Task Offloading with Service Replication for Scalable Mobile Edge Computing

by

Shimaa Abdelmeguid

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
December 2021

Abstract

Mobile Edge Computing (MEC) is a network architecture model designed to provide the end-user with fast and powerful computing. MEC accomplishes this by enabling context-aware applications over cloud computing services within the radio access network (RAN) at the edge of the network thus, closer to the mobile subscribers resulting in better performance of delay-sensitive applications. However, the number of individuals and groups of users offloading independent and inter-related computational tasks to MEC servers is rapidly increasing, thus overloading the servers, consequently increasing the risk of service interruptions.

This dissertation investigates how reactive service replication can enable individuals and groups of users to access services from remote edge servers, thus guaranteeing system scalability. We propose a task offloading and service replication scheme to improve the scalability of edge services in MEC systems from various perspectives. In particular, we dynamically manage task offloading for groups of users requesting to perform inter-related computations on local and remote MEC servers managed by the same home service provider. Our proposed scheme minimizes the average response time of all users while satisfying the delay requirements of user groups in traffic-heavy and multimedia-intensive applications (e.g., group AV/AR, multimedia conferencing). Furthermore, we propose a task offloading and resource allocation scheme using reactive service replication between multiple service providers considering prior information of available resources and pricing for each service provider. Our goal is to minimize the average response time and the cost to the underloaded service providers while satisfying the delay requirements. We also design a cost management scheme based on offloaded tasks, computing resources, and foreign service providers' reputations to maximize the social welfare of the home service providers and the foreign service providers. Simulation results demonstrate the merits of our proposed scheme: Utilizing reactive service replication to improve the scalability of MEC system services.

Co-Authorship

Journal Articles

1. **Shimaa A. Mohamed**, Sameh Sorour, and Hossam S. Hassanein, "Group-Delay Aware Scalable Mobile Edge Computing Using Service Replication," IEEE Transactions on Vehicular Technology. (submitted)

2. **Shimaa A. Mohamed**, Sameh Sorour, and Hossam S. Hassanein, "Cost and Group-Delay-Aware Scalable Mobile Edge Computing Using Service Replication Between Multiple Service Providers," IEEE Transactions on Vehicular Technology. (under review)

3. **Shimaa A. Mohamed**, Sameh Sorour, and Hossam S. Hassanein, "A Stackelberg game to Scalable Mobile Edge Computing Using Service Replication," IEEE Transactions on Vehicular Technology. (under review)

Conference Publications

1. **Shimaa A. Mohamed**, Sameh Sorour, and Hossam S. Hassanein, "Group-Delay Aware Task Offloading with Service Replication for Scalable Mobile Edge Computing," in Proc. of IEEE Global Communications Conference (GLOBECOM), December 2020. DOI: 10.1109/GLOBECOM42002.2020.9348241

Acknowledgments

In the name of God, the Most Gracious, the Most Merciful. Praises and thanks to God, who is allowing me to continue the post-graduation studies that enabled me to learn and explore new opportunities, and for helping me every step of my life.

A special thanks to my co-supervisor, Prof. Hossam Hassanein. Your support since day one has led me to this place, and your continuous guidance made a big difference in my research. I have learned a lot from you in the past five years. My experience at Queen's University would not be the same without your mentorship. Thank you for all your valuable insights on creating new research ideas and demonstrating an exceptional level of leadership and understanding towards many students who have different personalities, potentials, and needs. I have great respect for you. I am proud to have you as a good example of supervision. I will try to reach the same level of professional excellence while maintaining that much decency and grace in treating my students as you treat yours. I wish you all the best and more success to come your way.

I want to express my appreciation towards my co-supervisor, Dr. Sameh Sorour. Dr. Sorour passed away this month. I cannot thank him enough for his constant help and support throughout this journey. He was always there for me, and for that, I am eternally grateful. The level of knowledge and expertise he shared will always be a valuable asset and a source of appreciation and admiration. I am thankful that I got to work with Dr. Sorour, and I will always keep you in my heart.

I am also thankful to my thesis committee: Prof. Juergen Dingel and Prof. Mohammad Zulkernine. Your feedback during the last year helped to polish this work.

Special thanks to Dr. Sharief Oteafy (an Examiner) for your comments and advice throughout my COMP exam preparations. I would also like to express my gratitude and appreciation towards Ms. Basia Palmer, who provided me with all the help and support I needed and was extremely kind. Many thanks to Ms. Debby Robertson for always being there to help make my transition to Queen's easier since day one.

To the Telecommunications Research Lab (TRL) members, it has been a pleasure to have worked with some of you. The weekly presentations were very informative and useful. Many thanks to all TRL teams for all presentations and discussions we had during the regular TRL meetings. Words cannot express my gratitude towards my TRL friends Dr. Sara Abdelhamid , Ms. Faria Khandaker, Ms. Mary Zarif, Miss. Basma Mohamed, Miss. Habiba Asharaf, Miss. Heba Ouda, and Ms. Rawan Khatteb for their enormous support throughout this journey and for all the fruitful discussions and memorable times. Many thanks for the help and support of Mr. Amir Ibrahim, Mr. Sherif Azmy, and Mr. Ahmed Nagib. Thanks, Mr. Ibrahim, for always being so thoughtful, gracious, and kind. I don't remember seeking your help with anything without you trying everything you possibly could to do it. Good luck on your new journey. Thank you, Mr. Azmy, for your support with analytical solutions. Thank you, Mr. Nagib, for your help throughout seeking access to the resources of the School of Computing.

Special thanks to Dr. Shady Khalifa for offering his help throughout applying to housing companies before I came to Kingston. I am forever grateful for your help Dr. Khalifa.

I want to thank my Masters' supervisors, Prof. Hosam Shalaby and Prof. Ahmed Abdelnabi. Your constant direction and influence led me to this place. It has been a pleasure to work with you and all my professors in the City of Scientific Research and Technological Application, Prof. Walaa Sheta, Dr. Hanan Hassan, Dr. Mohamed Azab, and Dr. Ashraf Taha.

I want to extend my eternal gratitude towards my beloved parents, sisters, and brothers. All the success in my life is because of you. There is no way I can pay back all the love, care, guidance, and continuous prayers throughout my whole life. Thank you for always being behind my back and pushing me to achieve more in my life.

To my husband Mr. Ahmed Eltaras, I would never have survived the last 5 years without you. Your sacrifice, love, and support made this work possible. My beloved daughter Jasmin and son Yahia, you brought all the happiness to our small family, May God bless you always. This work is dedicated to you. Thanks to my second family Mr. Youssef Eltaras, Mrs. Sadia, Mrs. Jehan, Mrs. Mona, Mrs. Sabrin, and Mr. Mohamed, for all the support.

I am thankful to my friends in Kingston who made a living away from home bearable. Thanks to Ms. Mary Zarif, Dr. Sara Abdelhamid, Ms. Faria Khandaker , Ms. Samiha Mohsen, and Ms. Aya Aziz. Thanks for always being there when I need you to my lifetime friends Ms. Aya Ibrahim and Ms. Lamia Amar. I wish you the best of luck in your lives.

Finally, I am very grateful for the funding provided by the Government of Egypt. Special Thanks to the Egyptian Cultural and Educational employees in Canada, Mr.

Mostafa and Mrs. Mary, for their help and support.

Statement of Originality

I hereby certify that all the work presented in this Ph.D. thesis is original, and all notions and/or techniques attributed to others have been fully acknowledged in accordance with the proper referencing practices.

Contents

List of Tables

List of Figures

List of Acronyms

AP	Access Point
AO	Alternating Optimization
AR/VR	Augmented Reality/Virtual Reality
CAPEX	Capital Expenses
CC	Cloud Computing
CS	Centralized Solution
COTS	Commercial-off-the-shelf
CPU	Central Processing Unit
DB	Distance-based
DS	Distributed Solution
EC	Edge Computing
EU	End User
5GPPP	European 5G Infrastructure Public Private Partnership
ETSI	European Telecommunication Standards Institute
5G	Fifth Generation
IaaS	Infrastructure as a Service
ILP	Integer Linear Programming
IoT	Internet of Things
IP	Internet Protocol
KKT	Karush-Kuhn-Tucker
Lex	Lexicographic
LP	Linear Programming
LR	Linear Relaxation-based
LTE	Long-term Evolution
MB	Market-based
MRB	Market-reputation-based

MD	Mobile Device
MDP	Markov Decision Process
MEC	Mobile Edge Computing
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-linear Linear Programming
NE	Nash Equilibrium
NFV	Network Functions Virtualization
OPEX	Operational Expenses
QCQP	Quadratic Constrained Quadratic Problem
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource-based
RSU	Road-side Unit
SaaS	Software as a Service
SDN	Software-defined Networking
SDR	Semi-definite Relaxation
SDS	Software-defined Systems
ST	Sequential Tuning
SLA	Service Level Agreements
3G	Third Generation
UE	User's Equipment
USRA	User Service Replication Allocation
VM	Virtual Machine
VNF	Virtual Network Function
WAN	Wide Area Network
WSum	Weighted-sum

List of Symbols

B_u	The size of the required information for User U_u task
B	Channel bandwidth
b_{ig}	Binary replication indicator of a service of Group G_g at foreign Service Provider S_i^f
BG	Vector of replication decision variables
C_u	Number of cycles required for User U_u task
C_{SP}	System cost
C_i^S	Service cost of foreign Service Provider S_i^f
C_j	Cost of home service provider S_j^h
$C_{j,norm}$	Normalized cost of home service provider S_j^h
C_{ji}^S	Service cost of home Service Provider S_j^h will pay to foreign Service Provider S_i^f
C_{ji}^R	Replication cost of home Service Provider S_j^h will pay to foreign Service Provider S_i^f
d_{G_g}	Maximum delay difference between any two users belonging to Group G_g
d_{ui}	Distance between User U_u and Service Provider S_i^p edge server
$F^h_{S_j^h}$	The available computational capacity of home Service Provider S_j^h cluster
$F^f_{S_i^f}$	The available computational capacity of foreign Service Provider S_i^f cluster
$F^o_{S_i^o}$	The available computational capacity of each overloaded edge server S_i^o
$F^u_{S_i^u}$	The available computational capacity of each underloaded edge server S_i^u
$F^{min}_{G_g}$	Minimum computational capacity to process tasks of group G_g
f_u	The computational capacity assigned to the task of User U_u
$\mathcal{G}$	Set of all users' groups

$\mathcal{G}^j$	Set of users' groups of home service provider S_j^h
G_g	Group ID
$\mathcal{U}^{G_g}$	Set of users of Group G_g
H_{ui}	Channel power gain between User U_u and Service Provider S_i^p edge server
M_h	Number of home service providers
M_f	Number of foreign service providers
M_o	Number of overloaded edge servers
M_u	Number of underloaded edge servers
N	Number of users
N_j	Number of users who have contracted with home Service Provider or S_j^f
N_g	Number of users of Group G_g
P_u	User U_u transmission power
p_i	Probability of foreign Service Provider S_i^f serving incoming requests using foreign Service Provider S_i^f reputation
P	Reputation vector
R_{ui}	Data rate of User U_u at the Service Provider S_i^P edge server
RT_{ui}	Response time of User U_u at the Service Provider S_i^p edge server
RT_{uz}	Response time of User U_u at home Service Provider S_z^h edge server
RT	Average response time
RT_{Min}	Minimum response time in the system
RT_{Max}	Maximum response time in the system
$RT_{ui_{norm}}$	Normalized average response time of User U_u at the Service Provider S_i^p edge server
$RT_{j,norm}$	Normalized average response time of home service provider j
$RT_{uz_{norm}}$	Normalized response time of User U_u at home Service Provider S_z^h edge server
$\mathcal{S}^h$	Set of home service providers
$\mathcal{S}^f$	Set of foreign service providers
$\mathcal{S}^p$	Set of service providers to which users can offload tasks
$\mathcal{S}^o$	Set of overloaded edge servers

$\mathcal{S}^u$	Set of underloaded edge servers
$\mathcal{S}$	Set of edge servers to which users can offload tasks
S_j^o	Overloaded edge server with index j
S_i^u	Underloaded edge server with index i
S_i	Edge server with index i
S_j^h	Home service provider with index j
S_i^f	Foreign service provider with index i
S_i^p	Service Provider with index i
$T_{G_g}^{max}$	Maximum allowable delay all users in Group G_g
$\mathcal{T}_u$	User's task
T_{ui}^T	Transmission time of Task $\mathcal{T}_u$ of User U_u to Service Provider S_i^p edge server
T_{ui}^C	Computation time of Task $\mathcal{T}_u$ of User U_u to Service Provider S_i^p edge server
T_u^R	Replication time of Task $\mathcal{T}_u$ of User U_u to Service Provider S_i^p edge server
T^{RM}	Roaming time
$\mathcal{U}$	Set of all users who have contracts with home service providers
$\mathcal{U}^j$	Set of user devices attached to their local overloaded edge server S_j^o (Ch. 4)
$\mathcal{U}^j$	Set of user devices that have contracts with home service provider S_j^h (Ch. 5, 6)
U_u^j	User with index u with home service provider S_j^h
$U_u^{G_g}$	User U_u of Group G_g
UC_i^C	Unit cost of the computation rate of foreign Service Provider S_i^f
UC_i^T	Unit cost of the transmission rate of foreign Service Provider S_i^f
UC_g^R	Unit cost of replicating the service of Group G_g
$U_{home,j}$	Utility function of home Service Provider S_j^h
$U_{foreign,i}$	Utility function of foreign Service Provider S_i^f
x_{ui}	Binary task of loading indicator of User U_u at edge Server S_i (Ch. 4)
x_{ui}	Binary task of loading indicator of User U_u at Service Provider S_i^p (Ch. 5, 6)
X	Vector of task offloading decision variables
y_i	Unit cost of the computation rate

Y	Vector of Unit Prices
$Y_{i,min}$	Minimum price of foreign Service Provider S_i^f
$Y_{i,max}$	Maximum price of foreign Service Provider S_i^f
$\mathcal{Y}_i$	Set of all prices satisfying constraints of foreign Service Provider S_i^f
z	Home Service Provider $S_z^h, \ \forall \ S_z^h \in \mathcal{S}^h$
σ_i	Noise power spectral density to the Service Provider S_i^p edge server
α	Path loss exponent
Δ	Expected delay
δ_i	Reduction factor of foreign Service Provider S_i^f
γ_i	Step parameter of foreign Service Provider S_i^f
ϵ	Accuracy parameter

List of Algorithms

Chapter 1

Introduction

Edge computing (EC) is a computing paradigm that delivers computation and storage capacity from the core Wide Area Network (WAN) to the network's edge. Newer edge services have real-time requirements or interactive behavior with high Quality of Service (QoS) expectations. However, remote cloud computing (CC) resources might not support delay-sensitive applications because the WAN, used to access these cloud services, has latency constraints. Pushing these services to the network's edge becomes the optimal solution [1]. EC and CC are complementary paradigms that perform best when employed together. EC has two key advantages: it can overcome the mobile device's computational capacity limitation by providing computing resources closer to the user. Second, it avoids the high amount of latency caused by executing tasks on the remote CC resources.

In the literature, EC has several variants, including Cloudlets [2], Fog Computing [3], and Mobile Edge Computing (MEC) [1], each defining different practical implementations for EC. For example, fog computing enables new applications and services directly at routers, switches, access points (APs), and gateways such as future Internet of Things (IoT) applications. A cloudlet is a small-scale cloud infrastructure placed within close proximity to user devices in settings like coffee shops and company buildings. Our research focuses on the Mobile Edge Computing (MEC) EC model [4]. MEC is one of the most viable 5G EC implementations, which reduces processing delays and the energy consumed by the user's equipment (UE) [5]. MEC provides offloading computational tasks, data storage, caching and processes data more quickly [6]. In addition, MEC distributes request and delivery services from the remote cloud to the point on the edge closest to the UE; this reduces the network latency and bandwidth consumption of the UE. Also, MEC is more suitable for applications such as dynamic content optimization, computation offloading in IoT, mobile big data analysis, and smart transportation because these are computationally intensive and have substantial storage capacity requirements. However, MEC faces many challenges, including scalability, offloading granularity, impact on latency, and pricing models [4, 7].

1.1 Motivation

MEC's primary purpose is reducing latency by offloading the computations and data storage to edge servers [4]. Despite their significant potential to improve the performance of delay-sensitive applications, edge servers have limited computation resources compared to cloud servers. This limitation may cause edge servers to become overloaded with computations to such an extent that this may cause service interruptions for a subset of its subscribed users [4].

Looking ahead, as the number of users who offload independent and inter-related computational tasks to MEC servers increases, the risk of overloading, service interruptions, and Quality of Experience (QoE) degradation increases. This scenario is applicable during high-density user demand, such as viewing live sports events and concerts. Furthermore, as a business model, edge computing service providers aim to improve their profit while satisfying the Quality of Service (QoS) requirements. This profit involves setting good prices and not losing a good reputation by having too many blocked or long delays for incoming requests. Maximizing the profit of the edge computing service providers in highly competitive edge computing markets, consisting of more service providers, is a big challenge for several reasons. The main reason is that all service providers can selfishly work independently in a distributed system; no centralized authority exists, making their own decisions. Therefore, the primary focus of an edge computing service provider is to sell its services to as many clients as possible and increase the profit margin.

1.2 Research Statement and Thesis Contributions

The main objective of our research is to improve MEC services' scalability in response to excessive task offloading on MEC servers. We evaluate the potential advantages of applying an MEC service replication technique to an overloaded server with too many tasks. We aim to increase the MEC system's scalability while minimizing the delay requirements of multiple user groups, including individuals involved in the same traffic-heavy or multimedia-intense applications.

We assume that several service providers manage these MEC servers. Some of these service providers are referred to as home service providers; their users are clients. All other providers are considered foreign service providers. While each home service provider may manage one or more MEC servers at various locations, the nearest server to any given user is the user's local edge server. All other servers belonging to the same home service provider are considered remote edge servers.

In this thesis, we answer the following research questions:

1. From the user's perspective, how can the scalability of local edge services be improved in the case of massive offloading requests?

2. From the service provider's perspective, and with cost-aware knowledge, what is the best task offloading distribution to jointly minimize the average response time of all users and home service providers' charged fees?

3. What is the best price, while maintaining a good reputation of the foreign service provider, to maximize the social welfare between all service providers?

To answer these research questions: First, we propose a task offloading scheme involving service replication on local and remote MEC servers from the user's perspective, assuming that one home service provider manages all the MEC edge servers. Second, we dynamically manage task offloading and replication allocation for groups of users requesting to perform inter-related computations from home service providers, and foreign service providers considering cost information. Lastly, we propose a decentralized and dynamic task offloading and service pricing management scheme between the home service providers and foreign service providers based on the task offloading loads, availability of computing resources at the distant edge servers, and the reputation/capability of these distant servers in providing an acceptable level of QoE to users of home service providers in our open edge computing market.

The contributions of this thesis are summarized as follows:

1. To address the first research question, we dynamically manage task offloading

for groups of users requesting to perform inter-related computations on multiple overloaded and underloaded MEC servers. We formulate an integer linear problem that minimizes the average response time of all users while satisfying the time and time difference constraints of the user groups running the same applications. Next, we deal with the linear programming (LP) relaxation of our problem, which is used to derive the optimal decisions using Lagrangian analysis. We solve the proposed LP problem numerically and apply a proposed rounding algorithm to reach the integer values of the decision variables.

2. To address the second research question, we formulate a constrained multi-objective optimization problem that minimizes the average response time of all users and the charged fee to foreign service providers while satisfying the time and time difference constraints and requirements of user groups running the same applications. We use the Lexicographic method and Weight-sum method to solve our constrained multi-objective problem in the centralized allocation. Using the LP relaxation problem, we derive the optimal decision variables using Lagrangian analysis. In addition, we formulate a User Service Replication Allocation (USRA) game to solve our problem in a distributed manner. Our objective is to minimize the weight-sum of users' average response time and the charged cost to the foreign service providers from the home service provider's perspective and propose the distributed allocation algorithm for finding the Nash equilibrium in the USRA game.

3. To address the third research question, we formulate the task offloading and pricing problem as a Stackelberg model between home service providers and foreign service providers. We consider the reputation of the foreign service

providers by determining the best offloading decisions and replication indicators from each home service provider's perspective. We deal with the linear programming (LP) relaxation for the proposed problem of the home service providers; then, we apply a proposed rounding algorithm to reach the integer values of the decision variables. Lastly, we develop an algorithm to determine the Stackelberg equilibrium.

1.3 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 presents the background and overview. We begin by presenting an overview of MEC, including its definition, architecture, characteristics, enabling technologies, and applications. We also review existing solutions toward resource management in MEC. In addition, we present MEC architecture with economic consideration, including profit components, optimization factors, and existing economic and pricing models for resource management in cloud networking. Lastly, open research challenges are discussed.

In Chapter 3, we introduce our group-based scalable MEC architecture. First, we describe our system's components, system parameters, including the response time and the system cost, and performance metrics used in this thesis.

Chapter 4 introduces our proposed solution to achieve scalable MEC services from the user's perspective. We formulate the problem as an integer linear problem and use the linear relaxation programming to analyze using Lagrangian analysis and solve the problem using a numerical solver. In addition, we compare the optimized solution of the integer problem with distance-based and resource-based greedy approaches.

Chapter 5 introduces the proposed solution to achieve scalable MEC services from the service provider's perspective. In centralized allocation, we formulate the problem as a constrained multi-objective optimization problem and use linear relaxation programming to analyze using Lagrangian analysis and solve it using a numerical solver. In addition, we compare the optimized solution of the integer problem with the optimal solutions of the distance and resource-based approaches. Moreover, we formulate a user and service replication allocation (USRA) game in distributed allocation, a game-theoretic strategy that formulates the USRA problem as a potential game. We analyze the game and show that it admits a Nash equilibrium. We designed a distributed algorithm for finding a Nash equilibrium in the game to solve the USRA problem.

In Chapter 6, we propose a market-reputation-based (MRB) approach and a market-based (MB) approach using the Stackelberg economic model between home service providers and foreign service providers. On the one hand, a home service provider's goal is to improve the scalability of real-time offered services by replicating them on demand at underloaded foreign service providers edge servers in the MEC system so all users are served concurrently when the home service providers cannot sustain the load from all the user groups. On the other hand, a foreign service provider's revenue stems from setting profitable prices and maintaining a good reputation by not having incoming requests blocked or delayed in the MRB approach. However, in the MB approach, we do not consider foreign service providers' reputation. We prove the existence of Stackelberg equilibrium and develop an algorithm to reach the Stackelberg equilibrium.

Lastly, we highlight the conclusions of our work and outline potential future directions in Chapter 7.

Chapter 2

Background and Overview

Nowadays, mobile devices such as, smartphones, tablets, laptops, etc., are an essential tool for services such as education, entertainment, and social networking. However, due to the resource limitations of these mobile devices, (processing power, battery lifetime, and storage capacity), mobile users do not get the same quality of experience (QoE) as desktop users. Many proposed techniques to overcome these limitations, such as computational offloading, offloading some or all computations to remote cloud computing servers, aimed at saving processing power and energy. However, offloading computations to cloud servers may cause high latency for data exchange between the cloud and mobile devices. Mobile edge computing (MEC) enables mobile subscribers to access the cloud-like computing services at the network edge within their radio access network domain. MEC's primary objective is reducing the latency by supporting the computation and storage requirements of mobile subscribers at the edge of the mobile network [4].

The rest of this chapter is organized as follows; Section 2.1 presents a brief MEC overview, including MEC architecture, MEC objectives, MEC characteristics, MEC enabling technologies NFV and SDN, and MEC applications. Section 2.2 describes the resource management in MEC and summarizes the existing solutions, in addition to a brief introduction to service migration and service replication in MEC. Section 2.3 presents the MEC architecture with economic consideration, showing essential factors toward optimizing service providers' profits, states existing economic pricing techniques, and presents existing literature on economic and pricing models for resource management in cloud data center networking and edge computing. Section 2.4 states the main challenges and open issues. The summary of this chapter is presented in Section 2.5.

2.1 MEC Overview

The European Telecommunication Standards Institute (ETSI) defines the MEC as follows: "Mobile edge computing provides an IT service environment and cloud computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and close to mobile subscribers to serve delay-sensitive context-aware applications," [8]. Transferring the computation capabilities and storage capacity from the network's core to the network's edge reduces latency [4]. In addition, MEC can provide location awareness, a feature which many mobile device applications require. MEC servers typically are placed near base stations closest to the end mobile subscribers to access computing resources within the RAN [9, 1]. There are three components in the MEC system's architecture, edge devices, edge servers, and cloud servers, as shown in Figure 2.1. Edge devices include all devices connected to the network (e.g., desktop personal computers, laptops, and smartphones), and edge servers or MEC servers, these are usually placed at LTE or 5G base stations (eNodeB/gNodeB) or a multi-technology (3G/LTE/5G) cell aggregation site. They typically have fewer resources compared to the centralized cloud. The third component, the cloud servers are a centralized cloud infrastructure hosted somewhere across the network core and connected to edge networks via the Internet [10]. The communication between the edge devices and MEC servers is typically achieved via LTE, 5G, or other wireless communication protocols. A single MEC platform can serve several base stations located at a central aggregation point within the RAN. This approach reduces both capital expenses (CAPEX) and operational expenses (OPEX) [10].

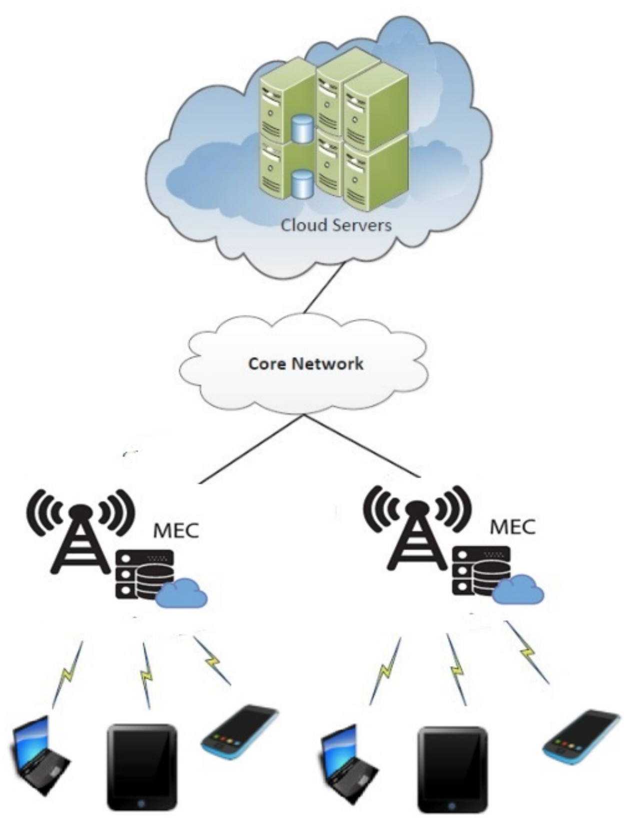

Figure 2.1: Mobile edge computing architecture

From an operational point of view, each edge device and network operator have different purposes. On the one hand, the edge device tries to save on its processing resources and energy consumption by offloading computation and storage to the MEC infrastructure. On the other hand, the network operator tries to minimize infrastructure costs and fulfill high throughput.

2.1.1 MEC Objectives

According to Ahmed and Ahmed in [4], the primary objectives of MEC are to:

1. Reduce the computing, storage, and energy demands on mobile devices by offloading compute-intensive and high bandwidth-demanding applications to the mobile network's edge server [9].

2. Reduce the disadvantages of CC by enabling cloud services and resources to be in closer proximity to the user. CC requires a high-speed network, its performance in terms of latency does not guarantee security and privacy, which require constant monitoring and enforcement of service level agreements (SLAs) [11].

3. Provide context-aware services with assistance from the collected RAN information.

2.1.2 MEC Characteristics

The main characteristics of MEC are as follows [4]:

- Proximity: MEC servers are hosted closer to mobile subscribers within the RAN range. RAN and MEC servers collect users' information to analyze user behavior to improve service.

- Low latency: MEC's primary goal is to reduce the latency of delay-sensitive applications by hosting these applications at edge servers.
- Location awareness: Base stations collect users' information from edge devices to locate the device and to predict the future network status. Based on this information, edge service providers deliver context-aware services to the users.
- Network context information: Using real-time RAN information (network load, radio condition, device location) helps to make future decisions for improving user satisfaction and QoE.

2.1.3 Enabling Technologies NFV and SDN

According to Hu et al., in [8], the European 5G Infrastructure Public Private Partnership (5GPPP) recognized MEC as one of the key emerging technologies for 5G networks together with Network Functions Virtualization (NFV) and Software-defined Networking (SDN). NFV is a technology that separates network functions, such as firewalls or load balancers, from the hardware and hands them over to a software-based application called Virtual Network Functions (VNFs) [12, 13]. Virtualization is an approach to decouple the software networking processing and applications from their supported hardware and implement the network services as software [14, 15, 16]. Virtualization has been developed to reduce CAPEX and OPEX and prevent the network ossification problem, which prevents the operation of service additions and network upgrades. ETSI Industry Specification Group proposed NFV decoupling network functions from the underlying hardware appliances and transferring them to software running on commercial-off-the-shelf (COTS) equipment (i.e., virtual machines). NFV provides flexible provisioning of software-based network functionalities on top of an optimally shared physical infrastructure, which reduces middleboxes deployed in the traditional networks to take advantage of cost savings [17, 12, 18]. Although SDN is independent of virtualization, it is complementary to it. SDN considers a subsystem of software-defined systems (SDS). SDN differs from traditional Internet Protocol (IP) networking, where packet forwarding depends on flows instead of packets. SDN's primary goal is to provide open user-controlled management of the forwarding hardware of a network element [19]. Utilizing an open user-controlled management architecture allows the network operator to configure the network hardware directly from the logically centralized controller and flexible the network. Therefore, network switches become simple forwarding devices, making management and operation smoother and less expensive [20, 21].

In [1], Baktir et al., proposed a clear picture of the EC domain to categorize and discuss some challenges and provide more insight into the complexities involved in the practical implementations of edge computing. Several areas benefited from utilizing SDN with edge computing technology: higher resolution and more effective control, greater flexibility enabling innovation, adaptability, interoperability with the OpenFlow protocol, producing lower-cost solutions, virtual machine mobility at the edge, and multiplicity of scope.

2.1.4 MEC Applications

In the section we present examples of use cases suggested by ETSI for the users and operators of MEC [22].

User-Oriented Use Cases

- ***Application computation offloading:*** The mobile edge server can process computationally intensive tasks in place of mobile devices (e.g., video analysis, sensor data processing, and language translation). Offloading to MEC servers is valuable to the IoT environment. Mobile devices consist of small sensors and micro-scale processors with a limited energy supply. With the option to offload heavy computation to nearby mobile edge servers for accelerated processing, the computational capability of the mobile device and energy consumption will no longer be the bottleneck for delivering rich applications.

- ***Gaming, virtual reality, and augmented reality:*** These applications require low latency and exceptionally large processing capability. On the one hand, mobile devices have limited computational capacities. On the other hand, offloading these tasks to a remote cloud server might incur too much latency. Therefore, users can offload part of the computational load to some mobile edge application running on mobile edge servers. MEC provides an appropriate balance between computational power and proximity to the end user. During a local event with a large audience, such as a sports game or a concert, numerous mobile devices concurrently access videos of the event. These videos are often rich in content and include multiple streams because they may consist of real-time and on-demand versions from various camera angles. However, they are locally generated. Instead of these videos being sent back and forward through the mobile core and the Internet to and from a video content server, they can be processed and delivered by the mobile edge servers.

- ***Vehicle-to-infrastructure communication:*** Vehicle-to-infrastructure communication services, such as those between cars and roadside units, are important to the safety and efficiency of public transportation systems. MEC enables the application of 5G to serve such communication functionalities, since it can more readily provide global coverage than can dedicate short-range communications. Applications that provide these functionalities can be loaded into mobile edge servers installed within the RAN along the roadway. Hazard warnings and other latency-sensitive messages can reach targeted vehicles within stringent timescale requirements due to the mobile edge server's proximity to the vehicles and roadside units.

Operator-Oriented Use Cases

- ***Local content caching at the mobile edge:*** The ubiquity of widely shared large files on social media, such as high-definition videos, stresses the backhaul network capacity. However, this content tends to be consumed by many users in the same geographical area and time zone. Video content can be cached locally at the mobile edge servers to reduce the traffic demand on the backhaul significantly. Moreover, the service quality of the mobile edge application in which the cached content proactively moves between mobile edge servers in awareness of the user's movement can improve. Similar benefits extend to broadcast videos, such as mobile TV, where many mobile users view the same video stream near a mobile edge server.

- ***Data aggregation and analytics:*** Operators provide many services, such as security monitoring and massive sensor information processing. They collect a large amount of data from sensors and mobile devices. Then, they send all of these raw data to the back-end servers. Therefore, a distributed mobile edge application running on multiple mobile edge servers can process this data to extract the required metadata before forwarding the metadata to the back-end servers.

- ***Location-based services:*** MEC provides effective location-based services because it allows user location tracking using advanced analytical techniques beyond measuring the received signal strength. In addition, mobile edge applications can provide more appropriate location-based service recommendations. For example, a mobile edge application can consider the context of the user's location, such as a shopping mall or museum, and the user's behavior pattern to make recommendations. It may also utilize advanced machine learning techniques and interface with big data analysis at back-end servers to further improve the accuracy and usefulness of its recommendations.

2.2 Resource Management in MEC

In general, resource management plays an important role in realizing energy-efficient and low latency in the MEC system. In this thesis, we focus on computational resources.

In MEC, a user decides whether to offload the computation task or not. Two types of computational offloading are proposed in the literature [23]:

- Binary offloading: a user chooses if entire computation is processed either locally by the UE (local execution) or is offloaded and processed by the MEC server (full offloading).

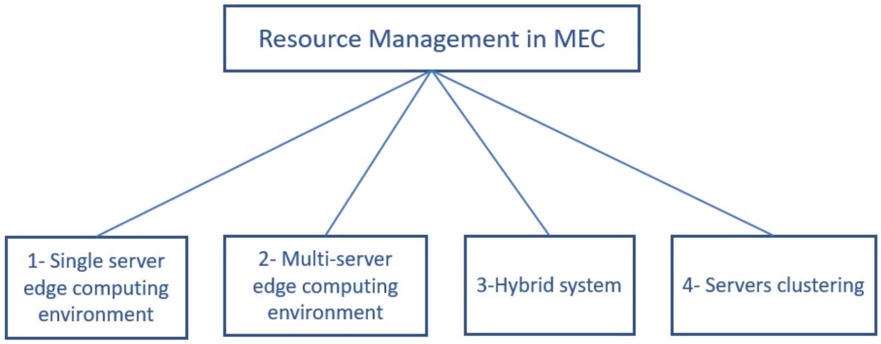

Figure 2.2: Categories of existing solutions based on MEC environment

- Partial offloading: A computation task is divided into sub-tasks and a part of these sub-tasks are processed locally while the remainder is offloaded to the MEC server.

In the literature, several works investigated task offloading and resource management for achieving system-level objectives, for example, efficient resource utilization, minimum latency, and total user device energy consumption. We can classify these solutions into four categories based on the following four MEC environments as shown in Figure 2.2: (1) Single server edge computing environment, (2) Multi-server edge computing environment, (3) Hybrid system such as edge-cloud system, (4) Server clustering.

2.2.1 Existing Solutions

Despite their significant potential to improve the performance of delay-sensitive applications, edge servers have comparatively limited computation resources than cloud servers. In typical scenarios, users' devices do not access an edge service simultaneously. In recent years, the number of edge devices has significantly grown, and massive numbers of devices may access several edge services simultaneously, causing network congestion and server overloads. The service may interrupt, for example, high-density demand scenarios such as sports events and live concerts.

Resource Management in MEC Single Server Edge Computing Environment

One way to mitigate overloading and prevent service interruption is to resort to partial task offloading and resource allocation strategies, extensively investigated in a single server edge computing environment. In [24], Liu et al., used a Markov Decision Process to solve a proposed stochastic optimization problem, which minimized the latency for one user with multiple tasks connected to one MEC server. In [25], Mao et al., studied task offloading, scheduling, and transmit power for the MEC system with numerous independent tasks by proposing a joint optimization problem that minimized the weighted sum of the latency and device energy consumption.

Resource Management in MEC Multi-Server Edge Computing Environment

Other works considered a multi-server MEC system. In [26], Liu et al., investigated the power-delay trade-off in a multi-user MEC system by optimizing the transmission power and allocated user tasks between local and on-server computations. In [27], Ranadheera et al., proposed an energy-efficient mechanism for computation offloading to MEC servers by activating a subset of servers and satisfying users' latency requirements. In [28], Dinh et al., proposed an optimization problem on offloading from a single mobile device to multiple access points, besides local computation on the mobile devices. Their goal is to minimize the mobile device's total latency and energy consumption while jointly optimizing the mobile device's task allocation decisions and frequency scaling. Two cases for the mobile device were considered, fixed computing capacity and elastic computing capacity. They studied how the computing capacity range can influence the task allocation decision.

Another line of research in the literature aims to jointly optimize task allocation and resource scheduling to reduce MEC service cost while assuming that the edge servers' capacity will always satisfy the users' demands. In the works that follow, the service cost is calculated based on the latency and the consumed power by the users' tasks at the edge server. In [29], Li et al., proposed a mixed-integer non-convex optimization problem to maximize the number of offloaded tasks and minimize the energy consumption of all the user devices and edge servers by selecting the best edge servers with capacity for offloading. In [30], Long et al., investigated the computation offloading and communication and computation resource allocation scheme for the MEC system. They designed a multi-objective computation offloading resource allocation algorithm, which optimized the uplink and downlink spectrum resources, computation resources, and offloading strategy. In [31], Zhang et al., aim to minimize the cost of an edge computing system consisting of multiple collocated edge servers by developing a task scheduling model and dynamic resource management problem for the delay-sensitive input tasks. They considered each job as an item and each server as a bin; the problem's goal is to pack all items into bins using the minimum number of bins. Given delay requirements, they proposed a two-stage task scheduling cost optimization algorithm to solve this cost optimization problem. In [32], Gu et al., jointly investigated base station association, task distribution, and virtual machine placement toward cost-efficient fog computing integrated with medical cyber-physical systems satisfying Quality of Service (QoS) requirements. In [33], Wang et al., proposed a data transfer and computation in monetary and time-cost minimization problem for several users over multiple heterogeneous cloudlets. In [34], Wang et al., minimized the consumption bandwidth and latency cost of using multi-edge servers while satisfying the delay constraint. They formulated the problem as an integer programming problem and proposed an optimal two-phase optimization algorithm to solve it in two phases.

Most of the above works did not consider reducing MEC service costs with the notion of scalability. In MEC, scalability guarantees the availability of MEC services regardless of the number of user devices and capacities of the local or closest edge servers. With the increasing numbers of edge devices and the complexity of applications that run on them offloading would typically impose a massive load on these local edge servers from these devices, resulting in network bottlenecks and service interruptions [4]. In [35], Du et al., proposed a cost-effective solution to serve many service clients by a multi-server MEC system. They presented a stochastic optimization problem with various objectives, aiming to maximize service capacity (defined by the number of served mobile devices) and minimize the service cost in the long term by measuring the service latency and power consumption experienced by users at the edge server. A joint online task offloading, and resource allocation algorithm is proposed based on Lyapunov optimization, which converts the stochastic optimization problem into a per time-slot deterministic optimization as a solution to maximize service capacity. However, this work assumed that some users could be left unserved due to a lack of resources.

Resource Management in MEC Hybrid System

In [36], Ma et al., proposed a cloud-assisted mobile edge computing solution to balance the load between the cloud and MEC servers and minimize the total users' delay, the MEC cost, and cloud computation. In [37], Chen et al., formulate a joint optimization problem of the user's device energy, latency, and edge server's utility cost to make the optimal decision between local computing on a mobile device against offloading computations to cloud servers. The work also optimized the allocation of communication resources. According to Chen et al., in [38], the work in [37] was improved by considering a third offloading layer, the cloudlet at the access point, between the local and cloud computing layers. In [39], Maheshwari et al., analyzed a hybrid edge-cloud system's scalability and capacity performance designed to support latency-sensitive applications, namely an augmented reality application with service time constraints. Their results presented a guide for striking the right balance between edge and cloud resources in such scenarios. In [40], Chen et al., proposed a task offloading decision scheme between a computing access point (local edge) and central cloud computation, which solves a joint optimization problem, minimizing the overall energy computation. They also minimized delay among all users using a game-theoretic approach.

On the other side of the spectrum, Yaqub et al., in [41], proposed distributing offloaded tasks between edge servers and peer user devices. The purpose of this work is to determine the best load partitioning between edge servers and peer devices to minimize the excess delay and energy consumption compared to the no-offloading alternative. In [42], Tong et al., proposed a multi-tier edge architecture, with higher tiers containing more powerful edge cloudlets than lower tiers. These higher tiers can receive migration requests from the lower tier when overloaded. Furthermore, a workload placement algorithm was proposed to execute mobile programs in the edge servers more efficiently.

Resource Management in MEC Server Clustering

Several researchers have suggested that load balancing mechanisms and server clustering can ensure scalability within the edge servers' layer. In [43], Liu et al., proposed a first-come-first-serve queuing and service model for scenarios with multiple MEC servers to improve the QoS of edge servers by clustering them together for load-sharing. When the buffer is full, the system processes the requests according to three load-sharing schemes: no sharing, random sharing, or least loaded sharing. In [44], Beraldi et al., proposed a cooperative load balancing scheme to deal with the resource limitations of the edge servers and minimize blocking states at each edge server. Each edge server has a buffer to store service requests for future implementation. The upcoming demands migrate to only one neighboring edge server in a full buffer case, which accepts them if their queue length is less than a specific threshold. In [45], Ndikumana et al., proposed a collaborative data caching and computation offloading among collaborating MEC servers. The system allocates the caching and computation resources to multiple service requesters based on their demands and payments. This work aims to maximize resource utilization from the MEC server. Table 2.1 summarizes selected related works regarding existing resource management solutions for achieving low latency and energy consumption. The table summarizes the system architecture, the goal, and the solution technique.

Table 2.1: Existing solutions (♣ Single server edge computing environment, ♠ Multi-server edge computing environment, ♡ Hybrid system, ♢ Servers clustering)

Ref.	Goal	Solution technique
[24] ♣	Minimize delay	Markov decision process
[25] ♣	Minimize execution delay, and device energy consumption	Optimization techniques (MINLP), and exhaustive search
[40] ♡	Minimize energy consumption, computation, and maximum delay among all users	Game theory
[26] ♠	Minimize computation and transmission power	Lyapunov stochastic optimization
[27] ♠	Minimize delay and energy consumption	The theory of minority games
[28] ♠	Minimize latency and energy consumption	Semi-definite relaxation (SDR)
[30] ♠	Optimize uplink and downlink spectrum resources	Game theory
[31] ♠	System cost	Heuristic algorithm (greedy best first search-BFS)
[32] ♠	System cost	Optimization techniques (MINLP) and LP-based two-phase heuristic algorithm
[33] ♠	Minimize system cost and delay	Heuristic algorithm
[36] ♡	Load balancing, minimize users' delay, and system cost	Analytic solution using KKT conditions
[34] ♠	Minimize energy consumption, bandwidth, and latency	Optimization techniques (ILP) and heuristic algorithm (two-phase optimization algorithm)

[37] ♠	Minimize energy computation, and users' delay	Optimization techniques (QCQP) and heuristic algorithm using SDR
[38] ♡	Minimize energy computation, and users' delay	Heuristic algorithm (SDR-AO-ST)
[35] ♠	Maximize service capacity, and minimize service cost	Lyapunov optimization technique
[39] ♡	Load balancing	Heuristic algorithm
[41] ♢	Load partitioning, minimize delay and, energy consumption	Optimization techniques and priority-based heuristic algorithm
[42] ♡	Efficient resource utilization	Optimization techniques (MILP) and heuristic algorithm (exhaustively search)
[43] ♠	Load balancing, minimize blocking probability, and user's waiting time	Three load sharing schemes: no sharing, random sharing, and least loaded sharing
[44] ♠	Load balancing, minimize service blocking probability, and service delay	Markov process
[45] ♢	Maximize resource utilization	Optimization techniques (ILP) and heuristic algorithm

2.2.2 Service Migration and Service Replication

Service Migration

Service migration in MEC is not new; however, it was combined with user mobility and the limited coverage of edge servers. The contradiction between the limited range of the single edge server and the unlimited mobility of users will result in significant network performance degradation, which can further lead to a drop in

QoS and interruption of ongoing edge services. Therefore, it is not easy to ensure service continuity. Thus, considering geographical proximity and QoS, the need to migrate a MEC service over MEC servers always to connect users to the optimal server [46, 47].

Taleb and Ksentini in [48] presented the Follow-Me Cloud concept and introduced its analytical model. They proposed a seamless migration of ongoing service between a data center and another optimal data center using the Markov Decision Process (MDP). MDP includes one dimensional MDP [49], where mobile users move in a straight line, and a two-dimensional MDP model [50] in which mobile users move randomly in an area, this is generally the case. Therefore, service mobility is enabled among data centers following the mobility of the mobile user.

Service Replication

The service migration may be costly for the operator as it consumes bandwidth to migrate services. In addition, it may introduce service disruption due to the downtime of the service during the migration. Thus, the low latency services may be interrupted. Therefore, service replication is proposed to avoid service migration and ensure low latency access. Service replication may prove very efficient, particularly if the service is very popular and shared by many users.

In [51], Frangoudis and Ksentini addressed such limitations, and they presented an analytical model based on Markov chains to capture the relation between the cost of service replication and the cost of service migration in terms of the service duration, size of the area where the service is replicated, the number of users, and the user mobility.

On the one hand, static mechanisms create a fixed number of replicas for each application; however, due to the variable mobile traffic in RANs [52], favored applications may not have enough replicas for their users. On the other hand, unfavored applications waste valuable MEC storage. Dynamic mechanisms allocate replicas based on network or storage resources of MEC servers to meet the requirements of mobile applications.

In [53], Chang and Wang analyzed the revenue, cost, and profit for the replicas in MEC servers, and presented algorithms of adaptive replication for MEC. They dynamically allocated replicas based on the number of read/write operations in each time interval. In addition, they proposed an algorithm that forwards requests to replicas in the neighboring MEC servers to balance the load of MEC servers.

In [54], Farris et al., proposed a multi-criteria optimization problem formulation to decide where to duplicate the service while minimizing the probability of service migration and the number of duplicate instances of the service at each handover. They considered proactive service replication a promising strategy to reduce migration time and guarantee the desired QoE. However, the provisioning of replicas over multiple neighbor edge nodes increases the resource consumption of constrained edge nodes and the relevant deployment cost.

2.3 MEC Architecture with Economic Considerations

Several works have investigated the MEC resource allocation and designed incentive-based pricing models. While MEC service providers can achieve certain payoffs from providing their resources to users, users can maximize their experience by paying more for these resources.

Similar to other businesses, the profit of the edge computing service provider is usually related to two components: revenue and expenditure. The revenue is the amount of money that the edge computing service provider receives from selling its resources to its clients.

2.3.1 Profit Components

Cloud provider' profit mainly consists of expenditures and revenue. A cloud providers' profit comes from revenue minus expenses. On the one hand, the cloud provider's expenditures for providing resources to cloud customers include energy consumption and VNF deployment. Therefore, improving the profitability of the cloud provider can be achieved by reducing these expenses. On the other hand, the revenue is related to service pricing strategies and service quality. The service quality has an indirect effect on the revenue of the cloud provider. When the QoS requirements are not met, cloud providers are obligated to pay fines to cloud users because the provider violated the SLA, which affects the current revenue of the cloud provider and has a negative impact on the future market share of the cloud provider. Thus, cloud providers need to increase their revenue by guaranteeing or improving QoS. In [55], Cong et al., summarize a number of profit optimization techniques to increase cloud provider profitability. These optimization techniques are based on service quality, service pricing, energy consumption, and VNF deployment.

In comparison, service pricing strategies have a direct effect on the revenue of the cloud provider. When cloud providers use appropriate pricing strategies, the value of their resources increases and more users will be attracted to the highly competitive cloud provider, thus increasing the revenue of the cloud provider. Table

2.2 shows profit consequences and user experience with varying service prices and service quality. While Table 2.3 shows profit consequences, and server performance with varying energy consumption, and VNF placement.

This thesis aims to maximize profits using a suitable pricing technique in competitive edge computing markets. In the next section, we present some of the pricing techniques used for resource management in cloud networking.

Table 2.2: Profit consequences and user experience from fluctuating service price and service quality.

	Service Pricing		**Service Quality**	
	Low	High	Superior	Inferior
User Experience	Favorable	Unfavorable	Favorable	Unfavorable
Profit Consequences	Less revenue for a single user	More revenue for a single user	Avoid fines	Pay fines

Table 2.3: Profit consequences and server performance from fluctuating energy consumption and VNF placement.

	Energy Consumption		**VNF Placement**	
	High	low	Superior	Inferior
Server Performance	Good	Bad	-	-
Delay	-	-	Low	High
Profit Consequences	Higher electricity bill	Lower electricity bill	-Serve more users -More VM rental fees -Higher electricity bill	-Serve less users -Less VM rental fees -Lower electricity bill

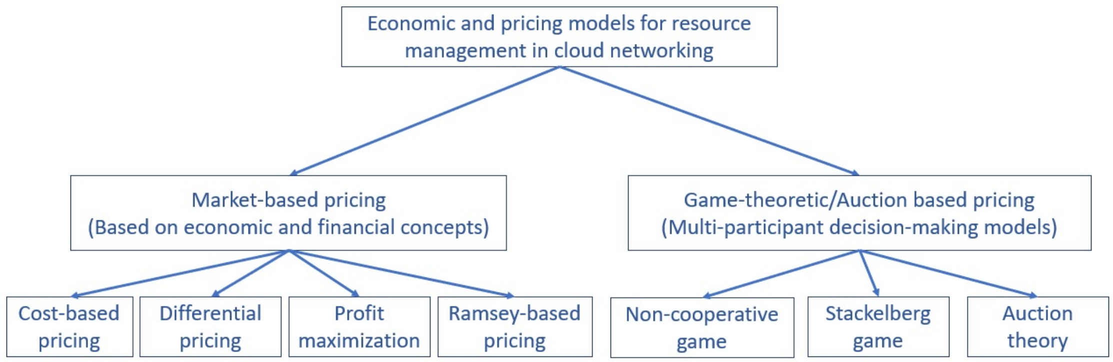

Figure 2.3: Economic and pricing models for resource management in cloud networking

2.3.2 Existing Economic and Pricing Models in Cloud Networking

Recently, economic and pricing models have received more attention because they achieve desirable performance such as social welfare (i.e., the overall well-being of the MEC system), profit, user satisfaction, and resource utilization. Economic and pricing approaches can determine optimal solutions for service providers or users using negotiation mechanisms. In [56], Luong et al., survey a variety of incentive mechanisms using the pricing strategies in sharing resources in edge computing. They classify the economic and pricing models for resource management in cloud networking based on how the prices are set, (i.e., market-based pricing) [57, 58, 59, 60] and game-theoretic and auction-based pricing [30, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70] as shown in Figure 2.3.

Market-Based Pricing

The pricing models are based on economic and financial concepts.

1. *Cost-Based Pricing:* A common pricing strategy determines the price of a service based on calculating the total cost of the service and adding a percentage of the cost as the desired profit. Its objective is to ensure that the price makes the service provider profitable (i.e., at least the fee covers the total cost of the service). In general words, the total cost consists of a fixed cost, which is the cost that does not change when the number of sales changes—for example, hardware costs, such as, servers and network devices, and a variable cost, which varies according to the number of sales of the services produced, for example, the resource costs such as, energy and bandwidth costs, the cost of data transfer between different data centers, and the cost of cloud server usage, are the variable cost for generating cloud services. Regarding the advantages and disadvantages, the benefit of cost-based pricing is the ease of setting the price since the price is a function of the internal cost, or the cost required to generate the service. However, this pricing strategy does not consider external market factors such as the pricing strategies of other providers and the perceived value and willingness consumers will pay [57].

2. *Differential Pricing:* The cloud provider may charge different prices to different cloud users based on their demand and willingness to pay. By setting higher prices for one user type, differential pricing transfers the user surplus to the provider, the user surplus is the difference between the total money users are willing to pay and the total money that they actually pay. Thus, this pricing guarantees a high revenue for the provider. However, it can be unfair because one type of user pays a greater price than another type of user [58].

3. *Profit Maximization:* This determines the output quantity and the corresponding price that yields the highest profit for a provider.

4. *Ramsey Pricing:* This is similar to differential pricing because it sets different prices of the same commodity to other markets depending on the demand elasticity of the commodity. While, unlike differential pricing that maximizes the provider's profit, Ramsey pricing aims to maximize the social welfare of users subject to a predefined threshold on the provider's profit. The threshold is a fixed profit value, which the cloud provider predefines.

In Table 2.4, we summarize and compare market-based pricing models in terms of advantages and disadvantages.

Table 2.4: Comparison between market-based pricing models: Advantages and disadvantages

Pricing Model	**Advantages**	**Disadvantages**
(1) Cost-based pricing	Easy to set the price	It ignores: a- The requirements and preferences of clients b- External market factors: (the pricing strategies of other providers)
(2) Differential pricing	It guarantees a high revenue for the provider	a- It can be unfair: one type of user pay a greater price than another type of user

(3) Profit Maximization	Maximize the provider's profit	a- It is difficult to determine demand curve b- In real market, user demand can be random or follow some distributions c- No market competition applied
(4) Ramsey pricing	Maximize the social welfare of users subject to a predefined threshold on the provider's profit	a- Determining demand elasticity in markets is challenging b- The cloud provider cannot apply this pricing in the long run since users charged with a higher price will seek alternatives

Game-theoretic models and auction mechanisms

Game-theoretic models and auction mechanisms have been widely used to determine resource prices in cloud networking. These models are multi-participant decision-making problems in which a participant's choice (player) potentially affects the interests of other participants. In cloud networking, participants can be cloud providers, service providers, cloud tenants, and users.

Beginning with terms used with Game-theoretic models and auction mechanisms:

- Player: A player is a participant who makes decisions in the game.
- Payoff: A payoff (e.g., a utility, a profit, or an interest) reflects the desired outcome of the player.
- Strategy: Player's strategy is a set of actions/instructions that the player can follow to achieve the desired outcome.

The payoff depends on not only the player's actions, but also the actions of other players.

1. *Non-cooperative game:* In the non-cooperative game, each player maximizes only its payoff, does not consider the other players' payoff nor the network's social welfare. Players are selfish and do not form coalitions or make agreements with each other. Regarding the non-cooperative game's drawback, it may not always achieve in real markets because the players simultaneously know each other's strategies.

   In [30], Long et al., designed a multi-objective computation offloading and resource allocation game considering the uplink, downlink spectrum resource, computation resource, and offloading strategy to minimize the delay and monetary cost of mobile devices (MDs) under the limitation of the computation and communication resources. In [61], Li et al., proposed a game-theoretic computation offloading and resource allocation scheme to optimize the offloading system cost, bandwidth, and resource allocation of the smart mobile terminal with connected multiple wireless access points (APs). They decomposed the offloading problem into two sub-problems (resource allocation and offloading decisions). In [62], Li et al., investigated computation offloading strategy optimization for multiple, heterogeneous, and competitive mobile users and multiple heterogeneous mobile edge clouds by using a non-cooperative game approach. Each player (user equipment) can selfishly minimize his payoff by choosing an appropriate strategy in his strategy space (i.e., all possible strategies that achieve player's constraints).

On the other hand, the cooperative game theory is another approach used to determine an ideal incentive structure between players. In [63], Kaewpuang et al., proposed a framework in which mobile cloud service providers can cooperate to create a resource pool to share their resources. They presented optimization models to obtain the optimal number of application instances that can be supported while meeting the resource requirements of the mobile applications to maximize the revenue of the service providers. Moreover, they applied the concepts of core and Shapley value from cooperative game theory as a solution to maximize the benefit of the mobile cloud service providers for sharing the revenue generated from the resource pool. In addition, the provider optimized the decision on the number of resources to contribute to the resource pool.

2. *Stackelberg game:* It is a sequential game in which players can announce their strategies following a certain predefined order considering the concept of leader and follower, where the player decides his own strategic choice after observing other players' strategies.

   In [65], Cardellini et al., considered an Infrastructure as a Service (IaaS) provider that offers services using cloud resources to multiple Software as a Service (SaaS) providers. They proposed a two-stage provisioning scheme to study the virtual machine provisioning and spot pricing strategies. The SaaS providers determine the optimal number of required flat and on-demand instances in the first stage. After that, they modeled the problem as a Stackelberg game. The IaaS provider sells its unused capacity, and the SaaS providers compete by submitting a bid in the second stage. The IaaS provider sets two different pricing strategies: the first considers that the IaaS provider sets a unique price; the second, the IaaS provider can set different prices for different customers. In [66], Yang et al., used game-theoretic approaches to formulate their problem as a Stackelberg game for minimizing energy consumption. They considered the system monitor was the leader, whose goal was to maximize profit by adjusting resource provisioning. At the same time, scheduler agents act as followers, who select resources to obtain optimal performance.

According to Chen et al., in [64], the MEC system consists of multiple end-users (EUs) and multiple resource-constrained heterogeneous MECs, and each MEC provides several types of resources. They proposed a multiple leaders and multiple followers Stackelberg game-based framework in which mobile edge cloud act as leaders who can set their prices and EUs act as followers who respond by selecting their demands. Their goal is to maximize the revenue for the mobile edge cloud while maximizing the number of resources assigned to users under budget constraints, this is the Stackelberg equilibrium solution. First, they decomposed the multiple resources allocation and the pricing problem into a set of sub-problems. Each sub-problem only considers a single resource type. The Stackelberg game framework is constructed for each sub-problem, whereas each player (i.e., an EU) can selfishly maximize its utility by selecting an appropriate strategy in the strategy space. Then, they proved the existence of the sub-game Stackelberg equilibrium and developed algorithms to determine the Stackelberg equilibrium for each resource type, including an optimal demand computation algorithm, to determine the best resource demand strategy for an EU. In addition, they developed an iterative algorithm to find an equilibrium price. Finally, the equilibrium solutions of all sub-games constitute the equilibrium solution of the original problem.

3. *Auction mechanism:* an auction is another economic mechanism used to investigate resource allocation and pricing in mobile edge computing. In an auction system, a bidding process is allocating commodities and establishing corresponding prices [67].

   First, some common terms used in an auction are the following:

   - Bidder: A bidder is a buyer who wants to purchase resources. In cloud networking, bidders can be end-users or cloud tenants.
   - Seller: A seller, for example, a cloud provider, offers its resources and services for sale.
   - Auctioneer: An auctioneer acts as an intermediate agent to conduct an auction, determine, and announce the winner. In many cases, an auctioneer is a seller as well.
   - Price: A price in an auction may be a bidding price or an asking price.

   The bidding price is the price that the bidder is willing to pay for a requested resource, and the asking price is the price of a resource that the seller is willing to accept. Recently, the auction theory was applied to resource management in cloud networking.

   In [68], Zhang et al., modeled the resource allocation process of a mobile cloud computing system as an auction mechanism and analyzed the individual rationality and incentive compatibility (truthfulness) properties of the users to reach the optimal solutions of the resource allocation and cost charging approaches. In [69], Zheng et al., investigated how the provider sets its price, what prices users should bid, and the bidding strategies of the computing users. They modeled the setting of the cloud provider spot price, derived optimal bidding strategies for different job requirements and interruption overheads, and adapted these strategies to specific kinds of jobs with master and slave nodes having different interruption overheads. In [70], Kiani et al., proposed a new hierarchical model by introducing the concept of field, shallow, and deep cloudlets in which the cloudlet tier architecture consists of three hierarchical levels based on the principle of LTE-advanced backhaul network. They offered and allocated the computing resources in an auction-based profit maximization manner while satisfying the users' QoS.

Table 2.5 summarizes selected existing literature on economic and pricing models for resource management in different paradigms: cloud data center networking, mobile cloud computing, and edge computing (i.e., mobile edge computing, cloudlets, fog computing).

In this thesis, we use the non-cooperative game in Chapter 5 to solve the proposed problem in a distributed manner. As well, we use the Stackelberg game in Chapter 6 to model our proposed problem.

Table 2.5: Existing literature of economic and pricing model for resource management (♣ Cloud data center networking, ♡ Mobile cloud computing, and ♠ Edge computing)

Ref.	**Market Structure**		**Goal**	**Pricing Model**
	Seller	Buyer		
[71] (♣)	Cloud provider	Cloud tenants	Bandwidth efficiency, and welfare social maximization	Auction theory
[58, 72] (♣)	Cloud provider	Cloud tenants	Payment minimization, and high utilization	Differential pricing

[73] (♣)	Cloud provider	Cloud tenants	High utilization, and social welfare maximization	Ramsey pricing
[74] (♣)	Cloud provider	Users	Profit maximization, and optimal accepted tasks	Profit maximization
[75] (♣)	Cloud provider	Application provider	High revenue, and bandwidth guarantee	Stackelberg game
[76] (♣)	Cloud provider	User	Seller's profit maximization	Cost-based pricing
[77] (♣)	Cloud provider	Service providers (players)	Social welfare maximization	Non-cooperative game
[78] (♣)	Cloud providers	Users	Profit maximization, and optimal request allocation	Profit maximization
[79] (♣)	Cloud providers	Cloud user	Buyer's cost minimization	Cost-based pricing
[80, 81] (♣)	Cloud provider	Tenants' virtual networks	Efficient and fair allocation, and seller's profit maximization	Stackelberg game
[66] (♣)	System monitor	Scheduler agents	Minimize energy consumption, and maximize profit	Stackelberg game
[65] (♣)	IaaS provider	SaaS providers	Optimal pricing strategy, and efficient VM provisioning	Stackelberg game
[64] (♠)	Edge servers	Users	Optimize demand of multiple resources, and maximize providers' profits	Stackelberg game
[82] (♠)	Cloudlet	Mobile users	Payment minimization, and profit maximization	Auction Theory
[83] (♠)	Mobile user	Mobile users	Payoff maximization, and resource efficiency	Profit maximization
[84] (♠)	Volunteered computer owner	Users	Payoff maximization, and resource efficiency (CPU, memory, and network bandwidth)	Profit maximization
[64] (♠)	Service providers	Users	Social welfare maximization	Stackelberg game

[30] (♠)	Edge server	MDs (players)	Minimize the delay and monetary cost on MDs	Non-cooperative game
[61] (♠)	Access Points (APs)	Mobile devices (players)	Optimize system cost, bandwidth, and resource allocation	Non-cooperative game
[62] (♠)	Edge servers	Users (players)	Minimize user's payoff	Non-cooperative game
[63] (♡)	Service providers (players)	Users	Maximize service provider revenue	Cooperative game
[85] (♡)	Cloudlet	Brokers (players)	Cost minimization for mobile users	Non-cooperative game

2.4 Challenges and Open Issues

According to Wong et al., in [22], several research challenges remain. In this section, we highlight some of the main challenges and open issues.

2.4.1 Computation Offloading

In MEC, multiple mobile devices share the same mobile edge servers. The limited resources of these edge servers in terms of communication and computation causing edge servers overloading is a major concern; because, edge services will be interrupted. One of the possible solutions is resource management and scheduling over existing MEC resources. Another challenge is that not all mobile edge servers can execute every job because of specific constraints of each user's task (e.g., execution deadlines, energy consumption, and dependency on other users' tasks). Consequently, these constraints will complicate the process of computation offloading [86].

2.4.2 Multi-Resource Scheduling

MEC provides edge users with multiple communication and computation resources. Scheduling tasks requiring various resources is challenging due to the combinatorial nature of allocating the amount of different resources. The majority of existing work has focused on either a single machine or an isolated group of servers. In contrast, MEC has a unique scheduling environment with mobile edge servers and remote CC servers. Therefore, a new multi-resource scheduling technique for MEC must account for the complexity of allocating multiple resources [87]. In addition, distributing jobs across multiple heterogeneous mobile edge servers is challenging because other users and applications share these servers with uncertain workloads. The access quality to these servers is also random due to wireless channel fluctuations, spectrum sharing, and user mobility [88].

2.4.3 Mobility Management

One of the main characteristics of MEC is location awareness of mobile device mobility. When a mobile user moves from one base station to another, its association with active applications running on the mobile edge and remote CC servers must remain intact. Furthermore, when the user moves too far away from its serving mobile edge host, the VMs created on that server for that user must migrate to a more suitable server. Thus, mobility management in MEC involves both communication handoff and computation handoff. Generally speaking, VM migration is costly. In MEC, the heterogeneity of user devices, mobile edge server capabilities, and the strict delay constraints of some applications are complicated. For optimal system design, it will be necessary to model the impact of mobile handoff accurately.

2.4.4 Resource Allocation and Pricing

Resource allocation and resource pricing are coupled together in the edge computing market. The edge provider may charge users a price for offloading user applications to mobile edge servers. The charging process may follow the pricing schemes used by current cloud computing service providers, for example charging higher amounts for on-demand VMs and giving discounts for existing VMs. However, joint optimization of resource pricing and resource allocation remains an open problem for general cloud computing systems. Therefore, the issue of offloading user applications in MEC and then having tasks forwarded further to remote cloud servers is considerably more complicated [89, 90].

2.5 Summary

MEC paradigm is highly useful for emerging resource-hungry and delay-sensitive mobile applications. In this chapter, we introduced the architecture of the MEC system. As well, we presented MEC objectives, MEC characteristics, and MEC enabling technologies. Moreover, we summarized the existing resource management solutions in MEC using task offloading and resource scheduling to minimize latency and energy consumption. In addition, we gave a brief introduction to service migration and service replication in MEC. We also introduced essential factors toward optimizing service providers' profits and stated existing economic pricing techniques. In addition, we presented existing literature on economic and pricing models for resource management in cloud data center networking and edge computing. Finally, we stated the main challenges and open issues.

Chapter 3

Group-based Scalable MEC Architecture

We propose a scalable MEC architecture that aims to improve the scalability for real-time applications that cannot tolerate queuing at the MEC servers to be executed. This proposed architecture forces under the second category of resource management in MEC multi-server edge computing environment 2.2.1.

The rest of this chapter is organized as follows; Section 3.1, the components of the proposed system are explained in detail, and we present our platoon of autonomous vehicles use case. Section 3.2 presents the system parameters used to derive the response time and cost. Section 3.3 describes the general architecture of the proposed system. The simulation setup is presented in Section 3.4. The summary of this chapter is presented in Section 3.5.

3.1 The System Components

The components of our proposed MEC architecture are shown in Figure 3.1. The MEC system consists of two sets of entities, the set of edge servers managed by several service providers and the set of users requesting services from these servers.

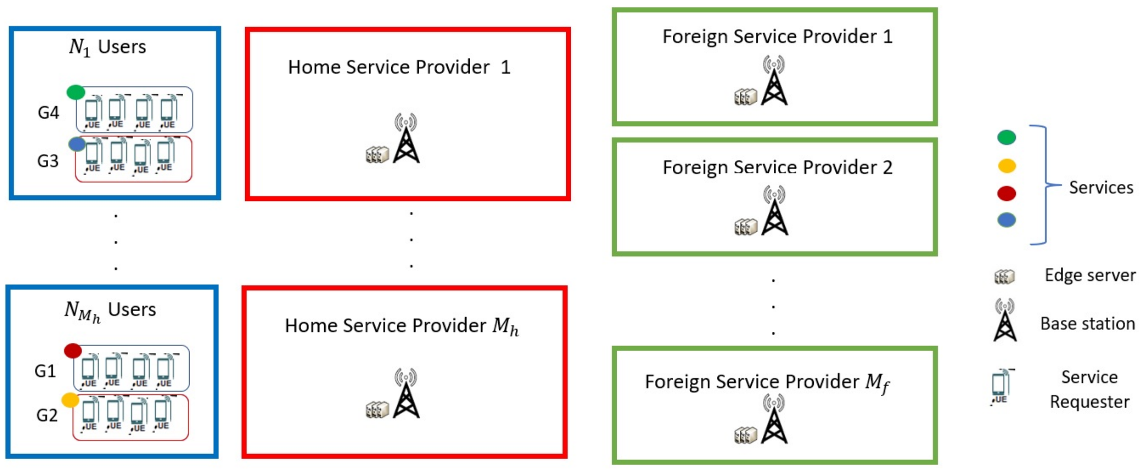

Figure 3.1: Components of group-based scalable MEC architecture

3.1.1 Edge Service Providers

Edge service providers are divided into two subsets, the home service providers (with whom the users have a contract) and the foreign service providers.

Home Service Provider

The home service provider has a set of edge servers deployed with base stations to serve users whenever needed. The user determines its home service provider based on an agreement (a monthly or yearly contract). Let $\mathcal{S}^h = \{S_1^h, S_2^h, \ldots, S_{M_h}^h\}$ be the set of home service providers in the system, and $M_h = |\mathcal{S}^h|$ denotes the number of home service providers. We assume that the home service provider manages a cluster of edge servers in the MEC system. This edge server is either a local edge server or a remote edge server. We assume that users can all connect to their local edge servers, which are typically collocated with their nearest base station, usually identified according to their received signal strength. Whenever needed, each user can also wirelessly connect to one of the remote servers with different channel qualities, which are impacted by its distance and fading conditions. Figure 3.2 shows an example in which the home service provider manages two edge servers. Edge server 1 considers the local edge server for users groups G1 and G2, while it is the remote edge server for groups G3 and G4. On the other hand, edge server 2 is the local edge server for group G3 and group G4, and the remote edge server for group G1 and group G2.

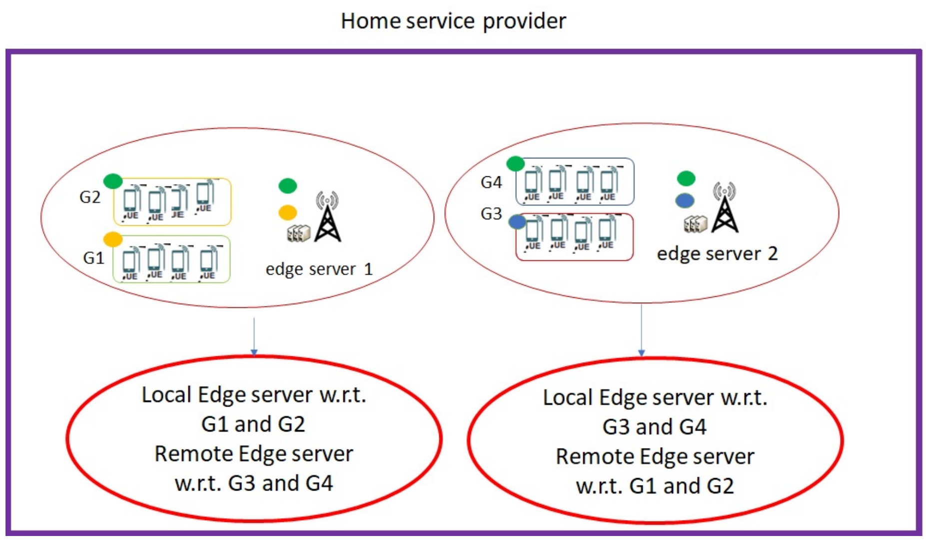

Figure 3.2: An example of one home service provider managing two edge servers

Foreign Service Provider

Other than home service providers, consider foreign service providers to User U_u. Let $\mathcal{S}^f = \{S^f_1, S^f_2, \ldots, S^f_{M_f}\}$ be the set of foreign service providers in the system, and $M_f = |\mathcal{S}^f|$ denotes the number of foreign service providers. All edge servers belonging to home service providers and foreign service providers in the MEC system are connected via backhaul links, used for service replications if needed. In our architecture, we assume the following:

- Each service provider manages a cluster of edge servers. In Chapter 4, these edge servers are collocated with different base stations. In Chapter 5 and Chapter 6, these edge servers are collocated together with the same base station.
- Foreign service providers have ample computing resources.
- We did not consider users of other foreign service providers in the MEC system.
- For the purpose of roaming, users can connect to other service providers when they are far from the base station belonging to their service providers with whom they have a contract. We can use the same roaming option for MEC purposes. Whenever a user cannot access the computing services from its home service provider, a user can wirelessly connect to any nearby available foreign service providers, considering various levels of channel qualities due to the fading conditions.

Furthermore, the provider's computational resources are divisible among multiple users. Let $F^h_{S^h_j}$ denotes the available computational capacity of each home Service Provider S^h_j, $S^h_j \in \mathcal{S}^h$ edge servers cluster, and $F^f_{S^f_i}$ denotes the available computational capacity of each foreign Service Provider S^f_i, $S^f_i \in \mathcal{S}^f$ edge servers cluster in the MEC system, their units are Gigacycle/sec.

3.1.2 Users and Grouping Concept

Let $\mathcal{U}^j = \{U^j_1, U^j_2, \ldots, U^j_{N_j}\}$ be the set of user devices that have a contract with the home service provider S^h_j, $S^h_j \in \mathcal{S}^h$. Moreover, let $\mathcal{U} = \bigcup_{j=1}^{M_h} \mathcal{U}^j$ represent the set of all such users $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$ with home service providers $\mathcal{S}^h$ contracts.

Subsets of these users may be participating in the same common traffic-heavy, real-time, and multimedia-intensive application/activity, such as group AR/VR and multimedia conferencing. We refer to each of these subsets of users as $\mathcal{U}^{G_g} = \{U_1^{G_g}, U_2^{G_g}, \dots, U_{N_g}^{G_g}\}$[1], and all such groups form a set $\mathcal{G} = \{G_1, G_2, \dots, G_{NG}\}$. Let $\mathcal{G} = \bigcup_{j=1}^{M_h} \mathcal{G}^j$, where $\mathcal{G}^j$ represents the groups' set of home service provider S_j^h.

Example:

In Figure 3.2, the system has four groups G1, G2, G3, and G4. Members of Group G_g are participating in the same activity. In this thesis, we assume that the group's members have the same local edge server and the same home service provider.

3.1.3 Use Case

This work can be beneficial in a wide range of applications that involve the cooperation of a group of users, including safety applications facilitated by vehicle platooning. Vehicle platooning is a coordinated vehicular movement where vehicles travel in close proximity to each other in the same lane without mechanical connections [91, 92]. A group of autonomous vehicles can utilize platooning. Autonomous vehicle platooning involves the cooperation of a platoon of vehicles to perform object recognition (e.g., detecting potholes, road surface anomalies, and animal crossings) to avoid road hazards [91, 92]. As depicted in Figure 3.3, groups of autonomous vehicles form platoons; each vehicle is responsible for collecting data from the surrounding environment. In each platoon, vehicles cooperate and send their collected data to the nearest roadside unit (RSU), which acts as an edge server. Note that the automated recognition of patterns and regularities, and cross-validating object identification from the collected

[1]This model captures a unique case scenario where any one or multiple users are solely involved in applications. In this case, each of these users forms a separate group of size 1 (i.e., $N_g = 1$)

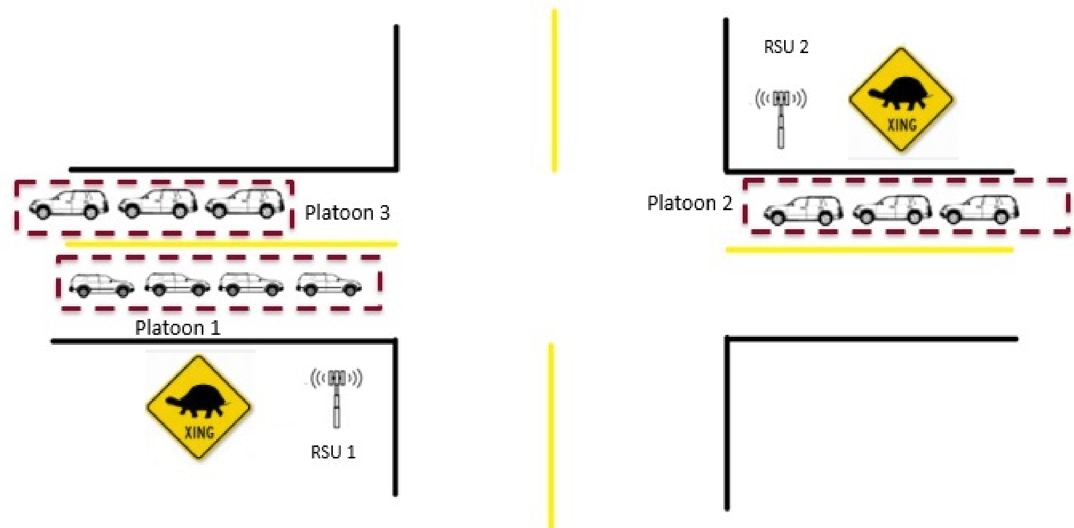

Figure 3.3: Platoon of autonomous vehicles

data, are computationally intensive processes which require powerful computational resources and low latency [91, 92, 93]. Thus, offloading such computationally intensive processes to edge servers (i.e., RSUs) is a more preferable alternative for all members of the platoon [91, 92].

3.2 System Parameters

Members of each group are typically constrained with application-dependent QoS requirements $T_{G_g}^{max}$, $F_{G_g}^{min}$, and d_{G_g}, denoting the maximum delay all members within the group can tolerate for having their tasks completed, the minimum computing capacity these tasks can be processed with, and maximum delay difference between any two users belonging to the same group, respectively. By delay difference, we mean the difference between the delay experienced by any pair of users in the same group. This parameter is extremely important in many of the considered applications. Users must get responses for their offloaded computations within a very small difference in time, or else the application fails. For example, platooning of autonomous vehicles, each vehicle needs to make classification early on to avoid hazards through its way. Therefore, vehicles must receive the results at the same time or within a bounded (typically very small) time difference to be synchronized. Otherwise, a subset of vehicles will make the wrong decisions to avoid hazards.

At execution time, each user sends its task to either its home service provider's edge server or one of its accessible foreign service providers' edge servers. Therefore, $\mathcal{S}^p = S_j^h \bigcup \mathcal{S}^f$ is the set of service providers to which users can offload tasks. A user's task $\mathcal{T}_u$ is defined by the following 3-tuple:

$$\mathcal{T}_u = (C_u, B_u, G_g), \tag{3.1}$$

where u refers to User U_u, $U_u \in \mathcal{U}$, C_u (measured in megacycles) denotes the total number of computing cycles required to complete user U_u's task, B_u (measured in megabytes) describes the size of the information that User U_u needs to send to the edge server for it to execute this task, and G_g, $G_g \in \mathcal{G}$ denotes the ID of the group user U_u belongs to.

3.2.1 Response Time Components

In typical MEC settings, users offload their tasks and access services to their assigned edge server through a direct wireless channel running from user to server. The rate at which the instructions and data of user U_u's computational task (i.e., $\mathcal{T}_u$) are sent to Server S_i^p, $S_i^p \in \mathcal{S}^p$, is expressed as [29]:

$$R_{ui} = B \log_2 \left(1 + \frac{P_u \times H_{ui}}{\sigma_i B}\right). \tag{3.2}$$

where B is channel bandwidth, P_u is the transmission power of User U_u, σ_i denotes the noise power spectral density at Server S_i^p, and H_{ui} represents the channel power gain between User U_u to Server S_i^p defined as:

$$H_{ui} = d_{ui}^{-\alpha}. \tag{3.3}$$

where d_{ui} is the distance between User U_u and Server S_i^p, and α is the path loss exponent, typically set to 4 in mobile edge environments [94].
Consequently, the transmission time of Task $\mathcal{T}_u$ of User U_u to Server S_i^p can be expressed as:

$$T_{ui}^T = \frac{B_u}{R_{ui}}. \tag{3.4}$$

In addition, the computation time of task $\mathcal{T}_u$ at Server S_i^p can be defined as:

$$T_{ui}^C = \frac{C_u}{f_u}. \tag{3.5}$$

where $f_u \geq F_{G_g}^{min}$ is the computing capacity assigned to the task of User U_u that belongs to Group G_g, $G_g \in \mathcal{G}$. In addition, if the task of User U_u is offloaded to foreign Service Provider $S_i^f \in \mathcal{S}^f$, an additional time T_u^R will be required to replicate and deploy this service on that foreign service provider's edge server. The value of the replication time depends on the required service to Group G_g that user U_u belongs to. The last measure required is the roaming time T^{RM} to connect user U_u with other foreign service providers affiliated with the users' home service provider. Thus, We mean the total time between a user submitting a task and getting a response by the response time. This response time includes the transmission time of the task to the edge server, the computation time at the server, the replication time of the application on a foreign service provider's edge server if needed, and the roaming time when a user accesses the required service(s) from a foreign service provider. Therefore, the response time of the user U_u's task $\mathcal{T}_u$ at the edge server of the service Provider S_i^p is given by:

$$RT_{ui} = T_{ui}^{T} + T_{ui}^{C} + T_{u}^{R} + T^{RM}. \tag{3.6}$$

3.2.2 Cost Components

The cost of foreign Service Provider S_i^f for providing any service is defined as the summation of service cost (computation plus transmission) and replication cost. We show the service cost equations in Chapter 5. In Chapter 5, all unit-cost were static and predefined before because we were looking at managing task offloading for groups of users requesting to perform inter-related computations between different service providers, home service providers and foreign service providers. While in Chapter 6, the price of a computation cycle for each foreign Service Provider was dynamic, which is denoted as y_i, $\forall\ S_i^f \in \mathcal{S}^f$. We show this because we aimed to set profitable prices and maximize social welfare between all service providers.

3.3 General System Architecture

In this section, we introduce how previously mentioned entities interact together to prevent edge servers from overloading thus, improve the scalability of MEC services.

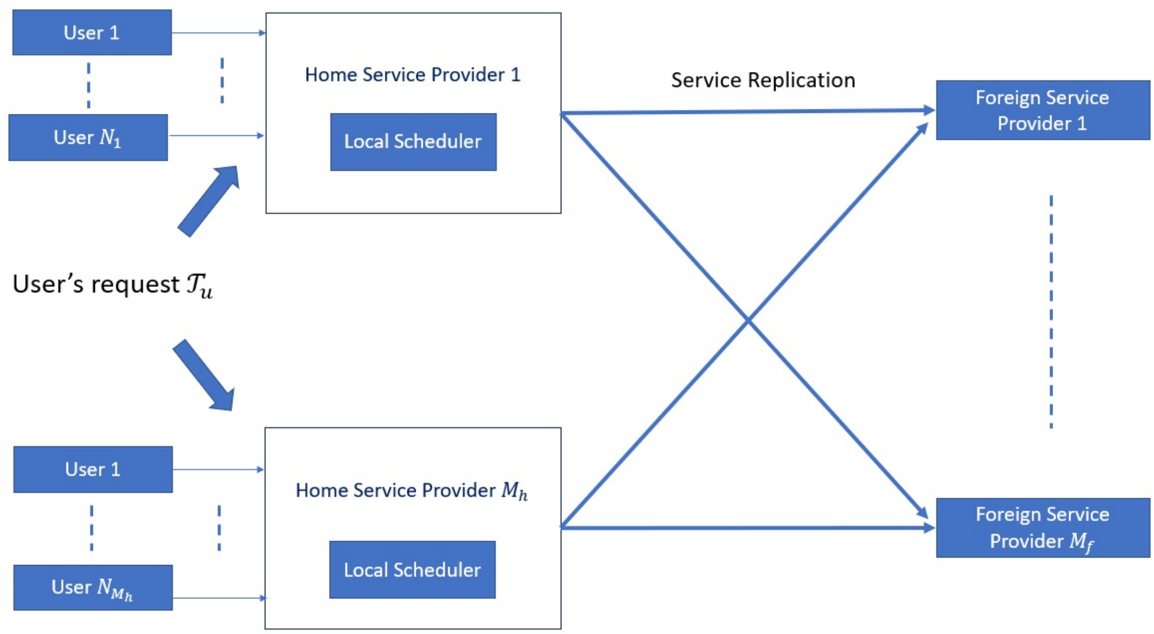

Figure 3.4: General system architecture

To serve users in the most efficient manner, each home service provider has a local scheduler to manage the available resources between their users, as shown in Figure 3.4.

In this thesis, if we solve the proposed problem in a centralized manner, one of these local schedulers will manage the scheduling process. The local scheduler collects users' tasks, group's QoS parameters and the available computational capacity for all home service providers and foreign service providers. The local scheduler solves the proposed problem as shown in Section 4.2 and Section 5.3. On the other hand, if we solve the same problem in a distributed manner, the local scheduler for each home service provider will manage its local users only. This local scheduler collects the contracted users' tasks of the home Service Provider S_j^h, $\forall\ S_j^h \in \mathcal{S}^h$, group's QoS parameters, the available computing capacity for home Service Provider S_j^h and foreign service providers. This approach solves the proposed problem as shown in

Section 4.3 and Section 5.3. The reason behind using a distributed allocation is that centralized allocation might be unrealistic in real scenarios because of the need for a central unit.

Regarding the users, each user has its task $\mathcal{T}_u$; however, these tasks are dependent on the group concept's explanation. User devices do not have enough resources (i.e., computational capacity) to process their tasks. Thus, they will be offloading their task (i.e, binary offloading) to their home service provider's local edge server unless it is overloaded. Regarding home service providers, each home Service Provider , S^h_j, $S^h_j \in \mathcal{S}^h$ receives several individual requests or many requests at once from a group for specific services. The service provider has a scheduler, that allocates the user's task to the proper edge server. If it has enough resources, it will serve all incoming requests. However, if this home service provider receives too many tasks, the server will be overloaded. In this case, the un-served users of the home service provider are allowed to access the foreign service providers' resources after the services of the home service provider have been replicated to the foreign service providers. Therefore, the home service provider replicates the required service(s) on one or more remote edge servers managed by foreign service providers to serve all remaining un-served users.

3.4 Simulation Setup

This section presents the simulations setup.

3.4.1 Service Provider Parameters

In Chapter 4, we have a total of eight edge servers deployed within an area measuring 1000X1000 m. We have one service provider managing all eight edge servers, four of them are designated as local edge servers and the other four are designated as remote edge servers because we wanted to manage task offloading for groups of users requesting to perform inter-related computations on multiple overloaded and under-loaded MEC servers from the users' perspective. The computational capacity of each edge server follows a uniform distribution in the range of [300-800] Gigacycle/sec.

In Chapter 5 and Chapter 6, we have a total of eight service providers, and each one manages a cluster of edge servers. Edge servers of the same cluster are deployed together in an area measuring 1000X1000 m. We designated four service providers as home service providers, and the other four as foreign service providers. In these chapters, we want to look at managing task offloading and replication allocation for groups of users requesting to perform inter-related computations from the service providers' perspective using static predefined pricing and dynamic pricing technique. The computational capacity of each service provider' edge servers follows a uniform distribution in the range of [300-800] Gigacycle/sec.

Figure 3.5 shows the locations of local and remote edge servers in the simulated area, which are managed by the same service provider. While, Figure 3.6 shows the locations of edge server cluster of home and foreign service providers in the simulated area.

3.4.2 User Parameters

The user devices' transmission power is set to $P_u = 20$ dBm. The data size of the tasks follows a uniform distribution in the range of [10-50] Kilo bit. The computing cycle of any one task follows a uniform distribution in the range of [100-700] cycles/byte.

All groups of user devices are deployed near to (yet at random distances from) the

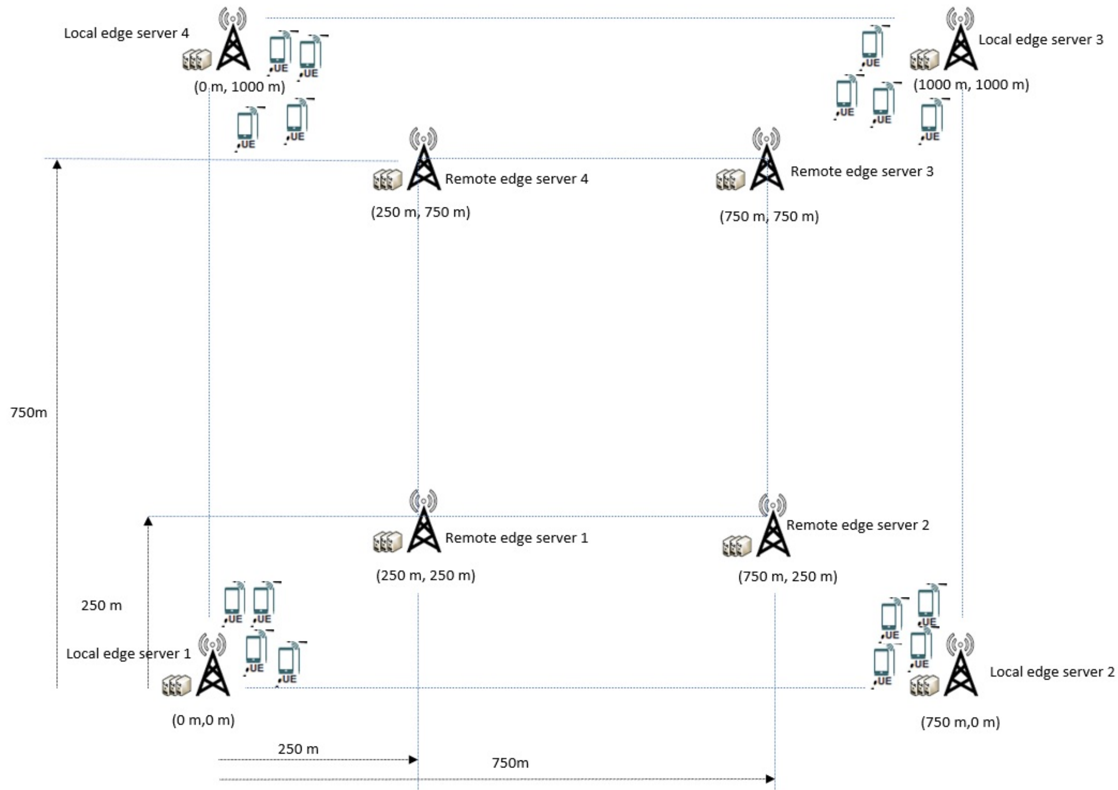

Figure 3.5: Simulation setup discussed in Chapter 4

number of specific service providers' edge servers designated as overloaded home service providers' edge servers, each group has a group AR/VR application with different QoS requirements. We assume all users belong to Group G_g, $G_g \in \mathcal{G}$, are assigned computing capacities $f_u = F_{G_g}^{min}$, $\forall\ U_u \in \mathcal{U}$, which follow a uniform distribution in the range of [15-20] Gigacycle/sec.

3.4.3 Channel Parameters

For a wireless access channel, the bandwidth for uplink and downlink channels is equal and is set to B = 1 MHz, and the noise power spectral density is set to -174 dBm/HZ. We assume any problems are going to be one-time problems as when

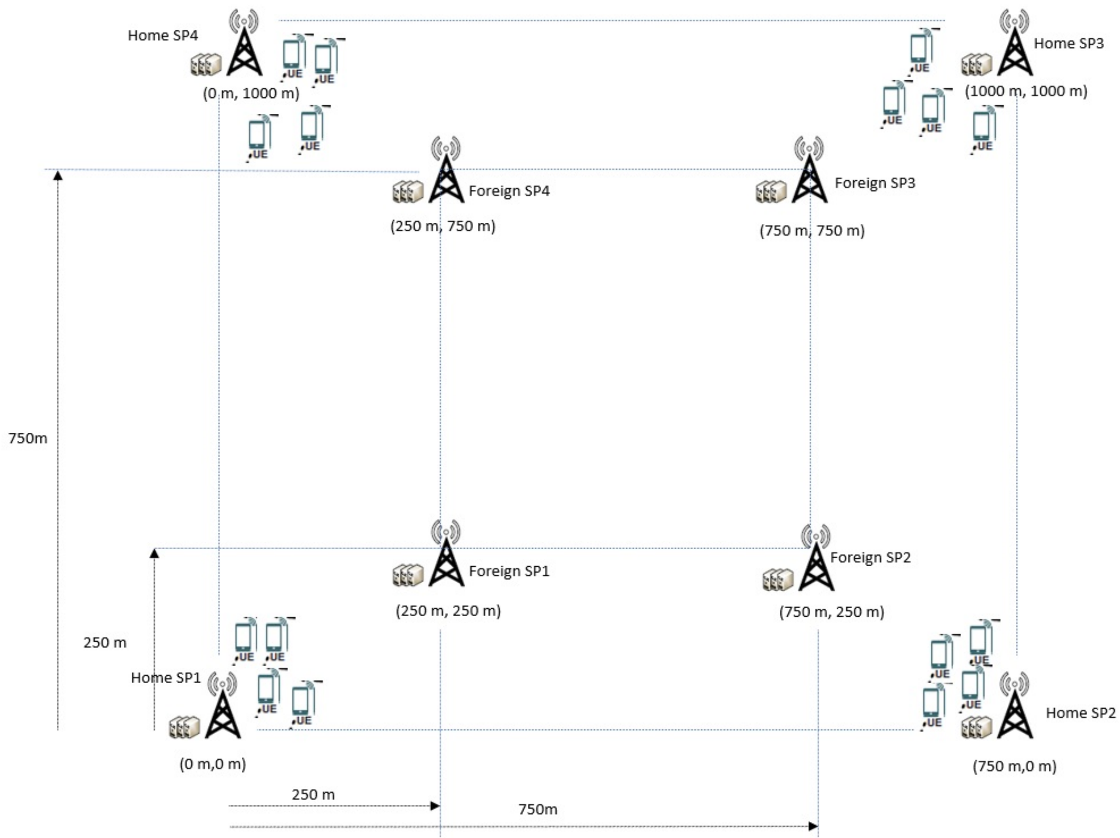

Figure 3.6: Simulation setup discussed in Chapter 5 and Chapter 6

channels remain unchanged during the task offloading, which is known as "quasi-static channel modeling."

The optimized solutions were derived using Linear programming and the corresponding solver from the MATLAB Optimization Toolbox [95].

3.5 Summary

Motivated by overloading servers and services interruptions of the MEC system, we proposed a scalable MEC architecture that utilizes remote edge servers, and foreign service providers resources to improve the MEC service scalability and solve the challenges of service interruptions and QoS degradation. The architecture consists of (1) a set of users and (2) a set of service providers. We described how these entities were interacting together. We reviewed the simulation set up. In the next three chapters we compare the performance of the proposed architecture in various scenarios.

Chapter 4

Scalable MEC from the Perspective of Users

We found that most of the literature did not consider the scenario where a group of users are split into groups due to participating in the same activity under time and time different constraints. In addition, our proposed work seeks to improve the scalability for real-time applications that cannot tolerate queuing at the MEC servers to be executed. From the perspective of users, un-served users will assign to other remote edge servers when their local edge servers are overloaded while satisfying the groups' QoS requirements.

In this chapter, we address these detailed research questions:

1. How can the scalability of local edge services be improved using their replication on remote edge servers?
2. How does the dependency between users affect the distribution of offloaded tasks?
3. What is the best distribution using service replication on remote edge servers to minimize users' response times?

In this chapter, we propose a scheme improves the scalability of real-time offered services by replicating them on-demand at underloaded remote edge servers in the MEC system to serve all users concurrently when the local servers cannot sustain the load of all user groups. We formulate an integer linear problem that minimizes the average response time of all users while satisfying the time and time difference constraints of the user groups running the same applications. We deal with the linear programming (LP) relaxation of our problem, which is used to derive optimal decisions using Lagrangian analysis. We solve the proposed LP problem numerically and apply a proposed rounding algorithm to reach the integer values of the decision variables. We compare the proposed solution with the ILP solution and show that the LP relaxation solution performs almost the same while saving on the time complexity. We compare the performance of our proposed LP relaxation solution with distance-based and resource-based greedy approaches.

The rest of this chapter is organized as follows; Section 4.1 details our system model and parameters, and Section 4.2 discusses the problem setup. Section 4.3 describes the solution using linear relaxation. Section 4.4 presents the performance evaluation and the comparison of the two greedy approaches. We conclude our findings in Section 4.5.

4.1 System Model

The system model is depicted in Figure 4.1. The details of our system model and parameters are described in Chapter 3, where we consider that the MEC system consists of a user set and one home service provider only. This service provider manages a cluster of edge servers, namely, local edge servers and remote edge servers, as shown in Figure 3.5. In this chapter, we assume the following:

- The local edge server of every User U_u always achieves the least response time.
- Remote edge servers have considerable amounts of computing resources.
- In this chapter, all edge servers have the home same service provider. Thus, the roaming option is not considered.
- The service provider manages a cluster of edge servers, these edge servers are collocated with different base stations.

We assume that local edge servers have demands that exceed their capacities. Thus, the un-served users of these local edge servers can access the remote edge servers' resources after the services of local edge servers have been replicated to the distant edge servers. Therefore, we divide the home service provider's cluster of edge servers into two subsets, overloaded and underloaded. The overloaded set typically consists of local servers whose local users have demands that exceed their capacities.

Let $\mathcal{S}^o = \{S_1^o, S_2^o, \ldots, S_{M_o}^o\}$ be the set of overloaded edge servers in the system, $\mathcal{S}^u = \{S_1^u, S_2^u, \ldots, S_{M_u}^u\}$ be the set of underloaded edge servers in the system, and $\mathcal{S} = S_j^o \bigcup \mathcal{S}^u$ be a set of several edge servers, of which one server a user can offload tasks to. Edge servers' computational resources are divisible among multiple users. Let $F_{S_i^o}^o$ denotes the available computational capacity of each overloaded edge server S_i^o, $S_i^o \in \mathcal{S}^o$, and $F_{S_i^u}^u$ denotes the available computational capacity of each underloaded edge server S_i^u, $S_i^u \in \mathcal{S}^u$ in the MEC system, their units are Gigacycle/sec.

Given the preceding system description, we can formulate our group-delay aware task offloading and service replication problem in the next section. This problem aims to minimize the average response time of all users served by both local and

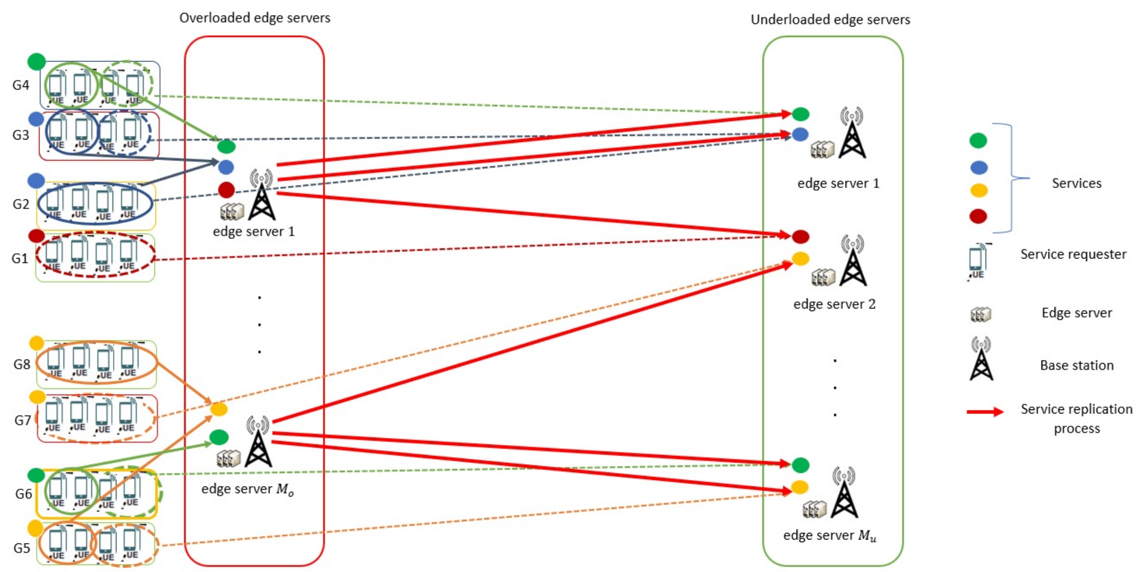

Figure 4.1: System model from the users' perspective

remote edge servers while satisfying all the QoS requirements of users and groups. Here, the response time includes the transmission time of the task to the edge server, the computation time at the server, and the replication time of the application on a remote edge server when a user is forced to send its task to a distant edge server if needed. Roaming time is not considered because one service provider manages all edge servers in the MEC system.

4.2 Problem Setup

In this section, we define objective variables and show the task offloading problem formulation.

4.2.1 Definition of Objective Variables

Let x_{ui}, $\forall\ u,\ i$, be the task offloading indicators, where u and i refer to User U_u, $U_u \in \mathcal{U}$, and Server S_i, $S_i \in \mathcal{S}$, respectively. In our formulation, x_{ui} is set to 1 when User U_u offloads its task to Server S_i and is set to 0 otherwise.

4.2.2 Problem Formulation

Given the preceding definition of the objective variables, the task offloading problem achieving our target can be expressed as follows:

$$\min_{x_{ui}} \ \frac{1}{N}\sum_{i\in\mathcal{S}}\sum_{u=1}^{N} x_{ui} RT_{ui} \tag{4.1}$$

$$S.T. \quad \sum_{i\in\mathcal{S}} x_{ui} = 1, \qquad \forall\ U_u \in \mathcal{U} \tag{4.2}$$

$$\sum_{u\in\mathcal{U}^j} x_{ui} f_u \le F^o_{S^o_j}, \qquad \forall\ S^o_j \in \mathcal{S}^o \tag{4.3}$$

$$\sum_{u=1}^{N} x_{ui} f_u \le F^u_{S^u_i}, \qquad \forall\ S^u_i \in \mathcal{S}^u \tag{4.4}$$

$$\sum_{i\in\mathcal{S}} x_{ui} RT_{ui} \le T^{max}_{G_g},\ \ \forall\ G_g \in \mathcal{G},\ \ \forall\ U^{G_g}_u \in \mathcal{U}^{G_g} \tag{4.5}$$

$$\sum_{i\in\mathcal{S}} x_{ui} RT_{ui} - \sum_{i\in\mathcal{S}} x_{li} RT_{li} \le d_{G_g}, \qquad \forall\ G_g \in \mathcal{G},\ \ \forall\ U^{G_g}_u, U^{G_g}_l \in \mathcal{U}^{G_g},\ \ l \ne u \tag{4.6}$$

$$x_{ui} \in \{0,1\}, \qquad \forall\ U_u \in \mathcal{U},\ \ \forall\ S_i \in \mathcal{S} \tag{4.7}$$

The objective function in (4.1) minimizes the average response time of all users' tasks by determining the best edge server each User U_u should offload their tasks. Constraint (4.2) ensures that each task is offloaded to only one server. Constraint (4.3) ensures that the sum of assigned computing capacities to the subset of users offloading their tasks to their local edge server does not exceed its total available computing capacity. Constraint (4.4) ensures that the sum of assigned computing capacities to all users offloading their tasks to each remote edge server does not exceed its total available computing capacity. Constraints (4.3) and (4.4) ensure that the maximum computational capabilities of the servers are not exceeded. Constraint (4.5) ensures that the total response time of a user's task belonging to any Group G_g does not exceed this group's maximum delay requirement. Constraint (4.6) ensures that the delay difference between two users belonging to the same Group G_g does not exceed its application's upper limit. Constraints (4.5) and (4.6) are to obeyed to the QoS demanded by each group. Finally, the last Constraint (4.7) imposes a binary decision value for the variable x_{ui} of all users and servers.

It is clear that the optimization problem is an integer linear problem (ILP). To solve this problem, we relax the ILP optimization problem into linear programming (LP) by relaxing the integer decision variables, deriving analytical and solver-based solutions, and then restoring the integer decision variables using a greedy approach. We then compare this solution to a solver-based solution to the ILP problem and two greedy approaches to satisfy the scalability and group-delay awareness settings considered in this chapter.

4.3 Solution Using Linear Relaxation

In this section, we relax the previous optimization problem into a linear programming (LP) formulation. The objective function and constraints are similar to the integer linear problem except for Constraint (4.7), replaced by $0 \leq x_{ui} \leq 1$. In [96], the output of a linear problem is an integer if each square submatrix of constraint coefficient matrices is unimodular, which can be determined by calculating the determinant of each coefficient matrix. We investigated whether this property applies to our problem, unfortunately, it failed for some sub-matrices.

4.3.1 Analytical Solutions

The linear problem version of (4.1)-(4.7) is a convex optimization problem. The Karush-Kuhn-Tucker (KKT) conditions thus provide necessary and sufficient conditions for an optimal solution [97]. Therefore, applying the KKT conditions to the constraints of the problem and the Lagrangian function's gradient allows us to find the analytical solution of the real offloading decision variables x_{ui}. The Lagrangian function associated with the linear optimization problem is given by (4.8), where X is the vector of offloading decisions (i.e., $X = [x_{ui}]$), u refers to user U_u, $\forall\ U_u \in \mathcal{U}$, and i refers to the edge server $S_i, \forall\ S_i \in \mathcal{S}$, and:

- $\lambda_{local} = [\lambda_{local_i}]$, $\lambda_{remote} = [\lambda_{remote_i}]$, $\alpha_p = [\alpha_u]$, $\sigma_p = [\sigma_{gul}]$, are the associated Lagrange multiplier to inequality constraints (4.3), (4.4), (4.5), (4.6), respectively.

- $\beta = [\beta_{ui}]$, $\gamma = [\gamma_{ui}]$ are the associated Lagrange multiplier to the lower bound and upper bound inequality constraint of x_{ui}, respectively.

- $\nu = [\nu_u]$ is the associated Lagrange multiplier to the equality constraint (4.2).

$$
\begin{aligned}
L(X, \lambda_{local}, \lambda_{remote}, \alpha_p, \sigma_p, \beta, \gamma, \nu) &= \sum_{i \in \mathcal{S}} \sum_{u=1}^{N} \frac{RT_{ui}}{N} x_{ui} + \sum_{i \in \mathcal{S}^o} \lambda_{local_i} \left(\sum_{u \in \mathcal{U}^j} f_u x_{ui} - F^o_{S^o_j} \right) \\
&+ \sum_{i \in \mathcal{S}^u} \lambda_{remote_i} \left(\sum_{u=1}^{N} f_u x_{ui} - F^u_{S^u_i} \right) + \sum_{g=1}^{NG} \sum_{u \in \mathcal{U}^{G_g}} \alpha_u \left(\sum_{i \in \mathcal{S}} RT_{ui} x_{ui} - T^{max}_{G_g} \right) \\
&+ \sum_{g=1}^{NG} \sum_{u \in \mathcal{U}^{G_g}} \sum_{l \in \mathcal{U}^{G_g}, l \neq u} \sigma_{guj} \left(\sum_{i \in \mathcal{S}} x_{ui} RT_{ui} - \sum_{i \in \mathcal{S}} x_{li} RT_{li} - d_{G_g} \right) \\
&- \sum_{i \in \mathcal{S}} \sum_{u=1}^{N} \beta_{ui} x_{ui} + \sum_{i \in \mathcal{S}} \sum_{u=1}^{N} \gamma_{ui} \left(x_{ui} - 1 \right) + \sum_{u=1}^{N} \nu_u \left(\sum_{i \in \mathcal{S}} x_{ui} - 1 \right) \\
&\forall\ S^o_i \in \mathcal{S}^o,\ \forall\ S^u_i \in \mathcal{S}^u,\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u, \in \mathcal{U}^{G_g}
\end{aligned}
\tag{4.8}
$$

KKT Conditions

In the following equations, we apply the KKT conditions on the equality and inequality constraints of the relaxed linear relaxation problem.

- Primal feasibility conditions:

$$\sum_{u \in \mathcal{U}^j} f_u x^*_{ui} - F^o_{S^o_j} \leq 0 \quad \forall\ S^o_j \in \mathcal{S}^o \tag{4.9a}$$

$$\sum_{u=1}^{N} f_u x^*_{ui} - F^u_{S^u_i} \leq 0 \quad \forall\ S^u_i \in \mathcal{S}^u \tag{4.9b}$$

$$\sum_{i \in \mathcal{S}} RT_{ui} x^*_{ui} - T^{max}_{G_g} \leq 0 \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u \in \mathcal{U}^{G_g} \tag{4.9c}$$

$$\sum_{i \in \mathcal{S}} x^*_{ui} RT_{ui} - \sum_{i \in \mathcal{S}} x^*_{li} RT_{li} d_{G_g} \leq 0, \qquad \forall G_g \in \mathcal{G},\ \forall\ U^{G_g}_l, U^{G_g}_u \in \mathcal{U}^{G_g} \tag{4.9d}$$

$$\sum_{i\in\mathcal{S}} x^*_{ui} = 1 \quad \forall\ U_u \in \mathcal{U} \tag{4.9e}$$

$$-\,x^*_{ui} \leq 0 \quad \forall\ S_i \in \mathcal{S},\ \forall\ U_u \in \mathcal{U} \tag{4.9f}$$

$$x^*_{ui} - 1 \leq 0 \quad \forall\ S_i \in \mathcal{S},\ \forall\ U_u \in \mathcal{U} \tag{4.9g}$$

- Dual feasibility conditions:

$$\lambda^*_{local_i}, \lambda^*_{remote_i},\ \alpha^*_u,\ \sigma^*_{guj},\ \beta^*_{ui},\ and\ \gamma^*_{ui} \geq 0, \qquad \forall\ S_i \in \mathcal{S},\ \forall\ U_u \in \mathcal{U} \tag{4.10a}$$

$$\begin{aligned}
&\frac{\partial L(X^*, \lambda^*_{local}, \lambda^*_{remote}, \alpha^*_p, \sigma^*_p, \beta^*, \gamma^*, \nu^*)}{\partial x^*_{ui}} = 0\\
&\frac{RT_{ui}}{N} + \lambda^*_{local_i} f_u + \alpha^*_u RT_{ui} + \sigma^*_{gul} RT_{ui} - \beta^*_{ui} + \gamma^*_{ui} + \nu^*_u = 0,\\
&\qquad\qquad \forall\ S^o_i \in \mathcal{S}^o,\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_l \in \mathcal{U}^{G_g}, l \neq u
\end{aligned} \tag{4.10b}$$

$$\begin{aligned}
&\frac{\partial L(X^*, \lambda^*_{local}, \lambda^*_{remote}, \alpha^*_p, \sigma^*_p, \beta^*, \gamma^*, \nu^*)}{\partial x^*_{ui}} = 0\\
&\frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} + \sigma^*_{gul} RT_{ui} - \beta^*_{ui} + \gamma^*_{ui} + \nu^*_u = 0\\
&\qquad\qquad \forall\ S^u_i \in \mathcal{S}^u,\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_l \in \mathcal{U}^{G_g}, l \neq u
\end{aligned} \tag{4.10c}$$

$$\begin{aligned}
&\frac{\partial L(X^*, \lambda^*_{local}, \lambda^*_{remote}, \alpha^*_p, \sigma^*_p, \beta^*, \gamma^*, \nu^*)}{\partial x^*_{li}} = 0\\
&\sigma^*_{gul} RT_{li} = 0 \qquad \forall\ S_i \in \mathcal{S},\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_j \in \mathcal{U}^{G_g}, l \neq u
\end{aligned} \tag{4.10d}$$

- Complementary slackness conditions:

$$\lambda^*_{local_j} \left(\sum_{u\in\mathcal{U}^j} f_u x^*_{ui} - F^o_{S^o_j} \right) = 0, \quad \forall\ S^o_j \in \mathcal{S}^o \tag{4.11a}$$

$$\lambda^*_{remote_i}\left(\sum_{u=1}^{N} f_u x^*_{ui} - F^u_{S^u_i}\right) = 0, \quad \forall\ S^u_i \in \mathcal{S}^u \tag{4.11b}$$

$$\alpha^*_u\left(\sum_{i\in\mathcal{S}} RT_{ui}x^*_{ui} - T^{max}_{G_g}\right) = 0, \qquad \forall\ G_g \in \mathcal{G},\ U^{G_g}_u \in \mathcal{U}^{G_g} \tag{4.11c}$$

$$\sigma_{gul}\left(\sum_{i\in\mathcal{S}} x_{ui}RT_{ui} - \sum_{i\in\mathcal{S}} x_{li}RT_{li} - d_{G_g}\right) = 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_l,\ U^{G_g}_u \in \mathcal{U}^{G_g} \tag{4.11d}$$

$$\beta^*_{ui}x^*_{ui} = 0, \quad \forall\ S_i \in \mathcal{S},\ \forall\ U_u \in \mathcal{U} \tag{4.11e}$$

$$\gamma^*_{ui}\left(x^*_{ui} - 1\right) = 0, \quad \forall\ S_i \in \mathcal{S},\ \forall\ U_u \in \mathcal{U} \tag{4.11f}$$

After applying the KKT conditions on the equality and inequality constraints, the following theorem illustrates the optimal solution for the linear problem.

Theorem 4.1: The optimal offloading decisions of the linear optimization problem can be expressed as shown in (4.12) and (4.13).

When $\forall\ S^o_i \in \mathcal{S}^o,\ \forall\ U_u, \in \mathcal{U}^i$,

$$x^*_{ui} = \begin{cases} 0, & if\ \gamma^*_{ui} = 0,\ \beta^*_{ui} > 0,\ \frac{RT_{ui}}{N} + \lambda^*_{local_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u \neq 0 \\ 1, & if\ \gamma^*_{ui} > 0,\ \beta^*_{ui} = 0,\ \frac{RT_{ui}}{N} + \lambda^*_{local_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u = -\gamma^*_{ui} \end{cases} \tag{4.12}$$

Otherwise, when $\forall\ S^u_i \in \mathcal{S}^u,\ \forall\ U_u, \in \mathcal{U}$,

$$x^*_{ui} = \begin{cases} 0, & if\ \gamma^*_{ui} = 0,\ \beta^*_{ui} > 0,\ \frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u \neq 0 \\ 1, & if\ \gamma^*_{ui} > 0,\ \beta^*_{ui} = 0,\ \frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u = -\gamma^*_{ui} \end{cases} \tag{4.13}$$

Proof. Using the equations we derived using KKT conditions in (4.9), (4.10), and (4.11), we can see from (4.10d) that $RT_{li} \neq 0$, and $\sigma^*_{gul} \geq 0$, and thus $\sigma^*_{gul} = 0$. By substituting in (4.10b) and (4.10c), we get:

$$\frac{RT_{ui}}{N} + \lambda^*_{local_j} f_u + \alpha^*_u RT_{ui} - \beta^*_{ui} + \gamma^*_{ui} + \nu^*_u = 0, \qquad \forall\ S^o_j \in \mathcal{S}^o,\ \forall\ U_u \in \mathcal{U} \tag{4.14}$$

$$\frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} - \beta^*_{ui} + \gamma^*_{ui} + \nu^*_u = 0, \qquad \forall\ S^u_i \in \mathcal{S}^u,\ \forall\ U_u,\ U_l \in \mathcal{U}, l \neq u \tag{4.15}$$

By Multiplying (4.14) by x^*_{ui} and using (4.11e), (4.11f), substitute in the result of multiplication, we get:

$$x^*_{ui} = \frac{-\gamma^*_{ui}}{\frac{RT_{ui}}{N} + \lambda^*_{local_j} f_u + \alpha^*_u RT_{ui} + \nu^*_u} = \frac{-\gamma^*_{ui}}{\beta^*_{ui} - \gamma^*_{ui}}, \qquad \forall\ S^o_j \in \mathcal{S}^o,\ \forall\ U_u, \in \mathcal{U}^j \tag{4.16}$$

Similarly, by multiplying (4.15) by x^*_{ui}, and using (4.11e), (4.11f), substitute in the result of multiplication, we get:

$$x^*_{ui} = \frac{-\gamma^*_{ui}}{\frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u} = \frac{-\gamma^*_{ui}}{\beta^*_{ui} - \gamma^*_{ui}}, \qquad \forall\ S^u_i \in \mathcal{S}^u,\ \forall\ U_u, \in \mathcal{U} \tag{4.17}$$

From (4.16), (4.11e), and (4.11f), we only have two viable cases:

- Case 1: $x^*_{ui} = 0$ when $\gamma^*_{ui} = 0$ and $\beta^*_{ui} > 0$, which means that $\frac{RT_{ui}}{N} + \lambda^*_{local_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u > 0$.

- Case 2: $x^*_{ui} = 1$ when $\gamma^*_{ui} > 0$ and $\beta^*_{ui} = 0$, which means that $\frac{RT_{ui}}{N} + \lambda^*_{local_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u = -\gamma^*_{ui}$. Consequently, ν^*_u should be negative.

Otherwise, when $\gamma^*_{ui} = \beta^*_{ui} = 0$, $\gamma^*_{ui} = \beta^*_{ui} > 0$, or $\gamma^*_{ui} > 0$ and $\beta^*_{ui} > 0$, x^*_{ui} has no valid value.

From (4.17), (4.11e), and (4.11f), we only have two viable cases:

- Case 1: $x^*_{ui} = 0$ when $\gamma^*_{ui} = 0$ and $\beta^*_{ui} > 0$, which means that $\frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u > 0$.

- Case 2: $x^*_{ui} = 1$ when $\gamma^*_{ui} > 0$ and $\beta^*_{ui} = 0$, which means that $\frac{RT_{ui}}{N} + \lambda^*_{remote_i} f_u + \alpha^*_u RT_{ui} + \nu^*_u = -\gamma^*_{ui}$. Consequently, ν^*_u should be negative.

Otherwise, when $\gamma^*_{ui} = \beta^*_{ui} = 0$ (because of the limitation of x^*_{ui} when γ^*_{ui} and β^*_{ui} tend to zero), $\gamma^*_{ui} = \beta^*_{ui} > 0$, or $\gamma^*_{ui} > 0$ and $\beta^*_{ui} > 0$, x^*_{ui} has no valid value. ■

4.3.2 Numerical Results

From (4.12) and (4.13), we found that the analytical solutions did not yield closed-form expressions for the relaxed problem. Therefore, we solve the linear problem using a numerical solver. We then employ the greedy rounding approach, described in Algorithm 1, to restore the binary values of the decision variables x_{ui}. This algorithm is based on the one defined in [98]. The worst-case time complexity of the rounding algorithm is $O(N^2 M_u^2)$.

The time complexity of the linear relaxation-based solution is the summation of the complexities of solving the linear problem and the rounding algorithm. This time complexity was shown to be less complex than that of solving the original ILP [99, 100, 101].

4.4 Performance Evaluation

This section presents the simulation results showing the performance of the ILP and linear relaxation-based solution and the merits of the latter.

Algorithm 1: Greedy rounding algorithm of task offloading indicators

Input : x^*_{ui}, $\forall\ S_i \in \mathcal{S}$, $\forall\ U_u \in \mathcal{U}$
Output: x_{ui}, $\forall\ U_u \in \mathcal{U}$, $\forall\ S_i \in \mathcal{S}$
Steps:
Sort the set of decision variables x^*_{ui}, $\forall\ S_i \in \mathcal{S}$, $\forall\ U_u \in \mathcal{U}$ in a descending order
Rounding $x^*_{ui} \in \{0, 1\}$
if *round*$(x^*_{ui}) = 1$ **then**
  if *constraints (4.2)-(4.7) are achieved* **then**
    $x_{ui} \leftarrow 1$
  else
    $x_{ui} \leftarrow 0$
  end
else
  $x_{ui} \leftarrow 0$
end

4.4.1 Performance Metrics

In this chapter, we interested in calculating the average response time (delay) and the satisfaction ration, which are:

1. **Average Response Time (Delay)**: This is calculated as the average transmission, computation, and replication time overall users in all edge servers. We do not consider the roaming time because one home service provider manages all edge servers in the system.

2. **Satisfaction Ratio**: This is the ratio of the number of users who satisfy the time difference constraint to all number of users in the system.

4.4.2 Simulation Setup

In our simulations, there are four overloaded local edge servers, four underloaded remote edge servers, we assume one service provider manages all edge servers. For each local edge server, there are up to three groups; each group has different QoS requirements. The maximum delay difference between any two users belonging to the same group follows a uniform distribution in the range of [4-11] msec and the maximum delay of the group is selected randomly between 25 and 40 msec.

The merits of the ILP and linear relaxation-based solutions are compared with two greedy approaches, as shown in [35, 29, 19]:

1. The distance-based approach in which users close to their overloaded local server offload their tasks to it and the remaining users offload their tasks to their closest available underloaded remote servers (i.e., the closest ones that can still satisfy both of their group's QoS requirements)

2. The resource-based approach in which users offload tasks to the overloaded local server then to underloaded remote servers in descending order of their computing capacities while satisfying the group's QoS requirements.

We solve two greedy baseline schemes using heuristic allocation. Moreover, we repeat the subsequent experiments 1000 times to compute the 95% confidence intervals[1].

4.4.3 Results and Discussion

In Figure 4.2 and Figure 4.3, we plot the average response time of the integer linear programming (ILP) and the linear relaxation-based (LR) solutions as the number of users increases. In Figure 4.2, the ILP and the LR solutions have almost the same average response time of all users. While the worst case of the average response time of the LR solution can achieve up to 5% more average response time, than the

[1] Since the confidence intervals are small, they are not explicitly depicted. The values of confidence intervals are around 3%

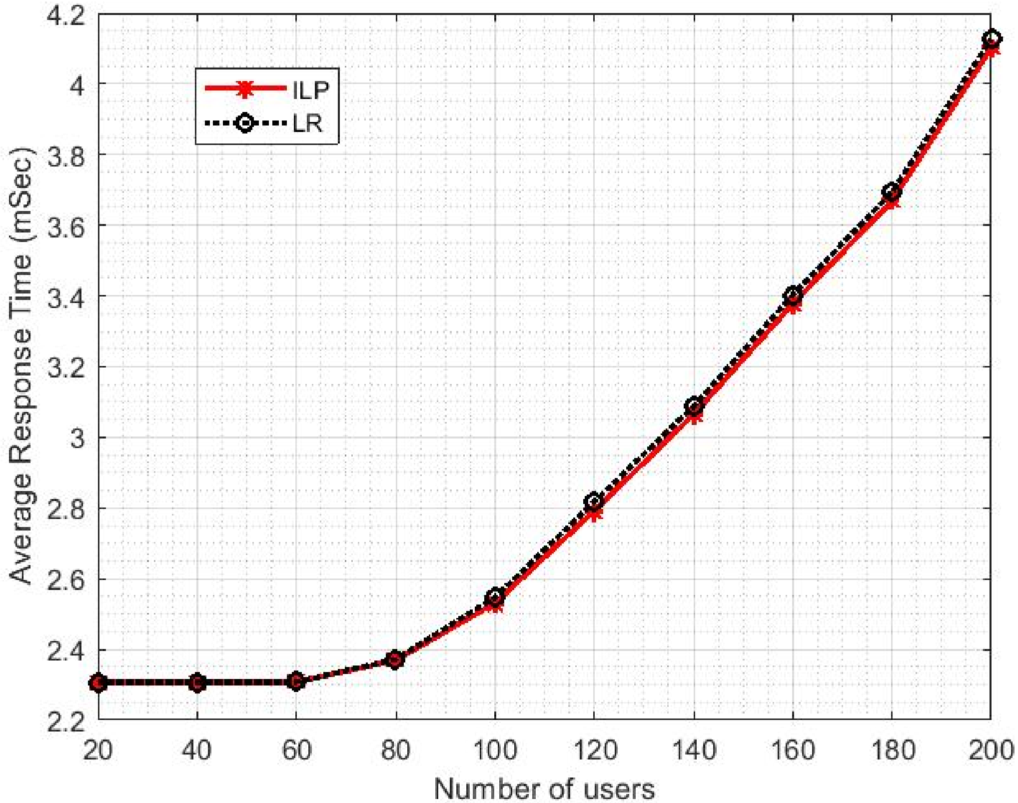

Figure 4.2: Average response time of integer linear programming (ILP) vs. linear relaxation-based (LR) in the case of all users

ILP solution of users whose tasks are assigned to the remote edge servers when the number of users equals 100, as shown in Figure 4.3. Therefore, we compare the linear relaxation-based solution with distance-based and resource-based greedy approaches in the following simulation results. Figure 4.4 shows the simulation time of the ILP and the LR solutions as the number of users increased. The ILP solution is more complex than the LR solution when more users (100 or more) are involved. The LR solution saves up to 89% of the simulation time compared to the ILP when the number of users equals 140.

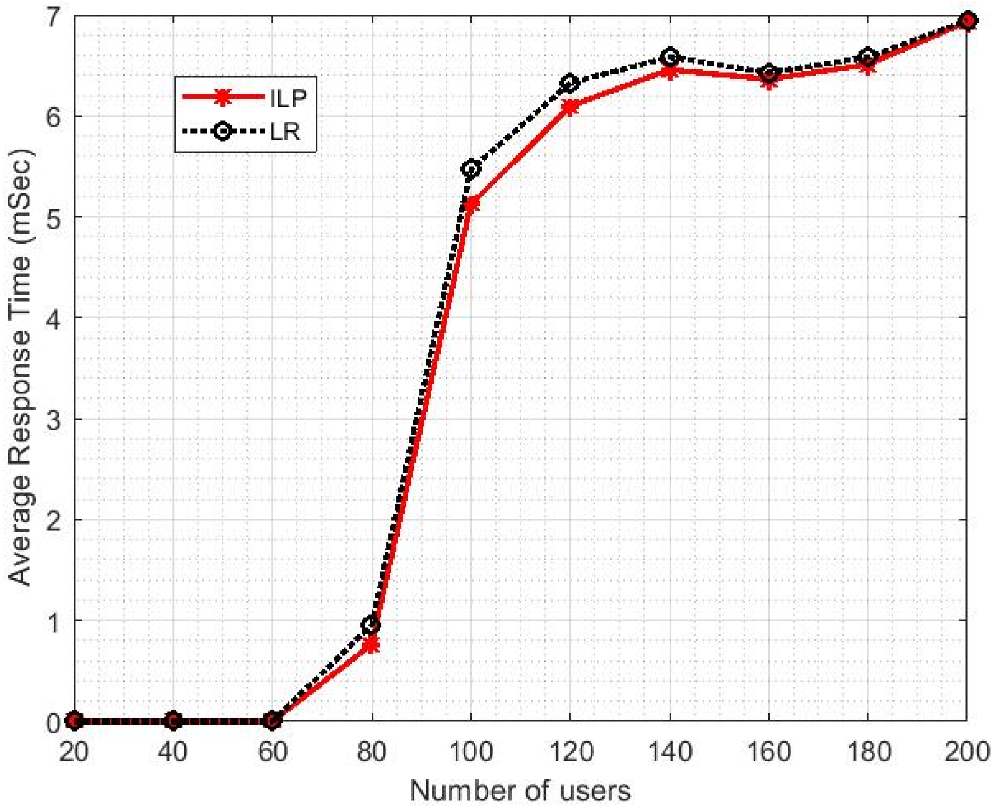

Figure 4.3: Average response time of integer linear programming (ILP) vs. linear relaxation-based (LR) in the case of users whose tasks are assigned to remote servers

Figure 4.5 and Figure 4.6 show that the LR solution outperforms the two greedy approaches in the average response time for a larger number of users, namely 100 users and above. As for the smaller number of users, figures show that the LR scheme does not provide performance gains compared to the two schemes, which can be explained by the fact that most users in this scenario will execute their tasks at their local servers, and only a very small number of users will be assigned to perform their tasks at remote servers. Thus, the impact of such assignment using the three schemes is quite close, unlike the larger number of users.

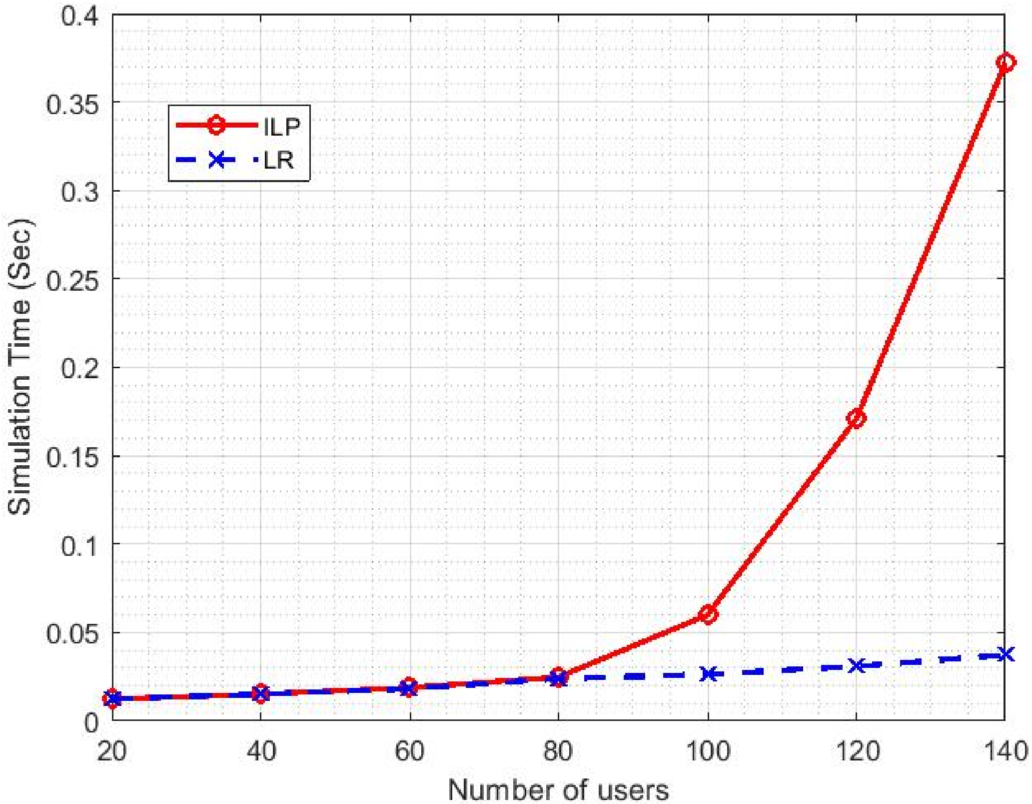

Figure 4.4: Simulation time of integer linear programming (ILP) vs. linear relaxation-based (LR)

In Figure 4.5, the LR solution can achieve 11% and 12% response time reduction over the solutions achieved by the distance-based and resource-based greedy schemes, respectively, for all users when the number of users equals 200. While the LR solution can achieve up to 9% and 10% response time reduction over the solutions achieved by distance-based and resource-based greedy schemes, respectively, for users whose tasks are assigned to remote edge servers when the number of users equals 200, as shown in Figure 4.6.

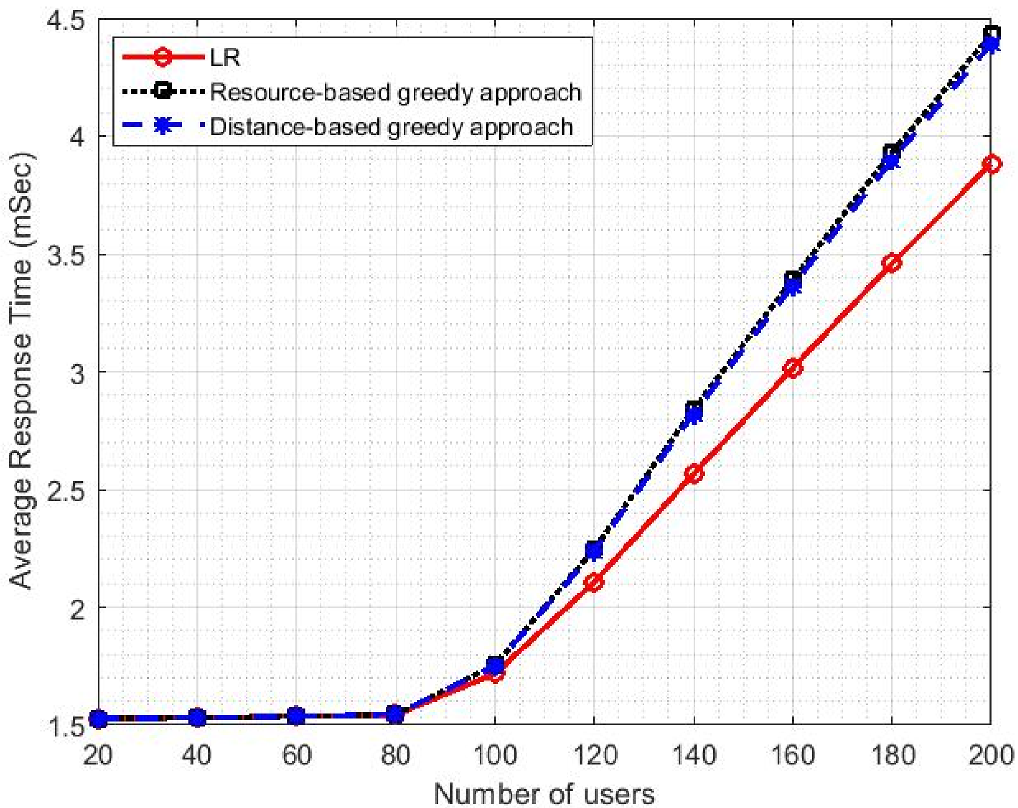

Figure 4.5: Average response time vs. the number of users in the case of all users

Figure 4.7 displays the satisfaction ratio for our proposed LR solution compared with the distance-based and resource-based greedy baseline solutions when the number of users varies from 20 to 200. The distance-based and resource-based greedy approaches do not meet all group's QoS requirements in terms of the delay difference (i.e., the delay difference between two users belonging to the same group does not satisfy the value d_{G_g} for a fraction of the users). This degradation becomes more evident for the larger number of users, in which case it is harder for these algorithms to satisfy such requirements. On the other hand, our proposed LR solution achieves this group's QoS delay difference requirements for all users.

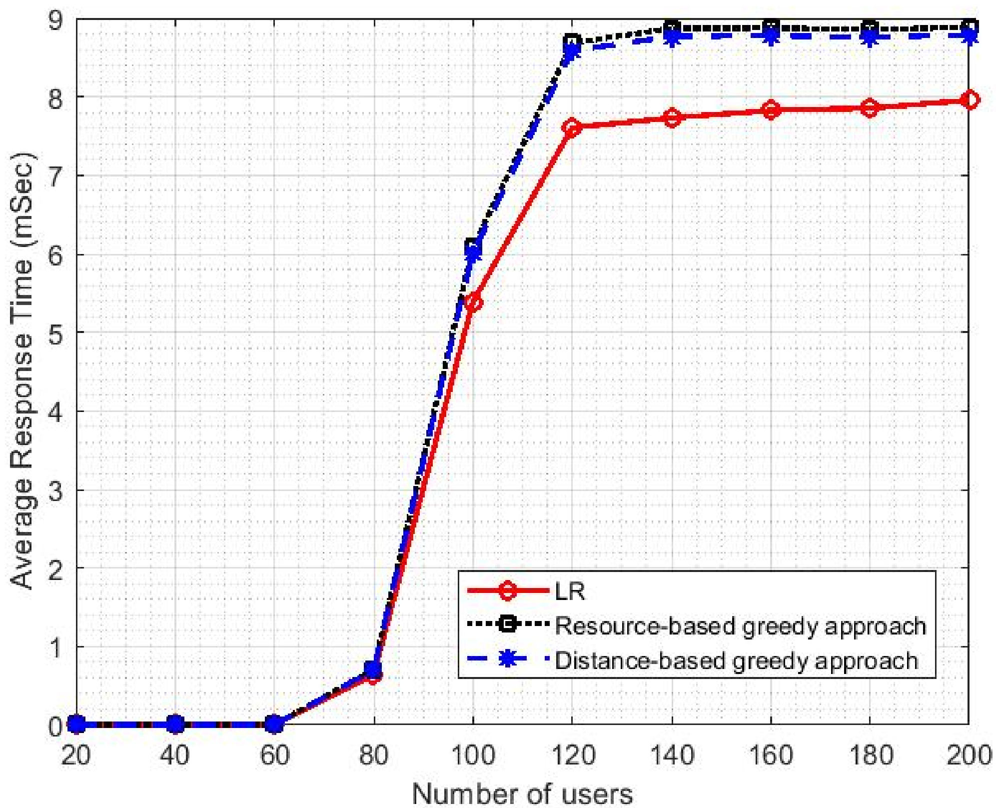

Figure 4.6: Average response time vs. the number of users in the case of users whose tasks are assigned to remote servers

4.5 Summary

In this chapter, we proposed the task offloading with service replications scheme in scalable MEC environments for groups of users involved in similar multimedia-intense applications. By scalable MEC, we mean MEC systems that allow local overloaded servers, which cannot immediately admit the tasks of all their local users, to replicate the services of the un-served users on remote underloaded servers so that these users get their tasks executed on them. Our study aimed to minimize the average response time of all users' offloaded tasks while respecting the application-imposed delays and

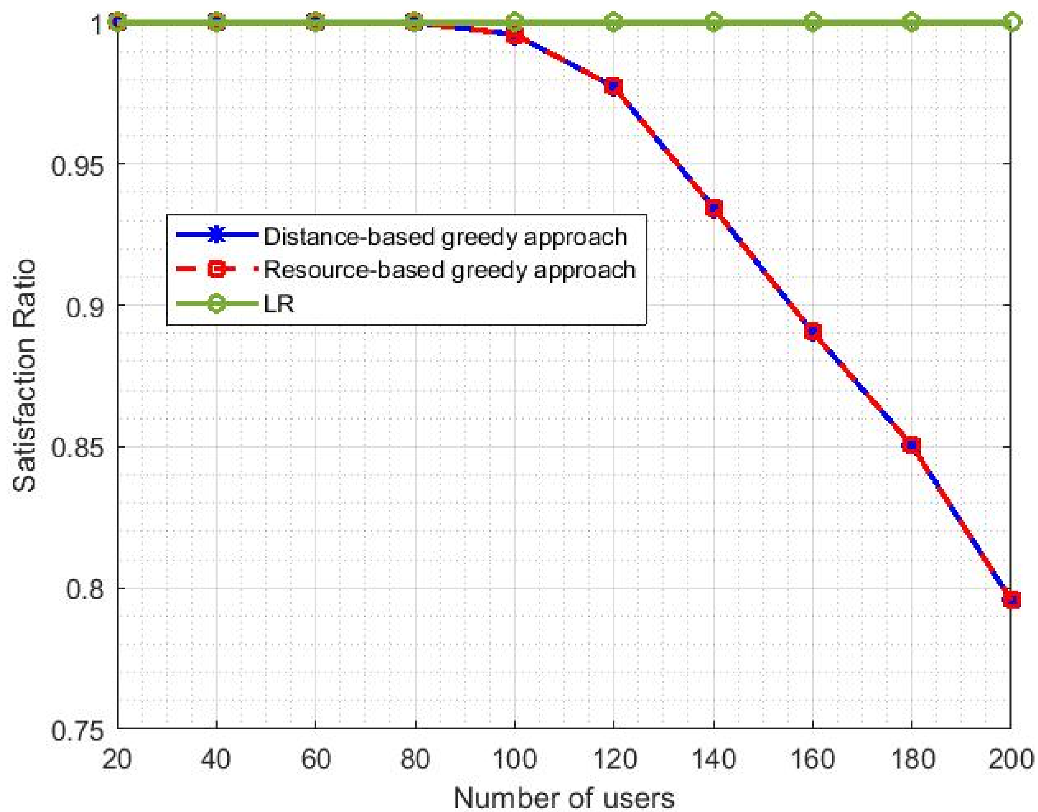

Figure 4.7: Satisfaction ratio vs. the number of users

delay difference requirements for each group of users involved in the same application. The problem was formulated as an integer linear program and was solved numerically using the MATLAB Optimization Toolbox. To reduce complexity, we proposed a linear relaxation-based solution, derived its optimal solution of its relaxed linear problem using Lagrangian analysis and KKT conditions, solved numerically using the MATLAB Optimization Toolbox, and presented a greedy rounding algorithm to restore binary values for the decision variables. Simulation results show that this linear relaxation-based solution both achieves almost the same performance as the ILP solution and outperforms the distance-based and resource-based greedy schemes that respect our model's scalability and group requirement settings. Our scheme achieved more gains than the two greedy schemes as the number of users increases. Our proposed scheme observed up to 11% and 12% compared to the distance-based and resource-based greedy approaches, respectively. Compared to the ILP solution, our proposed solution achieved up to 89% reduction in run time.

Chapter 5

Scalable MEC from the Perspective of Service Providers

In Chapter 4, we formulated our group-delay aware task offloading and service replication problem from the users' perspective. Now we look at scalable MEC from the service provider's perspective. Home service providers will absorb the costs for directing their un-served users to get service from other foreign service providers when they are overloaded while satisfying the groups' QoS requirements.

In this chapter, we address these detailed research questions:

1. What is the best task offloading distribution with cost-aware knowledge to minimize the home service provider's cost?

2. What is the best trade-off between the users' average response times and the home service provider's cost?

In this chapter, we dynamically manage task offloading for groups of users requesting to perform inter-related computations between different service providers, home service providers and foreign service providers based on service providers' predefined cost information. We formulate a constrained multi-objective optimization problem that minimizes the average response time of all users and the charged fee to foreign service providers while satisfying the time and time difference constraints and requirements of user groups running the same applications. To solve our constrained multi-objective problem in the centralized allocation, we use the Lexicographic and weight-sum methods. We deal with the linear programming (LP) relaxation of our problem because it saves time complexity compared with the integer linear programming (ILP). Using the LP relaxation problem, we derive the optimal decision variables using Lagrangian analysis. We solve the proposed LP problem numerically and apply a proposed rounding algorithm to reach the integer values of the decision variables. We compare the LP relaxation solution of Lexicographic Method and weight-sum method the LP relaxation solutions of distance-based and resource-based greedy approaches. We formulate a user service replication allocation (USRA) game to solve our problem in a distributed manner to minimize the weight-sum of users' average response time and the charged cost to the foreign service providers from the home service provider's perspective. We propose the distributed allocation algorithm for finding the Nash equilibrium in the USRA game. We compare the centralized and distributed solution experimentally.

The rest of this chapter is organized as follows; Section 5.1 details our system model and parameters. Section 5.2 presents definitions of objective variables. Section 5.3 presents the centralized solution, and Section 5.4 describes the solution using linear relaxation for the centralized allocation. Section 5.5 discusses the problem using a distributed solution. We figure out the performance evaluation and compare it with the two optimized distance-based and resource-based approaches in Section 5.6. We conclude our findings in Section 5.7.

5.1 System Model

Figure 5.1 depicts the system model. The detail of our system model is described in Chapter 3, where we consider that the MEC system consists of users set, M_h home service providers, and M_f foreign service providers, as shown in Figure 3.6. In our system, we assume the following:

- Each service provider manages a cluster of edge servers, these edge servers are collocated together with the same base station.
- Home service providers have demands that exceed their capacities, thus the un-served users of home service providers are allowed to access foreign service providers' resources after services of the home service providers have been replicated to foreign service providers.
- Home service providers always provide required services to their users without any additional cost unless overloaded. Thus, foreign service providers will invoice home service providers for serving their un-served users.

Given this system description, we formulate our cost and group-delay-aware task offloading and service replication problem between multiple service providers in the next section. This problem aims to minimize the average response time of all users and minimize the cost. Home service providers will pay foreign service providers when users have no choice but to send their task to foreign service providers while satisfying all the QoS requirements of the group's users.

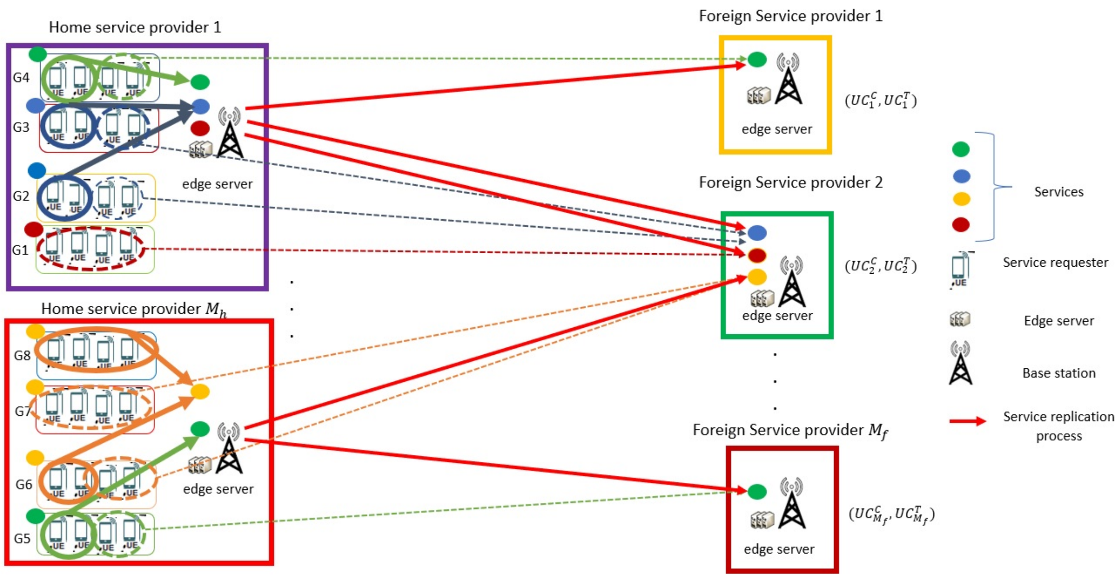

Figure 5.1: System model from the service providers' perspective

5.2 Problem Setup

Each foreign service provider S_i^f, $S_i^f \in \mathcal{S}^f$ has two-unit costs: $UC_i^C(\frac{\$}{Mcycle})$, $UC_i^T(\frac{\$}{MByte})$, which are the unit cost of the computation rate, and the cost of the transmission rate. In addition, $UC_g^R(\$)$ is a group-dependent parameter, denoting the unit cost of replicating the service of the group G_g on the foreign service providers.

5.2.1 Definition of Objective Variables

Let x_{ui}, $\forall\ u,\ i$ be a task offloading indicator, where u and i represent User U_u, $U_u \in \mathcal{U}$, and Service Provider S_i^p, $S_i^p \in \mathcal{S}^p$, respectively. In our formulation, x_{ui} is set to 1 when User U_u offloads its task to Service Provider S_i^p and is set to 0 otherwise. Let b_{ig}, $\forall\ i,\ g$ be a replication indicator, where i and g represent Foreign Service Provider S_i^f, $S_i^f \in \mathcal{S}^f$, and the group ID G_g, $G_g \in \mathcal{G}$, respectively. In our formulation, b_{ig} is set to 1 when at least one user of Group G_g takes a service from Foreign Service Provider

S_i^f and is set to 0 otherwise.

5.3 Centralized Allocation

This section looks at our problem from the centralized allocation for user tasks between multiple service providers.

5.3.1 Problem Formulation

Given the above definition of two objective variables, $x_{ui},\ b_{ig}$, the task offloading problem to achieve our goal can be expressed as a constrained multi-objective optimization problem, which is:

$$\min_{x_{ui},\ b_{ig}} \{RT, C_{SP}\} \tag{5.1}$$

$$S.T. \qquad \sum_{i \in \mathcal{S}^p} x_{ui} = 1, \qquad \forall\ U_u \in \mathcal{U}, \tag{5.2}$$

$$\sum_{u \in \mathcal{U}^j} x_{ui} f_u \le F^h_{S^h_j}, \qquad \forall\ S^h_j \in \mathcal{S}^h \tag{5.3}$$

$$\sum_{u=1}^{N} x_{ui} f_u \le F^f_{S^f_i}, \qquad \forall\ S^f_i \in \mathcal{S}^f \tag{5.4}$$

$$\sum_{i \in \mathcal{S}^p} x_{ui} RT_{ui} \le T^{max}_{G_g}, \quad \forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g} \in \mathcal{U}^{G_g} \tag{5.5}$$

$$\sum_{i \in \mathcal{S}^p} x_{ui} RT_{ui} - \sum_{i \in \mathcal{S}^p} x_{li} RT_{li} \le d_{G_g}, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U_l^{G_g}, U_u^{G_g} \in \mathcal{U}^{G_g},\ l \ne u \tag{5.6}$$

$$\frac{\sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui}}{N_g} - b_{ig} \le 0, \quad \forall\ G_g \in \mathcal{G},\ \forall\ S^f_i \in \mathcal{S}^f \tag{5.7}$$

$$b_{ig} - \sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui} \le 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ S^f_i \in \mathcal{S}^f \tag{5.8}$$

$$x_{ui} \in \{0,1\},\ \forall\ S^p_i \in \mathcal{S}^p,\ \forall\ U_u \in \mathcal{U} \tag{5.9}$$

$$b_{ig} \in \{0,1\},\ \forall\ S^f_i \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G} \tag{5.10}$$

where RT represents the average response time and C_{SP} represents the total cost defined in (5.11) and (5.12), respectively.

$$RT = \frac{1}{N} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{S}^p} x_{ui} RT_{ui} \tag{5.11}$$

$$C_{SP} = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{S}^f} C_i^S * x_{ui} + \sum_{i \in \mathcal{S}^f} \sum_{g \in \mathcal{G}} (b_{ig} * UC_g^R) \tag{5.12}$$

where $C_i^S = (C_u \times UC_i^C) + (B_u \times UC_i^T)$ is the service cost (computation plus transmission). The solution of the constrained multi-objective problem is characterized by a Pareto front formed by a set of possible solutions. Each solution represents a different trade-off between the average response time of all users and the cost to the home service providers [102, 103, 104].

Clearly, (5.1) is a multi-objective optimization function that minimizes the average response time of all users' tasks and home service providers' cost by determining the best service provider to which each User U_u should offload their task(s). Constraint (5.2) ensures that each task is offloaded to only one service provider. Constraint (5.3) ensures that the sum of assigned computing capacities to the subset of users offloading their tasks to their home service provider's MEC server does not exceed its total available computing capacity. Constraint (5.4) ensures that the sum of assigned computing capacities to all users offloading their tasks to the foreign service provider's MEC server does not exceed its total available computing capacity. Constraints (5.3) and (5.4) ensure that the maximum computational capabilities of the service providers' edge servers are not exceeded. Constraint (5.5) ensures that the total response time of a user's task belonging to any Group $G_g \in \mathcal{G}$ does not exceed this service's maximum delay requirement. Constraint (5.6) ensures that the delay difference between two users belonging to the same Group G_g does not exceed the predefined difference delay requirement. Constraints (5.5) and (5.6) are to obeyed to the QoS demanded by each group. Constraints (5.7) and (5.8) ensure that the group's service is deployed on the edge server of the foreign service provider (i.e., ensuring that $b_{ig} = 1$, if there is at least one user $U_u^{G_g}$ of Group G_g that accesses its service at foreign Service Provider S_i^f, $x_{ui} = 1$). Constraint (5.9) and (5.10) impose a binary decision value for the variable $x_{ui},\ b_{ig}$ of all users and services, respectively.

It is clear that the optimization problem is an integer linear multi-objective problem. To solve this problem, we relax the ILP optimization problem into linear programming (LP) by relaxing the integer decision variables, deriving analytical and solver-based solutions for the relaxed decision variables, and restoring the integer decision variables using a greedy rounding approach. Next, we compare this solution to both linear relaxation-based solutions of two greedy approaches that can satisfy the scalability and group-delay awareness settings considered. There are several mathematical methods to solve multi-objective problems [102]. We use the Lexicographic method [103, 102]; it is a unique method in which the aggregations performed are not scalar. In this method, the decision-makers rank the objectives in order of importance (best to worst). We can reach the optimum solution by minimizing the objective functions, starting with the most important one, the C_{SP}, and proceeding according to the order of importance of the objectives; in our case, the RT. Thus, the first problem is formulated as:

$$\min_{x_{ui},\ b_{ig}} \quad f_1 = C_{SP} \tag{5.13}$$

$$S.T. \qquad (5.2 - 5.10) \tag{5.14}$$

and its solution x^*_{ui}, b^*_{ig} and $f^*_1 = f_1(x^*_{ui}, b^*_{ig})$ is obtained.

Then the second problem is formulated as:

$$\min_{x_{ui},\ b_{ig}} \quad f_2 = RT \tag{5.15}$$

$$S.T. \qquad (5.2 - 5.10) \tag{5.16}$$

$$f_1 = f^*_1 \tag{5.17}$$

The solution obtained at the end is taken as the desired solution to the problem. Next, we solve two previous sub-problems (5.13) and (5.15) using a numerical solver. To assess the performance of the Lexicographic method, we use the weight-sum method [103] to convert the multi-objective problem into a scalar optimization problem in which the objective function is a weighted sum of the components of the original multi-objective function.

$$\min_{x_{ui},\ b_{ig}} \quad w\ \frac{C_{SP}}{C_{SP_{Max}}} + \frac{(1-w)}{N} \sum_{u=1}^{N} \sum_{i \in \mathcal{S}^p} RT_{ui_{norm}} x_{ui} \tag{5.18}$$

$$S.T. \qquad (5.2 - 5.10) \tag{5.19}$$

where $0 \le w \le 1$ is a scalar weighting factor associated with the objective function for all home service providers, and

$$RT_{ui_{norm}} = \frac{RT_{ui} - RT_{Min}}{RT_{Max} - RT_{Min}}$$

This weighting coefficient does not prioritize the objectives, but it is only a factor by varying it; the non-inferior solutions will change. From (5.18), we normalize the values of the objective functions using max-min normalization as follows:

$$I_{new} = \frac{I - Min}{Max - Min}$$

5.4 Solution Using Linear Relaxation

This section relaxes the previous optimization problem (5.18) into a linear programming (LP) formulation. The objective function and constraints are similar to the integer linear problem (5.1) except constraints (5.9) and (5.10), which are replaced by $0 \le x_{ui},\ b_{ig} \le 1$.

5.4.1 Analytical Solutions

The linear problem version of (5.18) is a convex optimization problem. The Karush-Kuhn-Tucker (KKT) conditions provide necessary and sufficient conditions for the optimal solution [97]. Therefore, applying the KKT conditions to the constraints of the problem, and the gradient of the Lagrangian function allows us to find the analytical solution of decision variables $x_{ui},\ b_{ig}$. The Lagrangian function associated with the linear optimization problem is given by (5.20). Whereas X is the vector of offloading decisions (i.e., $X = [x_{ui}]$), BG is the vector of placement replication decisions (i.e., $BG = [b_{ig}]$, u refers to User $U_u,\ \forall\ U_u \in \mathcal{U}$, and i refers to Service Provider $S_i^p,\ \forall\ S_i^p \in \mathcal{S}^p$, g refers to Group $G_g,\ \forall\ G_g \in \mathcal{G}$, and:

- $\lambda_{home} = [\lambda_{home_j}],\ \lambda_{foreign} = [\lambda_{foreign_i}],\ \alpha_p = [\alpha_u],\ \sigma_1 = [\sigma_{1_{ig}}],\ \sigma_2 = [\sigma_{2_{ig}}],\ \theta = [\theta_{gul}]$ are the associated Lagrange multiplier to inequality constraints (5.3), (5.4), (5.5), (5.7), (5.8), (5.6), respectively.

- $\beta = [\beta_{ui}],\ \gamma = [\gamma_{ui}]$ are the associated Lagrange multiplier to the lower bound and upper bound inequality constraint of x_{ui} and respectively.

- $\mu = [\mu_{ig}]$, $\eta = [\eta_{ig}]$ are the associated Lagrange multiplier to the lower bound and upper bound inequality constraint of b_{ig} and respectively.
- $\nu = [\nu_u]$ is the associated Lagrange multiplier to the equality constraint (5.2).

$$
\begin{aligned}
&L(X, B, \lambda_{home}, \lambda_{foreign}, \alpha_p, \theta, \sigma_1, \sigma_2, \beta, \gamma, \mu, \eta, \nu) = \\
&\quad \frac{w}{C_{SP_{Max}}}\left(\sum_{i\in\mathcal{S}^f}\sum_{u=1}^{N}(C_i^S * x_{ui}) + \sum_{i\in\mathcal{S}^f}\sum_{g=1}^{NG}(b_{ig} * UC_g^R)\right) + \frac{(1-w)}{N}\sum_{i\in\mathcal{S}^p}\sum_{u=1}^{N}(RT_{ui_{norm}}x_{ui}) \\
&\quad + \sum_{i\in\mathcal{S}^h}\lambda_{home_i}\left(\sum_{u\in\mathcal{U}^j}(f_u x_{ui}) - F^h_{S_j^h}\right) + \sum_{i\in\mathcal{S}^f}\lambda_{foreign_i}\left(\sum_{u=1}^{N}(f_u x_{ui}) - F^f_{S_i^f}\right) \\
&\quad + \sum_{u=1}^{N}\alpha_u\left(\sum_{i\in\mathcal{S}^p}(RT_{ui}x_{ui}) - T^{max}_{G_g}\right) + \sum_{i\in\mathcal{S}^f}\sum_{g=1}^{NG}\sigma_{1_{ig}}\left(\frac{\sum_{u\in\mathcal{U}^{G_g}} x_{ui}}{N_g} - b_{ig}\right) \\
&\quad + \sum_{i\in\mathcal{S}^f}\sum_{g=1}^{NG}\sigma_{2_{ig}}\left(b_{ig} - \sum_{u\in\mathcal{U}^{G_g}} x_{ui}\right) - \sum_{u=1}^{N}\sum_{i\in\mathcal{S}^p}\beta_{ui}x_{ui} + \sum_{u=1}^{N}\sum_{i\in\mathcal{S}^p}\gamma_{ui}(x_{ui}-1) \\
&\quad + \sum_{g=1}^{NG}\sum_{u\in\mathcal{U}^{G_g}}\sum_{l\in\mathcal{U}^{G_g}, l\neq u}\theta_{gul}\left((\sum_{i\in\mathcal{S}^p} x_{ui}RT_{ui}) - (\sum_{i\in\mathcal{S}^p} x_{li}RT_{li}) - d_{G_g}\right) \\
&\quad - \sum_{i\in\mathcal{S}^f}\sum_{g=1}^{NG}\mu_{ig}b_{ig} + \sum_{i\in\mathcal{S}^f}\sum_{g=1}^{NG}\eta_{ig}(b_{ig}-1) + \sum_{u=1}^{N}\nu_u\left(\sum_{i\in\mathcal{S}^p} x_{ui} - 1\right) \\
&\qquad\qquad\qquad \forall\ S_i^h \in \mathcal{S}^h,\ \forall\ S_i^f \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g}, \in \mathcal{U}^{G_g}
\end{aligned}
\tag{5.20}
$$

KKT Conditions

The following equations are derived by applying the KKT conditions on the equality and inequality constraints of the linear problem version of (5.18), we get:

- Primal feasibility conditions:

$$\sum_{i \in \mathcal{S}^p} x^*_{ui} = 1, \qquad \forall\ u \in \mathcal{U}. \tag{5.21a}$$

$$\sum_{u \in \mathcal{U}^j} x^*_{ui} f_u \le F^h_{S_j}, \qquad \forall\ S^h_j \in \mathcal{S}^h. \tag{5.21b}$$

$$\sum_{u=1}^{N} x^*_{ui} f_{ui} \le F^f_{S^f_i}, \qquad \forall\ S^f_i \in \mathcal{S}^f. \tag{5.21c}$$

$$\sum_{i \in \mathcal{S}^p} x^*_{ui} RT_{ui} \le T^{max}_{G_g}, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u \in \mathcal{U}^{G_g}. \tag{5.21d}$$

$$\frac{\sum_{U^{G_g}_u \in \mathcal{U}^{G_g}} x^*_{ui}}{N_g} - b^*_{ig} \le 0,\ \forall\ G_g \in \mathcal{G},\ \forall\ S^f_i \in \mathcal{S}_{foreign}. \tag{5.21e}$$

$$b^*_{ig} - \sum_{U^{G_g}_u \in \mathcal{U}^{G_g}} x^*_{ui} \le 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ S^f_i \in \mathcal{S}^f. \tag{5.21f}$$

$$\sum_{i \in \mathcal{S}^p} x^*_{ui} RT_{ui} - \sum_{i \in \mathcal{S}^p} x^*_{li} RT_{li} \le d_{G_g}, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_l \in \mathcal{U}^{G_g},\ l \ne u \tag{5.21g}$$

$$-x^*_{ui} \le 0, \quad \forall\ U_u \in \mathcal{U},\ \forall\ S_i \in \mathcal{S}. \tag{5.21h}$$

$$x^*_{ui} - 1 \le 0, \quad \forall\ U_u \in \mathcal{U},\ \forall\ S_i \in \mathcal{S} \tag{5.21i}$$

$$-b^*_{ig} \le 0, \quad \forall\ S^f_i \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G} \tag{5.21j}$$

$$b^*_{ig} - 1 \le 0, \quad \forall\ S^f_i \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G}. \tag{5.21k}$$

- Complementary slackness conditions:

$$\lambda^*_{home_j}\left(\sum_{u \in \mathcal{U}^j} x^*_{ui} f_u - F^h_{S^h_j}\right) = 0, \qquad \forall\ S^h_j \in \mathcal{S}^h. \tag{5.22a}$$

$$\lambda^*_{foreign_i}\left(\sum_{u=1}^{N} x^*_{ui} f_u - F^f_{S^f_i}\right) = 0, \qquad \forall\ S^f_i \in \mathcal{S}^f. \tag{5.22b}$$

$$\alpha_u^* \left(\sum_{i \in \mathcal{S}^p} x_{ui}^* RT_{ui} - T_{G_g}^{max} \right) = 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g} \in \mathcal{U}^{G_g}. \tag{5.22c}$$

$$\sigma_{1_{ig}}^* \left(\frac{\sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui}^*}{N_g} - b_{ig}^* \right) = 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ S_i^f \in \mathcal{S}^f. \tag{5.22d}$$

$$\sigma_{2_{ig}}^* \left(b_{ig}^* - \sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui}^* \right) = 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ S_i^f \in \mathcal{S}^f. \tag{5.22e}$$

$$\theta_{gul}^* \left(\sum_{i \in \mathcal{S}^p} x_{ui}^* RT_{ui} - \sum_{i \in \mathcal{S}^p} x_{li}^* RT_{li} - d_{G_g} \right) = 0,$$

$$\forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g},\ U_l^{G_g} \in \mathcal{U}^{G_g},\ l \neq u. \tag{5.22f}$$

$$-\beta_{ui}^* x_{ui}^* = 0, \qquad \forall\ U_u \in \mathcal{U},\ \forall\ S_i \in \mathcal{S}. \tag{5.22g}$$

$$\gamma_{ui}^* \left(x_{ui}^* - 1 \right) = 0, \qquad \forall\ U_u \in \mathcal{U},\ \forall\ S_i \in \mathcal{S}. \tag{5.22h}$$

$$-\mu_{ig}^* b_{ig}^* = 0, \qquad \forall\ S_i^f \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G}. \tag{5.22i}$$

$$\eta_{ig}^* \left(b_{ig}^* - 1 \right) = 0, \qquad \forall\ S_i^f \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G}. \tag{5.22j}$$

- Dual feasibility conditions:

$$\lambda_{home_j}^*,\ \lambda_{foreign_i}^*,\ \alpha_u^*,\ \theta_{gul}^*,\ \sigma_{1_{ig}}^*,\ \sigma_{2_{ig}}^*,\ \beta_{ui}^*,\ \gamma_{ui}^*,\ \mu_{ig}^*\ and\ \eta_{ig}^* \geq 0.$$

$$\forall\ S_i^p \in \mathcal{S}^p,\ \forall\ G_g \in \mathcal{G},\ \forall\ U_u \in \mathcal{U} \tag{5.23a}$$

$$\frac{\partial L(X^*, BG^*, \lambda_{home_j}^*, \ldots, \nu^*)}{\partial x_{ui}^*} = 0,$$

$$\frac{1-w}{N} RT_{ui_{norm}} + \lambda_{home_j}^* f_u + \alpha_u^* RT_{ui} + \theta_{gul}^* RT_{ui} - \beta_{ui}^* + \gamma_{ui}^* + \nu_u^* = 0.$$

$$\forall\ S_i^h \in \mathcal{S}^h,\ \forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g},\ U_l^{G_g} \in \mathcal{U}^{Gg},\ l \neq u \tag{5.23b}$$

$$\frac{\partial L(X^*, BG^*, \lambda^*_{home_j}, \ldots, \nu^*)}{\partial x^*_{ui}} = 0,$$

$$\frac{w}{C_{SP_{Max}}} C_{S_i} + \frac{1-w}{N} RT_{ui_{norm}} + \lambda^*_{foreign_i} f_u + \alpha^*_u RT_{ui} + \theta^*_{gul} RT_{ui} + \left(\frac{\sigma^*_{1_{ig}}}{N_g}\right)$$

$$- \sigma^*_{2_{ig}} - \beta^*_{ui} + \gamma^*_{ui} + \nu^*_u = 0$$

$$\forall\ S^f_i \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_l \in \mathcal{U}^{Gg},\ l \neq u \tag{5.23c}$$

$$\frac{\partial L(X^*, BG^*, \lambda^*_{home_j}, \ldots, \nu^*)}{\partial x^*_{li}} = 0,$$

$$- \theta_{gul} RT_{li} = 0$$

$$\forall\ S^h_j \in \mathcal{S}^h,\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_l \in \mathcal{U}^{Gg},\ l \neq u \tag{5.23d}$$

$$\frac{\partial L(X^*, BG^*, \lambda^*_{home_j}, \ldots, \nu^*)}{\partial b^*_{ig}} = 0,$$

$$\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1_{ig}} + \sigma^*_{2_{ig}} - \mu^*_{ig} + \eta^*_{ig} = 0$$

$$\forall\ S^f_i \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G},\ \forall\ U^{G_g}_u,\ U^{G_g}_l \in \mathcal{U}^{Gg},\ l \neq u \tag{5.23e}$$

After applying the KKT conditions on the equality and inequality constraints, the following theorem illustrates the optimal solution of the linear problem.

Theorem 5.1: The optimal offloading decisions of the linear optimization problem can be expressed as shown in (5.24), (5.25), and (5.26).

When $\forall\ S^h_i \in \mathcal{S}^h,\ \forall\ U^{G_g}_u \in \mathcal{U}^{G_g},\ \forall\ G_g \in \mathcal{G}$,

$$x^*_{ui} = \begin{cases} 0, & if\ \gamma^*_{ui} = 0,\ \beta^*_{ui} > 0,\ \frac{1-w}{N} RT_{ui_{norm}} + \lambda^*_{home_j} f_u + \alpha^*_u RT_{ui} + \nu^*_u \neq 0 \\ 1, & if\ \gamma^*_{ui} > 0,\ \beta^*_{ui} = 0,\ \frac{1-w}{N} RT_{ui_{norm}} + \lambda^*_{home_j} f_u + \alpha^*_u RT_{ui} + \nu^*_u = -\gamma^*_{ui} \end{cases} . \tag{5.24}$$

Otherwise, when $\forall\ S_j^f \in \mathcal{S}^f,\ \forall\ U_u^{G_g} \in \mathcal{U}^{G_g},\ \forall\ G_g \in \mathcal{G}$,

$$x_{ui}^* = \begin{cases} 0, & if\ \gamma_{ui}^* = 0,\ \beta_{ui}^* > 0, and \\ & \frac{w}{C_{SP_{Max}}} C_i^S + \frac{1-w}{N} RT_{ui_{norm}} + \lambda_{foreign_i}^* f_{ui} + \alpha_u^* RT_{ui} + \left(\frac{\sigma_{1_{ig}}^*}{N_g}\right) - \sigma_{2_{ig}}^* + \nu_u^* \neq 0 \\ 1, & if\ \gamma_{ui}^* > 0,\ \beta_{ui}^* = 0, and \\ & \frac{w}{C_{SP_{Max}}} C_i^S + \frac{1-w}{N} RT_{ui_{norm}} + \lambda_{foreign_i}^* f_{ui} + \alpha_u^* RT_{ui} + \left(\frac{\sigma_{1_{ig}}^*}{N_g}\right) - \sigma_{2_{ig}}^* + \nu_u^* = -\gamma_{ui}^* \end{cases} \tag{5.25}$$

and,

$$b_{ig}^* = \begin{cases} 0, & if\ \eta_{ui}^* = 0,\ \mu_{ui}^* > 0,\ \left(\frac{w}{C_{SP_{Max}}} UC_g^R - \sigma_{1_{ig}}^* + \sigma_{2_{ig}}^*\right) \neq 0 \\ 1, & if\ \eta_{ui}^* > 0,\ \mu_{ui}^* = 0,\ \left(\frac{w}{C_{SP_{Max}}} UC_g^R - \sigma_{1_{ig}}^* + \sigma_{2_{ig}}^*\right) = -\eta_{ui}^* \\ \frac{\sum_{u \in \mathcal{U}^{G_g}} x_{ui}^*}{N_g}, & if\ \eta_{ui}^* = 0,\ \mu_{ui}^* = 0,\ \left(\frac{w}{C_{SP_{Max}}} UC_g^R - \sigma_{1_{ig}}^* + \sigma_{2_{ig}}^*\right) = 0,\ \sigma_{1_{ig}}^* > 0, \\ & , \sigma_{2_{ig}}^* = 0. \qquad \forall\ S_i^f \in \mathcal{S}^f,\ \forall\ U_u^{G_g} \in \mathcal{U}^{G_g},\ \forall\ G_g \in \mathcal{G}. \end{cases} \tag{5.26}$$

Proof. Using the equations we derived using KKT conditions, we can see from (5.23d) that $RT_{li} \neq 0$ and $\theta_{gul}^* \geq 0$, and thus $\theta_{gul}^* = 0$. By substituting in (5.23b) and (5.23c), we get:

$$\frac{1-w}{N} RT_{ui_{norm}} + \lambda_{home_j}^* f_u + \alpha_u^* RT_{ui} - \beta_{ui}^* + \gamma_{ui}^* + \nu_u^* = 0.$$
$$\forall\ S_i^f \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g},\ U_l^{G_g} \in \mathcal{U}^{G_g},\ l \neq u. \tag{5.27}$$

$$\frac{w}{C_{SP_{Max}}} C_i^S + \frac{1-w}{N} RT_{ui_{norm}} + \lambda_{foreign_i}^* f_u + \alpha_u^* RT_{ui} + \left(\frac{\sigma_{1_{ig}}^*}{N_g}\right) - \sigma_{2_{ig}}^* - \beta_{ui}^* + \gamma_{ui}^* + \nu_u^* = 0,$$
$$\forall\ S_i^f \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G},\ \forall\ U_u^{G_g},\ U_l^{G_g} \in \mathcal{U}^{G_g},\ l \neq u \tag{5.28}$$

By multiplying (5.27) by x_{ui}^*, and using (5.22g) and (5.22h), substituting in the result of multiplication, we get:

$$x^*_{ui} = \frac{-\gamma^*_{ui}}{\frac{1-w}{N}RT_{ui_{norm}} + \lambda^*_{home_j}f_u + \alpha^*_u RT_{ui} + \nu^*_u} = \frac{-\gamma^*_{ui}}{\beta^*_{ui} - \gamma^*_{ui}}$$
$$\forall\ S^h_i \in \mathcal{S}^h,\ \forall\ U_u \in \mathcal{U}^i,\ \forall\ G_g \in \mathcal{G} \quad (5.29)$$

Similarly, by multiplying (5.28) by x^*_{ui}, and using (5.22g) and (5.22h), substituting in the result of multiplication, we get:

$$x^*_{ui} = \frac{-\gamma^*_{ui}}{\frac{w}{C_{SP_{Max}}}C^S_i + \frac{1-w}{N}RT_{ui_{norm}} + \lambda^*_{foreign_i}f_{ui} + \alpha^*_u RT_{ui} + \left(\frac{\sigma^*_{1_{ig}}}{N_g}\right) - \sigma^*_{2_{ig}} + \nu^*_u} = \frac{-\gamma^*_{ui}}{\beta^*_{ui} - \gamma^*_{ui}},$$
$$\forall\ S^f_i \in \mathcal{S}^f,\ \forall\ U^{G_g}_u \in \mathcal{U}^{G_g},\ \forall\ G_g \in \mathcal{G}. \quad (5.30)$$

From (5.29), (5.22h), and (5.22g), we only have two viable cases:

- Case 1: $x^*_{ui} = 0$ when $\gamma^*_{ui} = 0$, and $\beta^*_{ui} > 0$ which means that $\left(\frac{1-w}{N}RT_{ui_{norm}} + \lambda^*_{home_j}f_u + \alpha^*_u RT_{ui} + \nu^*_u\right) > 0$.
- Case 2: $x^*_{ui} = 1$ when $\gamma^*_{ui} > 0$ and $\beta^*_{ui} = 0$ which means that $\left(\frac{1-w}{N}RT_{ui_{norm}} + \lambda^*_{home_j}f_u + \alpha^*_u RT_{ui} + \nu^*_u\right) = -\gamma^*_{ui}$.

Otherwise, when $\gamma^*_{ui} = \beta^*_{ui} = 0$, $\gamma^*_{ui} = \beta^*_{ui} > 0$, or $\gamma^*_{ui} > 0$ and $\beta^*_{ui} > 0$, x^*_{ui} has no valid value.

From (5.30), (5.22h), and (5.22g), we only have two viable cases:

- Case 1: $x^*_{ui} = 0$ when $\gamma^*_{ui} = 0$, and $\beta^*_{ui} > 0$ which means that $(\frac{w}{C_{SP_{Max}}}C^S_i + \frac{1-w}{N}RT_{ui_{norm}} + \lambda^*_{foreign_i}f_u + \alpha^*_u RT_{ui} + (\frac{\sigma^*_{1_{ig}}}{N_g}) - \sigma^*_{2_{ig}} + \nu^*_u) > 0$.

- Case 2: $x^*_{ui} = 1$ when $\gamma^*_{ui} > 0$, and $\beta^*_{ui} = 0$ which means that $(\frac{w}{C_{SP_{Max}}} C^S_i + \frac{1-w}{N} RT_{ui_{norm}} + \lambda^*_{foreign_i} f_u + \alpha^*_u RT_{ui} + (\frac{\sigma^*_{1_{ig}}}{N_g}) - \sigma^*_{2_{ig}} + \nu^*_u) = -\gamma^*_{ui}$.

Otherwise, when $\gamma^*_{ui} = \beta^*_{ui} = 0$ (because of the limitation of x^*_{ui} when γ^*_{ui} and β^*_{ui} tend to zero), $\gamma^*_{ui} = \beta^*_{ui} > 0$, or $\gamma^*_{ui} > 0$ and $\beta^*_{ui} > 0$, x^*_{ui} has no valid value.

Regarding b^*_{ig}, multiply (5.23e) by b^*_{ig}, and use (5.22i), and (5.22j), substitute in the result of multiplication, we get:

$$b^*_{ig} = \frac{-\eta^*_{ig}}{\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1_{ig}} + \sigma^*_{2_{ig}}} = \frac{-\eta^*_{ig}}{\mu^*_{ig} - \eta^*_{ig}} \tag{5.31}$$

From (5.31) and (5.22j), we only have three cases:

- Case 1: $b^*_{ig} = 0$ when $\eta^*_{ig} = 0$, which means that $\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1_{ig}} + \sigma^*_{2_{ig}} > 0$.
- Case 2: $b^*_{ig} = 1$ when $\mu^*_{ig} = 0$, which means that $\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1ig} + \sigma^*_{2_{ig}} = -\eta^*_{ig}$.
- Case 3: $0 \leq b^*_{ig} \leq 1$, when $\eta^*_{ig} = 0$ and $\mu^*_{ig} = 0$, which means that $\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1_{ig}} + \sigma^*_{2_{ig}} = 0$, We thus have four cases for b^*_{ig} based on σ^*_{1ig} and σ^*_{2ig}:
  - Case 1: When $\sigma^*_{1_{ig}} = \sigma^*_{2_{ig}} = 0$, this case is invalid because $\left(\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1_{ig}} + \sigma^*_{2_{ig}}\right) \neq 0$
  - Case 2: When $\sigma^*_{1_{ig}} = 0$, $\sigma^*_{2_{ig}} > 0$, this case is invalid because $\left(\frac{w}{C_{SP_{Max}}} UC^R_g - \sigma^*_{1_{ig}} + \sigma^*_{2_{ig}}\right) \neq 0$
  - Case 3: When $\sigma^*_{1_{ig}} > 0$, $\sigma^*_{2_{ig}} = 0$, $b^*_{ig} = \frac{\sum_{U^{G_g}_u \in \mathcal{U}^{G_g}} x^*_{ui}}{N_g}$, $\forall\ S^f_i \in \mathcal{S}^f$, $\forall G_g \in \mathcal{G}$.
  - Case 4: When $\sigma^*_{1_{ig}} > 0$ and $\sigma^*_{2_{ig}} > 0$, this case is invalid because complementary slackness (5.22d) and(5.22e) will not achieve.

Therefore, we conclude that:

$$b_{ig}^{*} = \frac{\sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui}^{*}}{N_g}, \qquad \forall S_i^f \in \mathcal{S}^f, \ \forall G_g \in \mathcal{G}. \tag{5.32}$$

■

5.4.2 Numerical Results

From (5.24), (5.25), and (5.26), we found that the analytical solutions did not yield closed-form expressions for the relaxed problem. Therefore, we solve the linear problem using a numerical solver. In Algorithm 2, we employ the greedy rounding algorithm to restore the binary values of the decision variables x_{ui} and b_{ig}. This algorithm is based on the one defined in [98]. The worst-case time complexity of the rounding algorithm is $O(N^2 M_f^2)$.

The time complexity of the linear relaxation-based solution is the summation of the complexities of solving the linear problem and the rounding algorithm. This time complexity was shown to be less complex than that of solving the original ILP [101, 100, 99]. Therefore, we focus on solving the linear relaxation-based (LR) solution for all proposed methods in this chapter.

The centralized allocation might be unrealistic in real scenarios because of the need for a central unit. Therefore, we formulate the distributed allocation for solving our constrained multi-objective problem in the next section.

Algorithm 2: Greedy rounding algorithm of task offloading and replication indicators

Input : x^*_{ui}, $\forall\ S^p_i \in \mathcal{S}^p$, $\forall\ U_u \in \mathcal{U}$, and b^*_{ig}, $\forall\ S^f_i \in \mathcal{S}^f$, $\forall\ G_g \in \mathcal{G}$

Output: The integer linear decision variables x_{ui}, $\forall\ U_u \in \mathcal{U}$, $\forall S^p_i \in \mathcal{S}^p$, and b_{ig}, $\forall\ S^f_i \in \mathcal{S}^f$, $\forall\ G_g \in \mathcal{G}$

Steps:

Sort the set of decision variables x^*_{ui}, $\forall\ S^p_i \in \mathcal{S}^p$, $\forall\ U_u \in \mathcal{U}$ in a descending order

Rounding $x^*_{ui} \in \{0,1\}$

if *round*$(x^*_{ui}) = 1$ **then**
  if *constraints (5.2)-(5.9) are achieved* **then**
    $x_{ui} \leftarrow 1$
  else
    $x_{ui} \leftarrow 0$
  end
else
  $x_{ui} \leftarrow 0$
end

`/* Based on above decision variables` $X = [x_{ui}]$ `and` $\forall\ S^f_i \in \mathcal{S}^f$, $\forall\ G_g \in \mathcal{G}$: `*/`

if $(\sum_{u \in \mathcal{U}^{G_g}} x_{ui} > 0)$ **then**
  $b_{ig} \leftarrow 1$
else
  $b_{ig} \leftarrow 0$
end

5.5 Distributed Allocation

In this section, we present the USRA game and use a game-theoretic approach to solve the problem mentioned above from distributed allocation. The distributed allocation can reduce the burden of finding the centralized optimal solution by making allocation decisions for home service providers individually while achieving a collectively satisfactory allocation solution. It will also scale with the size of the USRA problem, for example, the number of home service providers and their local users to the number of available foreign service providers. Finally, compared with the centralized approach, a distributed game-theoretic approach can find a USRA solution quickly, which fulfills service needs for low latency in the MEC environment [105].

5.5.1 Game Formulation

The USRA game is built to find task allocation and service replication indicators for the home service provider(s) that offloads their local users' tasks to the home service provider's MEC server and the MEC servers of the other foreign service providers in a cost-efficient manner. In the USRA game, the home service providers are players who decide $[X,\ BG]$. Let us use $X_{-j} = [X_1, \ldots,\ X_{j-1},\ X_{j+1}, \ldots,\ X_{M_h}]$ and $BG_{-j} = [BG_1, \ldots,\ BG_{j-1},\ BG_{j+1}, \ldots,\ BG_{M_h}]$ to represent all home service providers allocation decisions except home service provider j. Given other home service provider decisions X_{-j}, home service provider j would like to make a proper decision X_j to minimize its utility function, which is defined as:

$$\min\ U_{home,j} = \min \left(w\ C_{j,norm} + (1 - w) RT_{j,norm} \right) \tag{5.33}$$

while achieving the user group's QoS requirements, and

$$C_{j,norm} = \frac{C_j}{C_{j,Max}} = \frac{1}{C_{j,Max}} \left(\sum_{u \in \mathcal{U}^j} \sum_{i=1}^{M_f} C_i^S * x_{ui} + \sum_{i=1}^{M_f} \sum_{g \in \mathcal{G}^j} (b_{ig} * UC_g^R) \right)$$

$$RT_{j,norm} = \frac{1}{N_j} \sum_{u \in \mathcal{U}^j} \sum_{i \in \mathcal{S}^p} x_{ui} RT_{ui_{norm}}$$

represent normalized cost and the normalized average response time of the home service provider j.

Based on (5.33), the USRA problem can be formulated as a game $\Gamma = (M_h, \{X_j, BG_j\}_{S_j^h \in \mathcal{S}^h}, \{U_{home,j}\}_{S_j^h \in \mathcal{S}^h})$. There might be conflicts among home service providers in this game because the users and services allocation to a particular foreign service provider might prevent some other home service provider from being allocated to the same foreign service provider or other foreign service providers. To prevent conflicts among the home service providers, we study whether the game admits at least one Nash equilibrium [105].

5.5.2 The Existence of Nash Equilibrium

For the USRA game, Γ, if the strategy profile $X^* = [X_1^*, \ldots, X_{M_h}^*]$ and $BG^* = [BG_1^*, \ldots, BG_{M_h}^*]$ are Nash equilibrium, any strategy x_j^*, BG_j^* is optimal for the strategic combination of other home service providers, this means:

$$U_j(X_j^*, BG_j^*, X_{-j}^*, BG_{-j}^*) \leq U_j(X_j, BG_j, X_{-j}^*, BG_{-j}^*).$$

The above definition implies that no home service provider can increase its utility by unilaterally changing its strategy at equilibrium. Next, we introduce the concept of a potential game to solve the problem, as mentioned in [30].

5.5.3 Potential Game

For the strategy profile $(X,\ B)$, there exists a function Φ, if meeting the following criteria, the game is the exact potential game.

$$U_j(X_j, BG_j, X_{-j}, BG_{-j}) - U_j(X_j^{'}, BG_j^{'}, X_{-j}, BG_{-j}) \\ = \Phi_{(X_{-j}, BG_{-j})}(X_j, BG_j) - \Phi_{(X_{-j}, BG_{-j})}(X_j^{'}, BG_j^{'}),$$

Next, we prove that the USRA game is an exact potential game.

Proof. In [106], player objective functions constitute a potential game with a potential function. We consider the following global objective function (potential function) for our problem:

$$\Phi(X, BG) := \sum_{j=1}^{M_h} \left(\frac{w\ C_j}{C_{Max_j}} + (1-w)RT_{j,norm} \right) \tag{5.34}$$

Now, we assign each player (home service provider) an objective function that captures the player's marginal contribution to the potential function.

$$U_j(X_j, BG_j, X_{-j}, BG_{-j}) = \sum_{S_i^h \in \mathcal{S}^h \setminus S_j^h} \left(\frac{w\ C_i}{C_{Max_i}} + (1-w)RT_{i,norm} \right) \tag{5.35}$$

Now, each player's objective function is only dependent on the actions of other players. From (5.34) and (5.35), the potential game of our problem is defined as:

$$\Phi_{(X_{-j}, BG_{-j})}(X_j, BG_j) = \sum_{S_j^h \in \mathcal{S}^h} \sum_{S_i^h \in \mathcal{S}^h \setminus S_j^h} \left(\frac{w\ C_i}{C_{Max_i}} + (1-w)RT_{i,norm} \right) + \\ \sum_{S_i^h \in \mathcal{S}^h \setminus S_j^h} \left(\frac{w\ C_i}{C_{Max_i}} + (1-w)RT_{i,norm} \right) \tag{5.36}$$

The change in the objective function of player $S_j^h \in \mathcal{S}^h$ by switching from action (X_j', BG_j') to action (X_j'', BG_j''), provided that all other players collectively play (X_{-j}, BG_{-j}), is

$$U_j(X_j'', BG_j'', X_{-j}, BG_{-j}) - U_j(X_j', BG_j', X_{-j}, BG_{-j}) = \\ \Phi_{(X_{-j},BG_{-j})}(X_j'', BG_j'') - \Phi_{(X_{-j},BG_{-j})}(X_j', BG_j') \qquad (5.37)$$

Therefore, the game is the exact potential game, which always has a Nash equilibrium.

■

5.5.4 Distributed Allocation Algorithm

Given the sets of $\mathcal{S}^h$ home service providers and $\mathcal{S}^f$ foreign service providers, the USRA game employs an iterative process for home service providers to reach the Nash equilibrium. We proposed a pseudo code as shown in Algorithm 3. This process starts with initializing, assigning all users to their home service provider S_j^h, $S_j^h \in \mathcal{S}^h$. Then, if the home service provider S_j^h is overloaded in the allocation process, the home service provider is looking for a user served by a foreign service provider with the lowest utility function. Then, the home service provider sends a computing resource request to the detected foreign service provider and waits for an acknowledgment from this foreign service provider. Each foreign Service Provider S_i^f receives multiple requests from all home service providers. On the one hand, foreign Service Provider S_i^f can serve users' requests, thus, foreign Service Provider S_i^f will send a positive acknowledgment to the corresponding home service providers. On the other hand, if these requests make foreign Service Provider S_i^f overloaded, it will select between them based on different selection criteria. In this case, foreign Service Provider S_i^f

will send a negative acknowledgment to the home Service Provider $S_j^h,\ S_j^h \in \mathcal{S}^h$ of the rejected tasks.

Thus, in the case of receiving a positive acknowledgment, the home service provider will release this user to this foreign service provider. On the other hand, the home service provider receives a negative acknowledgment, which means this foreign service provider will be overloaded if it accepts this request. Therefore, we remove this foreign service provider from future allocations by substituting users' utility functions at this foreign service provider equal zero. Finally, we check if the home service provider is still overloaded and repeat the process if it is overloaded. After allocating all tasks, detect the replication indicator b_{ig} for Group $G_g,\ G_g \in \mathcal{G}$ at the foreign Service Provider $S_i^f,\ S_i^f \in \mathcal{S}^f$. When at least one user is located on the foreign Service Provider S_i^f, b_{ig} is set to 1, (i.e., $b_{ig} = 1$) and 0 otherwise.

5.6 Performance Evaluation

Here, we present the simulation results showing the performance of the centralized solution of the linear relaxation-based multi-objective solutions using Lexicographic method, weight-sum method compared to the optimized solution of distance-based and resource-based greedy approaches. In addition, we compare distributed solution of the USRA game with the linear relaxation-based weight-sum method solution.

5.6.1 Performance Metrics

In this chapter, we interested in calculating the average response time (delay) and the system cost which are:

Algorithm 3: Distributed allocation algorithm

```
Input   : M_h, M_f, T_ui^resp, UC_i^C, UC_i^T, UC_g^R, w, 𝒯_u, 𝒢, 𝒰^{G_g}, 𝒮^h, 𝒮^f, 𝒰^j
Output: X_j and BG_j, ∀ S_j^h ∈ 𝒮^h
Initialization: x_ui ← 1, ∀ U_u^{G_g} ∈ 𝒰^{G_g}, ∀ G_g ∈ 𝒢, ∀ S_j^h ∈ 𝒮^h
Task Allocation Process:
for S_j^h ∈ 𝒮^h do
    while (∑_{u∈𝒰^j} f_u x_ui > F^h_{S_j^h}) do
        for S_i^f ∈ 𝒮^f do
            for U_u^{G_g} ∈ 𝒰^{G_g} do
                if constraints (5.2-5.10) are binding then
                    Compute U_{home,j} using (5.33)
                else
                    U_{home,j} ← 0
                end
            end
        end
        Determine S_i^f, which has min(U_{home,j}), x_ui = 1, and min(U_{home,j}) ≠ 0
        Send a request to that S_i^f /* asking for computing resources */
        Wait until S_j^h receives an acknowledgment from S_i^f
        if S_j^h receives a positive acknowledgment then
            x_ui ← 1, S_i^f ∈ 𝒮^f
            x_ui ← 0, S_j^h ∈ 𝒮^h
        else
            /* S_j^h receives a negative acknowledgment */
            U_{home,j} ← 0
        end
    end
end
Replication Allocation Process/* Based on the task allocation
    process */
if (∑_{U_u^{G_g} ∈ 𝒰^{G_g}} x_ui > 0) then
    b_ig ← 1
else
    b_ig ← 0
end
```

1. **Average Response Time (Delay)**: This is calculated as the average transmission, computation, replication, and roaming time overall users in all edge servers.

2. **System Cost**: This is the total fee that the home edge service providers pay the foreign edge service providers to serve their users.

5.6.2 Simulation Setup

In our simulations, there are four overloaded home service providers, four underloaded foreign service providers. For each foreign Service Provider S_i^f, $\forall\ S_i^f \in \mathcal{S}^f$, the price of the communication rate UC_i^T and the price of computation resource UC_i^C follow a uniform distribution in the range of [0.01-0.05] $\$(Mbit^{-1})$ and [0.1-1] $\$(Mcycle^{-1})$, respectively. And the unit cost of replicating the essential service of the group G_g, UC_g^R follows a uniform distribution in the range of [1-10]$.

For each home service provider's edge server, there are up to three groups; each group has different QoS requirements. The maximum delay difference between any two users (belonging to the same group) follows a uniform distribution in the range of 10 to 15 mSec. The maximum delay of the group is selected randomly between 25 and 40 mSec. This solution's performance of the linear relaxation-based solutions are compared with the linear relaxation-based solutions of two other greedy approaches: (1) the distance-based and (2) the resource-based greedy approaches, which are described in Chapter 4. We have also compared the centralized linear relaxation-based solution (weight-sum method) with the distributed solution of the USRA game handling different selection scenarios for the foreign service providers. We repeat the subsequent experiments 1000 times to compute the 95% confidence intervals.

5.6.3 Results and Discussion

Centralized Solutions

Figure 5.2 shows that the trade-off curve (Pareto optimal curve) between the two objective functions: the total cost, and the average response time when the weight coefficient varies from 0 to 1, and the number of users equals 200. As the number of local users increases, the un-served users have no choice but to offload their tasks on the foreign service providers. Therefore, the home service providers sustain a cost to these foreign service providers while trying to minimize the average response time of all users based on the value of w. Figure 5.2(a) represents the average response time of all such users in the MEC system. While Figure 5.2(b) represents the average response time for the subset of users whose tasks are assigned to foreign service providers. The difference in the range of the average response time values on the X-axis can be justified because most users will execute their tasks at their home service provider unless overloaded. Then the remaining un-served users' tasks will be assigned to foreign service providers. The total invoiced cost to the foreign service providers is the same because the value depends on assigning tasks to foreign service providers.

Figure 5.3 shows that the LR solutions for the weighted-sum (WSum) method and the Lexicographic (Lex) method outperform LR solutions for distance-based (DB) and resource-based (RB) greedy approaches in the total cost for a larger number of users, (i.e., 100 or more users). As for the smaller number of users, the figures show that the performance of the proposed schemes is quite poor, unlike when there are many users. We can interpret that most users in this scenario will execute their tasks to their home service providers. Only a minimal number of users will be assigned to

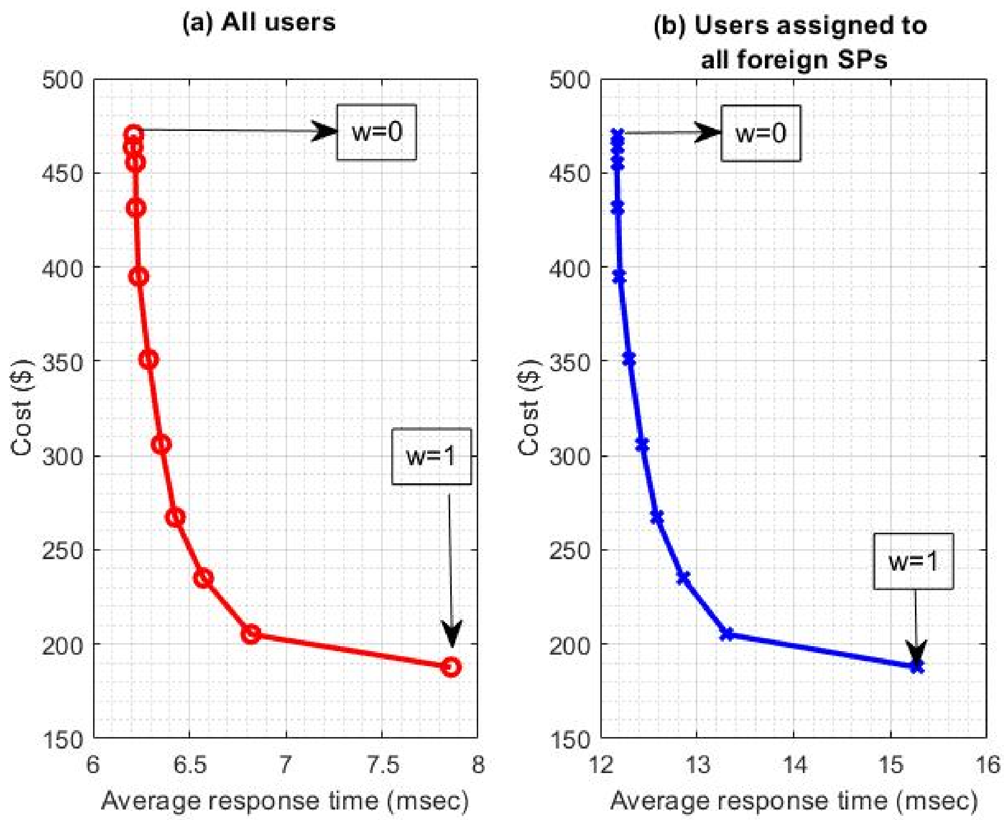

Figure 5.2: Pareto optimal curve

foreign service providers. Figure 5.3(a) shows the performance of the weight-sum method when the weight coefficient equals 0, 0.7, and 1. As the value of w increases, the WSum solution outperforms the DB and RB solutions. The WSum solution can achieve 42% and 38% cost reduction over the solutions achieved by the DB, RB solutions, respectively when $w = 0.7$ and the number of users equals 200. In contrast, Figure 5.2(b) shows that the performance of the Lex method LR solution is the best compared with the WSum method, the DB, and the RB solutions when $w < 1$. The Lex solution can achieve 59% and 56% cost reduction over the solutions achieved by the DB, and the RB solutions, respectively, when the number of users equals 200.

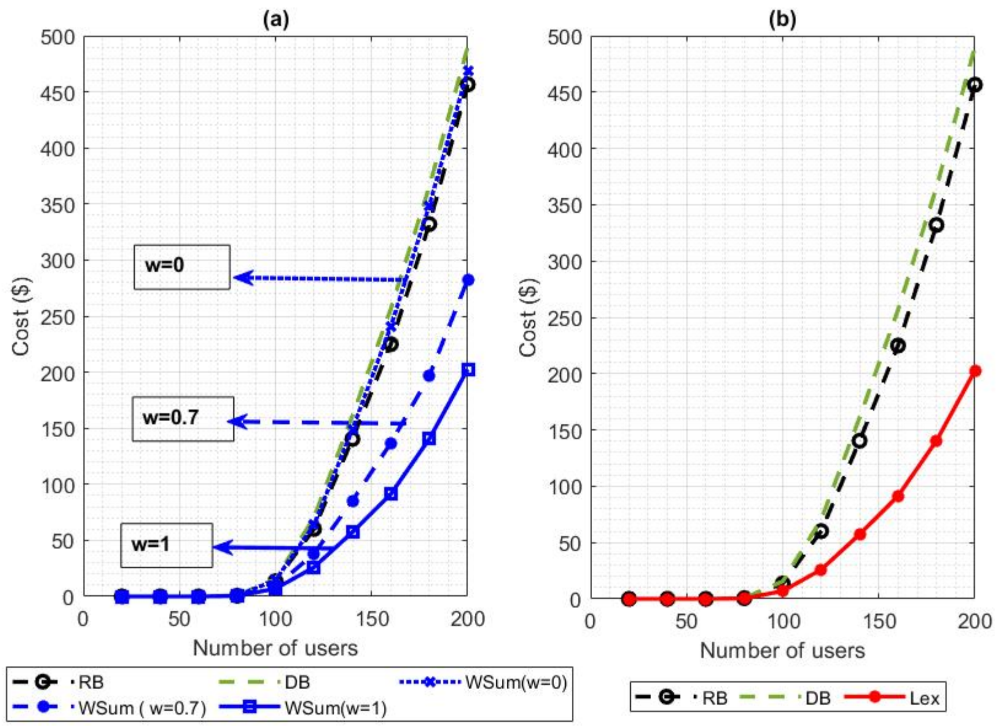

Figure 5.3: Cost ($) vs. the number of users

Figure 5.4 presents the performance of the WSum, and the Lex solutions compared to DB and RB greedy approaches in the average response time as the number of users increased. When the weight coefficient decreases, the WSum solution outperforms Lex, DB, and RB solutions, harmonizing with the trade-off curve in Figure 5.2. On the other hand, the performance of the Lex solution degrades due to the lower priority of the average response time objective function compared to the total cost objective function. For all such users in the MEC system, the average response time of the Lex solution will increase by 10% and 6% compared to the DB and the RB solution, when the number of users equals 200, as depicted in Figure 5.4(b). By comparison, users whose tasks are assigned to foreign service providers, the average response time of the WSum solution can achieve up to 12%, 6%, and 16% response time reduction compared with the Lex, the DB, and the RB solutions when the number of users equals 200 as shown in 5.4(c).

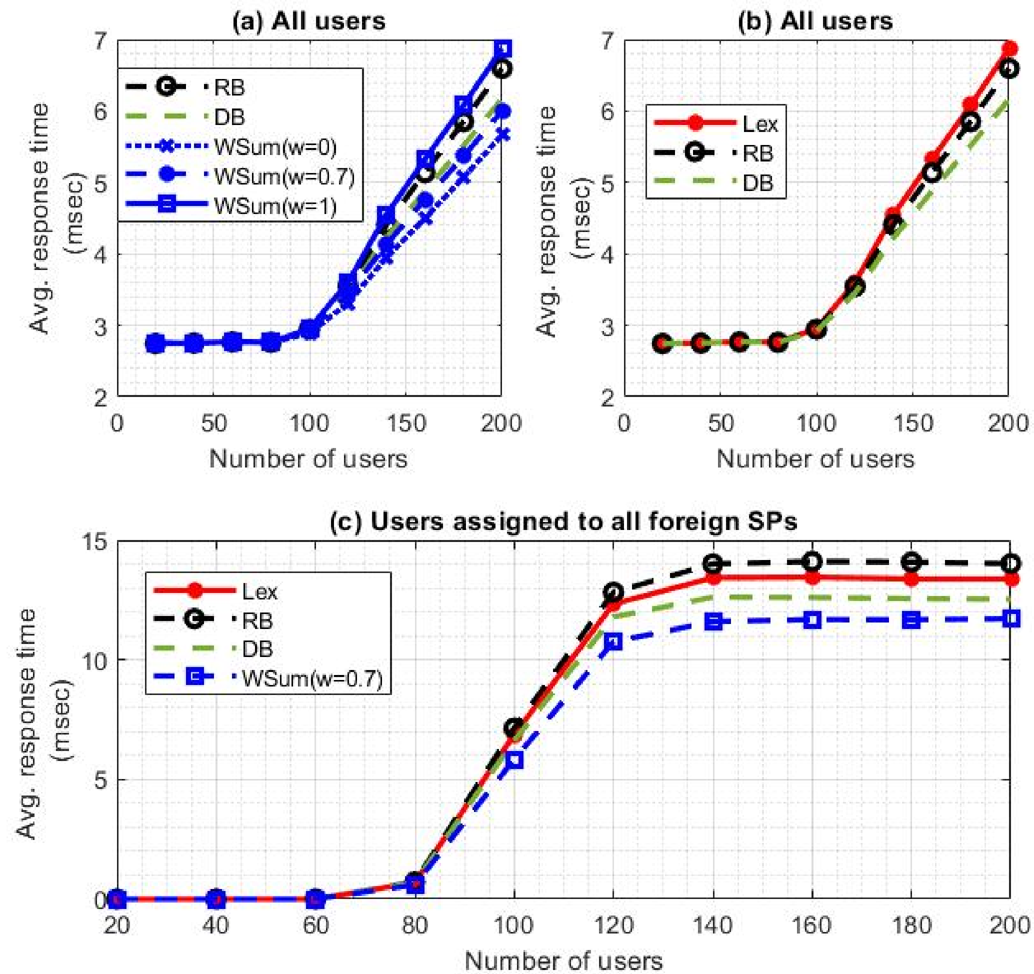

Figure 5.4: Average response time (msec) vs. the number of users

Distributed Solution

In the following simulation results, we demonstrate the effectiveness of the USRA game and the impacts of the increasing number of users on the charged cost to foreign service providers and the average response time of all users compared to the centralized WSum solution. We also evaluate the USRA game's efficiency by measuring the convergence time experimentally, representing the average number of iterations taken for the USRA game to reach a Nash equilibrium. By running the distributed algorithm, the foreign service provider has two cases either having the required resources to serve all incoming users' tasks or does not have enough resources. Thus, the foreign service provider will become overloaded when accepting all incoming users' tasks. Each foreign service provider chooses between the incoming tasks using two selection criteria: (1) select tasks requiring more computing resources to maximize revenue, and (2) randomly select from (1).

Figure 5.5 shows that the total cost of the USRA game is greater than the centralized weight-sum solution by 4% for the two selection criteria of the foreign service provider when the number of users equals 200 because the effect of two selection criteria that will appear when the foreign service provider is overloaded, which would be a rare scenario. Figure 5.6 shows that the performance of our distributed approach is almost the same as the centralized weight-sum solution when $w = 0.7$. Figure 5.6(a) shows that the average response time of the distributed solution (DS) is worst than the centralized solution (CS) by 4% for all users when the number of users equals 200. While Figure 5.6(b) shows that the difference of performance equals 1% for users whose tasks are assigned to foreign service providers when the number of users equals 200.

Figure 5.7 shows the number of iterations that the distributed algorithm will take for each home service provider versus the total number of users concerning two selection criteria of the foreign service providers. The result indicates that the random

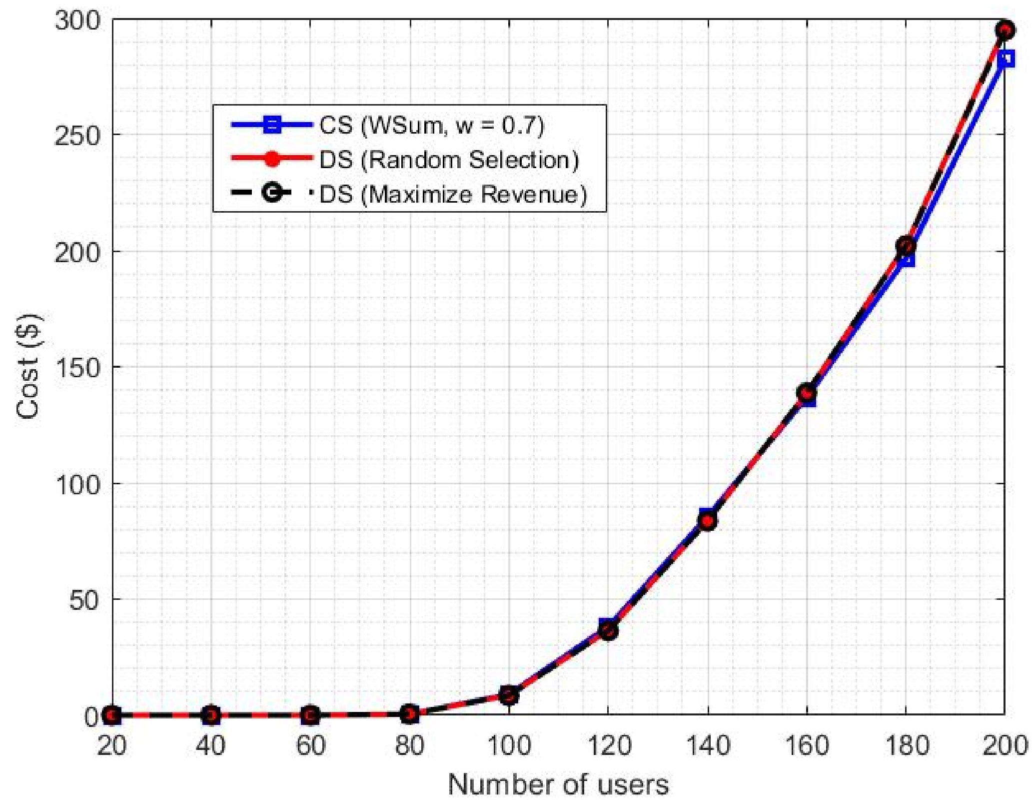

Figure 5.5: Distributed solution (DS) vs. centralized solution (CS) w.r.t. cost ($)

selection takes fewer iterations than the maximize revenue criteria for making the home service providers not overloaded.

5.7 Summary

We conclude, in terms of centralized solutions, simulation results show that the Lexicographic solution outperforms the weight-sum solution and the distance-based and resource-based schemes in the cost as the number of users increased, respecting our model's scalability and group service settings. While the weight-sum optimized solution achieves a better performance in the average response time of all users in the

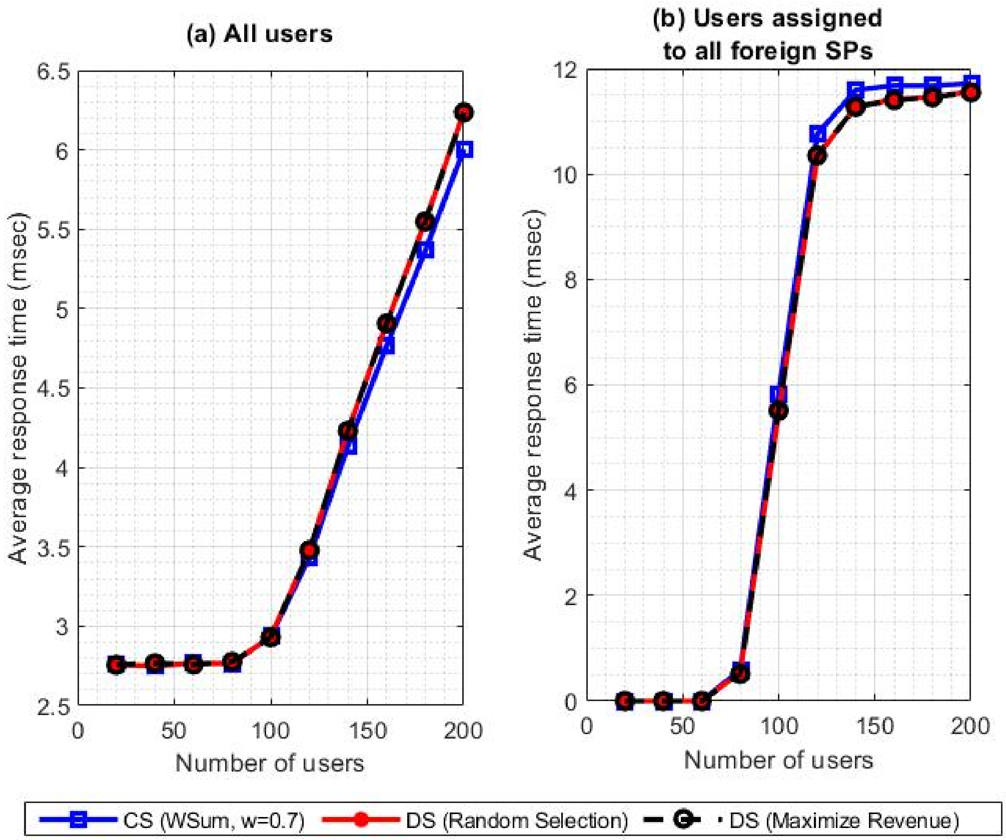

Figure 5.6: Distributed solution (DS) vs. centralized solution (CS) w.r.t. the average response time

MEC system. The results show that the Lexicographic solution is reduced by up to 59% and 56% compared to both distance-based and resource-based greedy approaches in the total cost. However, the distributed solution is more realistic and efficient, the total cost and the average response time are greater than the centralized optimal solution by 4% and 4%, respectively for all users.

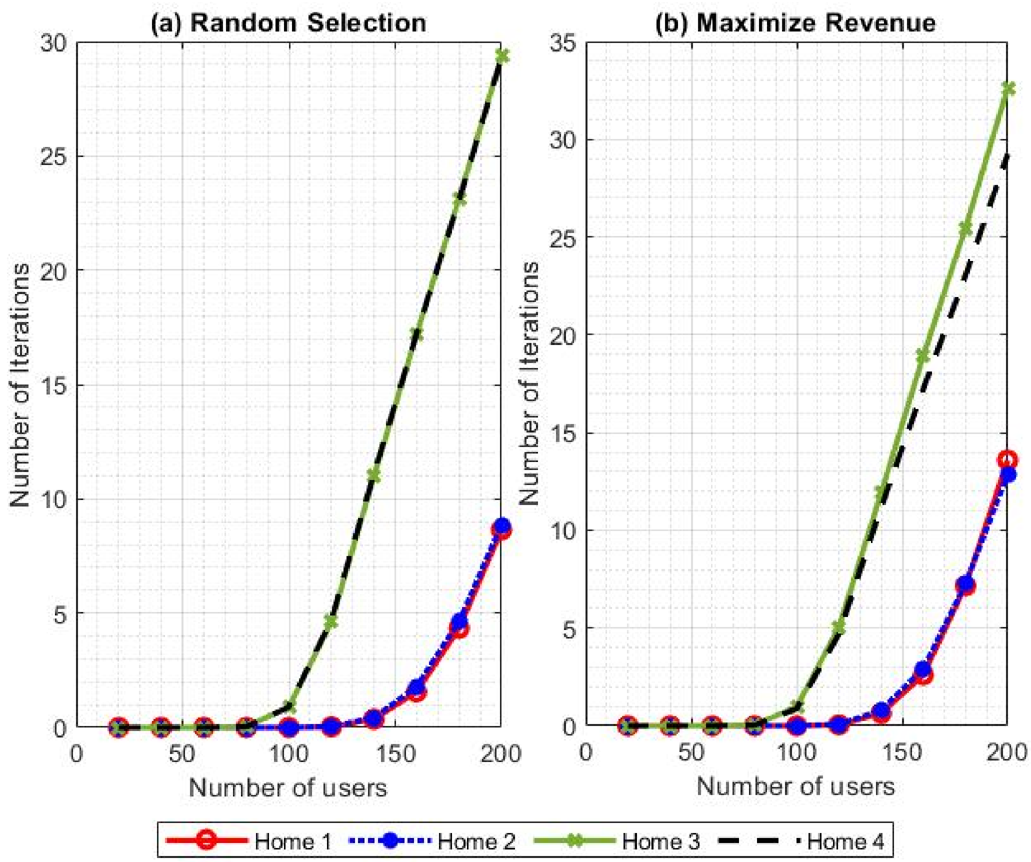

Figure 5.7: Number of iterations w.r.t the number of users

Chapter 6

Scalable and Profitable MEC System

In Chapter 5, we formulated our cost and group-delay-aware task offloading and service replication problem between multiple service providers. Based on service providers' predefined cost information, we dynamically managed task offloading for groups of users requesting to perform inter-related computations between different service providers, home service providers and foreign service providers, assuming the existing computational capacity always satisfies the user groups' computing resources demands. However, these groups may require more computing resources than the existing service providers have. Therefore, these users' groups will suffer from a processing delay during the task offloading process.

In this chapter, we address these detailed research questions:

1. What is the suitable price strategy that can be used to maximize the social welfare between home service providers and foreign service providers?

2. How does the foreign service provider's price affect the task offloading distribution?

3. How does the foreign service provider's reputation affect the task offloading

distribution?

In this chapter, we consider multiple sets of users forming service groups based on each group's participation in the same synchronized activity, such as group AR/VR. In addition, we propose two marketing approaches: market-reputation-based (MRB) and market-based (MB) approaches. For each approach, we formulate the task offloading and pricing problem as a Stackelberg model to improve the social welfare between all edge service providers. In MRB approach, each home service provider acts as a follower, which introduces the required service(s) to their user groups unless overloaded, thus, replicates the required service(s) to one or several foreign service providers' edge servers to minimize this home service provider's utility considering the foreign service provider's reputation and unit price of the computation cycle. Each foreign service provider acts as a leader, that can determine the unit price of its computing cycle to maximize its revenue by setting profitable price and not losing a good reputation (i.e., having too many blocked requests or incoming requests suffering long delays). The MB approach is similar MRB; however, we do not consider the foreign service providers' reputation in the task offloading and pricing problem. We consider the reputation of foreign service providers in determining the best offloading decisions and replication indicators for each home service provider utilizing the MRB approach. We apply a proposed rounding algorithm to reach the integer values of the decision variables to deal with the linear programming (LP) relaxation for the proposed problem of the home service providers. We develop an algorithm to determine the Stackelberg equilibrium. We compare the MRB solution to the MB solution experimentally.

The rest of this chapter is organized as follows; Section 6.1 details our system

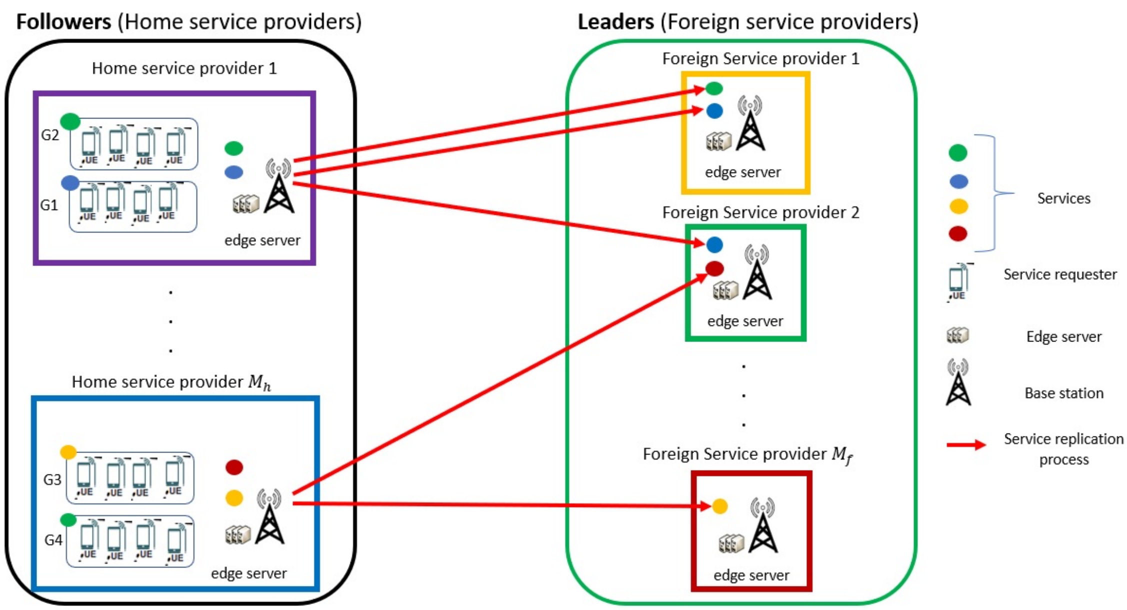

Figure 6.1: Proposed Stackelberg model

model and parameters. Section 6.2 discusses the problem setup. Section 6.3 presents the problem formulation of home service provider problems foreign service provider problems and presents a Stackelberg equilibrium algorithm to reach equilibrium. Section 6.4 presents the performance evaluation and the comparison between the MRB and the MB approaches. We conclude our findings in Section 6.5.

6.1 System Model

Figure 6.1 describes the system model. The details of our system model were described in Chapter 3, here we consider that the MEC system consists of a user set, M_h home service providers, and M_f foreign service providers, as shown in Figure 3.6. In addition, we assume the following:

- Each service provider manages a cluster of edge servers, these edge servers are collocated at the same base station.

- Home service providers have demands that exceed their capacities. Therefore, the un-served users of the home service providers would be allowed to access the foreign service providers' resources after replicating the needed service(s) to the foreign service providers.
- Foreign service providers have a predefined cost for the transmission but the cost of computation dynamically changes through our system.
- We assume that each foreign service provider sets the same price for all home service providers (uniform pricing), as proposed in [107].
- All foreign service providers publish their unit prices of the computation rate, and the home service provider determines the task offloading and the replication indicators based on these published prices and the known reputation information of these foreign service providers.

Given this system description, we focus on modeling an equilibrium solution that maximizes the social welfare between the home service providers and foreign service providers.

6.2 Problem Setup

In the MRB and the MB approaches, foreign service providers set their computing prices and announce them to home service providers. Subsequently, home service providers determine the best solution to minimize their utility based on foreign service providers' prices and reputations. Foreign service providers make choices before home service providers. Thus, this model replicates the Stackelberg model in economics [108].

Foreign service providers are leaders who can set their prices, and home service providers are followers who respond by determining the best solutions for their task offloading and service replication problem. The Stackelberg model consists of multiple leaders and multiple followers, and the leader foreign service providers set their prices on their expectations regarding the reactions of the follower home service providers to these prices. We assume that the follower home service provider aims to minimize the average response time of all users and minimize the total charged cost that all home service providers will pay when a user has no choice but to send its task to a foreign service provider who must satisfy all the QoS requirements. Therefore, the leader foreign service provider should consider the goals of the follower home service providers to make sensible economic decisions about their computing resource prices.

In addition to the description of the proposed architecture in Chapter 3, each foreign service provider $S_i^f,\ S_i^f \in \mathcal{S}^f$ has two-unit costs: $y_i(\frac{\$}{Mcycle}),\ UC_i^T(\frac{\$}{MByte})$, which are the adaptive unit price of the computation cycle, and the predefined unit price of the transmission rate. In addition, $UC_g^R(\$)$ is a group-dependent parameter, denoting the unit cost of replicating the service of the group G_g on the foreign service providers.

6.2.1 Definition of Objective Variables

Let $x_{ui},\ \forall\ u,\ i$ be a task offloading indicator, where u and i represent User $U_u,\ U_u \in \mathcal{U}$, and Service Provider $S_i^p,\ S_i^p \in \mathcal{S}^p$, respectively. In our formulation, x_{ui} is set to 1 when User U_u offloads its task to Service Provider S_i^p and is set to 0 otherwise. Let $b_{ig},\ \forall\ i,\ g$ be a replication indicator, where i and g represent Foreign Service Provider $S_i^f,\ S_i^f \in \mathcal{S}^f$ and the group ID $G_g,\ G_g \in \mathcal{G}$, respectively. In our formulation, b_{ig} is set to 1 when at least one user of Group G_g takes a service from Foreign Service Provider S_i^f and is set to 0 otherwise.

6.3 Problem Formulation

In this section, we formulate home service provider problems and foreign service provider problems for the MRB scheme.

6.3.1 Home service provider problem

Given the above definition of two objective variables, x_{ui}, b_{ig}, the home service provider j, $\forall\ S_j^h \in \mathcal{S}^h$ aims to minimize the following function: the weighted sum of the charged cost by the foreign service providers and the average response time while satisfying all the QoS requirements of all the user groups.

$$U_{home,j} = \frac{w_j C_j}{C_{j,Max}} + (1 - w_j) RT_{j,norm} \tag{6.1}$$

where $0 \le w_j \le 1$ is a scalar weighting factor. This weighting coefficient does not prioritize the objectives, but it is only a factor by varying it; the non-inferior solutions will change.

The follower home Service Provider S_j^h utility function is comprised of the average response time as shown in (6.2) for their contracted users $\mathcal{U}$, and the cost of offloading the home service providers' user tasks at the foreign service providers using service replication to improve the scalability of the introduces services, as shown in (6.4).

$$RT_{j,norm} = \frac{1}{N_j} \sum_{u \in \mathcal{U}^j} \left(RT_{uz_{norm}} x_{uz} + \sum_{i \in \mathcal{S}^f} \left(p_i RT_{ui_{norm}} + ((1 - p_i)(RT_{ui_{norm}} + \Delta)) \right) x_{ui} \right), \tag{6.2}$$

$$RT_{uz_{norm}} = \frac{RT_{uz} - RT_{Min}}{RT_{Max} - RT_{Min}}, \qquad RT_{ui_{norm}} = \frac{RT_{ui} - RT_{Min}}{RT_{Max} - RT_{Min}}, \tag{6.3}$$

$$C_j = \sum_{i \in \mathcal{S}^f} p_i (C^S_{ji} + C^R_{ji}) + (1 - p_i)(C^S_{ji} - (\delta_i \times C^S_{ji})) \tag{6.4}$$

where z represents home Service Provider S^h_j of User U_u, $U_u \in \mathcal{U}^j$, and

$$C^S_{ji} = \left((C_u \times y_i) + (B_u \times UC^T_i) \right) \times x_{ui}$$

is the service cost (computation plus transmission). The service cost depends on the unit price y_i of the computation cycle for the leader foreign Service Provider $S^f_i, S^f_i \in \mathcal{S}^f$, where $Y = [y_i, y_2, \dots, y_{M_f}]$ is the unit prices vector of computation cycle of all foreign Service Providers. $C^R_{ji} = \sum_{g \in \mathcal{G}^j} b_{ig} \times UC^R_g$ is the replication cost for replicating the required service(s) at foreign service providers' edge servers. δ_i is a reduction factor and Δ is the expected delay that the user's task might experience for taking the needed service from the leader foreign Service Provider S^f_i. Thus, foreign Service Provider S^f_i will charge the home service providers with a discounted service cost and not include the replication cost.

Moreover, p_i denotes the probability that the foreign Service Provider $S^f_i, \forall\ S^f_i \in \mathcal{S}^f$ will instantaneously serve incoming users' tasks from all home service providers. The value of these incoming tasks is expressed in the foreign Service Provider S^f_i reputation, which is the ratio between the number of immediately served users' tasks to the total number of all incoming requests from all home service providers. In other words, we can say the probability of serving for each foreign service provider expresses the user satisfaction or home service provider satisfaction from taking the required service from this foreign service provider, where $P = [p_i, p_2, \ldots, p_{M_f}]$ represents the reputation vector of all foreign service providers.

Based on the announced price Y vector and the reputation vector P, the home Service Provider S_j^h aims to find optimal offloading decisions between S_j^h and foreign service providers to minimize all users' average response time and the cost that all leader foreign service providers invoice the follower home service provider by solving its optimization problem, which is defined as:

$$\min_{x_{ui},\ b_{ig}} U_{home,j} \tag{6.6}$$

$$S.T. \sum_{i \in \mathcal{S}^p} x_{ui} = 1, \qquad \forall\ U_u^j \in \mathcal{U}^j, \tag{6.7}$$

$$\sum_{u \in \mathcal{U}^j} x_{ui} f_u \le F^h_{S_j^h} \tag{6.8}$$

$$\sum_{u \in \mathcal{U}^j} x_{ui} f_u \le F^f_{S_i^f}, \qquad \forall\ S_i^f \in \mathcal{S}^f \tag{6.9}$$

$$RT_{uz} x_{uz} + \sum_{i \in \mathcal{S}^f} \left[p_i RT_{ui} + \left((1 - p_i)(RT_{ui} + \Delta) \right) \right] x_{ui} \le T^{max}_{G_g},\ \forall\ U_u^j \in \mathcal{U}^j,\ \forall\ G_g \in \mathcal{G}, \tag{6.10}$$

$$\begin{aligned} & RT_{uz} x_{uz} + \sum_{i \in \mathcal{S}^f} \left[p_i RT_{ui} + \left((1 - p_i)(RT_{ui} + \Delta) \right) \right] x_{ui} - \\ & \quad RT_{lz} x_{lz} - \sum_{i \in \mathcal{S}^f} \left[p_i RT_{li} + \left((1 - p_i)(RT_{li} + \Delta) \right) \right] x_{li} \le d_{G_g}, \\ & \qquad \forall\ G_g \in \mathcal{G},\ \forall\ U_u^j,\ \forall\ U_l^j \in \mathcal{U}^j,\ l \ne u \end{aligned} \tag{6.11}$$

$$\frac{\sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui}}{N_g} - b_{ig} \le 0, \quad \forall\ G_g \in \mathcal{G},\ \forall\ S_i^f \in \mathcal{S}^f \tag{6.12}$$

$$b_{ig} - \sum_{U_u^{G_g} \in \mathcal{U}^{G_g}} x_{ui} \le 0, \qquad \forall\ G_g \in \mathcal{G},\ \forall\ S_i^f \in \mathcal{S}^f \tag{6.13}$$

$$x_{ui} \in \{0, 1\},\ \forall\ S_i^p \in \mathcal{S},\ \forall\ U_u^j \in \mathcal{U}^j \tag{6.14}$$

$$b_{ig} \in \{0, 1\},\ \forall\ S_i^f \in \mathcal{S}^f,\ \forall\ G_g \in \mathcal{G} \tag{6.15}$$

Constraint (6.7) ensures that each task is offloaded to only one service provider. Constraint (6.8) ensures that the sum of assigned computing capacities to the subset of users offloading their tasks to their home Service Provider's S_j^h MEC server does not exceed its total available computing capacity. Constraints (6.9) ensures that the sum of assigned computing capacities to the home Service Provider S_j^h users offloading their tasks to the foreign service provider's MEC server does not exceed its total available computing capacity. Constraints (6.8) and (6.9) ensure that the maximum computational capabilities of the service providers' edge servers are not exceeded. Constraint (6.10) ensures that the total response time of the home Service Provider S_j^h user's task belonging to Group $G_g \in \mathcal{G}^j$ does not exceed this service's maximum delay requirement. Constraint (6.11) ensures that the delay difference between two users belonging to the same Group G_g does not exceed the predefined difference delay requirement. Constraints (6.10) and (6.11) are to obeyed to the QoS demanded by each group. Constraints (6.12) and (6.13) ensure that the group's service is deployed on the edge server of the foreign service provider (i.e., ensuring that $b_{ig} = 1$, if there is at least one user $U_u^{G_g}$ of Group G_g accesses its service at foreign Service Provider S_i^f, $x_{ui} = 1$). Constraints (6.14) and (6.15) impose a binary decision value for the variable $x_{ui},\ b_{ig}$ of the home Service Provider S_j^h users and services, respectively.

It is clear that optimization problem (6.6)-(6.15) is an integer linear multi-objective problem. To solve this problem, we relax the ILP optimization problem into linear programming (LP) by relaxing the integer decision variables, assuming $0 \le x_{ui},\ b_{ig} \le 1$. After this, we use a numerical solver to solve solver-based solutions for the relaxed decision variables, then restore the integer decision variables using an integer rounding greedy approach, as described in Algorithm 2.

The home Service Provider S_j^h optimization problem depends on the prices announced by the foreign service providers and their reputation values. Thus, the relationship can be written as:

$$[X_j, BG_j] = f(Y, P) \tag{6.16}$$

where X_j is the vector of offloading decisions (i.e., $X_j = [x_{ui}]$), BG_j is the vector of placement replication decisions (i.e., $BG_j = [b_{ig}]$) for the home Service Provider S^h_j, u refers to User U^j_u, $\forall\ U^j_u \in \mathcal{U}^j$, and i refers to Service Provider S^p_i, $\forall\ S^p_i \in \mathcal{S}^p$, g refers to Group G_g, $\forall\ G_g \in \mathcal{G}^j$. This relationship (6.16) is a reaction function because it reveals how the home service providers will respond to the foreign service providers' determination.

6.3.2 Foreign service provider problems

Here, we discuss the leader foreign service provider's revenue maximization problem assuming they are aware that their prices and reputations influence the decision variables of follower service providers. The foreign Service Provider S^f_i, $S^f_i \in \mathcal{S}^f$ should set a suitable price for the computation cycle to maximize its revenue (i.e., the revenue is the amount of money that the foreign service provider gains from providing its computation resources to home service providers). If the foreign service provider sells its computation resources at a higher price than other foreign service providers, thus follower home service providers are more willing to offload un-served users' tasks at the lower-priced foreign service providers. We define the revenue function of each foreign Service Provider S^f_i, $S^f_i \in \mathcal{S}^f$ as:

$$U_{foreign,i}(y_i) = \sum_{j \in \mathcal{S}^h} \Big(p_i(C^S_{ji} + C^R_{ji}) + (1 - p_i)(C^S_{ji} - (\delta_i \times C^S_{ji})) \Big) \tag{6.17}$$

The foreign Service Provider S_i^f will select a suitable price to maximize its revenue. The revenue maximization problem for foreign Service Provider $S_i^f, S_i^f \in \mathcal{S}^f$ is formulated as:

$$\max_{y_i} \; U_{foreign,i} \tag{6.18}$$

$$S.T. \qquad \sum_{u \in \mathcal{U}} x_{ui} f_u \leq F^f_{S_i^f} \tag{6.19}$$

$$Y_{i,min} \leq y_i \leq Y_{i,max} \tag{6.20}$$

where $Y_{i,min}$, and $Y_{i,max}$ are the minimum and maximum acceptable prices for the foreign Service Provider S_i^f.

The objective function (6.18) refers to revenue (6.17) of the foreign Service Provider S_i^f, which is the summation of all payments from the home service providers to this foreign Service Provider S_i^f. Constraints (6.19) ensure the capacity limit of computation resources of the foreign Service Provider S_i^f. Constraint (6.20) ensures that the foreign Service Provider S_i^f should set a price greater than or equal to the minimum price and less than or equal to the maximum price. We assume that this minimum price $Y_{i,min} \neq 0$ because this would lead to foreign Service Provider S_i^f to be preferred by all home service providers for incoming iterations.

The relationship among the foreign service providers is non-cooperative and competitive, where each foreign service provider tries to set the proper price to attract more home service providers. Therefore, we can use a game-theoretic approach to represent the foreign service providers optimization problems as a non-cooperative game. The foreign service providers play the next non-cooperative game $\Gamma = (\mathcal{S}^f, \{\mathcal{Y}_i\}_{S_i^f \in \mathcal{S}^f}, \{U_{foreign,i}\}_{s_i^f \in \mathcal{S}^f}),$

in which:

- Players: M_f foreign Service providers.
- Strategies: each foreign Service Provider S_i^f will select the price $y_i \in \mathcal{Y}_i$ that maximizes its revenue, where $\mathcal{Y}_i$ is the set of all available prices that satisfy the constraint (6.20).
- Payoffs: The revenue function $U_{foreign,i}$, which is defined above.

Foreign service providers play the non-cooperative game to set their unit prices of computing cycles to maximize their revenue while not losing their reputation by blocking or delaying users' tasks, which is the Nash equilibrium (NE) point $Y^* = [y_1^*, \ldots, y_{M_f}^*]$. At this point, foreign service providers achieve a mutually satisfactory solution because there is no foreign service provider that can further increase its revenue by unilaterally changing its strategy.

$$U_{foreign,i}(y_i^*, y_{-i}^*) \geq U_{foreign,i}(y_i, y_{-i}^*), \ \forall S_i^f \in \mathcal{S}^f \tag{6.21}$$

where y_{-i}^* represents prices of all foreign service providers except the price of the foreign Service Provider S_i^f.

6.3.3 MB versus MRB approaches

In the MB approach, we apply the same previous Stackelberg model between home service providers and foreign service providers without considering the reputation of foreign service providers (i.e., we substitute $p_i = 1, \ \forall \ S_i^f \in \mathcal{S}^f$).

6.3.4 Stackelberg Equilibrium

Each home Service Provider S_j^h has the best response for a given foreign service providers' prices. Therefore, the Stackelberg game possesses a Stackelberg equilibrium (SE) if the price-setting game among the foreign service providers admits a Nash equilibrium (NE). Based on [64, 62, 109], the Nash equilibrium of the foreign service provider optimization game Γ always exists and is unique.

Theorem 1: A unique Nash equilibrium exists in the price-setting game among the foreign service providers, thereby a unique SE.

Proof. A Nash equilibrium exists for the foreign service providers in the price-setting game if:

- all players' strategy space is the product space $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_{M_f}$, where $y_i \in \mathcal{Y}_i,\ S_i^f \in \mathcal{S}^f$. Therefore, $\mathcal{Y}$ is a convex, closed, non-empty subset of some Euclidean space R^{M_f}.

- $U_{foreign,i}$ is continuous and concave in $y_i,\ \forall\ S_i^f \in \mathcal{S}^f$, where

$$\frac{\partial^2\ U_{foreign,i}}{\partial\ y_i^2} = 0, \forall\ S_i^f \in \mathcal{S}$$

The analogy to the work in [64], the game Γ is a concave N-person game. Therefore, according to Theorem 1 and 3 in [110], a unique NE exists in this game. ∎

Home service providers determine the task offloading and replication decision variables based on foreign service providers' announced prices and reputations. Thus, we model our problem as a Stackelberg game with multiple leaders and multiple followers. In the first stage, foreign service providers play the non-cooperative game to set the unit prices, a Nash equilibrium (NE) point and announce the prices to follower home service providers. Each home Service Provider S_j^h determines the $[X_j,\ B_j]$ decision variables in the second stage. The equilibrium strategy for the followers in a Stackelberg game is any strategy that constitutes an optimal response to the adopted leaders' prices.

Definition 1. Set $(X^*,\ B^*,\ Y^*)$ is called a Stackelberg equilibrium of the game between the foreign service providers and the home service providers if it satisfies the following conditions.

$$U_{foreign,i}(Y^*;[X^*,B^*]=f(Y^*,P)) \geq U_{foreign,i}(y_i,y^*_{-i};[X,B]=f(y_i,y^*_{-i},P)),\ \forall S_i^f \in \mathcal{S}^f$$
$$U_{home,j}(Y^*;[X^*,B^*]=f(Y^*,P)) \leq U_{home,j}(Y^*;[X_j,B_j,X^*_{-j},B^*_{-j}]),\ \forall S_j^h \in \mathcal{S}^h$$

Next, we develop an iterative algorithm to reach the equilibrium of the Stackelberg game.

6.3.5 Distributed Algorithm for Reaching Equilibrium

Here, we develop an iterative algorithm to reach the Stackelberg equilibrium. Each foreign Service Provider $S_i^f, \forall\ S_i^f \in \mathcal{S}^f$ sets its initial unit price y_i^0 based on its serving probability (reputation), then all foreign service providers publish these unit prices to all home service providers. Thereafter, each home Service Provider $S_j^h,\ \forall\ S_j^h \in \mathcal{S}^h$ determines its task offloading and replication decision variables by solving the problem (6.6)-(6.15) based on the published unit prices and the known reputations for all foreign service providers. Each foreign Service provider S_i^f updates its unit price and reputation probability based on the received tasks. When foreign Service Provider

S_i^f updates its unit price, all home service providers adjust their decision variables considering this new updated price by solving (6.6)-(6.15) for each home Service Provider S_j^h.

Algorithm 4: Stackelberg equilibrium algorithm

Input : $p_i,\ y_i^t \leftarrow 0,\ y_i^{t-1} \leftarrow y_i^o, t \leftarrow 1,\ \forall\ S_i^f \in \mathcal{S}^f$
Output: $X^*,\ BG^*$, and $Y^* \leftarrow [y_1^*, \dots, y_{M_f}^*]$
while $\left((|y_i^t - y_i^{t-1}|) < \epsilon,\ \forall\ S_i^f \in \mathcal{S}^f\right)$ **do**
  Publish $y_i^{t-1},\ \forall S_i^f \in \mathcal{S}^f$
  for $(S_j^h, \forall\ S_j^h \in \mathcal{S}^h)$ **do**
    $[X_j, BG_j] \leftarrow$Solve the problem (6.6)-(6.15)
    Send $[X_j, BG_j]$ to all $S_i^f, S_i^f \in \mathcal{S}^f$
  end
  for $(S_i^f,\ \forall\ S_i^f \in \mathcal{S}^f)$ **do**
    Compute p_i
    Update price: $y_i^t \leftarrow y_i^{t-1} + \left((\sum_{j\in\mathcal{S}^h}\sum_{u\in\mathcal{U}^j} x_{ui} f_u) - F^f_{S_i^f}\right) \times \gamma_i$
    if $y_i^t \neq y_i^{t-1}$ **then**
      Update $[X_j, BG_j],\ \forall\ S_j^h \in \mathcal{S}^f$
    else
    end
  end
  $t \leftarrow t + 1$
end

Each foreign Service Provider S_i^f follows a mathematical relation for updating its unit price, which is:

$$y_i^t = y_i^{t-1} + \left((\sum_{j\in\mathcal{S}^h}\sum_{u\in\mathcal{U}^j} x_{ui} f_u) - F^f_{S_i^f}\right) \times \gamma_i, \tag{6.22}$$

where γ_i is a step parameter of each foreign Service Provider S_i^f, which is a small number, and unit prices and home service providers' decision variables update iterative till prices of two consecutive iterations have satisfied the accuracy requirement.

The final converged prices Y^* and the corresponding home service providers' decision variables form the Stackelberg equilibrium.

6.4 Performance Evaluation

Here, we present the simulation results showing the performance of the MRB and MB approaches. We investigate how the home service providers select foreign service providers to purchase their computing resources to serve the remaining un-served users' tasks based on their unit prices of computation cycles and their reputation. In addition, we study how the foreign service providers determine the unit price of their computation resources to maximize their revenue.

6.4.1 Simulation Setup

In our simulations, we use MATLAB to examine the performance of the proposed Stackelberg model. Our simulation steps are performed using iterations to obtain the Stackelberg equilibrium. First, given the unit price of the computation cycle of each foreign service provider, each home service provider makes the best choice for offloading un-served users' tasks by solving the linear relaxation-based problem. After that, foreign service providers will update their prices based on the home service providers' allocations. Foreign service providers may receive more requests that require more than their computational capacity; thus, it selects between the incoming requests based on maximizing revenue, (i.e., select a request which requires more computing resources). We continue this process until the accuracy requirement for the unit price is achieved. Lastly, the best task offloading, and replication allocation of home service providers can be obtained when the unit prices are determined.

Table 6.1: Simulation parameters

Parameters	**Values**
M_h	4
M_f	4
UC_i^T	$[0.01\text{-}0.05]\frac{\$}{Mbit}$
$UC_g^R,\ \forall\ g \in \mathcal{G}$	[1-10]$
d_{G_g}	[20-25] msec
$T_{G_g}^{max}$	[40-60] msec
y_i^0	$0.5\ \frac{(\$)}{Mcycle}$
$Y_{i,min}$	$0.01\frac{(\$)}{Mcycle}$
$Y_{i,max}$	$1\frac{(\$)}{Mcycle}$
ϵ	10^{-2}
$\gamma_i,\ \forall\ S_i^f \in \mathcal{S}^f$	10^{-4}
$p_i,\ \forall\ S_i^f \in \mathcal{S}^f$	1
$w_j,\ \forall\ S_j^h \in \mathcal{S}^h$	0.7
Δ	5 msec
$\delta_i,\ \forall\ S_j^f \in \mathcal{S}^f$	0.3
$F_{S_j^h}^h,\ \forall\ S_j^h \in \mathcal{S}^h$	300 Gigacycle/sec
$F_{S_i^f}^f,\ \forall\ S_i^f \in \mathcal{S}^f$	300, 450, 600, and 750 Gigacycle/sec, i =1, 2, 3, and 4, respectively
$N_j,\ \forall\ S_j^h \in \mathcal{S}^h$	50, 60, 70, 80, 90, 100, and 110 users, j =1, 2, 3, and 4, respectively

Unless mentioned otherwise, Table 6.1 summarizes simulation parameters used in this chapter. All foreign service providers have the same probability of serving assigned users' tasks, which equals 1, meaning all foreign service providers serve all incoming users' tasks. We repeat the subsequent experiments 1000 times to compute the 95% confidence intervals.

6.4.2 Performance Metrics

In this chapter, we interested in calculating the average response time (delay), the system cost, revenue, and best price which are:

1. **Average Response Time (Delay)**: This is calculated as the average transmission, computation, replication, and roaming time overall users in all edge servers.

2. **System Cost**: This is the total fee that the home edge service providers pay the foreign edge service providers to serve their users.

3. **Revenue**: This is the amount of money that the foreign Service Provider S_i^f, $\forall$ $S_i^f \in \mathcal{S}^f$ might invoice all home service providers for serving their unserved users.

4. **Best Price**: This is the convergence unit sale price of the computation cycle that the foreign Service Provider S_i^f, $\forall$ $S_i^f \in \mathcal{S}^f$ determines while satisfying users' QoS requirements.

6.4.3 Results and Discussion

Optimal Prices and Revenue of Foreign Service Providers at The Stackelberg Equilibrium

Figure 6.2 shows the optimal price of unit computation cycle for each foreign service provider at the Stackelberg equilibrium when the number of users varies from 200 to 440. On the one hand, foreign service providers increase their prices while the demand increases. On the other hand, they reduce their prices if the demand is low to attract more home service providers to purchase their computing resources. The prices will be increased as the number of users increases because the demand of computing resources increases. In Figure 6.2(a), (b), (c), and (d), the MRB solution can achieve 21%, 30%, 27%, and 31%, respectively, unit price reduction over the MB

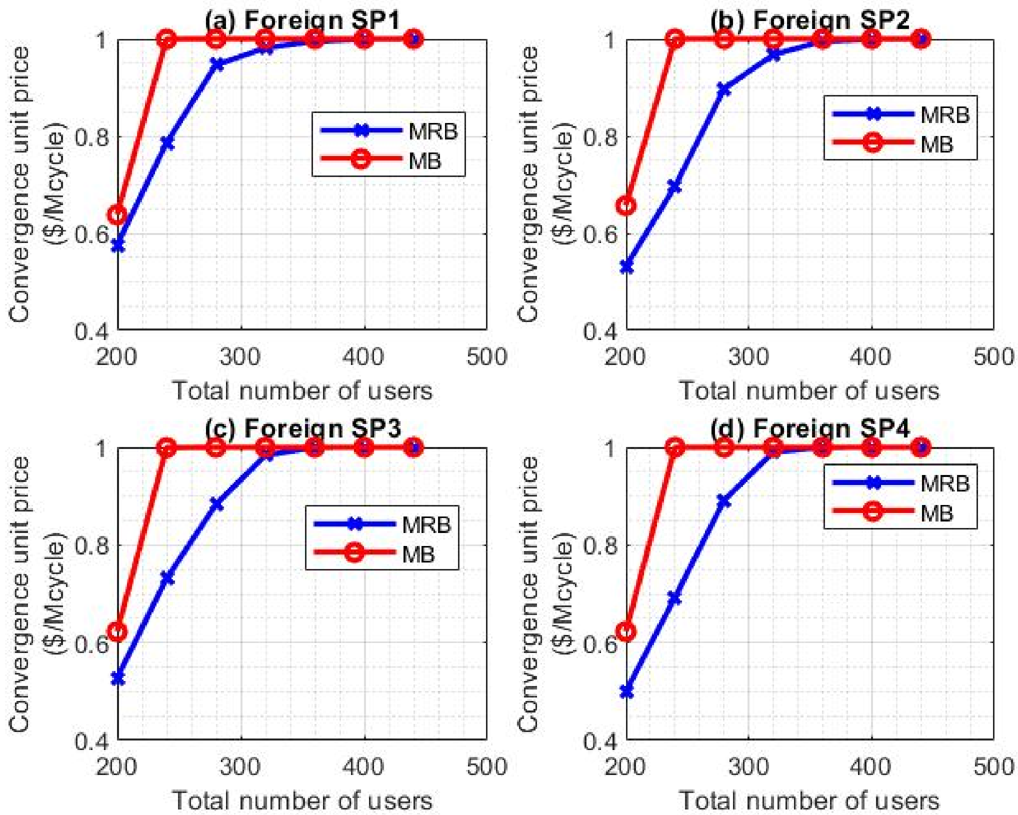

Figure 6.2: Optimal prices of foreign service providers at the Stackelberg equilibrium vs. the total number of users in the system

solution, respectively, when the total number of users equals 240. As the number of users increases, the difference in unit price between the MRB and the MB solutions decreases due to increasing the incoming requests, and the convergence price nears the maximum value.

Figure 6.3 shows the steady state serving probability for each foreign service provider as the total number of users in the system changes from 200 to 440. As the serving probability increases, the foreign service provider can process more users' tasks instantaneously. In general, the foreign service provider's serving probability

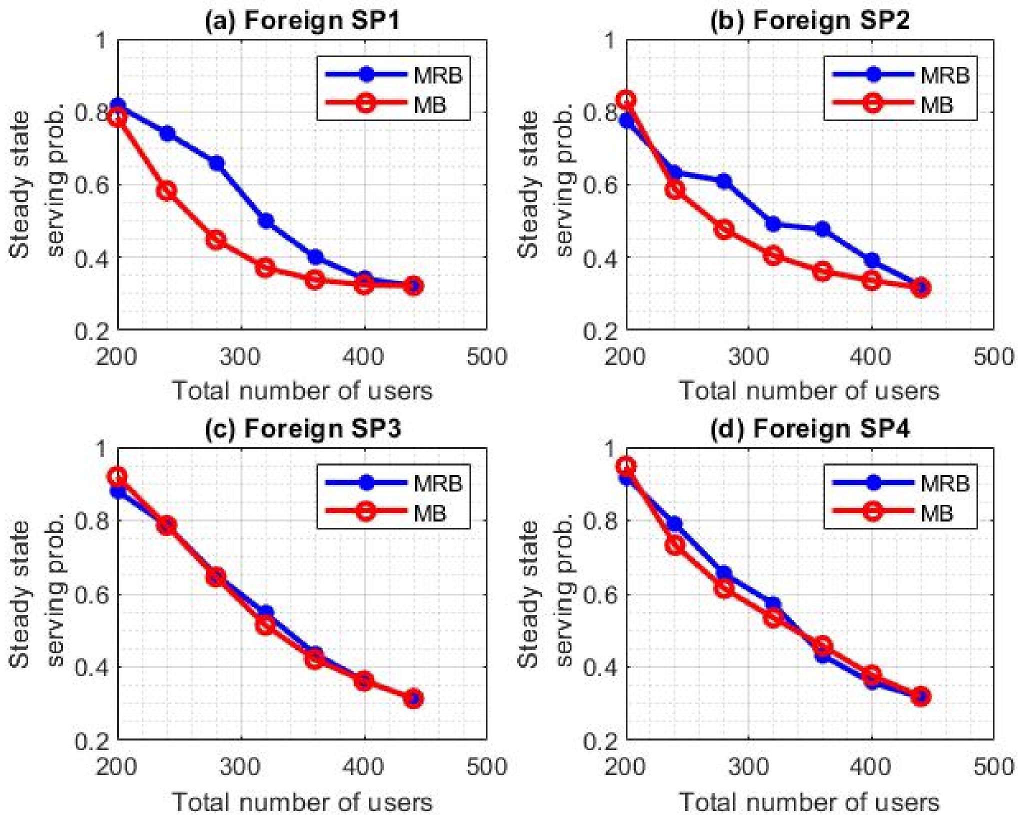

Figure 6.3: Steady state serving probability of foreign service providers at the Stackelberg equilibrium vs. the total number of users in the system

increases with increased computing resources. Figure 6.3(a) and Figure 6.3(b) display the serving probability for foreign service provider 1 and 2 with fewer computing capacities; therefore, their serving probability is lower than foreign service providers 3 and 4 with higher computing capacities, as shown in Figure 6.3(c) and Figure 6.3(d). As for the smaller number of users, the MRB approach outperforms MB. However, both MRB and MB approaches almost have the same performance for a larger number of user, (i.e., 400 or more users). We can interpret that as the number of users' tasks increases and computing resources of foreign service providers becomes limited,

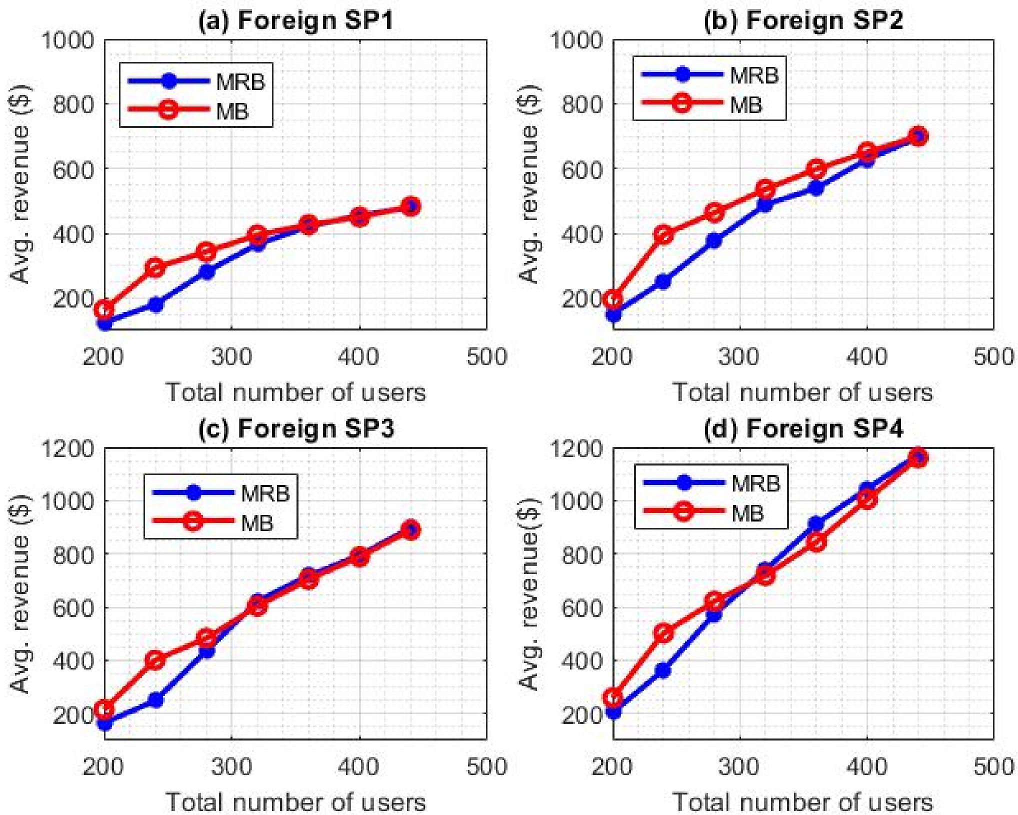

Figure 6.4: The average revenue of foreign service providers at the Stackelberg equilibrium vs. the total number of users in the system

users experience an additional delay due to waiting for service from other back-end servers.

Figure 6.4 shows the revenue of each foreign service provider when the number of users varies from 200 to 440, which corresponds to the situations in Figure 6.2(a), (b), (c), and (d) and Figure 6.3(a), (b), (c), and (d), respectively. Foreign service providers' revenue increases as the number of users' tasks increases. Figure 6.4(a) displays the corresponding revenue of foreign service provider 1. Although foreign

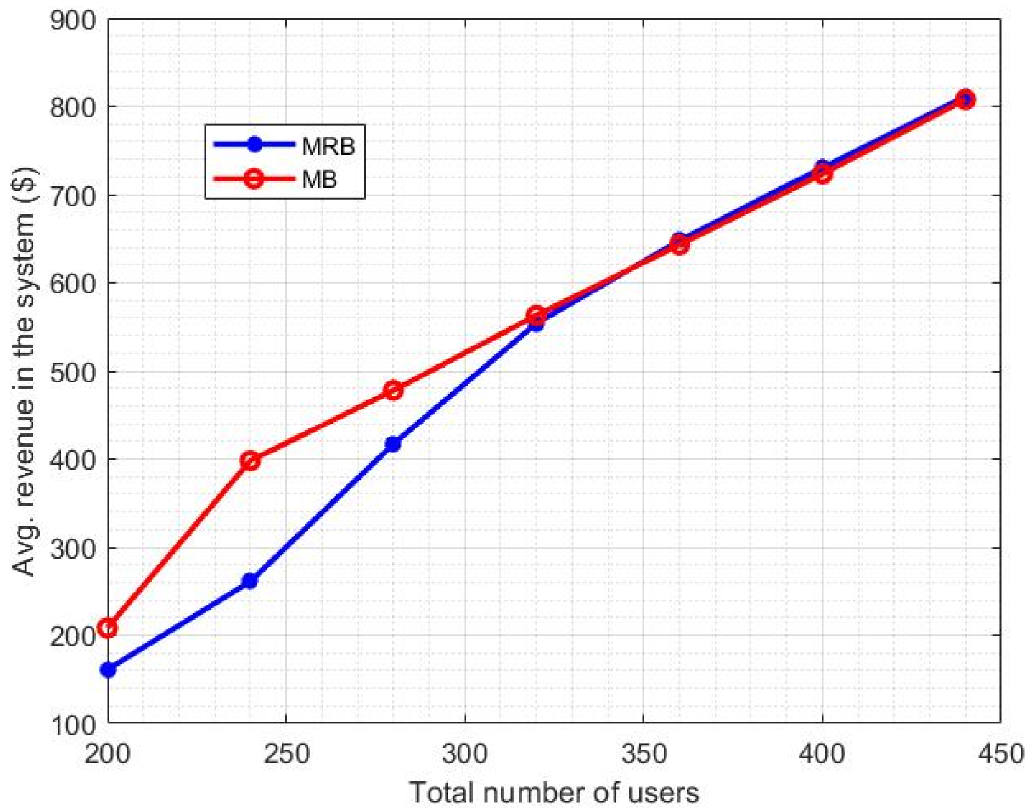

Figure 6.5: The average revenue of all foreign service providers at the Stackelberg equilibrium vs. the total number of users in the system

service provider 1 charges the highest unit price, the earned revenue is the lowest because the foreign service provider manages the lowest amount of computing resources. In contrast, the revenue of foreign service provider 4 is the highest due to managing the highest computing capacity, as shown in Figure 6.4(d). In Figure 6.5, we see the MB approach outperforms MRB. The MB approach can achieve around 34% more revenue than the MRB approach, when the number of users equals 240. This difference in revenue is reduced as the total number of users increases; thus, each foreign service provider has the same revenue for both the MEB and MB approach because

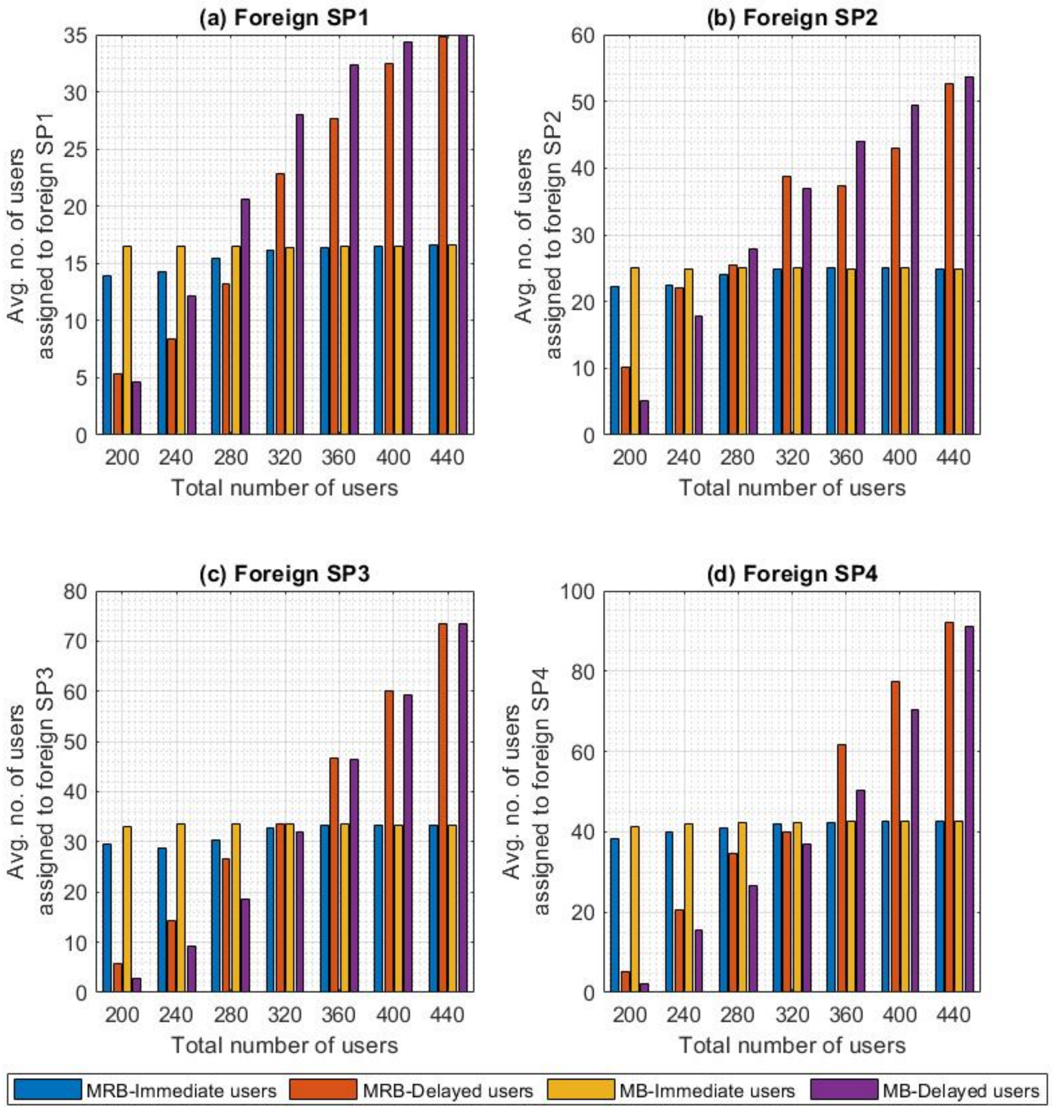

Figure 6.6: Number of users whose tasks are assigned to foreign service providers at the Stackelberg equilibrium vs. the total number of users in the market-based (MB) approach and the market-reputation-based (MRB) approach

this foreign service provider sets the price to the maximum profitable value.

Figure 6.6 shows the number of users assigned to the foreign service providers in the MB and MRB when the total number of users varies from 200 to 440 in the system. Home service providers forward more users' tasks to foreign service providers as the number of users increases due to the limitation of their computation capacity. Both approaches have the same number of users whose tasks are assigned to the foreign service providers as the total number of users increases. However, the number of delayed users' tasks and those whose tasks are served immediately at the foreign service providers slightly fluctuates in the MRB approach compared with MB, where the number of delayed users' tasks increases almost linearly. Regarding the number of users whose tasks are processed with a delay, the MRB approach outperforms MB for a fewer number of users because home service providers used the serving probability with the unit price of foreign service providers to determine the best solution that minimizes their utility in task offloading and replication process. In contrast, home service providers determine their best solution based on the unit price only in the MB approach. Therefore, the home service providers' satisfaction will be increased. In contrast, the MRB and MB approaches have the same performance in the average numbers of delayed users' tasks for a larger number of users, (i.e., 360 and more users), due to the limited computing capacity of foreign service providers; thus, any additional users' tasks are going to be delayed.

Utilities of Home Service Providers at The Stackelberg Equilibrium

Figures 6.7(a), (b), (c), and (d) show the average response time of each home service provider when the number of users for each changes from 50 to 110. Figure 6.8 shows the average response time of all users in the system when the number of users changes from 200 to 440. The average response time increases as the number of users increases. The performance of the MRB approach is a little poor for a fewer number of users,

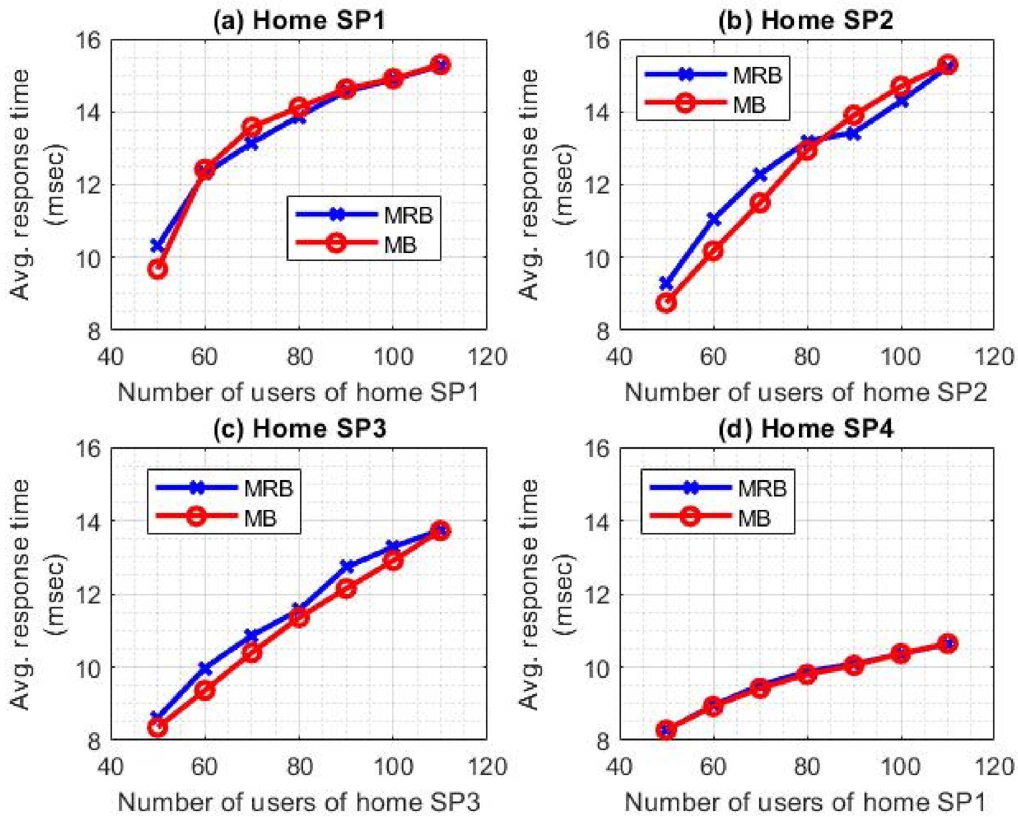

Figure 6.7: Average response time of home service providers at the Stackelberg equilibrium vs. the total number of users

unlike when there are many users. We can interpret that home service providers prefer a foreign service provider with a good serving probability while satisfying the QoS requirements. While home service providers do not consider foreign service providers' serving probability in the MB approach. The MB solution can achieve around a 4% response time reduction over the MRB solution when the number of users equals 200. In the case of a larger number users, the MRB and MB solutions have the same average response time.

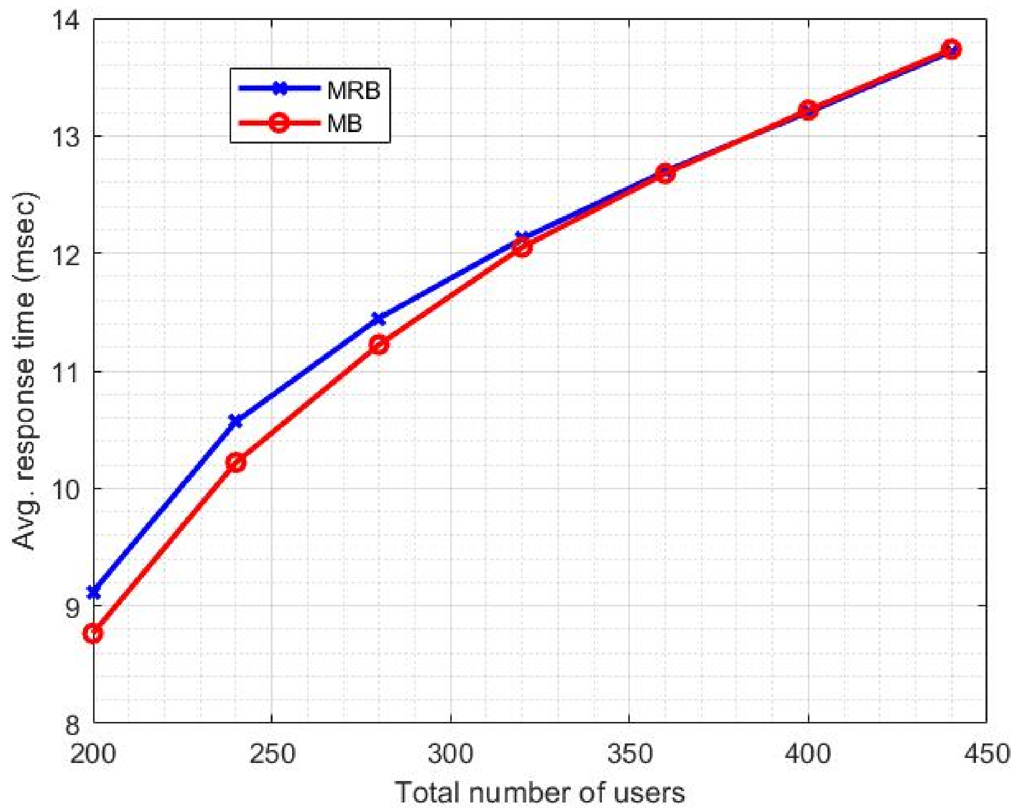

Figure 6.8: Average response time of home service providers at the Stackelberg equilibrium vs. the total number of users

Figure 6.9 presents the revenue of each foreign service provider, showing the contribution of home service providers' cost in the MB approach and MRB as the total number of users changes from 200 to 440, that are corresponding to the situations in Figure 6.2(a), (b), (c), and (d), Figure 6.3(a), (b), (c), and (d), and Figure 6.4(a), (b), (c), (d), respectively. The MRB approach outperforms MB from the home service providers' perspective for fewer users (i.e., 200 and 280 users) due to the convergence unit prices of the foreign service providers in the MRB approach are less than in the

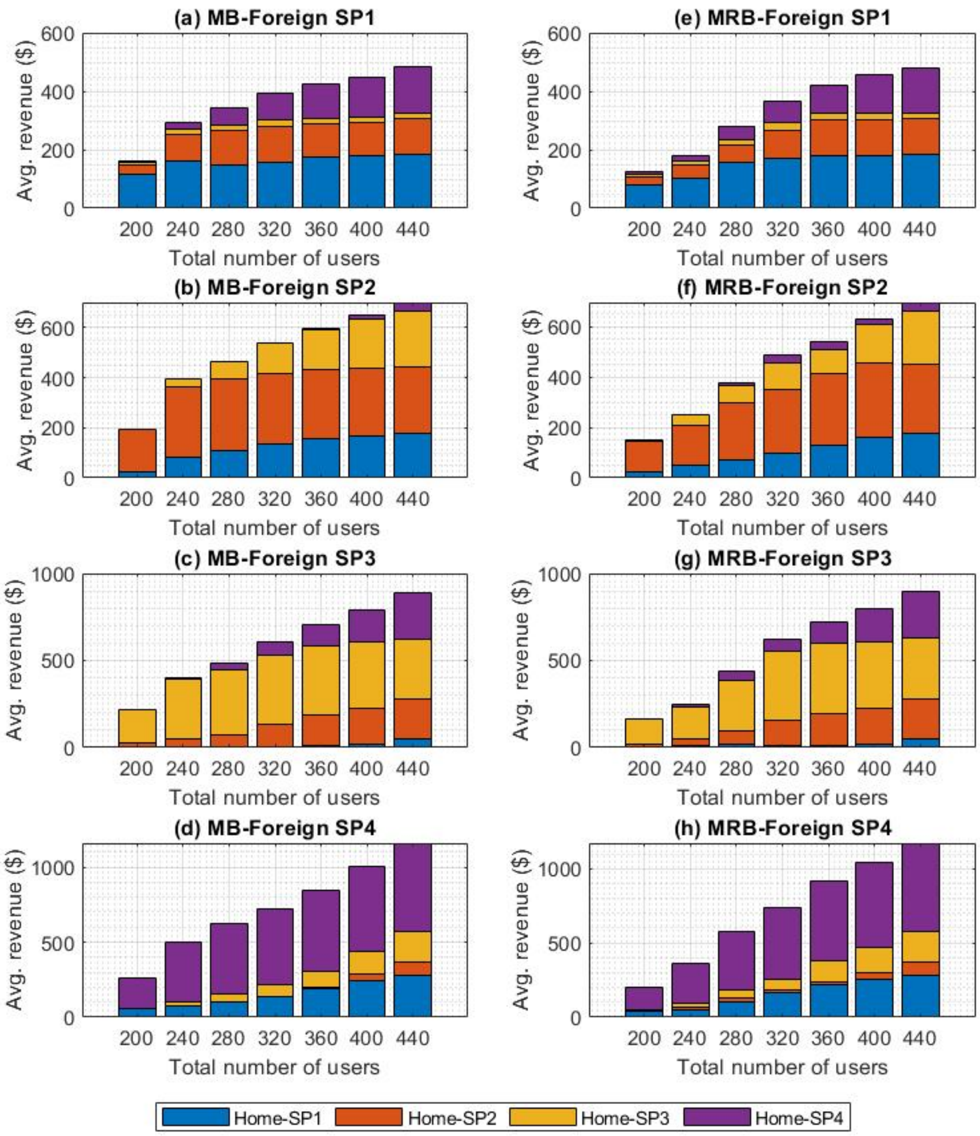

Figure 6.9: Foreign service providers' revenue at the Stackelberg equilibrium vs. the total number of users in the market-based (MB) approach and the market-reputation-based (MRB) approach

MB approach. As for the larger number of users the figures show that the performance of the proposed approaches are almost the same because both approaches have the same convergence unit price. In addition, foreign service providers succeeded in attracting slightly more home service providers' unserved users' tasks in the MRB approach than MB.

Convergence of Algorithm 4

The convergence of algorithm 4 is affected by three parameters: the initial price, the accuracy parameter ϵ, and the step parameter γ_i, where we set the initial price of all foreign service providers by the same value. We also set the step parameters of all foreign service providers by the same value, and fix the number of users in the system to be 280. Next, we present the effect of these parameters on foreign service provider 1 (other foreign service providers have the same performance).

1. Initial unit price:

   We set $\epsilon = 10^{-2}$ and $\gamma_i = 10^{-4}, \ \forall \ S_i^f \in \mathcal{S}^f$. Figure 6.10 shows the fluctuation of the unit price of the foreign service provider 1 to reach the convergence at the Stackelberg equilibrium, when the initial unit price of foreign service provider 1 $= 0.1 \ \frac{\$}{Mcycle}$ and $0.5 \frac{\$}{Mcycle}$. The number of steps to equilibrium price decreases as the initial price's value is closer to the convergence unit price.

2. Accuracy parameter ϵ:

   We set $\epsilon = 10^{-2}$ and $\gamma_i = 10^{-4}, \ \forall \ S_i^f \in \mathcal{S}^f$. Figure 6.11 shows the fluctuation of the unit price of foreign service provider 1 to reach the convergence at the Stackelberg equilibrium, when the accuracy parameter $= 10^{-2}, \ 10^{-3}$, and 10^{-4}.

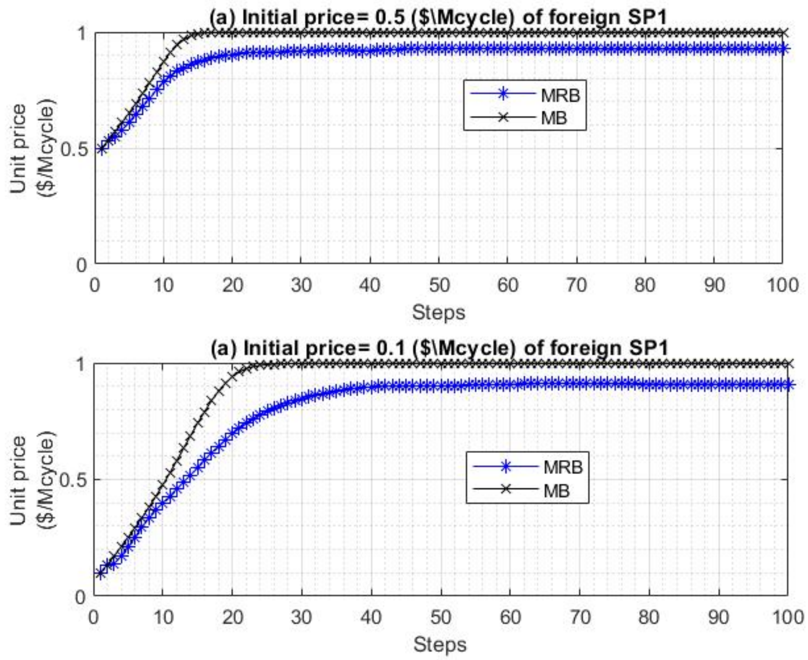

Figure 6.10: Unit price of the foreign service provider 1 with different initial prices vs. the number of steps to reach the steady-state

The number of steps to equilibrium price increases as the accuracy parameter's value decreases.

3. Step parameter γ_i:

   We set $\epsilon = 10^{-2}$ and initial price of all foreign service providers equals $0.5\ \frac{\$}{Mcycle}$. Figure 6.12 shows the fluctuation of the unit price of foreign service provider 1 to reach the convergence at the Stackelberg equilibrium when the step parameter $= 10^{-3}$ and 10^{-4}. The number of steps to equilibrium price increases as the step parameter's value decreases.

Lastly, Figure 6.13 shows that the fluctuation of the serving probability for foreign service provider 1 to reach the steady state when $\gamma = 10^{-4},\ \epsilon = 10^{-2}$, the initial unit price $= 0.5\ \frac{\$}{Mcycle}$, and the initial serving probability = 0.5 and 1 for all foreign

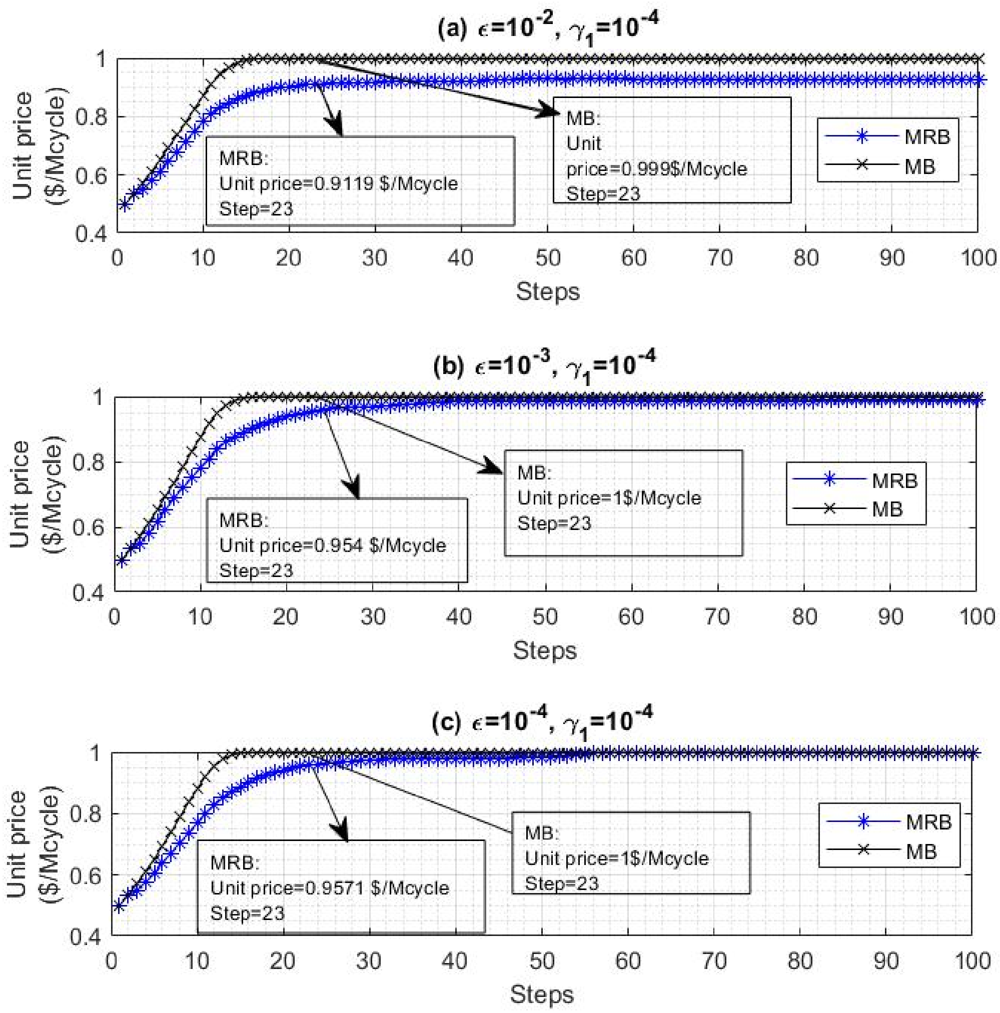

Figure 6.11: Unit price of the foreign service provider 1 with different ϵ vs. the number of steps to reach the steady-state

service providers. The steady-state serving probability does not change with the different initial value because serving probability changes according to the number of incoming requests, which depends on the unit price of foreign service provider 1.

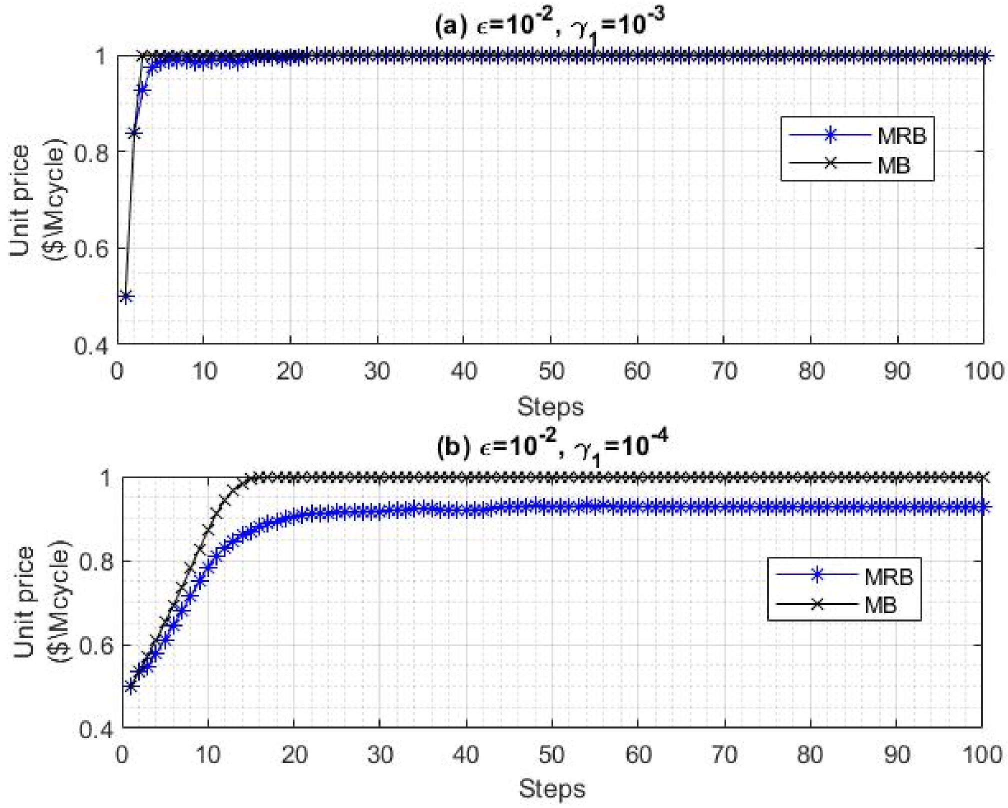

Figure 6.12: Unit price with different γ_i vs. the number of steps to reach the steady-state

6.5 Summary

In this chapter, we investigated the problem of task offloading with service replications and pricing issues in the MEC market. We proposed the MRB and the MB approaches using the Stackelberg game in which the home service providers act as followers and the foreign service providers act as leaders. We proved the existence of Nash equilibrium and a Stackelberg equilibrium. Regarding the home service provider problems, we proposed a linear relaxation-based solution. We used a greedy rounding algorithm to restore binary values for the decision variables to find the best values

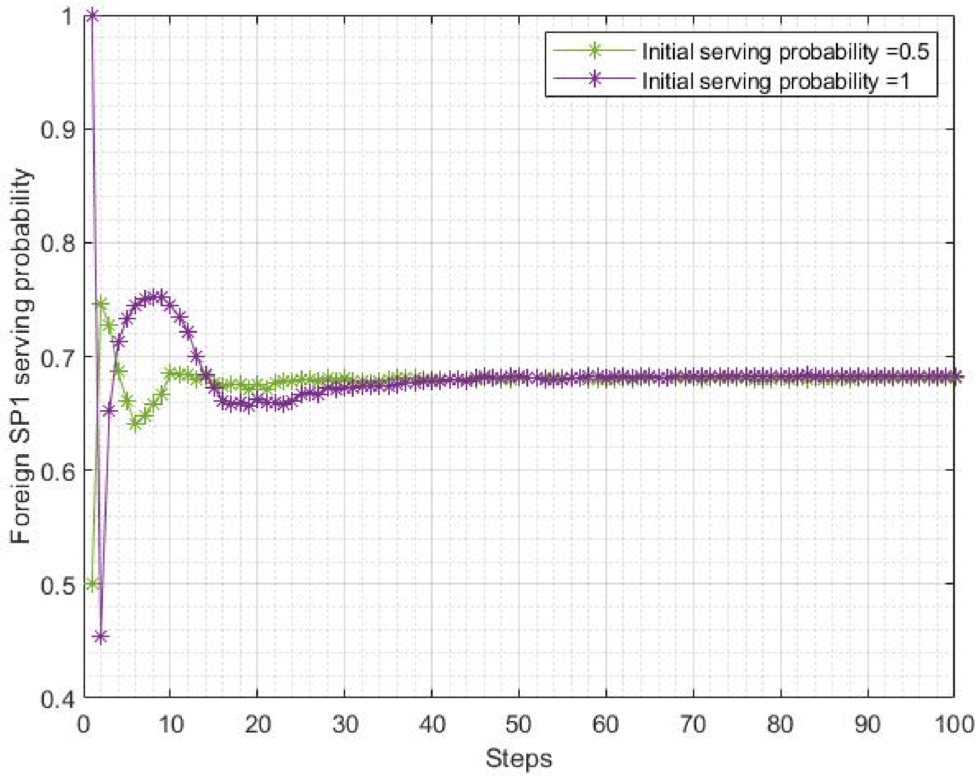

Figure 6.13: Serving probability fluctuation of foreign service provider 1 vs. the number of steps to reach the steady-state

considering the reputation of the foreign service providers. On the other side, we developed an iterative algorithm to reach equilibrium prices for the foreign service provider problems. In addition, we compared the MRB approach with the MB approach without considering foreign service providers' reputations.

The results show that the MRB approach is the best from the user satisfaction perspective by ensuring that foreign service providers serve users with a good reputation compared with the MB approach. Moreover, the MRB approach reduces the total cost that the home service provider pays to foreign service providers and attracts more home service providers to purchase foreign service providers' computing resources. However, the MRB and MB almost have the same average response time. Also, the MB approach succeeds in gaining more revenue for the foreign service providers with a smaller number of users due to the higher value of the convergence price. The convergence of the Stackelberg equilibrium algorithm depends on the initial price, accuracy parameter, and step parameter. However, the convergence does not depend on the initial value of the serving probability.

Chapter 7

Conclusion and Future Directions

This thesis explored improving MEC service scalability using service replication during high-traffic, high-density user demand applications and multi-user synchronized activities, such as viewing live sports events. This thesis also investigated maximizing the profit of edge computing service providers in highly competitive edge computing markets.

7.1 Summary

In Chapter 1, we provided an overview of the research problem and contributions. Chapter 2 presented an overview of the MEC paradigm, including its objectives, characteristics, applications, and enabling technologies SDN and NFV. We summarized existing MEC resource management solutions. Looking at MEC from an economic point of view, we discussed profit components and compared select existing economic and pricing models of resource management in cloud networking.

Chapter 3 proposed an MEC architecture aimed to improve scalability for real-time applications that cannot tolerate queuing at the MEC servers. The components of our system model and the system interactions between these architecture components were explained. Next, we presented our platoon of autonomous vehicles use case, and stated the system parameters used to define the response time and the system cost. The chapter concluded with performance evaluations, including performance metrics and simulation setup.

Chapter 4 proposed a scheme to improve the scalability of real-time offered services: by replicating them on-demand at underloaded remote edge servers in the MEC system, when local servers cannot sustain the load of all user groups concurrently. We dynamically managed task offloading for groups of users requesting to perform inter-related computations on multiple overloaded and underloaded MEC servers. This problem was formulated as an integer linear problem to minimize the average response time of all users while satisfying the time and time difference constraints of the user groups running the same applications. To reduce complexity, we proposed a linear relaxation-based solution, derived its optimal solution of its relaxed linear problem using Lagrangian analysis and KKT conditions, solved numerically, and presented a greedy rounding algorithm to restore binary values for the decision variables. Simulation results show this linear relaxation-based solution achieves close to the same performance as the ILP solution and outperforms the distance-based and resource-based greedy schemes that respect our model's scalability and group requirement settings. We also saw that, as the number of users increased, our scheme achieved more gains than the two greedy schemes.

Chapter 5 investigated the problem of task offloading with service replications between home service providers and foreign service providers in scalable MEC environments for groups of users involved in similar multimedia-intense applications.

In a scalable MEC, the home service providers that cannot immediately accept the tasks of all their users replicate the services of the un-served users onto foreign service providers to be completed. We formulated a constrained multi-objective optimization problem to minimize the average response time and the fee charged to foreign service providers while satisfying the time and time difference constraints and requirements of user groups running the same applications. We also reduced the complexity of the ILP solution by proposing the linear relaxation-based solution and presented a greedy rounding algorithm to restore binary values for the decision variables. We derived optimal solutions of the relaxed linear problem using Lagrangian analysis and KKT conditions for the weight-sum solution, in addition to solving the problem numerically. In addition, we formulated the USRA game to solve the problem in a distributed manner. At the same time, realizing centralized allocation might not be applicable in real scenarios because of the need for a central unit, we proposed the distributed allocation algorithm to solve the problem, aiming to minimize the weight-sum of users' average response time and the cost to the foreign service providers from the home service provider's perspective. The simulation results show that the Lexicographic solution outperforms the weight-sum solution and the distance-based and resource-based schemes in the cost as the number of users increased respecting our model's scalability and group service settings. Conversely, the weight-sum optimized solution achieved better average response times of all users in the MEC system. Therefore, the distributed solution is more realistic and efficient; however, its performance is less than the centralized solution.

Chapter 6 proposed the MRB to maximize the social welfare between all edge service providers. A home service provider's goal is to improve the scalability of real-time offered services by replicating them on demand at underloaded foreign service providers edge servers in the MEC system so all users are served concurrently when the home service providers cannot sustain the load from all the user groups. The foreign service provider's revenue stems from setting profitable prices and maintaining a good reputation by not having incoming requests blocked or delayed. In the MRB model, we formulated offloading and pricing problems between home and foreign service providers as a Stackelberg model, using foreign service providers' reputation in determining the best offloading decisions for all home service providers. We reduced the complexity of the ILP solution by suggesting the linear relaxation-based solution for the proposed problem of the home service providers. A proposed rounding algorithm to reach the integer values of the decision variables and the Stackelberg equilibrium algorithm to reach the equilibrium point was applied. We compared this MRB model with the MB approach (which does not consider the reputation). The simulation results indicate the MRB approach is the best from the perspective of user satisfaction. Moreover, the MRB approach reduces the home service provider's total cost to foreign service providers and attracts more home service providers to purchase foreign service providers' computing resources. However, the MRB and MB have close to the same average response time. We concluded that the MB approach gained more revenue for foreign service providers because of the higher value of the convergence price.

7.2 Future Directions

To fully capture the benefit of using service replication to improve MEC services' scalability there are several future directions to explore and open issues to be resolved.

1. Our schemes assume that users of the same group have the same local edge server/home service provider. To fully capture the concept of grouping between users, we need to investigate group members having different home service providers.

2. In our schemes, we assumed that users remain in their positions during the offloading and processing tasks. However, there is a dynamic nature associated with users as they are moving. Thus, their local servers will change and might not be covered by their home service providers. Thus, we need to discuss our schemes in a dynamic user environment.

3. Our schemes focused on the response time of users' tasks and the cost that a home service provider pays to foreign service providers. Next step would be to consider the energy efficiency as well.

4. We proposed the Stackelberg game to maximize the social welfare between service providers by setting profitable prices for computing cycles while assuming that the transmission rates for foreign service providers were predefined. Therefore, one could consider dynamic prices of transmission for transmission rate as well.

5. We proposed that foreign service provides set uniform prices for their computing cycles, which means all home service providers paid the same unit price for the

same foreign service provider. One could consider a scenario where each foreign service provider will set a different price for each home service provider.

6. We used the probability of serving users' tasks for all foreign service providers as their reputation. However, we need to explore the QoE metrics to guarantee perceived performance.

7.3 Concluding Remarks

We conclude this thesis with the following remarks:

1. Task offloading between different service providers using reactive service replication improves the scalability of the MEC services. However, this improvement depends on the idle computing resources of the underloaded service providers.

2. Applying a grouping concept is a promising approach because many real-time applications, such as group AR/VR, deal with multiple user groups and have different QoS requirements. In the proposed architecture, we managed the grouping idea and studied the effect of groups' constraints on the task offloading between service providers' edge servers.

3. The users' group has specific constraints: the maximum allowable time and the time difference constraints. These constraints affected users' assignments at foreign service providers. Users should be assigned at the proper provider's edge servers while the time difference between two users belonging to the same group is less than or equals a specific threshold. All these constraints made solving our proposed problems challenging.

4. We used the probability of serving users' tasks for all foreign service providers to indicate their reputations. Home service providers used these reputations with the published prices to determine the best task offloading and replication indicators.

5. Using the reputation changed the assignment of un-served users at foreign service providers' edge servers because the foreign service provider with a good reputation will attract more users; thus, the foreign service provider will be over-loaded. Therefore, setting profitable prices is the key to maintaining a good reputation.

Bibliography

[1] Ahmet Cihat Baktir, Atay Ozgovde, and Cem Ersoy. “How can edge computing benefit from software-defined networking: A survey, use cases, and future directions”. In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2359–2391.

[2] Mahadev Satyanarayanan et al. “The case for vm-based cloudlets in mobile computing”. In: *IEEE pervasive Computing* 8.4 (2009), pp. 14–23.

[3] Flavio Bonomi et al. “Fog computing and its role in the internet of things”. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, pp. 13–16.

[4] Arif Ahmed and Ejaz Ahmed. “A survey on mobile edge computing”. In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE. 2016, pp. 1–8. DOI: 10.1109/ISCO.2016.7727082.

[5] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. “Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges”. In: *Future Generation Computer Systems* 78 (2018), pp. 680–698.

[6] Min Chen and Yixue Hao. "Task offloading for mobile edge computing in software defined ultra-dense network". In: *IEEE Journal on Selected Areas in Communications* 36.3 (2018), pp. 587–597.

[7] Yifan Yu. "Mobile edge computing towards 5G: Vision, recent progress, and open challenges". In: *China Communications* 13.Supplement2 (2016), pp. 89–99.

[8] Yun Chao Hu et al. "Mobile edge computing—A key technology towards 5G". In: *ETSI white paper* 11.11 (2015), pp. 1–16.

[9] Nasir Abbas et al. "Mobile edge computing: A survey". In: *IEEE Internet of Things Journal* 5.1 (2017), pp. 450–465.

[10] Dimas Satria, Daihee Park, and Minho Jo. "Recovery for overloaded mobile edge computing". In: *Future Generation Computer Systems* 70 (2017), pp. 138–147.

[11] Mohammad Sajid and Zahid Raza. "Cloud computing: Issues & challenges". In: *International Conference on Cloud, Big Data and Trust*. 2013, pp. 13–15.

[12] Rashid Mijumbi et al. "Network function virtualization: State-of-the-art and research challenges". In: *IEEE Communications surveys & tutorials* 18.1 (2015), pp. 236–262.

[13] Abdallah Moubayed et al. "Edge-enabled V2X service placement for intelligent transportation systems". In: *IEEE Transactions on Mobile Computing* 20.4 (2020), pp. 1380–1392.

[14] Hassan Hawilo, Manar Jammal, and Abdallah Shami. “Network function virtualization-aware orchestrator for service function chaining placement in the cloud”. In: *IEEE Journal on Selected Areas in Communications* 37.3 (2019), pp. 643–655.

[15] Dimitrios Michael Manias et al. “Machine learning for performance-aware virtual network function placement”. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[16] Dimitrios Michael Manias et al. “Depth-optimized delay-aware tree (DO-DAT) for virtual network function placement”. In: *IEEE Networking Letters* 2.3 (2020), pp. 149–153.

[17] Yong Li and Min Chen. “Software-defined network function virtualization: A survey”. In: *IEEE Access* 3 (2015), pp. 2542–2553.

[18] Ibrahim Tamim et al. “Introducing virtual security functions into latency-aware placement for NFV applications”. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE. 2020, pp. 1–7.

[19] Qiang Duan, Yuhong Yan, and Athanasios V Vasilakos. “A survey on service-oriented network virtualization toward convergence of networking and cloud computing”. In: *IEEE Transactions on Network and Service Management* 9.4 (2012), pp. 373–392.

[20] Manar Jammal et al. “Software defined networking: State of the art and research challenges”. In: *Computer Networks* 72 (2014), pp. 74–98.

[21] Mohamed Abu Sharkh, Abdelkader Ouda, and Abdallah Shami. “A resource scheduling model for cloud computing data centers”. In: *2013 9th International*

Wireless Communications and Mobile Computing Conference (IWCMC). IEEE. 2013, pp. 213–218.

[22] Vincent WS Wong et al. *Key technologies for 5G wireless systems*. Cambridge university press, 2017.

[23] Pavel Mach and Zdenek Becvar. “Mobile edge computing: A survey on architecture and computation offloading”. In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1628–1656.

[24] Juan Liu et al. “Delay-optimal computation task scheduling for mobile-edge computing systems”. In: *2016 IEEE international symposium on information theory (ISIT)*. 2016, pp. 1451–1455.

[25] Yuyi Mao, Jun Zhang, and Khaled B Letaief. “Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems”. In: *2017 IEEE wireless communications and networking conference (WCNC)*. IEEE. 2017, pp. 1–6.

[26] Chen-Feng Liu, Mehdi Bennis, and H Vincent Poor. “Latency and reliability-aware task offloading and resource allocation for mobile edge computing”. In: *2017 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2017, pp. 1–7.

[27] Shermila Ranadheera, Setareh Maghsudi, and Ekram Hossain. “Computation offloading and activation of mobile edge computing servers: A minority game”. In: *IEEE Wireless Communications Letters* 7.5 (2018), pp. 688–691.

[28] Thinh Quang Dinh et al. “Offloading in mobile edge computing: Task allocation and computational frequency scaling”. In: *IEEE Transactions on Communications* 65.8 (2017), pp. 3571–3584.

[29] Peifeng Li et al. "Energy minimization and offloading number maximization in wireless mobile edge computing". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2018, pp. 1–6.

[30] Long Long et al. "Delay optimized computation offloading and resource allocation for mobile edge computing". In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE. 2019, pp. 1–5.

[31] Yongchao Zhang et al. "Cost efficient scheduling for delay-sensitive tasks in edge computing system". In: *2018 IEEE International Conference on Services Computing (SCC)*. IEEE. 2018, pp. 73–80.

[32] Lin Gu et al. "Cost efficient resource management in fog computing supported medical cyber-physical system". In: *IEEE Transactions on Emerging Topics in Computing* 5.1 (2015), pp. 108–119.

[33] Lin Wang et al. "Reconciling task assignment and scheduling in mobile edge clouds". In: *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE. 2016, pp. 1–6.

[34] Wenjie Wang and Wei Zhou. "Computational offloading with delay and capacity constraints in mobile edge". In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6.

[35] Wei Du et al. "Service capacity enhanced task offloading and resource allocation in multi-server edge computing environment". In: *arXiv preprint arXiv:1903.04709* (2019).

[36] Xiao Ma et al. “Cost-efficient workload scheduling in cloud assisted mobile edge computing”. In: *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*. IEEE. 2017, pp. 1–10.

[37] Meng-Hsi Chen, Ben Liang, and Min Dong. “Joint offloading decision and resource allocation for multi-user multi-task mobile cloud”. In: *2016 IEEE International Conference on Communications (ICC)*. IEEE. 2016, pp. 1–6.

[38] Meng-Hsi Chen, Ben Liang, and Min Dong. “Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point”. In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[39] Sumit Maheshwari et al. “Scalability and performance evaluation of edge cloud systems for latency constrained applications”. In: *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE. 2018, pp. 286–299.

[40] Meng-Hsi Chen, Min Dong, and Ben Liang. “Multi-user mobile cloud offloading game with computing access point”. In: *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*. IEEE. 2016, pp. 64–69.

[41] Umair Yaqub and Sameh Sorour. “Multi-objective resource optimization for hierarchical mobile edge computing”. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2018, pp. 1–6.

[42] Liang Tong, Yong Li, and Wei Gao. “A hierarchical edge cloud architecture for mobile computing”. In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE. 2016, pp. 1–9.

[43] Li Liu et al. “Performance modeling of representative load sharing schemes for clustered servers in multiaccess edge computing”. In: *IEEE Internet of Things Journal* 6.3 (2018), pp. 4880–4888.

[44] Roberto Beraldi, Abderrahmen Mtibaa, and Hussein Alnuweiri. “Cooperative load balancing scheme for edge computing resources”. In: *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE. 2017, pp. 94–100.

[45] Anselme Ndikumana et al. “Collaborative cache allocation and computation offloading in mobile edge computing”. In: *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE. 2017, pp. 366–369.

[46] Shangguang Wang et al. “A survey on service migration in mobile edge computing”. In: *IEEE Access* 6 (2018), pp. 23511–23528.

[47] Hassan Hawilo, Manar Jammal, and Abdallah Shami. “Orchestrating network function virtualization platform: Migration or re-instantiation?” In: *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*. IEEE. 2017, pp. 1–6.

[48] Tarik Taleb and Adlen Ksentini. “An analytical model for follow me cloud”. In: *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2013, pp. 1291–1296.

[49] Adlen Ksentini, Tarik Taleb, and Min Chen. “A Markov decision process-based service migration procedure for follow me cloud”. In: *2014 IEEE International Conference on Communications (ICC)*. IEEE. 2014, pp. 1350–1354.

[50] Kuo-Hsing Chiang and Nirmala Shenoy. “A 2-D random-walk mobility model for location-management studies in wireless networks”. In: *IEEE Transactions on Vehicular Technology* 53.2 (2004), pp. 413–424.

[51] Pantelis A Frangoudis and Adlen Ksentini. “Service migration versus service replication in multi-access Edge Computing”. In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE. 2018, pp. 124–129.

[52] Shuo Wang et al. “An approach for spatial-temporal traffic modeling in mobile cellular networks”. In: *2015 27th International Teletraffic Congress*. IEEE. 2015, pp. 203–209.

[53] Wan-Chi Chang and Pi-Chung Wang. “Adaptive replication for mobile edge computing”. In: *IEEE Journal on Selected Areas in Communications* 36.11 (2018), pp. 2422–2432.

[54] Ivan Farris et al. “Optimizing service replication for mobile delay-sensitive applications in 5G edge network”. In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6.

[55] Peijin Cong et al. “A survey of profit optimization techniques for cloud providers”. In: *ACM Computing Surveys (CSUR)* 53.2 (2020), pp. 1–35.

[56] Nguyen Cong Luong et al. “Resource management in cloud networking using economic analysis and pricing models: A survey”. In: *IEEE Communications Surveys & Tutorials* 19.2 (2017), pp. 954–1001.

[57] Costas Courcoubetis and Richard Weber. “Cost-based Pricing”. In: *WILEY-INTERSCIENCE SERIES IN SYSTEMS AND OPTIMIZATION* (2003), p. 163.

[58] Dinil Mon Divakaran, Mohan Gurusamy, and Mathumitha Sellamuthu. "Bandwidth allocation with differential pricing for flexible demands in data center networks". In: *Computer Networks* 73 (2014), pp. 84–97.

[59] Konstantinos Tsakalozos et al. "Flexible use of cloud resources through profit maximization and price discrimination". In: *2011 IEEE 27th International Conference on Data Engineering*. IEEE. 2011, pp. 75–86.

[60] William G Shepherd. "Ramsey pricing: Its uses and limits". In: *Utilities Policy* 2.4 (1992), pp. 296–298.

[61] Qiuping Li, Junhui Zhao, and Yi Gong. "Cooperative computation offloading and resource allocation for mobile edge computing". In: *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE. 2019, pp. 1–6.

[62] Keqin Li. "A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing". In: *IEEE Transactions on Sustainable Computing* (2018).

[63] Rakpong Kaewpuang et al. "A framework for cooperative resource management in mobile cloud computing". In: *IEEE Journal on Selected Areas in Communications* 31.12 (2013), pp. 2685–2700.

[64] Yifan Chen et al. "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing". In: *Future Generation Computer Systems* 108 (2020), pp. 273–287.

[65] Valeria Cardellini, Valerio Di Valerio, and Francesco Lo Presti. “Game-theoretic resource pricing and provisioning strategies in cloud systems”. In: *IEEE Transactions on Services Computing* 13.1 (2016), pp. 86–98.

[66] Bo Yang et al. “Stackelberg game approach for energy-aware resource allocation in data centers”. In: *IEEE Transactions on Parallel and Distributed systems* 27.12 (2016), pp. 3646–3658.

[67] R Preston McAfee and John McMillan. “Auctions and bidding”. In: *Journal of economic literature* 25.2 (1987), pp. 699–738.

[68] Yang Zhang, Dusit Niyato, and Ping Wang. “An auction mechanism for resource allocation in mobile cloud computing systems”. In: *International Conference on Wireless Algorithms, Systems, and Applications*. Springer. 2013, pp. 76–87.

[69] Liang Zheng et al. “How to bid the cloud”. In: *ACM SIGCOMM Computer Communication Review* 45.4 (2015), pp. 71–84.

[70] Abbas Kiani and Nirwan Ansari. “Toward hierarchical mobile edge computing: An auction-based profit maximization approach”. In: *IEEE Internet of Things Journal* 4.6 (2017), pp. 2082–2091.

[71] Yang Gui et al. “SOAR: Strategy-proof auction mechanisms for distributed cloud bandwidth reservation”. In: *2014 IEEE International Conference on Communication Systems*. IEEE. 2014, pp. 162–166.

[72] Dinil Mon Divakaran and Mohan Gurusamy. “Probabilistic-bandwidth guarantees with pricing in data-center networks”. In: *2013 IEEE International Conference on Communications (ICC)*. IEEE. 2013, pp. 3716–3720.

[73] Bassem Wanis, Nancy Samaan, and Ahmed Karmouch. "Efficient modeling and demand allocation for differentiated cloud virtual-network as-a service offerings". In: *IEEE Transactions on Cloud Computing* 4.4 (2015), pp. 376–391.

[74] Jian Zhao et al. "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers". In: *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE. 2014, pp. 118–126.

[75] Wenxin Li et al. "iDaaS: Inter-datacenter network as a service". In: *IEEE Transactions on Parallel and Distributed Systems* 29.7 (2015), pp. 1515–1529.

[76] K Hima Prasad et al. "Resource allocation and SLA determination for large data processing services over cloud". In: *2010 IEEE International Conference on Services Computing*. IEEE. 2010, pp. 522–529.

[77] Qi Zhang et al. "Dynamic service placement in geographically distributed clouds". In: *IEEE Journal on Selected Areas in Communications* 31.12 (2013), pp. 762–772.

[78] Makhlouf Hadji and Djamal Zeghlache. "Mathematical programming approach for revenue maximization in cloud federations". In: *IEEE transactions on cloud computing* 5.1 (2015), pp. 99–111.

[79] Jörn Altmann and Mohammad Mahdi Kashef. "Cost model based service placement in federated hybrid clouds". In: *Future Generation Computer Systems* 41 (2014), pp. 79–90.

[80] Ying Yuan, Cui-rong Wang, and Cong Wang. "A game based approach for sharing the data center network". In: *International Symposium on Neural Networks*. Springer. 2012, pp. 641–649.

[81] Cong Wang, Ying Yuan, and Cong Wan. "Lease data center in the light of network resources: An economic model". In: *2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control*. IEEE. 2014, pp. 606–610.

[82] Jin Zhang, Tao Xiong, and Wei Lou. "Community clinic: Economizing mobile cloud service cost via cloudlet group". In: *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE. 2014, pp. 208–216.

[83] Yu Wu and Lei Ying. "A cloudlet-based multi-lateral resource exchange framework for mobile users". In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 927–935.

[84] Tridib Mukherjee et al. "Risc: Robust infrastructure over shared computing resources through dynamic pricing and incentivization". In: *2015 IEEE International Parallel and Distributed Processing Symposium*. IEEE. 2015, pp. 1107–1116.

[85] Zhangyu Guan and Tommaso Melodia. "The value of cooperation: Minimizing user costs in multi-broker mobile cloud computing networks". In: *IEEE Transactions on Cloud Computing* 5.4 (2015), pp. 780–791.

[86] Matthew Andrews, Yue Jin, and Martin I Reiman. “A truthful pricing mechanism for sponsored content in wireless networks”. In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE. 2016, pp. 1–9.

[87] Liang Zhang, Weijie Wu, and Dan Wang. “Sponsored data plan: A two-class service model in wireless data networks”. In: *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. 2015, pp. 85–96.

[88] Joohyun Lee et al. “Economics of WiFi offloading: Trading delay for cellular capacity”. In: *IEEE Transactions on Wireless Communications* 13.3 (2014), pp. 1540–1554.

[89] Mohammad Hassan Lotfi et al. “The economics of quality sponsored data in wireless networks”. In: *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE. 2015, pp. 395–402.

[90] Stefano Paris et al. “A bandwidth trading marketplace for mobile data offloading”. In: *2013 Proceedings IEEE INFOCOM*. IEEE. 2013, pp. 430–434.

[91] Jihene Rezgui et al. “Platooning of autonomous vehicles with artificial intelligence V2I communications and navigation algorithm”. In: *2020 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE. 2020, pp. 1–6.

[92] Jiadai Wang, Jiajia Liu, and Nei Kato. "Networking and communications in autonomous driving: A survey". In: *IEEE Communications Surveys & Tutorials* 21.2 (2018), pp. 1243–1274.

[93] Abdallah Moubayed et al. "Cost-optimal v2x service placement in distributed cloud/edge environment". In: *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)(50308)*. IEEE. 2020, pp. 1–6.

[94] Luo Yuchong et al. "Task scheduling in mobile edge computing with stochastic requests and m/m/1 servers". In: *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE. 2019, pp. 2379–2382.

[95] https://www.mathworks.com/products/optimization.html.

[96] Benjamin Doerr. "Linear discrepancy of totally unimodular matrices". In: *Combinatorica* 24.1 (2004), pp. 117–125.

[97] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[98] Rama Cont and Massoud Heidari. "Optimal rounding under integer constraints". In: *arXiv preprint arXiv:1501.00014* (2014).

[99] Christos H Papadimitriou. "On the complexity of integer programming". In: *Journal of the ACM (JACM)* 28.4 (1981), pp. 765–768.

[100] Nimrod Megiddo. “On the complexity of linear programming”. In: *Advances in economic theory: Fifth world congress*. Cambridge University Press London. 1987, pp. 225–268.

[101] Clovis C Gonzaga. “On the complexity of linear programming”. In: *Resenhas do Instituto de Matemática e Estatıstica da Universidade de São Paulo* 2.2 (1995), pp. 197–207.

[102] Gerald W Evans. “An overview of techniques for solving multiobjective mathematical programs”. In: *Management Science* 30.11 (1984), pp. 1268–1282.

[103] Matthias Ehrgott. *Multicriteria optimization*. Vol. 491. Springer Science & Business Media, 2005.

[104] SS Rao. “Multiobjective optimization in structural design with uncertain parameters and stochastic processes”. In: *AIAA journal* 22.11 (1984), pp. 1670–1678.

[105] Qiang He et al. “A game-theoretical approach for user allocation in edge computing environment”. In: *IEEE Transactions on Parallel and Distributed Systems* 31.3 (2019), pp. 515–529.

[106] Jason R Marden, Gürdal Arslan, and Jeff S Shamma. “Cooperative control and potential games”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.6 (2009), pp. 1393–1407.

[107] Mengyu Liu and Yuan Liu. “Price-based distributed offloading for mobile-edge computing with computation capacity constraints”. In: *IEEE Wireless Communications Letters* 7.3 (2017), pp. 420–423.

[108] Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

[109] Sabita Maharjan et al. "Dependable demand response management in the smart grid: A Stackelberg game approach". In: *IEEE Transactions on Smart Grid* 4.1 (2013), pp. 120–132.

[110] J Ben Rosen. "Existence and uniqueness of equilibrium points for concave n-person games". In: *Econometrica: Journal of the Econometric Society* (1965), pp. 520–534.