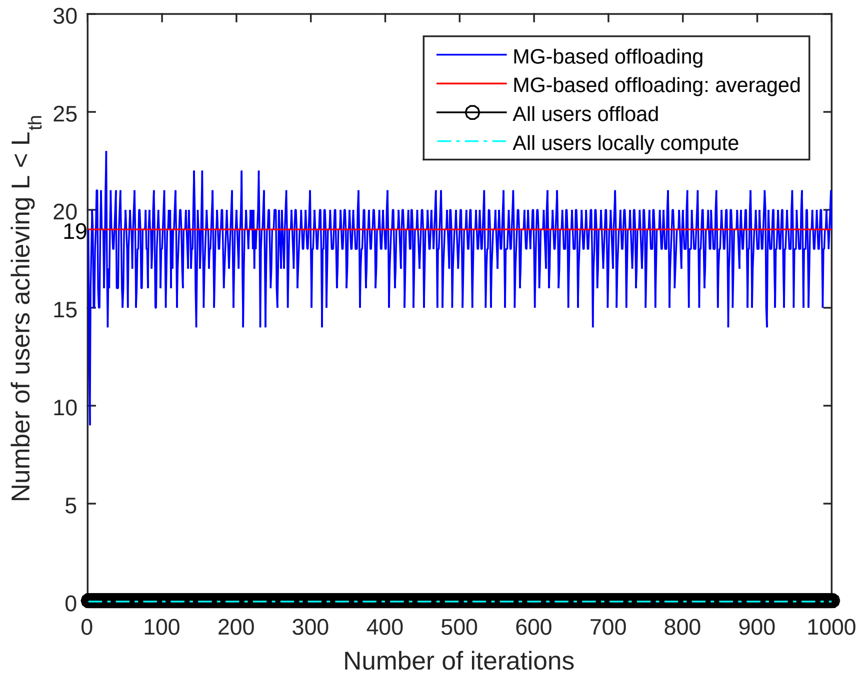


Figure 3.6: Number of users achieving below-threshold latency.

Chapter 4

Efficient Activation of MEC Servers Using Minority Games and Reinforcement Learning

4.1 Introduction

4.1.1 Overview

In MEC, efficient utilization of MEC servers is vital, since they have limited computational resources and power, compared to the traditional centralized cloud computing servers. To this end, one solution is to activate only a specific number of servers, while keeping the rest in the energy saving mode. At the same time, users' latency requirements should be taken into account, as overloading the servers with computational tasks can result in unacceptable delay. Therefore, addressing this trade-off is a major issue in developing efficient MEC systems. This becomes challenging in the presence of uncertainty in task arrival and/or in the absence of any central controller.

Therefore, in this chapter, I formulate the efficient edge server activation problem for a MEC network considering a pool of edge servers. To date, the great majority of the existing literature focuses on the user-centric objectives of efficient computation and radio resource allocation for multiple users, meeting users' delay constraints, and minimizing users' energy consumption. On the contrary, this work presents a hybrid view where both servers' and users' standpoints for a computation offloading environment are considered.

4.1.2 Contribution

Below I summarize the main contributions of this chapter.

1. First, I address the uncertainty caused by the randomness in channel quality and users' requests in an MEC offloading environment. There, I analyze the statistical characteristics of the offloading delay.

2. Then, I model the computation offloading system as a planned market, where the price of computational services is determined by a central governor.

3. Next, by using the theory of MG, I develop a novel distributed approach for efficient mode selection at the servers' side.

4. Then, using the proposed pricing framework in combination with the designed mode selection mechanism, I guarantee a minimal server activation to ensure energy efficiency, while meeting the users' delay constraints with adjustable certainty.

5. Numerical results are presented to illustrate the trade-off between the efficient server activation and meeting users' expected latency threshold.

6. Furthermore, formulated model is used to study the performance of different distributed learning methods including reinforcement learning and stochastic learning algorithms in an MG-setting. The performances of the learning algorithms are compared in terms of social and individual welfare of the servers as well as the QoE measure of the users.

4.2 System Model and Problem Formulation

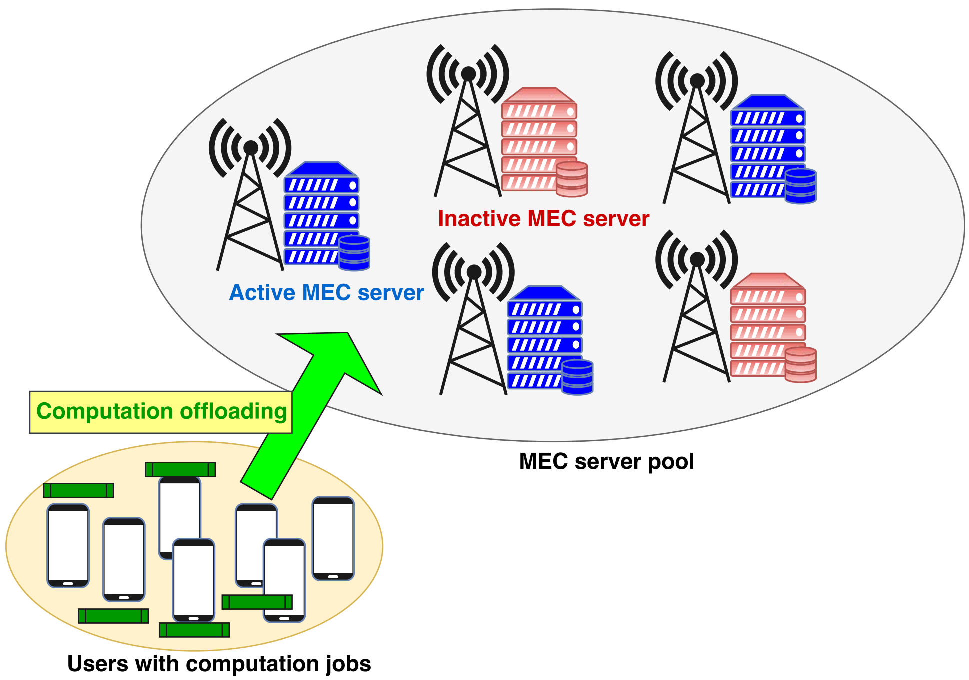


Figure 4.1: System model.

In order to increase the readability, all the symbols that are used throughout this chapter are gathered in Table 4.1. System model is illustrated in Fig. 4.1.

I consider an MEC system consisting of a virtual pool of M edge computational servers (e.g., small base stations), denoted by a set $\mathcal{M}$, and a set of users (e.g., mobile

Table 4.1: Symbols used in this chapter.

Symbol	Description
β	Desired QoE level for users
θ	Offloading delay
τ	Offloading delay for the last job in the queue of any active server
$a_i(t)$	Action of server i at time t
$c(t)$	Number of active servers at time t
c_{th}	System threshold for number of active servers
e_{f}	Fixed activation cost
e_{j}	Cost spent by a server per job
e_{p}	Price charged by a server per job
h	Channel power gain
K_{T}	Total number of offloaded jobs at every offloading period
$k(t)$	Number of jobs per active server at time t
k_{min}	Minimum required number of jobs per active server
k_{max}	Maximum allowed number of jobs per active server
$\mathcal{M}$	Set of servers in the pool
M	Number of servers
$\mathcal{N}$	Set of users
N	Number of users
$R(t)$	Reward received by a server at time t
R_{th}	Minimum desired reward
S	Number of strategies per server
t_c	Processing time, Truncated normal with parameters μ and σ.
t_0	Transmission time
T	Job deadline (maximum acceptable delay)
$U_{i,a}(t)$	Utility received by active servers
$U_{i,p}(t)$	Utility received by inactive servers
$w(t)$	Winning choice

devices). Each user has some delay sensitive computational jobs to be completed in consecutive offloading periods. Each offloading period is referred to as one *time slot*. In every time slot t, the users offload a total number of K_{T} computational jobs to the edge server pool. Prior to job arrival, every server independently decides whether to

- accept computation jobs (*active* mode); or
- not to accept any computation job (*inactive* mode).

To become active, each server incurs a fixed energy cost represented by e_{f} (dimensionless value). In addition, doing *each job* yields an extra e_{j} units of energy cost. By processing each job, a server receives a reimbursement (benefit) equal to $e_{\mathrm{p}} > e_{\mathrm{j}}$. Let $c(t)$ be the number of servers that decide to become active at time slot t. The total K_{T} jobs are equally divided among the active servers, so that the number of jobs per active server is given by

$$k(t) = K_{\mathrm{T}}/c(t). \tag{4.1}$$

Hence, each active server processes $k(t)$ jobs, and thus earns a total reward given by

$$R(t) = k(t)(e_{\mathrm{p}} - e_{\mathrm{j}}) - e_{\mathrm{f}}. \tag{4.2}$$

For each server, being in active mode is attractive only if a minimum desired reward, denoted by $R_{\mathrm{th}} > 0$ can be obtained. Then each active server has to receive at least

$$k_{\min} = \frac{R_{\mathrm{th}} + e_{\mathrm{f}}}{e_{\mathrm{p}} - e_{\mathrm{j}}} \tag{4.3}$$

jobs to achieve the minimum desired reward. Each computational job requires a random time to be processed by a server, denoted by t_c. I assume that t_c lies within the interval $(0, T)$ and has a truncated normal distribution with parameters μ and σ.

Moreover, considering Rayleigh fading, the channel power gain (h) is exponentially distributed with parameter ν. I model the round trip transmission delay (from the user to the server pool) as a linear function of the channel gain. The channel gains in both directions are assumed to be equal. Thus formally,[1]

$$t_0 = 2(ah + b), \tag{4.4}$$

where $a < 0$ and $b > 0$ are constants chosen such that, $t_0 \geq 0$.

The total offloading delay θ is, the sum of processing delay at the server t_c and the round trip transmission delay t_0. Thus,

$$\theta = t_c + t_0. \tag{4.5}$$

The following proposition characterizes θ statistically.

Proposition 4.2.1. *The probability distribution function (pdf) of offloading delay θ can be calculated as*

$$f_\Theta(\theta) = \left(\frac{\Omega\nu}{2a} e^{\left(-\frac{\nu(\theta-2b)}{2a} - \frac{\mu^2 - \left(\mu - \frac{\nu\sigma^2}{2a}\right)^2}{2\sigma^2} \right)} \right) \times \left(\Phi\left(\frac{T - \mu + \frac{\nu\sigma^2}{2a}}{\sigma} \right) - \Phi\left(\frac{-\mu + \frac{\nu\sigma^2}{2a}}{\sigma} \right) \right) \tag{4.6}$$

Moreover, the mean and variance of θ can be derived as,

$$\mu_\theta = \mu + \sigma \frac{\phi(\frac{-\mu}{\sigma}) - \phi(\frac{T-\mu}{\sigma})}{\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})} + \frac{2(a + b\nu)}{\nu}, \tag{4.7}$$

[1] Assuming a normal distribution for the time required to perform each job does not limit the applicability, and similar analysis can be performed with any other distribution. The same holds for the linear model of the transmission delay. Any other model can be used at the expense of additional calculus steps. Also, note that since the required energy to perform each job is proportional to the required time to perform that job, one might consider $e_j \propto \kappa\mu$ with $\kappa > 0$.

and

$$\sigma_\theta^2 = \sigma^2 \left(1 + \left(\frac{\frac{-\mu}{\sigma}\phi(\frac{-\mu}{\sigma}) - \frac{T-\mu}{\sigma}\phi(\frac{T-\mu}{\sigma})}{\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})} \right) - \left(\frac{\phi(\frac{-\mu}{\sigma}) - \phi(\frac{T-\mu}{\sigma})}{\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})} \right)^2 \right) + \frac{4a^2}{\nu^2}$$

respectively where $\Omega = \frac{1}{\sigma\left(\Phi(\frac{T-\mu}{\sigma})-\Phi(\frac{-\mu}{\sigma})\right)}$, *and* $\Phi(\epsilon) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{\epsilon}{\sqrt{2}}\right)\right)$ *is the cumulative distribution function (cdf) of a standard normal distribution.*

Proof. The proof follows by simple probability rules given the independence of t_c and t_0. The pdf of processing delay t_c is given by

$$f_{T_c}(t_c) = \begin{cases} \frac{\phi(\frac{t_c-\mu}{\sigma})}{\sigma\left(\Phi(\frac{T-\mu}{\sigma})-\Phi(\frac{-\mu}{\sigma})\right)}, & \text{if } 0 < t_c < T \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

where, $\phi(\epsilon) = \frac{1}{\sqrt{2\pi}}\exp(\frac{-\epsilon^2}{2})$ and $\Phi(\epsilon) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{\epsilon}{\sqrt{2}}\right)\right)$ are the pdf and cdf of the standard normal distribution respectively. Moreover, by using simple probability rules, the pdf of the round trip transmission delay t_0 yields

$$f_{T_0}(t_0) = \begin{cases} \frac{\nu e^{-\frac{\nu(t_0-2b)}{2a}}}{2|a|}, & \text{if } \frac{2b}{a} - \frac{t_0}{a} \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

Let Ω be

$$\Omega = \frac{1}{\sigma\left(\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})\right)}. \tag{4.10}$$

Since t_c and y are independent random variables, the pdf of offloading delay $\theta = t_c + t_0$

can be calculated as

$$
\begin{aligned}
&f_\Theta(\theta) = f_{T_c}(\theta) * f_{T_0}(\theta) \\
&= \int_0^T \Omega \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t_c-\mu)^2}{2\sigma^2}} \frac{\nu}{2a} e^{-\nu\left(\frac{\theta - t_c - 2b}{2a}\right)} \, dt_c \\
&= \left(\frac{\Omega\nu}{2a} e^{\left(-\frac{\nu(\theta-2b)}{2a} - \frac{\mu^2 - \left(\mu - \frac{\nu\sigma^2}{2a}\right)^2}{2\sigma^2}\right)} \right) \times \left(\Phi\left(\frac{T - \left(\mu - \frac{\nu\sigma^2}{2a}\right)}{\sigma} \right) - \Phi\left(\frac{-\left(\mu - \frac{\nu\sigma^2}{2a}\right)}{\sigma} \right) \right).
\end{aligned}
\tag{4.11}
$$

Given the pdf of t_c, the expected value and variance of t_c can be derived as

$$
\mathrm{E}[t_c] = \mu + \sigma \frac{\phi(\frac{-\mu}{\sigma}) - \phi(\frac{T-\mu}{\sigma})}{\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})} \tag{4.12}
$$

and

$$
\mathrm{Var}[t_c] = \sigma^2 \left(1 + \left(\frac{\frac{-\mu}{\sigma}\phi(\frac{-\mu}{\sigma}) - \frac{T-\mu}{\sigma}\phi(\frac{T-\mu}{\sigma})}{\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})} \right) - \left(\frac{\phi(\frac{-\mu}{\sigma}) - \phi(\frac{T-\mu}{\sigma})}{\Phi(\frac{T-\mu}{\sigma}) - \Phi(\frac{-\mu}{\sigma})} \right)^2 \right). \tag{4.13}
$$

Similarly, using pdf of t_0, the expected value and variance of t_0 are given as

$$
\mathrm{E}[t_0] = \frac{2(a + b\nu)}{\nu} \tag{4.14}
$$

and

$$
\mathrm{Var}[t_0] = \frac{4a^2}{\nu^2}. \tag{4.15}
$$

Therefore, due to the independence of t_c and t_0, the mean and variance θ are given as

$$\begin{aligned} \mu_\theta &= \mathrm{E}[t_c + t_0] \\ &= \mathrm{E}[t_c] + \mathrm{E}[t_0] \end{aligned} \quad (4.16)$$

and

$$\begin{aligned} \sigma_\theta^2 &= \mathrm{Var}[t_c + t_0] \\ &= \mathrm{Var}[t_c] + \mathrm{Var}[t_0]. \end{aligned} \quad (4.17)$$

□

Every user requires its offloaded job(s) be completed by a deadline T. Therefore, in every round t and for every server, the total processing time of all jobs, i.e.,

$$\tau = \sum_{i=1}^{k(t)} \theta_i, \quad (4.18)$$

should be less than T, so that the delay experienced by the last user in the queue does not exceed the deadline T as well. In other words, the condition $\tau \leq T$ ensures that all users receive their jobs completed before the deadline. Since θ_i are independent and identically distributed (i.i.d.), τ is the sum of $k(t)$ i.i.d. random variables. Therefore, the expected value and variance of τ are given by $k(t)\mu_\theta$ and $k(t)\sigma_\theta^2$, respectively. For large enough $k(t)$ (e.g., $k(t) \geq 30$), and by using the central limit theorem, the distribution of τ can be approximated as

$$\tau \sim \mathrm{Nor}(k(t)\mu_\theta, k(t)\sigma_\theta^2). \quad (4.19)$$

Due to the uncertainty caused by the randomness, deterministic performance guarantee in terms of delay is not feasible. Thus I resort to a probabilistic guarantee of users' QoE requirement. Formally, let $\Pr[\tau > T]$ be the probability that τ exceeds T, i.e., the likelihood that the delay requirement of some offloading user(s) is not satisfied. It is required that $\Pr[\tau > T]$ remains below a predefined threshold β. That is,

$$\Pr[\tau > T] \leq \beta. \tag{4.20}$$

Considering both the servers' and users' perspectives, the trade-off in the system can be seen as follows: On one hand, for each server it is beneficial to be active only if the number of active servers is less than a certain threshold $c_{\max}$, so that every active server receives the minimum number of jobs required to achieve the threshold reward (as stated by (4.3)). On the other hand, the users prefer that the number of jobs per server $k(t)$ is small enough so that their desired QoE is fulfilled with high probability, i.e., the number of active servers shall be larger than a certain threshold $c_{\min}$. In what follows, I will derive the values of $c_{\min}$ and $c_{\max}$ analytically. Therefore, for the offloading system to perform efficiently, the number of active servers at any offloading round t, i.e., $c(t)$ should be determined in way that both servers and users are satisfied. I denote this value by c_{th}.

4.2.1 Condition for Servers

Recalling (4.3), to achieve minimum desired reward R_{th}, each active server has to receive at least $k_{\min}$ jobs. Consequently, at most

$$c_{\max} = K_{\text{T}}/k_{\min} \tag{4.21}$$

servers can be in the active mode so that every active server receives the threshold reward R_{th}, while inactive servers receive no reward. Thus, the condition below should be satisfied when selecting the cut-off c_{th}:

$$\text{Condition I:} \qquad c_{\text{th}} \leq c_{\max}. \tag{4.22}$$

4.2.2 Condition for Users

Recall that the users' QoE requirement given by (4.20). Then, from (4.19) and (4.20), I have

$$1 - \Phi\left(\frac{T - k(t)\mu_\theta}{\sqrt{k(t)}\sigma_\theta}\right) \leq \beta, \tag{4.23}$$

which, by definition, is equivalent to

$$\text{erf}\left(\frac{T - k(t)\mu_\theta}{\sqrt{2k(t)}\sigma_\theta}\right) \geq 1 - 2\beta. \tag{4.24}$$

Since $\text{erf}(x)$ is an increasing function, $\text{erf}^{-1}(x)$ is also an increasing function. Therefore, (4.24) results in

$$k(t)\mu_\theta + \sqrt{2k(t)}\sigma_\theta \text{erf}^{-1}(1 - 2\beta) - T \leq 0. \tag{4.25}$$

Solving the quadratic inequality, I obtain;

$$\frac{-\sqrt{2}\sigma_\theta \text{erf}^{-1}(1 - 2\beta) - \sqrt{\Delta}}{2\mu_\theta} \leq \sqrt{k(t)} \leq \frac{-\sqrt{2}\sigma_\theta \text{erf}^{-1}(1 - 2\beta) + \sqrt{\Delta}}{2\mu_\theta} \tag{4.26}$$

where $\Delta = 2\sigma_\theta^2 \left(\mathrm{erf}^{-1}(1-2\beta)\right)^2 + 4\mu_\theta T$. Since $k(t) \geq 0$, considering only the right hand side of the inequality (4.26), I have

$$k(t) \leq \left(\frac{-\sqrt{2}\sigma_\theta \mathrm{erf}^{-1}(1-2\beta) + \sqrt{\Delta}}{2\mu_\theta} \right)^2. \tag{4.27}$$

Therefore,

$$k_{\max} = \left(\frac{-\sqrt{2}\sigma_\theta \mathrm{erf}^{-1}(1-2\beta) + \sqrt{\Delta}}{2\mu_\theta} \right)^2 \tag{4.28}$$

is the maximum allowable number of jobs per active server so that the users' QoE (i.e., latency) requirement is satisfied with probability $1-\beta$. Thus by (4.1), the minimum number of active servers $c_{\min}$ to guarantee the users' QoE satisfaction is

$$c_{\min} = \frac{K_{\mathrm{T}}}{k_{\max}}. \tag{4.29}$$

Therefore, the condition below should be satisfied when selecting the threshold c_{th}.

$$\text{Condition II:} \qquad c_{\mathrm{th}} \geq c_{\min}. \tag{4.30}$$

By conditions (4.22) and (4.30), the optimal number of active servers, c_{th}, is determined by solving the following equation:

$$c_{\min} = c_{\max}. \tag{4.31}$$

Or equivalently, the system performs optimally in terms of servers' energy and users' delay when c_{th} servers are active so that

$$k_{\min} = k_{\max}. \tag{4.32}$$

Thus, I obtain the threshold c_{th} using (4.21), (4.29), and (4.32) as

$$c_{\mathrm{th}} = \frac{K_{\mathrm{T}}}{k_{\max}}. \tag{4.33}$$

4.2.3 Modeling as a Planned Market

I model the proposed offloading system by a *planned market*, where the buyers correspond to the offloading users and the sellers map to the active servers. Unlike a free market, the demand of buyers is fixed here, since the total number of jobs offloaded at every time slot is equal to K_{T}. Moreover, the price of receiving computing services, i.e., e_{p} does not result from a competition among the sellers, but is determined by a central controller (for instance, macro base station or network planner) to ensure that the entire system works efficiently, i.e., (4.32) is satisfied. More precisely, by (4.2), if the price is too high, a server can achieve the threshold reward R_{th}, even by working under capacity. However, the network planner would like to determine the price so that the minimum number of servers are active. Therefore, the planner desires to ensure that every active server receives enough jobs (K) to fulfill its capacity, while keeping K under the maximum value allowed by the users' QoE requirement. Moreover, every active server working at its capacity should achieve R_{th}. To address this trade-off, e_{p} is determined by a central controller with respect to the the ideal number of sellers (c_{th}) to ensure system efficiency, i.e., to ensure that (4.32) is satisfied. Then, using (4.3), I have

$$e_{\mathrm{p}} = \frac{R_{\mathrm{th}} + e_{\mathrm{f}}}{k_{\max}} + e_{\mathrm{j}}, \tag{4.34}$$

with $k_{\max}$ given by (4.28).

Now the challenge is to activate c_{th} servers in a self-organized manner, which is addressed in the following section.

4.3 Modeling the Problem as a Minority Game

I model the formulated server mode selection problem as an MG, where the M servers represent the players, with a cut-off value c_{th} for the number of active servers. In each offloading period, the servers decide between the two actions, i.e., being *active* or *inactive*, denoted by 1 and 0 respectively. I denote the action of a given player i in the time slot t by $a_i(t)$. The number of active servers $c(t)$ maps to the *attendance*. Each player has S strategies. According to our formulated servers' mode selection problem and analysis in Section 4.2,

- If $c(t) \leq c_{\text{th}}$, each of the $c(t)$ active servers (the minority) earns a reward higher than or equal to the minimum desired reward, R_{th}.
- If $c(t) > c_{\text{th}}$, $c(t)$ active servers cannot achieve R_{th}. In this case, inactivity (i.e., the action of the minority) is considered as the winning choice, since inactive servers spend no cost without being properly reimbursed.

4.3.1 Control Information

After each round of play, a central unit (e.g., a macro base station) broadcasts the winning choice to all servers by sending a one-bit control information:

$$w(t) = \begin{cases} 1, & \text{if } c(t) \leq c_{\text{th}} \\ 0, & \text{otherwise.} \end{cases} \tag{4.35}$$

Note that neither the actual attendance value $c(t)$ nor the system cut-off c_{th} is known by the players.

4.3.2 Utility

Let $U_{i,\mathrm{a}}(t)$ and $U_{i,\mathrm{p}}(t)$ denote the utility that server i receives for being active and being inactive, respectively. Based on the discussion above, I define

$$U_{i,\mathrm{a}}(t) = \begin{cases} 1, & \text{if } c(t) \leq c_{\mathrm{th}} \\ 0, & \text{otherwise} \end{cases} \tag{4.36}$$

and

$$U_{i,\mathrm{p}}(t) = \begin{cases} 1, & \text{if } c(t) > c_{\mathrm{th}} \\ 0, & \text{otherwise.} \end{cases} \tag{4.37}$$

4.3.3 Distributed Learning Algorithm

Every player applies the seminal strategy reinforcement technique [24] given in the original MG formulation. The algorithm is summarized in **Algorithm 2** for some

Algorithm 2 Distributed learning algorithm to solve server mode selection MG [28]

1: **Initialization**: Randomly draw S strategies from the universal strategy pool, gathered in a set $\mathcal{S}$. Moreover, For every $s \in \mathcal{S}$, set the score $V_{i,s}(0) = 0$.
2: **for** $t = 1, 2, ...$ **do**
3: If $t = 1$, select the current strategy, $s_i(1)$, uniformly at random from the set $\mathcal{S}$. Otherwise, select the best strategy so far, defined as

$$s_i(t) = \arg\max_{s \in \mathcal{S}} V_{i,s}(t). \tag{4.38}$$

4: Select the action $a_i(t)$, predicted by $s_i(t)$ as the winning choice.
5: The central unit broadcasts the control information (winning choice), $w(t)$.
6: Update the score of the strategy $s_i(t)$ as

$$V_{i,s}(t+1) = \begin{cases} V_{i,s}(t) + 1, & \text{if } a_i(t) = w(t) \\ V_{i,s}(t), & \text{otherwise} \end{cases} \tag{4.39}$$

7: **end for**

Table 4.2: Simulation parameters.

Parameter	**Value**
β	0.05
μ	7
ν	1
σ^2	2
a	-1
b	2.5
e_{f}	50
e_{j}	5
K_{T}	500
M	21
R_{th}	100
S	2
T	0.35s

player i. Details can be found in [28].

4.4 Numerical Results

For numerical analysis, simulation is carried out for 32 runs and in each run, the servers randomly draw S strategies and repeatedly execute the MG for 10000 offloading periods. All simulation parameters are given in Table 4.2. For these values, using (4.33), the cut-off value yields $c_{\mathrm{th}} = 15$. Random choice game where each server selects its action uniformly at random is also simulated for comparison purposes.

In Fig. 4.2, I present the variation of important system parameters as a function of users' QoE index β (see Section 4.2). From the figure, the following can be concluded: As β increases, the number of required active servers (c_{th}) decreases, thereby allowing a larger number of offloading tasks to be processed per server. Similarly, the maximum allowable number of tasks per active server ($k_{\max}$) increases with increasing β. In fact, with larger β, larger delay (τ) is tolerable, or in other words, longer task queue ($k_{\max}$)

is allowed. Naturally, in this case, the price per task, e_{p}, reduces, as intuitively expected for a weaker service.

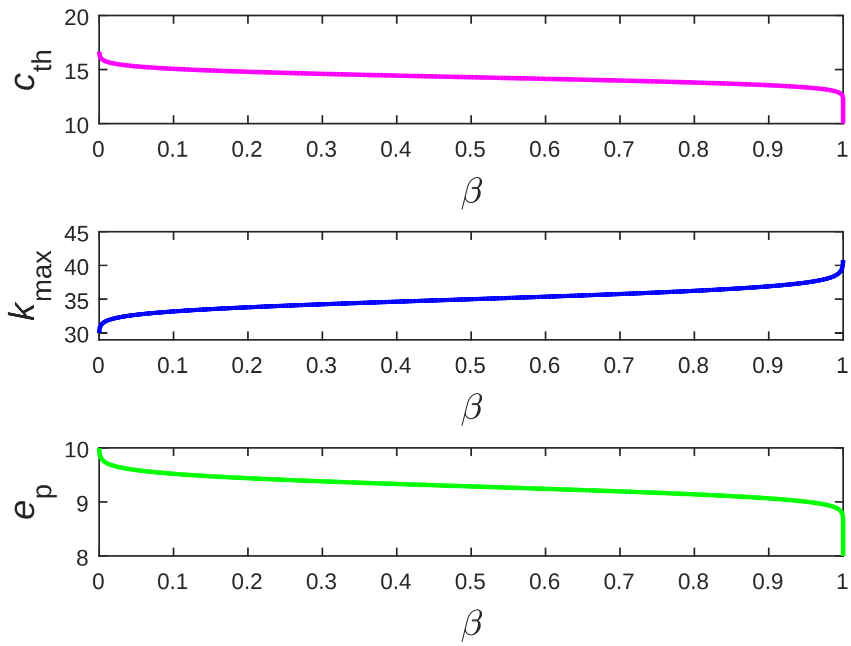


Figure 4.2: Variation of c_{th}, k_{max}, and e_{p} as a function of β.

Figure 4.3 shows the fluctuation of the attendance around the threshold c_{th} for different β values. From the figure, it can be seen that the servers are able to self-organize so that the attendance remains near the maximum allowed number of active servers (i.e., c_{th}), although this value is unknown to servers. Thus, the computational pool operates at an energy-efficient point while the delay requirement of users are satisfied with high probability.

In MG, the fluctuation of attendance around the threshold is quantified using a parameter known as the *volatility*, formally defined as the standard deviation of the attendance. As mentioned previously, this value also corresponds to the social welfare of the system. In Figure 4.4, the variation in volatility is shown as a function of the

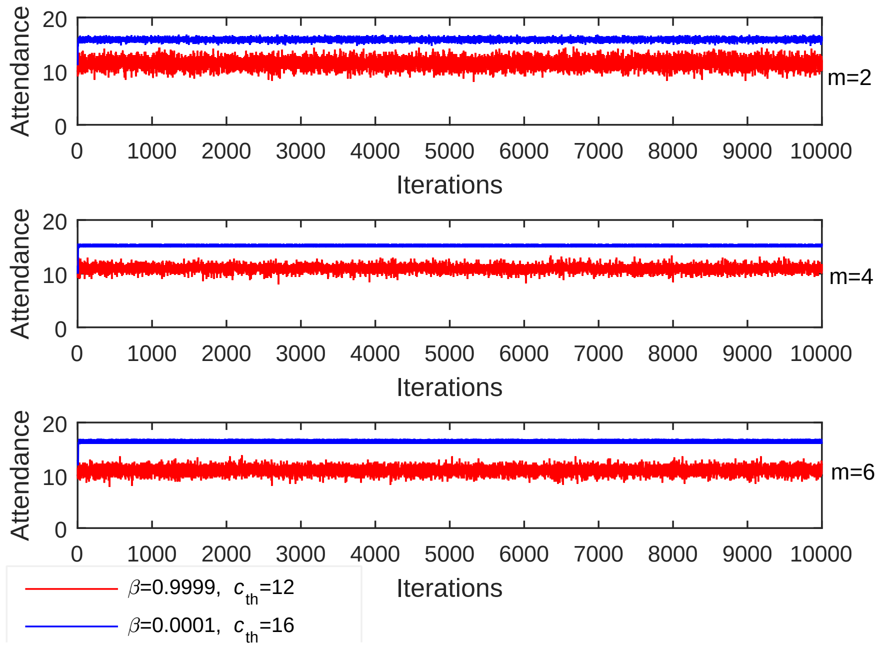


Figure 4.3: Chapter 4: Time evolution of attendance.

brain size (m). Naturally, it is desired to have small volatility, implying that the fluctuations near c_{th} is small, and the number of winners is very often close or equal to the maximum possible number of active servers (c_{th}). A more detailed explanation on how the brain size affects volatility can be found in [22, 38].

Figure 4.5 illustrates that as the number of active servers in the pool increases, users' probability measure, i.e., $\Pr[\tau > T]$ decreases. This is because when a larger number of servers choose to become active, the number of tasks per server becomes smaller thus making the offloading delay for the last task in the queue become smaller. The figure also shows that when the number of active servers exceed the cut-off $c_{th} = 15$, the desired $\beta = 0.05$ can be achieved.

Fig. 4.6 shows the changes in users' probability measure, i.e., $\Pr[\tau > T]$. The users meet their QoE certainty requirement whenever $c(t) > c_{\text{th}}$. As the attendance fluctuates near c_{th}, the probability value also remains near the desired certainty.

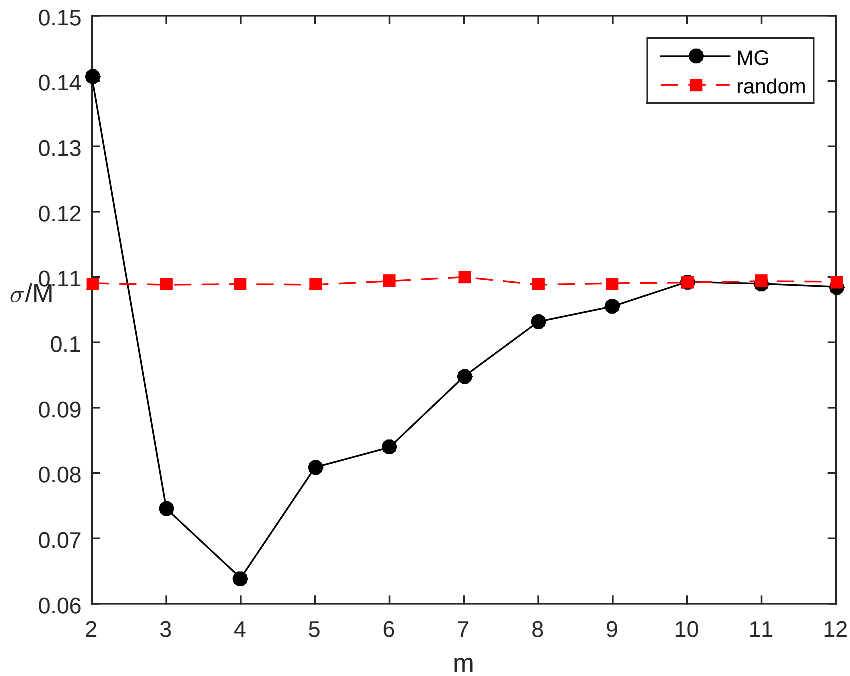


Figure 4.4: Chapter 4: Variation of volatility with m.

The average utility per user is depicted in Fig. 4.7. It can be seen that the utility of MG-based strategy is higher than that of random selection. Yet, it is below the average utility of the optimal scenario. This is due to the fact that in MG-based method, servers make decisions under minimal external information, and without any coordination with other servers.

Figure 4.8 shows the average number of servers that achieve a reward greater than or equal to the desired threshold reward, R_{th}. According to the numerical analysis, while the maximum possible number of servers that can obtain $R \geq R_{\text{th}}$ remains c_{th}, MG based procedure performs superior to the random selection though it is still less than the desired maximum. Moreover, alternative approaches such as all servers choosing to become active or all servers choosing to become inactive result in zero servers being able to receive $R > R_{\text{th}}$. In other words, if all servers make the same

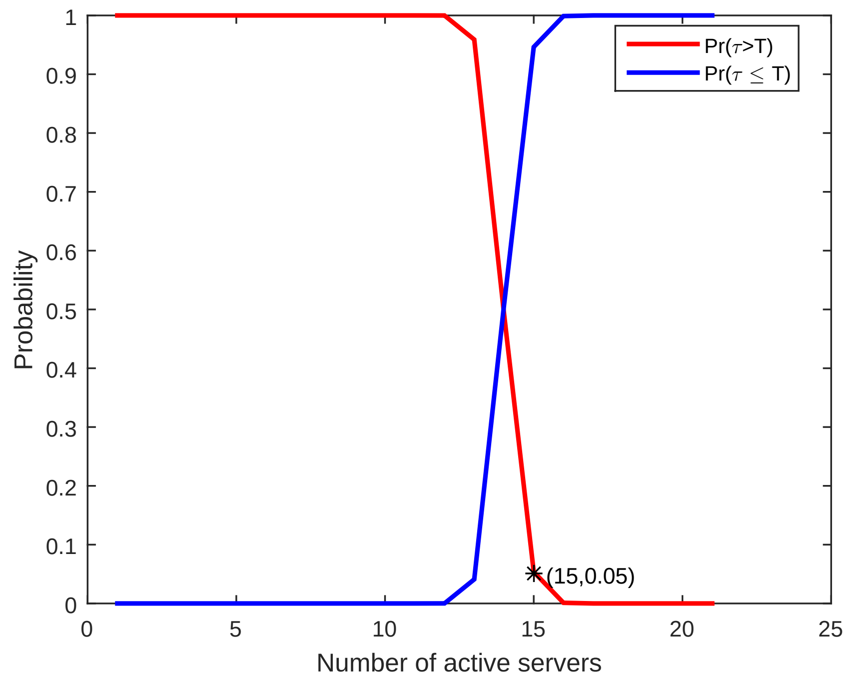


Figure 4.5: Variation of users' QoE measure vs number of active servers.

decision to be either active or inactive, all servers land in the majority thus resulting them to lose.

4.5 Applying Different Learning Algorithms

In an MG, the agents apply an algorithm to learn the best action to be played in the next round of play. Apart from the seminal reinforcement learning mechanism (given in Algorithm 2), a variety of learning algorithms are available in the MG literature that can be used by agents to learn the best action. Many of these algorithms fall into the category of reinforcement learning, where the learners balance the exploration-exploitation trade-off in order to maximize their utilities. In addition, learning methods based on stochastic strategies are also available where agents choose their actions with some probability.

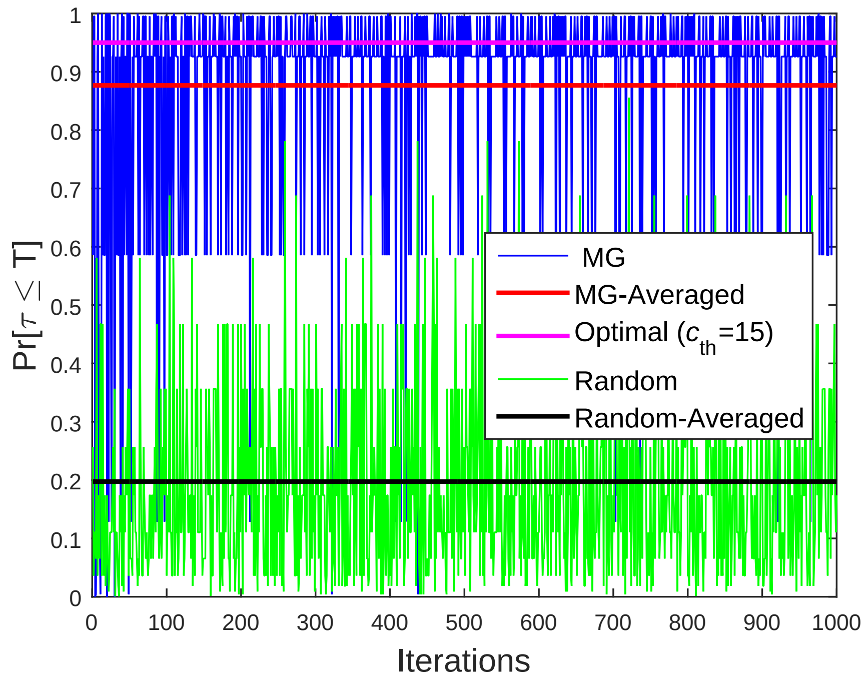


Figure 4.6: Users' QoE measure.

Reinforcement learning is a well-known technique applicable to distributed decision making. In reinforcement learning, autonomous agents learn the best action by using the rewards and penalties received in each round of play. Since agents do not know which action is the best, they learn by balancing exploration of unknown actions and exploitation of the current knowledge of used actions. In other words, agents use trial and error approach to maximize their utilities over the horizon. Reinforcement learning mechanisms are very-well suited for learning in MG, since adaptation to the collective action of the other agents in the presence of information scarcity can be achieved using such methods. In what follows, I introduce some of these algorithms and their applicability in an MG setting using the same system model given in Section 4.2.

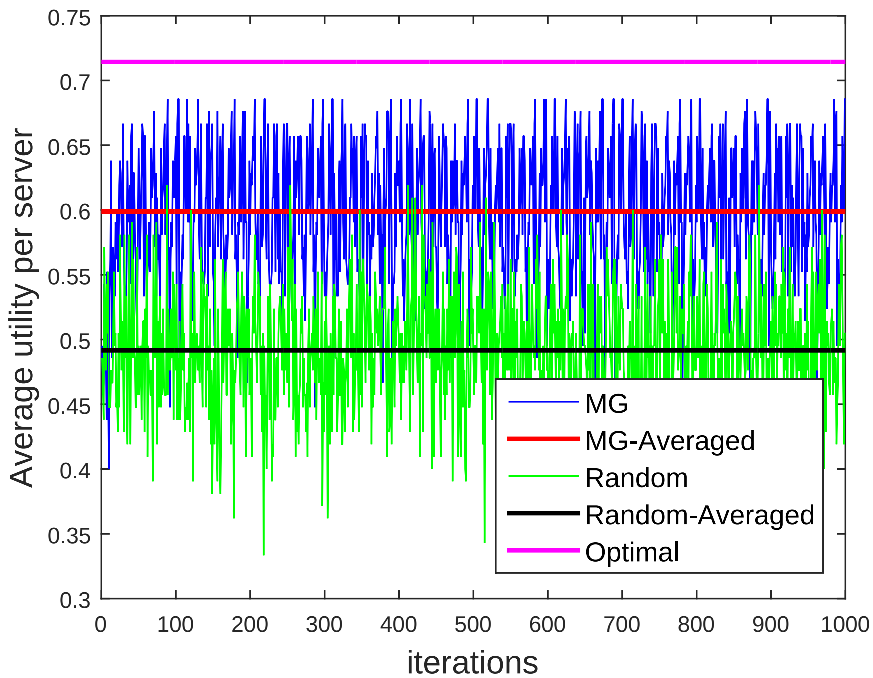


Figure 4.7: Average utility of a server.

4.5.1 Exponential Learning

In [27], exponential learning is applied in MG. Each agent is given S strategies, and the agent scores each of these strategies based on the accuracy of its prediction of the winning action. Each agent i selects a strategy s with some probability $p_{i,s}(t)$, defined as: $p_{i,s}(t) = e^{\gamma_i V_{s,i}(t)}/(\sum_{s'=1}^{S} e^{\gamma_i V_{s',i}(t)})$, where $V_{s,i}(t)$ is the score of strategy s at time slot t. Moreover, γ_i is the *learning rate* of each agent. Note that, $\gamma_i = \infty$ corresponds to selecting the strategy with the highest score which is the seminal MG learning algorithm.

4.5.2 Q-Learning

In [39], Q-learning is applied in an MG, where each agent keeps track of the Q-value of two actions. Every agent i uses the following rule to update the Q-values, where

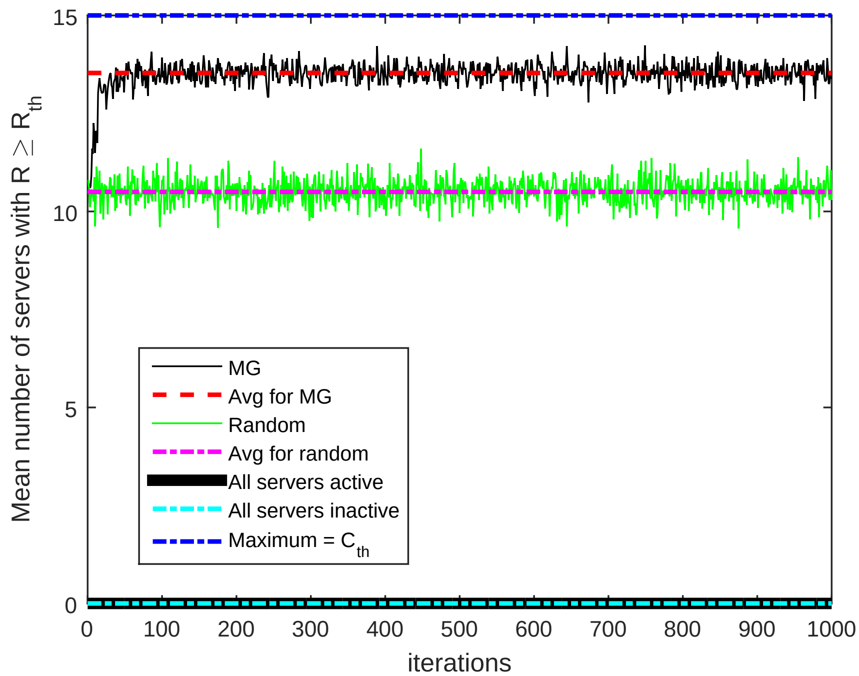


Figure 4.8: Average number of servers with $R \geq R_{\text{th}}$.

$U_{i,a}$ is the utility received by agent i as a result of some action a. This rule makes use of the utility information (U_i) possessed by the agents in order to learn the best action (i.e., exploitation of the available information). The Q-learning in MG is two fold; (i) Q-values are determined for the two actions (I refer to this as *Action-based Q-learning*) and (ii) Q-values are determined for agents' strategies (I refer to this as *Strategy-based Q-learning*). In the second scenario, an agent keeps track of the Q-values for each of her *strategies*:

$$Q_{i,a}(t+1) = \begin{cases} Q_{i,a}(t) + \gamma_i(U_i(t) - Q_{i,a}(t)), \;\; a_{i,t} = a \\ Q_{i,a}(t), \;\; \text{O.w.} \end{cases}$$

Given Q-values and some $\epsilon > 0$, every agent selects the action with the highest Q-value with probability $1 - \epsilon$, and with probability ϵ selects an action uniformly at random (i.e., exploration).

4.5.3 Adaptive Strategy

Authors in [40] developed an adaptive learning strategy for MG. Therein, for each actions a, each agent i calculates a parameter called *attractiveness* ($t_{i,a}$) defined as: $t_{i,a} = (1 - x_{i,a})h_a + x_{i,a}U_{i,a}$, where $x_{i,a}$ is the *attitude* of action a, which is initially selected randomly from $[0, 1]$. Moreover, h_a is the fraction of rounds in which action a has won in a given history of the game. The action with the highest attractiveness is chosen by the agent in the next round of the play. As the game evolves, in each round of play, an agent adapts her attitude values such that if agent selects action a and wins, $x_{i,a}$ will be increased by some constant $a+$ whereas if agent selected action a and lost, $x_{i,a}$ will be decreased by some constant $a-$.

4.5.4 Win-Stay Lose-Shift Strategy

In [41], this learning method is presented as a simple behavioral model for the agents playing an MG. This is a stochastic strategy-based learning method. If an agent wins in the current round of the game, she selects the same action in the next round. In contrast, if the agent loses, she will choose the other action with some probability p. Authors analytically showed that for small enough p values, the social welfare (i.e., volatility) of the system approaches the optimal value. More precisely, for the MG with N odd players and $N/2$ cutoff value, p is chosen such that $p = x/(N/2)$ where $x \ll N$.

4.5.5 Roth-Erev Learning

This learning method is applied in MG in [39]. Similar to the Q-learning, an agent determines a weight for each of her actions, denoted by q_a and referred to as *action weights*. However, unlike Q-learning, q_a is defined as the sum of the initial action weight and the discounted sum of all past utility values received for playing action a (λ is referred to as the *discount factor*). Agents use the following rule to update the actions' weight:

$$q_a(t+1) = \begin{cases} \lambda q_a(t) + U_{i,a}(t), \ a_{i,t} = a \\ \lambda q_a(t), \ \text{O.w.} \end{cases}$$

Given the values of q_a, the selection probability of action a is defined as $p_a = \frac{q_a}{\sum_{a'} q_{a'}}$.

4.5.6 Learning Automata

According to [39], learning automata can be applied as an MG learning mechanism, by using the following rule to update the probability of playing every action a, denoted by p_a, after each round of play:

$$p_a(t+1) = \\ \begin{cases} p_a(t) + \gamma U_{i,a}(1 - p_a(t)) - \delta(1 - U_{i,a})p_a(t), \ a_{i,t} = a \\ p_a(t) - \gamma U_{i,a} p_a(t) + \delta(1 - U_{i,a})(\frac{1}{2} - p_a(t)), \ \text{O.w.} \end{cases}$$

Here γ and δ are known as the *reward rate* and *penalty rate*, respectively.

4.5.7 Random selection

In a random selection scenario, agents simply select one of the two actions uniformly at random.

4.5.8 Numerical Results for Different Learning Algorithms

I choose $M = 21$ and $c_{\text{th}} = 10$. Simulations are carried out for 32 runs and in each run, the servers repeatedly execute the MG for 10000 offloading periods. I compare all aforementioned learning methods based on the social- and individual welfare of servers as well as users' QoE measure. For different learning schemes, the parameters are selected as follows, using the best values as suggested in the literature:

- Exponential learning: $\gamma = 100$.
- Q-learning: $\gamma = 0.1$, $\epsilon = 0.01$.
- Adaptive strategy: Initial attitude values $x_{i,0} = x_{i,1} = 0.5$, and $a+ = a- = 0.5$, $\forall i \in \{1, ..., M\}$.
- Win-stay lose-shift strategy: $p = 0.005$.
- Roth-Erev learning: $\lambda = 0.2$.
- Learning automata: $\gamma = 0.2$ and $\delta = 0.3$.
- Seminal MG: $S = 2$.

In Fig. 4.9, I show the variations in the volatility (i.e. inverse global efficiency) as a function of the parameter $\alpha = 2^s/M$, with s being the *memory size*, i.e., the length of the historical data used by the agents for learning. It can be seen that exponential learning method achieves the best social welfare (inversely proportional to the volatility), with its lowest volatility approaching to 0. In addition to examining the social welfare of the system, I also investigate the performance of each learning method in terms of individual welfare of the servers. In doing so, I illustrate the average utility per server during the entire the game for each learning algorithm in

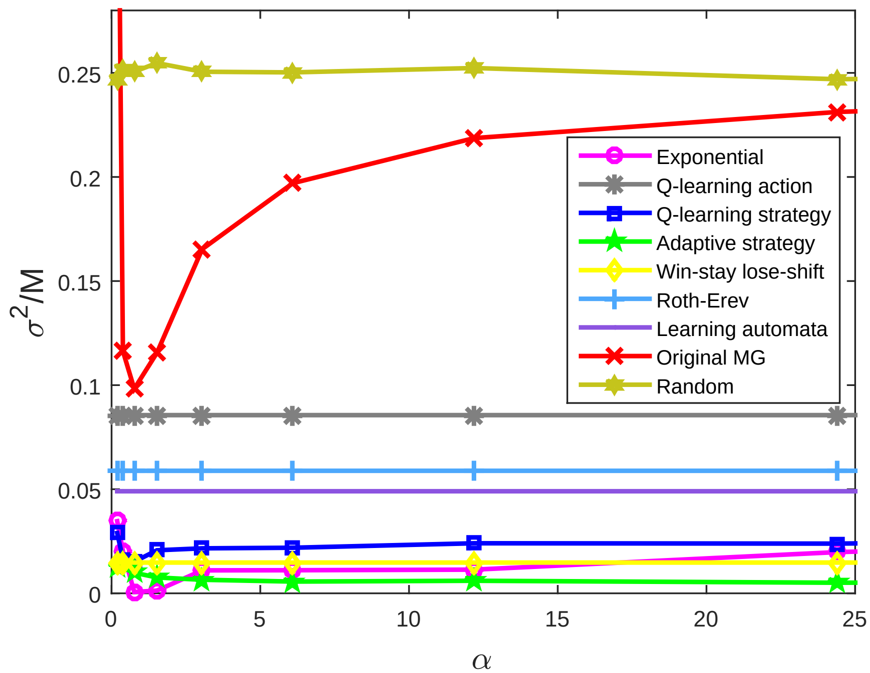
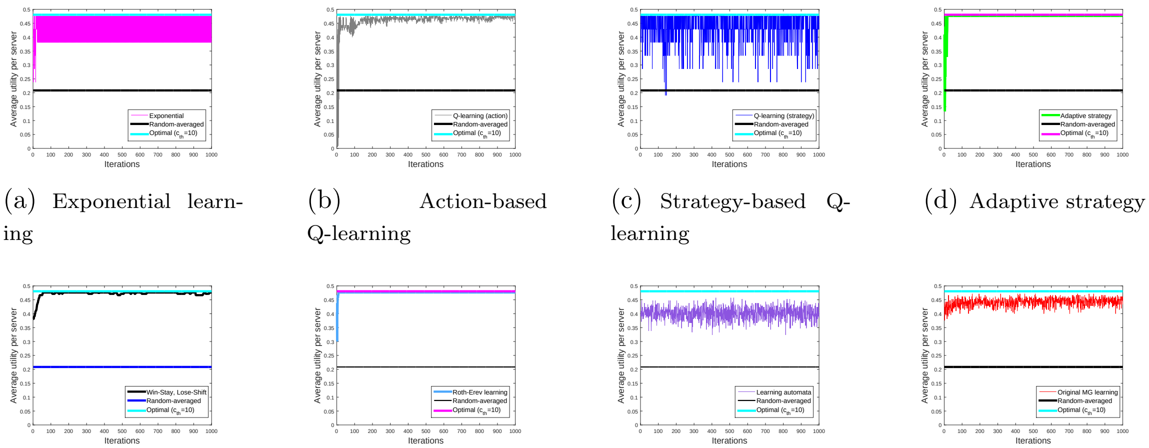


Figure 4.9: Performance comparison in terms of (inverse) global efficiency.

Fig. 4.10. Moreover, in Fig. 4.11, I compare the performances of algorithms in terms of the average utility. It can be concluded that using an appropriate learning method, a near-optimal average utility is achievable by the servers, despite not having any prior information.

In Fig. 4.12, I present the performance of each learning algorithm in terms of the users' experienced QoE measure. Finally, in Fig. 4.13, I compare the performance of different learning methods in terms of users' experience. To this end, I simulate the probability that the total time required by each server to perform all tasks, denoted by τ, exceeds the deadline T. Naturally, $\Pr[\tau \leq T]$ is then the probability that no user, even the last one in the queue, would experience a delay larger than T to have its offloaded task done.

The learning methods such as exponential learning, adaptive strategy, win-stay



(a) Exponential learning (b) Action-based Q-learning (c) Strategy-based Q-learning (d) Adaptive strategy

(e) Win-stay lose-shift strategy (f) Roth-Erev learning (g) Learning automata (h) Original MG algorithm

Figure 4.10: Average utility per server for different learning algorithms

lose-shift strategy and Q-learning that exhibit better performance than the seminal inductive learning method help servers achieve better coordination and thus form larger minorities. This reduces wastage of computation server resources and hence improves the resource allocation efficiency. Therefore, these methods can be recommended as more efficient and sophisticated learning rules for the formulated MG-based server selection problem.

4.6 Summary

This chapter presents an energy efficient, distributed edge server activation scheme focusing on the perspectives of both servers and users. In doing so, I address the uncertainty caused by the randomness in channel quality and users' requests. I first analyze the statistical characteristics of the offloading delay. Based on this, I model the computation offloading problem as a planned market, where the price of computational services is determined by a central governor. Afterward, by using the theory of MG, I develop a novel approach for efficient mode selection at the servers' side. The

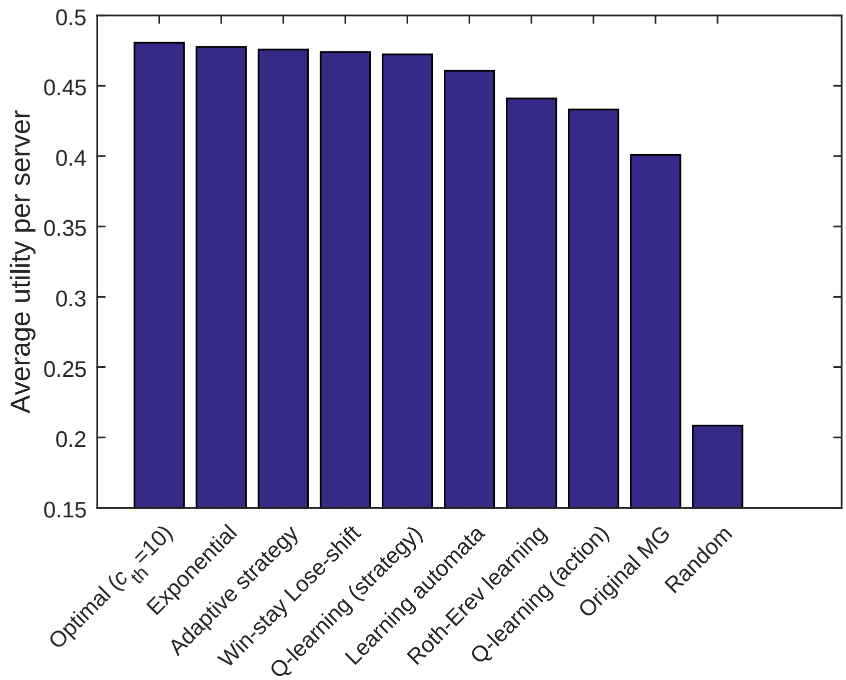
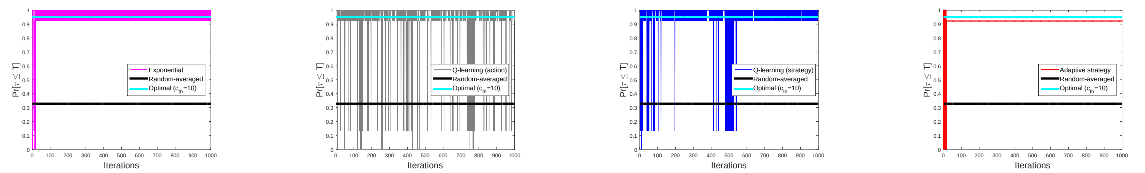
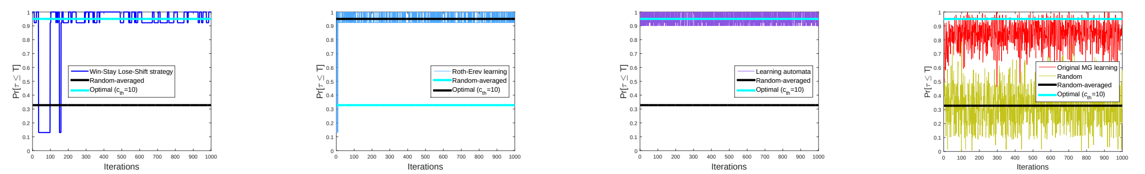


Figure 4.11: Performance comparison in terms of average utility for each server.

proposed pricing framework in combination with the designed mode selection mechanism guarantee a minimal server activation to ensure energy efficiency, while meeting the users' delay constraints with adjustable certainty. Moreover, the mode selection scheme is distributed, and does not require any prior information at the servers' side. Numerical results are presented to illustrate the trade-off between the efficient server activation and meeting users' expected latency threshold. Then, I apply a number of different distributed learning algorithms to solve the formulated MG. I obtain the simulation results to compare the performance of these learning methods in terms of the social and individual welfare of servers as well as the QoE measure of users.



(a) Exponential learning (b) Action-based Q-learning (c) Strategy-based Q-learning (d) Adaptive strategy



(e) Win-stay lose-shift strategy (f) Roth-Erev learning (g) Learning automata (h) Original MG algorithm

Figure 4.12: Users' QoE measure for different learning algorithms

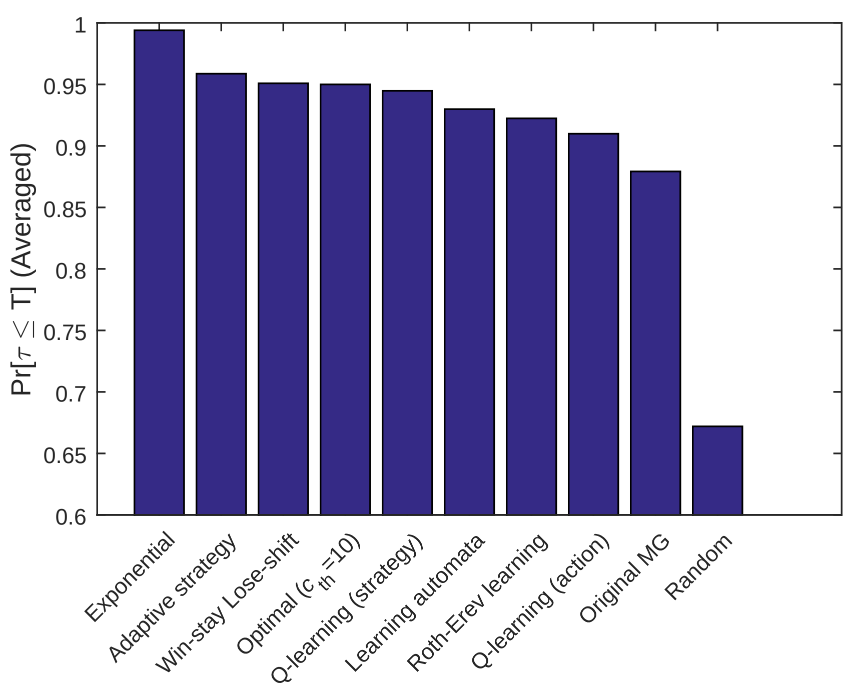


Figure 4.13: Performance comparison in terms of users' QoE measure.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

In this thesis, I have provided a brief tutorial on minority games and discussed the applicability of the minority game as a tool to mathematically model the distributed resource allocation problems in dense wireless networks. As its main contributions, this thesis provided distributed solutions for the problem of efficient edge server utilization problem while focusing on both users and servers perspectives. To this end, distributed algorithms have been developed using the minority game for two different scenarios: (i) multiple user offloading decision making problem and (ii) efficient edge server activation problem.

I have investigated the distributed mobile computation offloading problem in a small cell network with MEC capability. I have formulated the offloading decision making problem of users accessing a given SBS such that the computational resources of the SBS are utilized to its maximum capacity while the users' latency remain below a specific threshold. Thus, in this work, both users' and server's perspectives have been considered. I have formulated the problem using a minority game theoretic framework and provide a distributed offloading decision making mechanism that does not require any amount of communication among users. I have numerically investigated the performance of the proposed algorithm against the random offloading decision making mechanism.

I have formulated the efficient edge server activation problem for a MEC network considering a pool of edge servers. Therein, I have analyzed the statistical characteristics of the offloading delay. Based on this, I have modeled the computational offloading system as a planned market, where the price of computational services is determined by a central governor. Afterward, by using the theory of minority games, I have developed a novel approach for efficient mode selection at the servers' side. The proposed pricing framework in combination with the designed mode selection mechanism has been shown to guarantee a minimal server activation to ensure energy efficiency, while meeting the users' delay constraints with adjustable certainty. Thus, both users' and servers' requirements have been taken into account. The proposed mode selection scheme is distributed, and does not require any prior information at the servers' side. Moreover I have applied different reinforcement learning and stochastic learning methods to solve the formulated minority game and numerically investigate their performances to examine which methods provide the best performance in terms of both users' and servers' utility. Numerical results have been presented to illustrate the trade-off between the efficient server activation and meeting users' expected latency threshold.

5.2 Future Research Directions

5.2.1 Edge Server Resource Utilization in MEC

The edge server resource utilization problem is multi-faceted and it is a fundamental problem in the next generation of MEC networks. Related to the proposed models given in the Chapter 3 and Chapter 4 of this thesis, below we provide few possible extensions and improvements as future research directions.

- Apart from meeting the desired latency threshold for the users, minimizing users' offloading energy cost need to be addressed for different MEC offloading use cases.

- In MEC offloading, users typically have access to multiple computational resources such as, local device, SBS, MBS, D2D offloading or even the remote cloud [42]. To model such multi-option offloading decision making scenarios, multiple option MG (simplex game) [29] can be used.

- Though we consider homogeneous user devices, in practice, different types of devices are available with varying computational and battery capacities (e.g., smart phones, tablets, laptops); thus MG models that incorporate different user types can be employed to model such scenarios. Moreover, the offloading tasks can also be heterogeneous and also users may have the options of full and partial offloading. Hence, the system model can be extended to accommodate such requirements.

- Since MEC networks typically consist of a variety of edge nodes such as SBS, MBS, wireless access points, etc., the edge servers are not homogeneous in practice. Therefore, heterogeneities in their computational capability, power and storage should be taken into account when developing efficient resource allocation mechanisms. To model such scenarios, games that incorporate different types of players could be applied.

- In addition, to ensure fairness among the servers, analyses using various equilibrium notions need to be carried out.
- Moreover, mathematical tools such as queuing theory and Markov decision processes can be used to more accurately model the randomness in the offloading system such as random arrival of computation tasks and the users' status change.

5.2.2 Other Potential Research Areas for MEC

Below we point out some potential research directions for the area of MEC.

- Information-Centric MEC: Inspired by the concept of caching of popular files, in information-centric MEC, the data and/or services can be saved at different edge servers to promote an efficient computation. In fact, by using this concept, the amount of data which should be uploaded/downloaded dramatically reduces. Naturally, not all the data/services can be cached at every server. In addition, the service demand for users might change over time. Thus the problem to address is as follows: How much and which data/services shall be saved at each server? In addition, the servers should be motivated to cooperate with each other, so that if necessary, the tasks/data/services can be exchange among servers. Such problems can be addressed by using cooperative game theory, repeated auctions, and exchange economy.

- Economics of MEC Server Virtualization: Mobile network operators (MNOs) or service providers (SPs) may lease the MEC servers/resources from infrastructure providers (InPs). The InPs then will need to virtualize their MEC resources among different MNOs/SPs. The economics of the virtualization of MEC resources can be modeled and analyzed using game theory models. As an example, for a scenario with multiple InPs and multiple MNOs/SPs, a multi-leader and multi-follower Stackelberg game model can be formulated to determine the equilibrium prices that the MNOs/SPs need to pay to the InPs. In a more general scenario, virtualization of MEC resources/servers can be combined with virtualization of other resources including infrastructures (e.g., base stations), spectrum resources, as well as caching storage. Modeling and analysis of such a general virtualized network under users' quality of experience (QoE) constraints is an interesting research problem.

Bibliography

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[2] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct 2009.

[4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[5] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Jan 2016, pp. 1–8.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[7] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, April 2015.

[8] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, October 2016.

[9] H. Cao and J. Cai, "Distributed multi-user computation offloading for cloudlet based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[10] S. Josilo and G. Dan, "A game theoretic analysis of selfish mobile computation offloading," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.

[11] M. A. Messous, H. Sedjelmaci, N. Houari, and S. M. Senouci, "Computation offloading game for an uav network in mobile edge computing," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[12] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *2016 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–5.

[13] J. Zheng, Y. Cai, Y. Wu, and X. S. Shen, "Stochastic computation offloading game for mobile cloud computing," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, July 2016, pp. 1–6.

[14] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[15] C. Tekin and M. van der Schaar, "An experts learning approach to mobile service offloading," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sept 2014, pp. 643–650.

[16] S. O. Aliyu, F. Chen, Y. He, and H. Yang, "A game-theoretic based qos-aware capacity management for real-time edgeiot applications," in *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, July 2017, pp. 386–397.

[17] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, Sept 2017.

[18] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *2017 IEEE International Conference on Edge Computing (EDGE)*, June 2017, pp. 47–54.

[19] T. Y. He, N. Zhao, and H. Yin, "Integrated networking, caching and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[20] S. Ranadheera, S. Maghsudi, and E. Hossain, "Computation offloading and activation of mobile edge computing servers:A minority game," *CoRR*, vol. abs/1710.05499, 2017. [Online]. Available: http://arxiv.org/abs/1710.05499

[21] W. Kets, "The minority game: An economics perspective," Tilburg University, Center for Economic Research, Discussion Paper 2007-53, 2007.

[22] E. Moro, "The minority game: an introductory guide," in *Advances in Condensed Matter and Statistical Physics*, E. Korutcheva and R. Cuerno, Eds. New York: Nova Science Publishers, 2004, pp. 263–286.

[23] W. B. Arthur, "Inductive reasoning and bounded rationality," *American Economic Review*, vol. 84, no. 2, pp. 406–11, 1994.

[24] D. Challet and Y. C. Zhang, “Emergence of cooperation and organization in an evolutionary game,” *Physica A: Statistical Mechanics and its Applications*, vol. 246, no. 3, pp. 407–418, 1997.

[25] C. H. Yeung and Y. C. Zhang, “Minority games,” in *Encyclopedia of Complexity and Systems Science*, A. R. Meyers, Ed. New York, NY: Springer New York, 2009, pp. 5588–5604.

[26] N. Johnson, P. Hui, D. Zheng, and C. Tai, “Minority game with arbitrary cutoffs,” *Physica A: Statistical Mechanics and its Applications*, vol. 269, no. 24, pp. 493–502, 1999.

[27] M. Marsili, D. Challet, and R. Zecchina, “Exact solution of a modified El Farol’s bar problem: Efficiency and the role of market impact,” *Physica A: Statistical Mechanics and its Applications*, vol. 280, no. 34, pp. 522–553, 2000.

[28] D. Challet, M. Marsili, and Y. C. Zhang, *Minority Games: Interacting Agents in Financial Markets*. Oxford, UK: Oxford University Press, 2014.

[29] P. Mertikopoulos and A. L. Moustakas, “The simplex game: Can selfish users learn to operate efficiently in wireless networks?” in *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, ser. ValueTools ’07, 2007, pp. 7:1–7:10.

[30] K. Hamidouche, W. Saad, M. Debbah, J. B. Song, and C. S. Hong, “The 5g cellular backhaul management dilemma: To cache or to serve,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4866–4879, Aug 2017.

[31] Y. Saito, K. Yamamoto, H. Murata, and S. Yoshida, “Robust interference management to satisfy allowable outage probability using minority game,” in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2010, pp. 2314–2319.

[32] M. Elmachkour, I. Daha, E. Sabir, A. Kobbane, and J. Ben-Othman, "Green opportunistic access for cognitive radio networks: A minority game approach," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 5372–5377.

[33] M. Elmachkour, E. Sabir, A. Kobbane, J. Ben-Othman, and M. E. koutbi, "The greening of spectrum sensing: a minority game-based mechanism design," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 150–156, December 2014.

[34] P. Mähönen and M. Petrova, "Minority game for cognitive radios: Cooperating without cooperation," *Physical Communication*, vol. 1, no. 2, pp. 94–102, 2008.

[35] M. Michalopoulou, M. Petrova, M. Rappaport, and P. Mähönen, "Employing minority games in self-organizing wireless networks: Dynamic channel allocation," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sept 2016, pp. 665–671.

[36] P. Mertikopoulos and A. L. Moustakas, "Correlated anarchy in overlapping wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 7, pp. 1160–1169, September 2008.

[37] H. B. A. Sidi, W. Chahin, R. El-Azouzi, and F. D. Pellegrini, "Coordination minority games in delay tolerant networks," in *Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on*, May 2013, pp. 667–674.

[38] S. Ranadheera, S. Maghsudi, and E. Hossain, "Minority games with applications to distributed decision making and control in wireless networks," *IEEE Wireless Communications*, vol. PP, no. 99, pp. 2–10, 2017.

[39] D. Catteeuw and B. Manderick, *Heterogeneous Populations of Learning Agents in the Minority Game*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 100–113.

[40] K.-m. Lam and H.-f. Leung, "An adaptive strategy for minority games," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '07. New York, NY, USA: ACM, 2007, pp. 194:1–194:3.

[41] G. Reents, R. Metzler, and W. Kinzel, "A stochastic strategy for the minority game," *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 1, pp. 253 – 261, 2001, application of Physics in Economic Modelling.

[42] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, "Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, p. 974, 2016.